SecureMetric Technology ST3 ACE Token Hardware Version: 1.0.0

FIPS 140-2 Non-Proprietary Security Policy





Prepared by:



KUALA LUMPUR (HQ)
SecureMetric Technology Sdn. Bhd.
2-2 Incubator 2 Technology Park Malaysia Lebuhraya Sg Besi -

2-2, Incubator 2, Technology Park Malaysia, Lebuhraya Sg Besi - Puchong, Bukit Jalil, 57000 Kuala Lumpur, Malaysia T +603 8996 8225 F +603 8996 7225



JAKARTA SINGAPORE HANOI HO CHI MINH CITY MANILA

Table of Contents

| 1 Introduction | 4 |
|---|----|
| 1.1 Purpose | 4 |
| 1.2 References | 4 |
| 2 ST3 ACE Token | 5 |
| 2.1 Overview | 5 |
| 2.2 Module Specification | 6 |
| 2.3 Module Interfaces | 7 |
| 2.4 Roles and Services. | 8 |
| 2.4.1 Crypto-Officer Role | 10 |
| 2.4.2 User Role | 15 |
| 2.4.3 Additional Services | 18 |
| 2.5 Physical Security | 20 |
| 2.6 Operational Environment | 20 |
| 2.7 Cryptographic Key Management | 21 |
| 2.8 EMI/EMC | 28 |
| 2.9 Self-Tests | 28 |
| 2.9.1 Power-Up Self-Tests | 28 |
| 2.9.2 Conditional Self-Tests | 28 |
| 2.10 Mitigation of Other Attacks | 28 |
| 3 SECURE OPERATIONS | 29 |
| 3.1 Detecting a FIPS Cryptographic Module | 29 |
| 3.2 Initial Setup | 30 |
| 3.2.1 Zeroization | 30 |
| 3.3 Non-Approved Mode | 30 |
| A ACRONYMS | 31 |

Figures List

| Figure 1 – SECUREMETRIC's ST3 ACE Token | 5 |
|--|----|
| Figure 2 – Physical Cryptographic Boundary | |
| Figure 3 - "FIPS" Label Location | |
| Figure 4 - "FIPS-Mode-Detect" Tool | |
| Tables List | |
| Table 1 – Security Level Per FIPS 140-2 Section | |
| Table 2 – Operator Authentication Mechanism | 8 |
| Table 3 – APDU Command Structure | 9 |
| Table 4 – APDU Command Response Structure | 9 |
| Table 5 – Mapping of Crypto-Officer Role's Services to Inputs, Outputs, CSPs, and Type of Access | |
| Table 6 – Mapping of User Role's Services to Inputs, Outputs, CSPs, and Type of Access | |
| Table 7 – Mapping of Unauthenticated Services to Inputs, Outputs, CSPs, and Type of Access | |
| Table 8 – FIPS-Approved Algorithm Implementations | |
| Table 9 – FIPS-Allowed Algorithm Implementations | |
| Table 10 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs | |
| Table 11 – Non-Approved Services | |
| Table 12_ Acronyms | 31 |

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the ST3 ACE Token from SecureMetric Technology Sdn. Bhd. This Security Policy describes how the ST3 ACE Token meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 3 FIPS 140-2 validation of the module. The ST3 ACE Token is referred to in this document as ST3 ACE Token, crypto-module, or the module.

1.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The SecureMetric website (http://www.securemetric.com) contains information on the full line of products from SECUREMETRIC.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains contact information for individuals to answer technical or sales-related questions for the module.

2 ST3 ACE Token

2.1 Overview

SECUREMETRIC is a leading innovator of smart card and Chip Operating System (COS) based security technologies and applications. Their product offerings include devices that provide software protection, strong authentication, and smart card operating systems. Evidence of SECUREMETRIC's continued leadership and innovation is demonstrated within this Security Policy, which specifies their second FIPS 140-2 validated cryptographic module. This new module, referred to as the ST3 ACE Token, is a USB token containing SECUREMETRIC's own SECUREMETRIC-FIPS-COS cryptographic operating system. The SECUREMETRIC-FIPS-COS is embedded in an ST23YT66 Integrated Circuit (IC) chip and has been developed to support SECUREMETRIC's ST3 ACE USB token (Figure 1). The ST3 ACE Token is designed to provide strong authentication and identification and to support network login, secure online transactions, digital signatures, and sensitive data protection. SECUREMETRIC's ST3 ACE Token guarantees safety of its cryptographic IC chip and other components with its hard, semi-transparent, polycarbonate shell.



Figure 1 - SECUREMETRIC's ST3 ACE Token

The ST3 ACE Token is validated at the following FIPS 140-2 Section levels (Table 1):

Table 1 – Security Level Per FIPS 140-2 Section

| Section | Section Title | Level |
|---------|---|-------|
| 1 | Cryptographic Module Specification | 3 |
| 2 | Cryptographic Module Ports and Interfaces | 3 |
| 3 | Roles, Services, and Authentication | 3 |
| 4 | Finite State Model | 3 |

2014 SECUREMETRIC TECHNOLOGY SDN BHD

USB – Universal Serial Bus

| Section | Section Title | Level |
|---------|------------------------------|-------|
| 5 | Physical Security | 3 |
| 6 | Operational Environment | N/A |
| 7 | Cryptographic Key Management | 3 |
| 8 | EMI/EMC2 | 3 |
| 9 | Self-tests | 3 |
| 10 | Design Assurance | 3 |
| 11 | Mitigation of Other Attacks | N/A |

2.2 Module Specification

The ST3 ACE Token is a hardware module with a multi-chip standalone embodiment. The overall security level of the module is 3. The logical and physical cryptographic boundaries of the ST3 ACE Token are defined by the hard, semi-transparent, polycarbonate casing of the USB token. The ST3 ACE Token is comprised of a STMicroelectronics ST23YT66 serial access microcontroller sitting atop a Printed Circuit Board (PCB). The PCB carries the signals and instructions of the microcontroller to the other components contained within the ST3 ACE Token. All cryptographic functions and firmware are stored within the microcontroller package and executed by an 8/16-bit ST23 CPU (Core Processing Unit). A LED contained within the USB token shows power, initialization, and operation status through the semi-transparent casing of the USB token. All other logical functions take place through the USB connector, covered in Section 2.3 of this document. Please refer to Figure 2 below for a depiction of the physical cryptographic boundary and logical flows of the ST3 ACE Token.

-

²EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

³LED – Light Emitting Diode

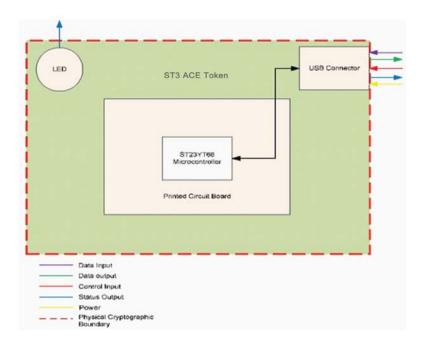


Figure 2 – Physical Cryptographic Boundary

The ST3 ACE Token is shipped in a FIPS-Approved mode of operation, as indicated on the module and will always operate in a FIPS-Approved mode of operation. Section 3 details how to tell if the module is a FIPS module and is running in a FIPS approved mode of operation. Section 2.7 gives a complete list of FIPS-Approved algorithms within the module.

2.3 Module Interfaces

The cryptographic boundary of the ST3 ACE Token is the outer polycarbonate casing of the USB token. There is only one physical point, the USB connector, at which the module interfaces with equipment outside of the physical boundary. The USB connector facilitates the following logical interfaces:

- Data Input
- Data output
- Control Input
- Status Output
- Power

The USB connector contains 4 pins: Data+ (D+), Data-(D-), VCC⁴, and Ground (GND). These 4 pins carry out the logical interfaces as defined by FIPS 140-2 and are described below:

- The D+ and D-pins carry all Data Input, Data Output, Control Input, and Status Output signals to and from the module.
- The VCC pin handles up to 5V DC power input from whatever source the USB connector is plugged into.

⁵V -Volt

⁴VCC – Common Collector Voltage

The GND pin also handles up to 5V DC power and helps to regulate the power consumed by the USB token.

An LED contained within the USB token is used for status output. This LED shows power, initialization, and operational status through the semi-transparent casing of the USB token.

2.4 Roles and Services

The module supports the two roles required by FIPS 140-2: Crypto-Officer and User. The Crypto-Officer is the role responsible for module initialization, including file system management, key management, and access control management. The User role is the everyday user of the device. Once authenticated, the Crypto-officer and User role is implicitly selected, allowing the operator to access services from both roles. Please see Table 2 for details regarding the authentication mechanism. Table 5 and Table 6 below specify the full list of services per supported role. Unauthenticated services are also supported by the module. The services not requiring authentication are listed in Table 7.

Table 2 - Operator Authentication Mechanism

| Authentication | Authentication Data | Authentication Mechanism | | |
|----------------|-------------------------------------|--|--|--|
| Mechanism | | | | |
| Identity-based | 128-bit AES ⁷ Key Shared | The AES key is 128 bits in length. The probability that a random | | |
| | Secret | attempt will succeed or a false acceptance occur is no greater than | | |
| | | 1/2^128, which is less than 1/1,000,000. | | |
| | | The module will allow fewer than 600 authentication attempts in a one minute period. Therefore, the random success rate for multiple retries is 600/2^128, which is less than 1/100,000. | | |
| Identity-based | 3-key Triple-DES Shared | Each Triple-DES key is effectively 56 bits in length, resulting in a | | |
| identity bused | Secret Secret | total of 168 bits of total keying material. The probability that a | | |
| | | random attempt will succeed or a false acceptance occur is no | | |
| | | greater than 1/2^168, which is less than 1/1,000,000. | | |
| | | The module will allow fewer than 600 authentication attempts in a one minute period. Therefore, the random success rate for multiple retries is 600/2^168, which is less than 1/100,000. | | |

⁷ AES – Advanced Encryption Standard

⁶ DC – Direct Current

| Authentication Mechanism | Authentication Data | Authentication Mechanism |
|-----------------------------|---------------------|---|
| Identity-based | RSA Key Pairs | The modules supports RSA public key authentication. Using conservative estimates and equating a 2048-bit RSA key to an 112-bit symmetric key, the probability for a random attempt to succeed is 1/2 ¹¹² . The module will allow fewer than 600 authentication attempts in a one minute period. Therefore, the random success rate for multiple retries is 600/2^112, which is less than 1/100,000. |

All services provided by ST3 ACE Token are implemented in accordance with ISO 7816-4, which defines the interface available as a command and response pair referred to as an Application Protocol Data Unit (APDU). The module will process only one command at a time, per channel (of four available logical channels), and must process and respond before allowing another command to be processed over any given channel. Table 3 and Table 4 show a typical ADPU command structure and command response structure used by the module, respectively.

Table 3 - APDU Command Structure

| Hea | ıder | Lc Field | Data Field | Le Field |
|-----|------|----------|---------------------------|----------|
| CLA | INS | 1 byte | Input Data (1 or 3 bytes) | 1 byte |

ADPU command structure descriptions:

- CLA The Class byte indicates the class of the command as follows:
 - If the class of the command is inter-industry or not
 - If secure messaging is required
 - Logical channel 0-3
- INS The Instruction byte indicates the command to process as follows:
 - Command word
 - Data encoding
- Lc Length in bytes of the data field
- Data Field Data input with command for processing
- Le-Maximum number of bytes expected in the response

Table 4 - APDU Command Response Structure

| Data Field | Trailer |
|---------------|--------------|
| Response data | Status bytes |

ADPU command response structure descriptions:

- Data Field Data output, if applicable
- Trailer Status bytes (e.g. 9000, 64XX)

⁸ ISO – International Organization for Standardization

⁹IEC – International Electrotechnical Commission

2.4.1 Crypto-Officer Role

This section provides a list of all services accessible to a Crypto-Officer (Table 5). The list includes a full description of each service, and in addition, it describes the type of access that each service has to a CSP¹⁰.

NOTE:

- R Read: The CSP is read.
- W Write: The CSP is established, generated, modified, or zeroized.
- X Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism.

Table 5 - Mapping of Crypto-Officer Role's Services to Inputs, Outputs, CSPs, and Type of Access

| Service | INS | Description | Input | Output | CSP and Type of |
|---------|-----|--------------------------------|---|------------------------------|----------------------|
| | | | | | Access |
| Read | В0 | Allows read access to a binary | Offset address of the | • File data or "Nonexistent" | No CSPs are accessed |
| Binary | | file. A binary file is a file | binary file to read | • Status (e.g. 9000, 6283, | via this service. |
| | | whose content is a sequential | • Length of the data to be | 6284, 6A80, 6A81, 6A82, | |
| | | string of bits. | read | 6A86, 6A87) | |
| Update | D6 | Allows write access to a | Offset address of the | • Status (e.g. 9000, 6283, | No CSPs are accessed |
| Binary | | binary file. | binary file to read | 6284, 6A80, 6A81, 6A82, | via this service. |
| | | | • Length of the data to be | 6A86, 6A87) | |
| | | | read | | |
| Read | B2 | Allows read access to a | Record number | Record data or | No CSPs are accessed |
| Record | | record. A record is a type of | • Read parameter (i.e, all | "Nonexistent" | via this service. |
| | | data storage structure as | records starting at | • Status (e.g. 9000, 6283, | |
| | | defied within ISO 7816. | specified record number, | 6284, 6A80, 6A81, 6A82, | |
| | | Records are stored in files. | or just one record) | 6A86, 6A87) | |
| Update | DC | Allows write access to a | Record number | • Status (e.g. 9000, 6283, | No CSPs are accessed |
| Record | | record. | Length of record | 6284, 6A80, 6A81, 6A82, | via this service. |
| | | | Record data | 6A86, 6A87) | |
| | | | • Read parameter (i.e, | | |
| | | | update the record | | |
| | | | specified by the record | | |
| | | | number) | | |

.

¹⁰CSP – Critical Security Parameter

| Service | INS | Description | Input | Output | CSP and Type of |
|--------------|-----|-------------------------------|--|-----------------------------|----------------------------------|
| | | | | | Access |
| Append | E2 | Allows a record to be | • Record number | • Status (e.g. 9000, | No CSPs are accessed |
| Record | | appended | • Current file | 6283, 6284, 6A80, | via this service |
| | | | • Length of record | 6A81, 6A82, 6A86, | |
| | | | Record data | 6A87) | |
| | | | Read parameter | | |
| | | | (i.e, update the | | |
| | | | record specified by | | |
| | | | the record | | |
| | | | number) | | |
| External | 82 | Authenticates an external | Initiate a secure sessions | • Status (e.g. 9000) | Initiate a secure |
| Authenticate | | entity to the cryptographic | | • Retry number for the | session: |
| | | module. This service may | Authentication data of | referenced key incremented | • INIT_KEY _{enc} : R, X |
| | | also be used to both | external entity (32 bytes) plus | by one. | • INIT_KEY _{mac} : R, X |
| | | authenticate and initiate a | the MAC ¹¹ (8 bytes) | | • K _{enc} : R, X |
| | | secure session with an | | NOTE: If successful, this | • K _{mac} : R, X |
| | | external entity. | Or | number is then reset to the | • K _{Senc} : W |
| | | | | maximum | • KS _{mac} :W |
| | | NOTE: Prerequisite to this | Authenticate only: | | |
| | | service is the use of Get | • Algorithm type (AES, | | Or |
| | | Challenge service. The key | Triple-DES ¹² , RSA ¹³) | | Authenticate Only: |
| | | as referenced within the | • Key ID (Key Index) | | • Symmetric key: R, X |
| | | service call exists under the | • Length of data in the field | | • RSA Private Key: R, |
| | | current file. | Authentication data (data | | X |
| | | | field) | | |
| Internal | 88 | Authenticates the | Algorithm type | Authentication data | Authenticate Only: |
| Authenticate | | cryptographic module to | (AES,Triple-DES, | • Status (e.g. 9000, | • Symmetric key: R, |
| | | an external entity | RSA) | 6300, 62CX, 6581, | X |
| | | | • Key ID (Key | 6700, 6982, 6984, | • RSA Private Key: |
| | | NOTE: In order for this | Index) | 6A81, 6A2, 6A86, | R, X |
| | | service to be utilized, the | • Length of data in | 6A88) | |
| | | external entity must have | the field | | |
| | | privileged access to the | • Random data (data | | |
| | | referenced key. | field) | | |
| | | | , | | |

¹¹ MAC – Message Authentication Code

¹²DES – Data Encryption Standard

¹³ RSA – Rivest, Adleman, and Shamir

| Service | INS | Description | Input | Output | CSP and Type of |
|--------------|-----|--|-----------------------------------|----------------------------|----------------------|
| | | | | | Access |
| Verify | 20 | Provides PIN ¹⁴ verification. | • Reference to the PIN | • Status (e.g. 9000, 6300, | • PIN: R, X |
| | | | • PID ¹⁵ | 62CX, 6581, 6700, 6982, | |
| | | NOTE: In order for this | Data to be verified | 6984, 6A81, 6A2, 6A86, | |
| | | service to be utilized, the | | 6A88) | |
| | | external entity must have | | | |
| | | privileged access to the | | | |
| | | referenced PIN. | | | |
| Change | 24 | Modify the PIN | • Old PIN | • Status (e.g. 9000, | • PIN: R, W, X |
| Reference | | | • New PIN | 6300, 62CX, 6581, | |
| Data | | NOTE: In order for this | • Reference to the PIN | 6700, 6982, 6984, 6A81, | |
| | | service to be utilized the | • PID | 6A2, 6A86, 6A88) | |
| | | external entity must have | | | |
| | | privileged access to the | | | |
| | | referenced PIN. | | | |
| Enable | 28 | Modifies a PIN's state | • Reference to the | • Status (e.g. 9000, | No CSPs are accessed |
| Verification | | from invalid to valid. | PIN | 6300, 62CX, 6581, | via |
| Requirement | | | • PID | 6700, 6982, 6984, 6A81, | this service. |
| | | NOTE: Utilization of this | | 6A2, 6A86, 6A88) | |
| | | service requires permission | | | |
| | | to activate the PIN. | | | |
| | | | | | |
| | | | | | |
| Disable | 26 | Modifies a PINs state | Reference to the | • Status (e.g. 9000, | No CSPs are accessed |
| Verification | | from valid to invalid. | PIN | 6300, 62CX, 6581, | via |
| Requirement | | | • PID | 6700, 6982, 6984, 6A81, | this service. |
| | | NOTE: Utilization of this | | 6A2, 6A86, 6A88) | |
| | | service requires permission | | | |
| | | to invalidate the PIN. | | | |
| Reset Retry | 2C | Resets the retry counter of | Reset parameter (resets | • Status (e.g. 9000, 6300, | No CSPs are accessed |
| Counter | | the PIN to its initial value. | recount maximum number and | 62CX, 6581, 6700, 6982, | via this service. |
| | | | remaining count to default) | 6984, 6A81, 6A2, 6A86, | |
| | | NOTE: Utilization of this | • Restore parameter (restores | 6A88) | |
| | | service requires permission | recount to initial default value) | | |
| | | to modify PIN. | • Reference to PIN | | |
| | | | • PID | | |

¹⁴ PIN – Personal Identification Number

 $^{^{15}}$ PID - Personal Identification number index

| Generate Asymmetric Rey Pair Generates an Asymmetric Rey Pair Asymmetr | Service | INS | Description | Input | Output | CSP and Type of |
|---|------------|-----|---------------------------------|---------------------------------|----------------------------|----------------------|
| Asymmetric Key Pair Algorithm ID | | | | | | Access |
| Asymmetric Key Pair Key Pair Key Pair Aggratum Aggratum Angoroum of Checksum Aggratum Angoroum of Checksum Angoroum of Compute Angoroum of Checksum Angoroum of Chicksup of Chicksum Angoroum of Chicksup of Chicksum Angoroum of Chicksup of Chicksup of Chicksum Angoroum of Chicksup o | Generate | 46 | Generates an Asymmetric | •Key parameter information | • Status (e.g. 9000, 6300, | |
| Modulus Leagth Private Key File Identifier GPRD Performs an encrypt operation using an Approved security function. Plaintext data Ciphertext data * Status (e.g. 9000, 6300, 62CX, Kangheric key: R. 24 Performs an encrypt operation using an Approved security function. Plaintext data Ciphertext data * Status (e.g. 9000, 6300, 62CX, Kangheric key: R. 25 Plaintext data Ciphertext data * Status (e.g. 9000, 6300, 62CX, Kangheric key: R. 25 Plaintext data Plai | Asymmetric | | key pair | •Algorithm ID | 62CX, 6581, 6700, 6982, | • RSA Public Key: W |
| Encrypt 2A Performs an encrypt operation using an Approved security function. NOTTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation Pecrypt 2A Performs a decrypt operation. NOTTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation NOTTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation Verify 2A Verifies a digital signature using RSA PKCS ⁷⁷ #1 using RSA PKCS ⁷⁷ #1 signature Verify 2A Computes a digital signature pigital signature signature using RSA PKCS ⁷⁷ #1 signature signature using RSA performs AES or Triple-OES checksum verification. Verify 2A Computes an AES or The data used to compute the cryptographic checksum The length of the cryptographic checksum Palaintext data *Ciphertext data *Status (e.g. 9000, 6300) *Symmetric key: R, X *RSA Public key: R, X *RSA Private Key: R, X *RSA Private Ke | Key Pair | | | Modulus Length | 6984, 6A81, 6A2, 6A86, | |
| Encrypt | | | | Private Key File Identifier | 6A88) | |
| operation using an Approved security function. NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation. Decrypt Operation Verify 2A Verifies a digital signature using RSA PKCS ⁷ #1 digital signature Unigital Signature Compute 2A Computes a digital signature PKCS#1. Verify 2A Performs AES or Triple DES checksum verification. Verify 2A Performs AES or Triple DES checksum. Triple-DES checksum. Compute the cryptographic checksum object plus the cryptographic checksum of the | | | | (FID) | | |
| Approved security function. NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation. Decrypt Decrypt ZA Performs a decrypt operation. Verify Lorify Lorif | Encrypt | 2A | Performs an encrypt | Plaintext data | • Ciphertext data • Status | • Symmetric key: R, |
| Function. NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation. *Ciphertext | | | operation using an | | (e.g. 9000, 6300, 62CX, | X |
| NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation Decrypt 2A Performs a decrypt operation NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation Verify 2A Verifies a digital signature using RSA PKCS*#1 Signature Compute 2A Computes a digital signature signature using RSA PKCS*#1 PRCS#1. Verify 2A Performs AES or Triple* DES checksum DES checksum Other Cryptograph ic checksum Compute 2A Computes and signature using RSA PKCS** DES checksum Other Cryptograph ic checksum Triple-DES checksum. Compute 2A Computes and Service data push the digital signature PRCS#1. Verify 2A Performs AES or Triple* DES checksum Other Cryptograph ic checksum data Triple-DES checksum. Triple-DES checksum. The length of the cryptographic checksum Triple-DES checksum. | | | Approved security | | 6581, 6700, 6982, 6984, | • RSA Public Key: R, |
| must have previously been utilized to choose the algorithm and key for the security operation. Decrypt | | | function. | | 6A81, 6A2, 6A86,6A88) | X |
| Decrypt ZA | | | NOTE: The MSE service | | | |
| Decrypt ZA | | | must have previously been | | | |
| Decrypt 2A Performs a decrypt operation. *Ciphertext *Plaintext *Symmetric key: R, X operation *NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation. *Data Object of the signed data plus the digital signature *Status of the verification *RSA Private Key: R, X | | | | | | |
| Security operation. Security operation. Performs a decrypt operation Performs a decry | | | algorithm and key for the | | | |
| Decrypt 2A Performs a decrypt operation NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation. Verify 2A Verifies a digital signature using RSA PKCS ¹⁷ #1 data plus the digital signature Signature using RSA pKCS ¹⁷ #1 digital signature by Compute ApkCS#1. Verify 2A Computes a digital signature using RSA pKCS ¹⁷ #1 digital signature by Cryptograp hic checksum data Verify 2A Performs AES or Triple by Plaintext data object plus the cryptographic checksum data Compute 2A Computes an AES or Triple or Triple-DES checksum. The length of the cryptographic checksum Compute Cryptograp hic checksum The length of the cryptographic checksum The length of the cryptographic checksum The length of the cryptographic checksum PKCSH1. Plaintext A Plaintext PRIaintext PRIATURE Plaintext PRIATURE | | | | | | |
| operation NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation. Verify 2A Verifies a digital signature using RSA PKCS¹*#1 data plus the digital signature signature using RSA pkcS**#1 PKCS#1. Verify 2A Performs AES or Triple DES checksum checksum Compute 2A Computes a AES or Triple DES checksum. The length of the cryptographic checksum Compute 2A Computes a DES checksum. The length of the cryptographic checksum Compute 2A Computes an AES or Triple ompute the cryptographic checksum The length of the cryptographic checksum The length of the cryptographic checksum Verify 2A Computes an AES or Triple ompute the cryptographic checksum The length of the cryptographic checksum Compute the cryptographic checksum The length of the cryptographic checksum Tombut AES or Triple ompute the cryptographic checksum The length of the cryptographic checksum Tombut AES or Triple ompute the cryptographic checksum The length of the cryptographic checksum Tombut AES or Triple ompute the cryptographic checksum Tombut AES or Triple ompute the cryptographic checksum The length of the cryptographic checksum Tombut AES or Triple ompute the cryptographic checksum Tombut AES or Triple of the cryptogra | Decrypt | 2A | | Ciphertext | Plaintext | • Symmetric key: R, |
| NOTE: The MSE service must have previously been utilized to choose the algorithm and key for the security operation. Verify 2A Verifies a digital signature using RSA PKCS ¹⁷ #1 vata plus the digital signature signature using RSA pKCS ¹⁷ #1 visinature a digital signature signature using RSA pKCS ¹⁷ #1 visinature signature using RSA pKCS ¹⁷ #1 visinature vising RSA pKCS ¹ #1 visinature vising RSA private Key: RSA Private Key: RSA Private Key: A pKCS pKCS private Key: A pKCS private Key: A pKCS pKCS pKCS pKCS pKCS pKCS pKCS pKCS | • • | | | • | | |
| must have previously been utilized to choose the algorithm and key for the security operation. Verify 2A Verifies a digital signature using RSA PKCS ¹⁷ #1 data used to cryptographic checksum tilized to choose the algorithm and key for the security operation. **Note of the verification using RSA PKCS ¹⁷ #1 data used to compute the cryptographic checksum **Data Object of the signed data plus the digital signature **Data Object of the signed data plus the digital signature **PKSA Private Key: **R, X **PKSA Private Key: **R, X **PKSA Public Key: R, X **Jubical Signature **Status (e.g. 9000, 6300) **Symmetric Key: **R, X **PKCS#1.** **Plaintext data object plus the cryptographic checksum data **Cryptographic checksum data **Cryptographic checksum data **Cryptographic checksum data **Cryptographic checksum ompute the cryptographic checksum **Triple-DES checksum. **Tr | | | • | | | • RSA Private Key: |
| Utilized to choose the algorithm and key for the security operation. *Data Object of the signed using RSA PKCS ¹⁷ #1 *Data Object of the signed data plus the digital signature Using RSA PKCS ¹⁷ #1 *Input data for generating the digital signature using RSA gignature using RSA pkcS#1. *Digital Signature *Digital Signature *Digital Signature *RSA Public Key: R, Digital Signature *Digital Signature *Note of the verification *RSA Private Key: R, X *Input data for generating the digital signature *Digital Signature *RSA Public Key: R, X *X *X *X *X *X *X *X | | | NOTE: The MSE service | | | - |
| Algorithm and key for the security operation. Status of the verification Status of the verificati | | | must have previously been | | | |
| Security operation. Security operation. Security operation. | | | utilized to choose the | | | |
| Verify Digital using RSA PKCS ¹⁷ #1 data plus the digital signature using RSA PKCS ¹⁷ #1 object of the signed data plus the digital signature PKCS private Key: Compute 2A Computes a digital signature signature using RSA digital signature object plus the digital signature object plus the cryptograph checksum object plus the cryptograph plus data or generating the digital signature object plus the cryptographic checksum data Compute 2A Computes an AES or Triple object plus the cryptographic checksum data Compute 2A Computes an AES or The data used to compute the fire the length of the cryptographic checksum Triple-DES checksum. Triple-DES checksum. The length of the cryptographic checksum | | | algorithm and key for the | | | |
| Digital Signature using RSA PKCS ¹⁷ #1 data plus the digital signature R, X Compute 2A Computes a digital signature signature using RSA digital signature using RSA pKCS#1. Verify 2A Performs AES or Triple Plaintext data object plus the cryptographic checksum data Checksum Compute 2A Computes an AES or Triple object plus the cryptographic checksum data Compute 2A Computes an AES or Trible object plus the cryptographic checksum data Compute 2A Computes an AES or Trible object plus the cryptographic checksum data Compute 2A Computes an AES or Trible object plus the cryptographic checksum data Cryptograp object plus the cryptographic checksum data Compute 2A Computes an AES or Trible data used to object plus the cryptographic checksum object plus the checksum object plus the cryptographic checksum object plus the ch | | | security operation. | | | |
| Signature Compute 2A Computes a digital signature using RSA digital signature PKCS#1. Verify 2A Performs AES or Triple object plus the cryptographic checksum Compute 2A Computes an AES or Triple object plus the cryptographic checksum Compute 2A Computes an AES or Triple object plus the cryptographic checksum data Compute 2A Computes an AES or object plus the cryptographic checksum data Compute 2A Computes an AES or object plus the cryptographic checksum data Cryptograp bic object plus the cryptographic checksum data Compute 2A Computes an AES or object plus the cryptographic checksum object plus the cryptographic checksum data Compute 2A Computes an AES or object plus the cryptographic checksum object | Verify | 2A | Verifies a digital signature | Data Object of the signed | Status of the verification | • RSA Private Key: |
| Compute 2A Computes a digital signature signature using RSA digital signature signature using RSA performs AES or Triple object plus the hic verification. Compute 2A Computes an AES or Triple object plus the cryptographic checksum data Checksum 2A Computes an AES or Triple object plus the cryptographic checksum data Compute 2A Computes an AES or Triple object plus the cryptographic checksum data Compute 2A Computes an AES or Triple object plus the cryptographic checksum data Cryptograp object plus the cryptographic checksum data Compute 2A Computes an AES or Triple-DES checksum. Triple-DES checksum. Compute the cryptographic checksum The length of the cryptographic checksum | Digital | | using RSA PKCS ¹⁷ #1 | data plus the digital signature | | R, X |
| Digital signature using RSA PKCS#1. Verify 2A Performs AES or Triple¬ Plaintext data object plus the verification. Checksum Compute 2A Computes an AES or Triple¬ • The data used to Triple¬ • The length of the Cryptographic checksum The length of the Cryptographic checksum Signature using RSA digital signature X X X X Status (e.g. 9000, 6300) • Symmetric Key: R,X • Cryptographic checksum data • Cryptographic checksum • Cryptographic checksum R,X • Cryptographic checksum R,X | Signature | | | | | |
| Signature PKCS#1. Verify 2A Performs AES or Triple Plaintext data object plus the cryptographic checksum data Checksum Compute 2A Computes an AES or Triple Other Cryptographic checksum. Cryptograp Other Cryptographic checksum data Triple-DES checksum. Triple-DES checksum. The length of the cryptographic checksum Cryptographic checksum Cryptographic checksum Triple-DES checksum. Cryptographic checksum | Compute | 2A | Computes a digital | • Input data for generating the | Digital Signature | • RSA Public Key: R, |
| Verify 2A Performs AES or Triple¬ Plaintext data object plus the cryptographic checksum data Checksum Compute 2A Computes an AES or Triple¬ • The data used to cryptographic checksum. Triple-DES checksum. The length of the cryptographic checksum Compute 2A Computes an AES or the data used to cryptographic checksum. The length of the cryptographic checksum • Status (e.g. 9000, 6300) • Symmetric Key: R,X Cryptographic checksum • Cryptographic checksum • Cryptographic checksum • R,X | Digital | | signature using RSA | digital signature | | X |
| Cryptograp hic verification. Checksum Compute 2A Computes an AES or Triple-DES checksum. Cryptographic checksum Compute bic Triple-DES checksum. The length of the cryptographic checksum Cryptographic checksum Object plus the cryptographic checksum data R,X Cryptographic checksum Cryptographic checksum Cryptographic checksum Cryptographic checksum Cryptographic checksum | Signature | | PKCS#1. | | | |
| hic verification. cryptographic checksum data Checksum Compute 2A Computes an AES or Triple-DES checksum. compute the hic The length of the cryptographic checksum Triple-DES checksum. cryptographic checksum Triple-DES checksum. cryptographic checksum | Verify | 2A | Performs AES or Triple¬ | Plaintext data | • Status (e.g. 9000, 6300) | Symmetric Key: |
| Checksum Compute 2A Computes an AES or • The data used to • Cryptographic checksum • Symmetric Key: Cryptograp | Cryptograp | | DES checksum | object plus the | | R,X |
| Compute 2A Computes an AES or • The data used to • Cryptographic checksum • Symmetric Key: Cryptograp hic Triple-DES checksum. The length of the cryptographic checksum | hic | | verification. | cryptographic checksum data | | |
| Cryptograp Triple-DES checksum. compute the hic The length of the cryptographic checksum | Checksum | | | | | |
| hic The length of the cryptographic checksum | Compute | 2A | Computes an AES or | • The data used to | Cryptographic checksum | Symmetric Key: |
| | Cryptograp | | Triple-DES checksum. | compute the | | R,X |
| Checksum checksum is 8 bytes. | hic | | The length of the | cryptographic checksum | | |
| | Checksum | | checksum is 8 bytes. | | | |

¹⁷PKCS -Public-Key Cryptography Standards

| Service | INS | Description | Input | Output | CSP and Type of |
|-------------|-----|------------------------------|--------------------------------|-----------------------------|----------------------|
| | | | | | Access |
| Create File | E0 | Creates a file | •File control parameters (data | • Status (e.g. 9000) | No CSPs are |
| | | | field) | | accessed via this |
| | | | •Length of data field | | service. |
| Delete File | E4 | Deletes a file and all files | •File ID | • Status (e.g. 9000) | No CSPs are |
| | | which exist within that | | | accessed via this |
| | | file | | | service. |
| Terminate | FE | Terminates all | •None | • None | No CSPs are |
| Card | | applications on the card | | | accessed via this |
| | | | | | service. |
| Install | E3 | This service is used to | •Encrypted PIN or Key data | • Status (eg. 9000, 6700, | • Kenc: W |
| Secret | | enter AES keys, | • "Final" secret or "Not | 6982, 6986, 6A8, 6A82, | • Kmac: W |
| | | Triple-DES keys, and | Final" secret flag | 6B00, 6CXX) | • Internal Auth key: |
| | | PINs. The keys which | | | W |
| | | may be entered are as | | | • External Auth key: |
| | | follows: | | | W |
| | | • Kenc | | | • Symmetric Key: W |
| | | • Kmac | | | • PIN: W |
| | | Internal Auth key | | | |
| | | • External Auth key | | | |
| | | Symmetric Key | | | |
| | | • PIN | | | |
| Update | E5 | Allows the updating of | •INIT_KEYs •Secret Key | • Status (eg. 9000, 6700, | Symmetric Key: W |
| Key | | the INIT_KEYs or secret | data | 6982, 6986, 6A8, 6A82, | • INIT_KEYenc: W |
| | | file keys. | •New error counter plus the | 6B00, 6CXX) | • INIT_KEYmac: W |
| | | | key value | | • Kenc: W • Kmac: W |
| | | | | | • Internal Auth key: |
| | | | | | W |
| | | | | | • External Auth key: |
| | | | | | W |
| Get File | 34 | Allows the reading of the | None | • FID list or "Nonexistent" | No CSPs are |
| List | | FID list of child files of | | • Status (eg. 9000, 6700, | accessed via this |
| | | the current file. | | 6982, 6986, 6A8, 6A82, | service. |
| | | | | 6B00, 6CXX) | |
| | | | | | |

 $^{^{^{16}}}DRBG-Deterministic\ Random\ Bit\ Generator$

| Service | INS | Description | Input | Output | CSP and Type of Access |
|------------|-----|------------------------|---------------------------|---------------------------|---------------------------|
| Read | B4 | Allows the output of a | • FID of the public key | Public Key data or | No CSPs are |
| Public Key | | public key | Public Key component read | "Nonexistent" | accessed via this |
| | | | parameter (Read all | • Status (eg. 9000, 6700, | service. |
| | | | component, read E | 6982, 6986, 6A8, 6A82, | |
| | | | component, or read N | 6B00, 6CXX) | |
| | | | component) | | |
| Import | E7 | Allows the input of an | Encrypted key data | • Status (eg. 9000, 6700, | • RSA key pair: W |
| RSA Key | | RSA key. | • FID of the RSA Key | 6982, 6986, 6A8, 6A82, | |
| | | | | 6B00, 6CXX) | |
| | | | | | |
| | | | | | |

2.4.2 User Role

This section provides a list of all services accessible to a User (Table 6). The list includes a full description of each service and, in addition, it describes the type of access that each service has to CSPs.

NOTE:

 \cdot R – Read: The CSP is read.

 \cdot W – Write: The CSP is established, generated, modified, or zeroized.

 \cdot X – Execute: The CSP is used within an Approved or allowed security function or authentication mechanism.

Table 6 - Mapping of User Role's Services to Inputs, Outputs, CSPs, and Type of Access

| Service | INS | Description | Input | Output | CSP and Type of |
|---------|-----|-------------------------|---------------------------------|------------------------------|----------------------|
| | | | | | Access |
| Read | В0 | Allows read access to a | Offset address of the binary | • File data or "Nonexistent" | No CSPs are accessed |
| Binary | | binary file. | file to read | • Status (e.g. 9000, 6283, | via this service. |
| | | | • Length of the data to be read | 6284, 6A80, 6A81, 6A82, | |
| | | | | 6A86, 6A87) | |

| Service | INS | Description | Input | Output | CSP and Type of |
|--------------|-----|----------------------------|---------------------------------|-----------------------------|---------------------|
| | | | | | Access |
| Read Record | B2 | Allows read access to a | •Record number | Record data or | No CSPs are |
| | | record. | •Read parameter (i.e, all | "Nonexistent" | accessed via this |
| | | | records starting at specified | • Status (e.g. 9000, 6283, | service. |
| | | | record number, or just one | 6284, 6A80, 6A81, 6A82, | |
| | | | record) | 6A86, 6A87) | |
| External | 82 | Authenticates an | Initiate a secure | • Status (e.g. 9000) | Initiate a secure |
| Authenticate | | external entity to the | session: | • Retry number for the | session: |
| | | cryptographic module. | •Authentication data of | referenced key incremented | • Kenc: R, X |
| | | This service may also be | external entity (32 bytes) plus | by one. | • Kmac: R, X |
| | | used to both authenticate | the MAC (8 bytes) | | • KSenc: W |
| | | and initiate a secure | | NOTE: If successful this | • KSmac :W |
| | | session with an external | OR | number is then reset to the | |
| | | entity. | | maximum. | Or |
| | | | •Authenticate only: | | |
| | | NOTE: Prerequisite to | •Algorithm type (AES, | | Authenticate Only: |
| | | this service is the use of | Triple-DES, RSA) | | • Symmetric key: R, |
| | | Get Challenge service. | •Key ID (Key Index) •Length | | X • RSA Private |
| | | The key as referenced | of data in the field | | Key: R, X |
| | | within the service call | •Authentication data (data | | |
| | | exists under the current | field) | | |
| | | file. | | | |
| Internal | 88 | Authenticates the | •Algorithm type (AES, | Authentication data | • Symmetric key: R, |
| Authenticate | | cryptographic module to | •Triple-DES, RSA) | • Status (e.g. 9000, 6300, | X |
| | | an external entity. | •Key ID (Key Index) •Length | 62CX, 6581, 6700, 6982, | • RSA Private Key: |
| | | | of data in the field •Random | 6984, 6A81, 6A2, 6A86, | R, X |
| | | NOTE: In order for this | data (data field) | 6A88) | |
| | | service to be utilized the | | | |
| | | external entity must have | | | |
| | | privileged access to the | | | |
| | | referenced key. | | | |
| Verify | 20 | Provides PIN | •Reference to the PIN | • Status (e.g. 9000, 6300, | • PIN: R, X |
| | | verification. | •PID | 62CX, 6581, 6700, 6982, | |
| | | | •Data to be verified | 6984, 6A81, 6A2, 6A86, | |
| | | NOTE: In order for this | | 6A88) | |
| | | service to be utilized the | | | |
| | | external entity must have | | | |
| | | privileged access to the | | | |
| | | referenced PIN. | | | |
| | | = 1 1 1 1 1 1 | | | |

| Service INS | | Description | Input | Output | CSP and Type of | |
|-------------|----|----------------------------|-----------------------------|---------------------------|---------------------|--|
| | | | | | Access | |
| Change | 24 | Modifies the PIN. | • Old PIN | •Status (e.g. 9000, | • PIN: R, W, X | |
| Reference | | | • New PIN | 6300, 62CX, 6581, | | |
| Data | | NOTE: In order for this | Reference to the password | 6700, 6982, 6984, 6A81, | | |
| | | service to be utilized the | • PID | 6A2, 6A86, 6A88) | | |
| | | external entity must have | | | | |
| | | privileged access to the | | | | |
| | | referenced PIN. | | | | |
| Reset Retry | 2C | Resets the retry counter | Reset parameter (resets | •Status (e.g. 9000, 6300, | No CSPs are | |
| Counter | | of the PIN to its initial | recount maximum number | 62CX, 6581, 6700, 6982, | accessed via this | |
| | | value. | and remaining count to | 6984, 6A81, 6A2, 6A86, | service. | |
| | | | default) | 6A88) | | |
| | | NOTE: Utilization of this | Restore parameter (restores | | | |
| | | service requires | recount to initial default | | | |
| | | permission to modify | value) | | | |
| | | PIN. | Reference to PIN | | | |
| | | | • PID | | | |
| Generate | 46 | Generates an asymmetric | Key parameter information • | •Status (e.g. 9000, 6300, | • RSA Private Key: | |
| Asymmetric | | key pair. | Algorithm ID | 62CX, 6581, 6700, 6982, | W | |
| Key Pair | | | Modulus Length | 6984, 6A81, 6A2, 6A86, | • RSA Public Key: | |
| | | | Private Key File Identifier | 6A88) | W | |
| | | | (FID) | | • DRBG Seed: R,W, | |
| | | | | | X | |
| Encrypt | 2A | Performs an encrypt | Plaintext data | •Ciphertext data Status | • Symmetric key: R, | |
| | | operation using an | | (e.g. 9000, 6300, 62CX, | X | |
| | | Approved security | | 6581, 6700, 6982, 6984, | • RSA Public Key: | |
| | | function. | | 6A81, 6A2, 6A86, 6A88) | R, X | |
| | | NOTE: The MSE service | | | | |
| | | must have previously | | | | |
| | | been utilized to chose the | | | | |
| | | algorithm and key for the | | | | |
| | | security operation. | | | | |
| Decrypt | 2A | Performs a decrypt | • Ciphertext | •Plaintext | • Symmetric key: R, | |
| | | operation. | | | X | |
| | | NOTE: The MSE service | | | • RSA Private Key: | |
| | | must have previously | | | R, X | |
| | | been utilized to chose the | | | | |
| | | algorithm and key for the | | | | |
| | | security operation. | | | | |
| | | | | | | |

| Service | INS | Description | Input | Output | CSP and Type |
|-----------------|-----|----------------------------|----------------------------|--------------------------------|-------------------|
| | | | | | of Access |
| Verify Digital | 2A | Verifies a digital | •Data Object of the signed | •Status of the verification | • RSA Public |
| Signature | | signature using RSA | data plus the digital | | Key: R, X |
| | | PCKS#1. | signature | | |
| Compute Digital | 2A | Computes a digital | •Input data for generating | •Digital Signature | • RSA Private |
| Signature | | signature using RSA | the digital signature | | Key: R, X |
| | | PCKS#1. | | | |
| Verify | 2A | Performs and AES | •Plaintext data object | •Status (e.g. 9000, | Symmetric |
| Cryptographic | | or Triple-DES | plus the cryptographic | 6300) | Key: R, |
| Checksum | | checksum verification | checksum data | | X |
| Compute | 2A | Performs an AES or | •The data used to | •Cryptographic | Symmetric |
| Cryptographic | | Triple-DES | compute the | checksum | Key: R, X |
| Checksum | | checksum. The length | cryptographic checksum | | |
| | | of the checksum is 8 | | | |
| | | bytes. | | | |
| Get File List | 34 | This command is used | •None | •FID list or "Nonexistent" | No CSPs are |
| | | to read the FID list of | | •Status (eg. 9000, 6700, 6982, | accessed via this |
| | | child files of the current | | 6986, 6A8, 6A82, 6B00, | service. |
| | | file. | | 6CXX) | |
| Read Public Key | B4 | Allows the output of a | •FID of the public key | •Public Key data or | No CSPs are |
| | | public key. | •Public Key component | "Nonexistent" | accessed via this |
| | | | read parameter (Read all | •Status (eg. 9000, 6700, 6982, | service. |
| | | | component, read E | 6986, 6A8, 6A82, 6B00, | |
| | | | component, or read N | 6CXX) | |
| | | | component) | | |
| Import RSA Key | E7 | Allows the input of an | •Encrypted key data | •Status (eg. 9000, 6700, 6982, | • RSA key pair: |
| | | RSA key. | •FID of the RSA Key | 6986, 6A8, 6A82, 6B00, | W |
| | | | | 6CXX) | |

2.4.3 Additional Services

The module provides a limited amount of services for which the operator does not have to assume an authorized role. Table 7 provides the list of services for which the operator is not required to assume an authorized role. The list includes a full description of each service and, in addition, it describes the type of access that each service has to CSPs. None of the services listed in the table disclose cryptographic keys and CSPs or otherwise affect the security of the module

NOTE:

- R Read: the CSP read.
- W Write: The CSP is established, generated, modified, or zeroized.
- X Execute: The CSP is used within an Approved or allowed security function or authentication mechanism.

Table 7 – Mapping of Unauthenticated Services to Inputs, Outputs, CSPs, and Type of Access

| Service INS | | Description | Input | Output | CSP and Type | |
|-------------|----|-------------------------------|--------------------------|---------------------------|-------------------|--|
| | | | | | of Access | |
| Put Data | DA | Allows data to be received | • Data object tag ('81' | •Status (e.g. 9000, 6283, | No CSPs are | |
| | | and stored by the | which indicates OEM | 6284, 6A80, 6A81, 6A82, | accessed via this | |
| | | cryptographic module. In the | info, followed by up to | 6A86, 6A87) | service. | |
| | | Put Data service, only the | 32 bits of OEM info. | | | |
| | | OEM information is allowed | • Length of object data | | | |
| | | to be set. | | | | |
| Get Data | CA | This service allows data to | • Data object tag (e.g., | •Content of object | No CSPs are | |
| | | be retrieved. Data refers to | '80' which indicates | •Status (e.g. 9000, 6283, | accessed via this | |
| | | global data, which belongs to | card serial number) | 6284, 6A80, 6A81, 6A82, | service. | |
| | | the cryptographic module, | | 6A86, 6A87) | | |
| | | such as the serial number, | | | | |
| | | OEM information, chip | | | | |
| | | information which includes | | | | |
| | | algorithm support, RAM | | | | |
| | | size. | | | | |
| Get | 84 | Requests a random value that | • None | •Random value | • DRBG Key | |
| Challenge | | will be used as a challenge | | •Status (e.g. 9000, 6283, | Value: R, W, X | |
| | | within the External | | 6284, 6A80, 6A81, 6A82, | • DRBG 'V' | |
| | | Authenticate service. | | 6A86, 6A87) | Value: R; W, X | |
| Manage | 22 | Prepares the | • CRDO ¹⁹ | •Status (e.g. 9000, | No CSPs are | |
| Security | | cryptographic | Algorithm Reference | 6300, 62CX, 6581, | accessed via | |
| Environment | | module for the | Key Reference | 6700, 6982, 6984, | | |
| (MSE) | | subsequent commands, SET, | • File Reference | 6A81, 6A2, 6A86, 6A88) | | |
| | | STORE, RESTORE, SEID, | • Length of CRDOs | | | |
| | | and ERASE. | | | | |
| Select | A4 | Allows the selection of a | File identifier | •File control information | No CSPs are | |
| | | specified file. | Dedicated file Name | •Status (e.g. 9000, 6283, | accessed via | |
| | | | File path starting at | 6284, 6A80, 6A81, 6A82, | this service. | |
| | | | master file | 6A86, 6A87) | | |
| | | | File path starting at | | | |
| | | | dedicated file | | | |

¹⁹CRDO – Control Reference Data Object

| Service | INS | Description | Input | Output | CSP and Type of Access |
|---------|-----|----------------------------------|-------------------------|----------------------|--------------------------|
| Manage | 70 | Allows the assignment; | Number of logical | • Status (e.g. 9000, | No CSPs are accessed via |
| Channel | | opening, and closing of a | channel to be assigned, | 6283, 6284, 6A80, | this service. |
| | | logical channel. A logical | opened, or closed | 6A81, 6A82, 6A86, | |
| | | channel is a logical link | (01-03). | 6A87) | |
| | | between the host system and | | | |
| | | a file on the smart card. | | | |
| Hash | 2A | Performs a hash using | Input data | Hash result or None | No CSPs are accessed via |
| | | SHA ²⁰ -1 or SHA-256. | | | this service. |

2.5 Physical Security

The ST3 ACE Token is a multi-chip standalone cryptographic module as defined by FIPS 140-2 and is designed to meet Level 3 physical security requirements.

The ST3 ACE Token is a made of a completely hardened, production-grade polycarbonate. The colored polycarbonate obscures a clear view of the hardware components within. There is a removable cap that reveals the plastic USB connector and a hard, non-malleable metal casing surrounding the USB connector. The USB connector is made of hard production-grade, black plastic.

The coloring of the module obscures any visible writing on the PCB. The visible critical components within the module are further covered to meet FIPS 140-2 level 3 physical security requirements. The ST23YT66 microcontroller is covered with a black, opaque, tamper-resistant, epoxy encapsulate, thus completely covering all critical cryptographic components from plain view. All other non-critical viewable components are unmarked and unidentifiable. The USB connector located outside of the plastic casing of the USB token is made of a hardened, production grade plastic and prevents access to the rest of the USB token.

Any attempt at removal or penetration of the plastic enclosure has a high probability of causing serious damage to the module and the hardware components within the enclosure, which will reveal clear tamper evidence. Removal of the metal surrounding the USB connector will result in the physical damage of the USB connector and its associated pins, rendering the entire cryptographic module useless. If the USB connector is exposed, there is no power going to the USB token. Once power is removed from the cryptographic module, all plaintext keys and unprotected CSPs are zeroized.

2.6 Operational Environment

The operational environment for the ST3 ACE Token includes the ST23YT66 microcontroller containing an 8/16-bit ST23 CPU. The token's operational environment is non-modifiable and does not possess a general purpose operating system.

2.7 Cryptographic Key Management

The module implements the FIPS-Approved algorithms show in Table 8:

Table 8 – FIPS-Approved Algorithm Implementations

| Algorithm | Certificate Number |
|--|--------------------|
| AES in ECB ²¹ , CBC ²¹ modes using 128-bit key sizes | 1473 |
| Triple-DES in ECB, CBC modes using Keying | 991 |
| Option 1 | |
| RSA PKCS#1 v1.5 signature generation—using | 720 |
| 2048-bit keys | |
| RSA PKCS#1 v1.5 signature verification – using | 720 |
| 1024-and 2048bit keys | |
| ANSI ²³ X9.31 Key Pair Generation | 720 |
| SHA-1 and SHA-256 | 1332 |
| SP24 800-90 CTR25_DRBG | 58 |

Caveat:

Additional information concerning SHA-1 and specific guidance on transitions to the use of stronger cryptographic keys and more robust algorithms is contained in NIST Special Publication 800-131A.

Table 9 lists the non-Approved algorithms implemented in the module which are allowed in a FIPS-Approved mode of operation.

Table 9 - FIPS-Allowed Algorithm Implementations

| Algorithm |
|---|
| Non-Deterministic Random Number Generator (NDRNG) |
| RSA PKCS#1v1.5 2048-bit (Key establishment methodology provides 112 bits of security; |
| non-compliant less than 112 bits of encryption strength) |

²¹ ECB –Electronic Codebook

²² CBC – Cipher-Block Chaining

²³ ANSI – American National Standards Institute

²⁴ SP – Special Publication

²⁵ CTR – Counter

The module supports the critical security parameters (CSPs) listed below in Table 10. Internally generated keys are generated following scenario 1 of Implementation Guidance number 7.8.

Table 10 - List of Cryptographic Keys, Cryptographic Key Components, and CSPs

| Key | Key Type | Use | Generation / | Output | Storage | Zeroization | Key To |
|---------------|--------------|-------------------|-----------------|---------------|---------------|----------------|----------------|
| | | | Input | | | | Entity |
| Symmetric | AES | These keys are | Generation: | N/A: The | These keys | Procedurally | Storage: 4-bit |
| Key | 128-bit key; | used to | This key is not | module does | are stored in | overwrite | key ID |
| | Triple-DES | encrypt/decrypt | generated | not support | EEPROM26 in | keys with | |
| | 168-bit Key | data, or within a | within the | the output of | special files | arbitrary data | Input/Output: |
| | | symmetric MAC | module. | this key. | used to store | using the | This key is |
| | | algorithm to | | | symmetric | Update Key | associated |
| | | generate | Input: This | | keys and | service. | with the |
| | | authentication | key may be | | PINs. | | Crypto-Office |
| | | data. | input encrypted | | | | r role during |
| | | | within a secure | | | | Input. |
| | | | channel. | | | | |
| Internal Auth | AES | These keys are | Generation: | N/A: The | These keys | Procedurally | Storage: 4-bit |
| Key | 128-bit | used to | This key is not | module does | are | overwrite | key ID |
| | key; | authenticate the | generated | not support | stored in | keys with | |
| | Triple-DES | module to an | within the | the output of | EEPROM in | arbitrary data | Input/Output: |
| | 168-bit Key | external entity. | module. | this key. | special files | using the | This key is |
| | | | | | used to store | Update Key | associated |
| | | | Input: This | | symmetric | service. | with the |
| | | | key may be | | keys and | | Crypto-Office |
| | | | input encrypted | | PINs. | | r role during |
| | | | within a secure | | | | Input |
| | | | channel. | | | | |

²⁶ EEPROM - Electronically Erasable Programmable Read-Only Memory

| Key | Key Type | Use | Generation / | Output | Storage | Zeroization | Key To |
|-------------------------|--------------|-----------------------------|-------------------|---------------|---------------|----------------|----------------|
| | | | Input | | | | Entity |
| External Auth | AES 128-bit | These keys are | Generation: | N/A: The | These keys | Procedurally | Storage: 4-bit |
| Key | key; | used to modify | This key is not | module does | are | overwrite | key ID |
| | Triple-DES | the security | generated | not support | stored in | keys with | |
| | 168-bit Key; | state of the | within the | the output of | EEPROM in | arbitrary data | Input/Output: |
| | RSA 2048-bit | currently | module. | this key. | special files | using the | This key is |
| | key | selected DF ²⁷ . | | | used to store | Update Key | associated |
| | | | Input: This | | symmetric | service. | with the |
| | | | key may be | | keys and | | Crypto-Office |
| | | | input encrypted | | PINs. | | r role during |
| | | | within a secure | | | | Input |
| | | | channel. | | | | |
| INIT_KEY _{enc} | AES 128-bit | This key is | Generation: | N/A: The | This key is | Procedurally | Storage: 4-bit |
| | key | used to derive a | This key is not | module does | stored under | overwrite key | key ID |
| | | session key | generated | not support | in the | with arbitrary | |
| | | which is then | within the | the output of | reserved file | data using the | Input/Output: |
| | | used to | module. It is a | this key. | in EEPROM. | Update Key | N/A |
| | | encrypt/decrypt | factory-set key | | | service. | |
| | | data over a | which is used | | | | |
| | | secure session | only in the | | | | |
| | | between an | initialized state | | | | |
| | | authorized | of the module. | | | | |
| | | external entity | | | | | |
| | | and the | Input: This | | | | |
| | | module. | key is | | | | |
| | | | factory-set and | | | | |
| | | | cannot be | | | | |
| | | | modified or | | | | |
| | | | input outside | | | | |
| | | | of | | | | |
| | | | manufacturing. | | | | |

²⁷ DF – Dedicated File

| Key | Key Type | Use | Generation / | Output | Storage | Zeroization | Key To |
|-------------------------|-------------|------------------|-------------------|---------------|-----------------|----------------|----------------|
| | | | Input | | | | Entity |
| INIT_KEY _{mac} | AES | This key is used | Generation: | N/A: The | This key is | Procedurally | Storage: 4-bit |
| | 128-bit key | to derive a | This key is not | module does | stored under | overwrite keys | key ID |
| | | session key | generated | not support | in the reserved | with arbitrary | |
| | | which is then | within the | the output of | file in | data using the | Input/Output: |
| | | used to | module. It is a | this key. | EEPROM. | Update Key | N/A |
| | | authenticate an | factory-set key | | | service. | |
| | | operator or data | which is used | | | | |
| | | over a secure | only in the | | | | |
| | | session between | initialized state | | | | |
| | | an authorized | of the module. | | | | |
| | | external entity | | | | | |
| | | and the module. | Input: This | | | | |
| | | | key is | | | | |
| | | | factory-set and | | | | |
| | | | cannot be | | | | |
| | | | modified or | | | | |
| | | | input outside of | | | | |
| | | | manufacturing. | | | | |
| K _{enc} | AES | This key is used | Generation: | N/A: The | These keys | Procedurally | Storage: 4-bit |
| | 128-bit key | to derive a | This key is not | module does | are stored | overwrite keys | key ID |
| | | session key | generated | not support | index 0x00 of | with arbitrary | |
| | | which is then | within the | the output of | the currently | data using the | Input/Output: |
| | | used to | module. | this key. | selected DF. | Update Key | N/A |
| | | encrypt/decrypt | | | | service. | |
| | | data over a | Input: This | | | | |
| | | secure session | key may be | | | | |
| | | between an | input encrypted | | | | |
| | | authorized | within a secure | | | | |
| | | external entity | channel. | | | | |
| | | and the module. | _ | | | | |

| Key | Key Type | Use | Generation / Input | Output | Storage | Zeroization | Key To Entity |
|-------------------|--------------------|---|--|--|--|--|--|
| K _{mac} | AES 128-bit key | This key is used to derive a session key which is then used to authenticate an operator or data over a secure session between an authorized external entity and the module. | Generation: This key is not generated within the module. Input: This key may be input encrypted within a secure channel. | N/A: The module does not support the output of this key. | These keys are stored index 0x00 of the currently selected DF. | Procedurally overwrite keys with arbitrary data using the Update Key service. | Storage: 4 bit key ID Input/Output: N/A |
| KS _{enc} | AES 128-bit key | This key is used to encrypt/decryp t data over a secure session. | Generation: Generated from the INIT_KEYenc or Kenc key as part of the Secure Channel Protocol v01 as specified within Global Platform v2.1. Input: This key cannot be input. | N/A: The module does not support the output of this key. | Stored in module RAM. | Power cycle the module. | Storage: This key is associated with a logical channel ID (0-3) for which it is being used to secure messaging. Input/Output: N/A, this key is not output |

| Key | Key Type | Use | Generation / | Output | Storage | Zeroization | Key To |
|-------------------|-------------|-------------------|----------------------------|---------------|-------------|----------------|----------------|
| | | | Input | | | | Entity |
| KS _{mac} | AES | This key is used | Generation: | N/A: The | Stored in | Power cycle | Storage: This |
| | 128-bit key | to authenticate | Generated from | module does | module RAM. | the module. | key is |
| | | data over a | the | not support | | | associated |
| | | secure session. | INIT_KEY _{mac} | the output of | | | with a logical |
| | | | or K _{mac} key as | this key. | | | channel ID |
| | | | part of the | | | | (0-3) for |
| | | | Secure Channel | | | | which it is |
| | | | Protocol v01 as | | | | being used to |
| | | | specified | | | | secure |
| | | | within Global | | | | messaging. |
| | | | Platform v2.1. | | | | |
| | | | | | | | Input/Output: |
| | | | Input: This | | | | N/A, this key |
| | | | key cannot be | | | | is not output |
| | | | input. | | | | |
| Personal | 6-16 byte | This key is used | Generation: | N/A: The | EEPROM in | Procedurally | Storage: 4-bit |
| Identification | secret | to modify the | This key is not | module does | plaintext | overwrite keys | key ID |
| Number (PIN) | | security state of | generated | not support | | with arbitrary | |
| | | the currently | within the | the output of | | data using the | |
| | | selected DF. | module. | this key. | | Update Key | |
| | | | | | | service. | |
| | | | Input: This | | | | |
| | | | key may be | | | | |
| | | | input encrypted | | | | |
| | | | within a secure | | | | |
| | | | channel. | | | | |

| key is used ecrypt or by data. | Input Generation: This key is generated using | N/A: The | EEPROM in | Procedurally | Entity |
|--------------------------------|---|--|---|--|---|
| ecrypt or | This key is | | EEPROM in | Procedurally | |
| | _ | 1 1 1 | | 1 Toccuurany | Storage: 4-bit |
| y data. | generated using | module does | plaintext | overwrite keys | File ID |
| | generated using | not support | | with arbitrary | |
| | the Approved | the output of | | data using the | NOTE: Only |
| | SP800-90 | this key. | | Import RSA | one RSA |
| | DRBG. | | | Key service. | Private key |
| | | | | | may be stored |
| | Input: This | | | | in an RSA |
| | key may be | | | | Private Key |
| | input encrypted | | | | file. |
| | within a secure | | | | |
| | channel. | | | | |
| key is used | Generation: | Output in | EEPROM in | N/A: this key | Storage: 4-bit |
| erypt or | This key is | plaintext | plaintext | is a public key | File ID |
| data. | generated using | using the | | and therefore | |
| | the Approved | Read Public | | does not have | NOTE: Only |
| | SP800-90 | key | | to be zeroized. | one RSA |
| | DRBG. | command. | | | Public key |
| | | | | | may be stored |
| | Input: This | | | | in an RSA |
| | key may be | | | | Public Key |
| | input encrypted | | | | file. |
| | within a secure | | | | |
| | channel. | | | | |
| l for SP | Internally | Never | Plaintext in | Power Cycle | Associated |
| 90 | Generated | | volatile | | with an |
| _DRBG | | | memory | | internal |
| | | | | | module |
| | | | | | variable |
| l for SP | Internally | Never | Plaintext in | Power Cycle | Associated |
| 90 | Generated | | volatile | | with an |
| _DRBG | | | memory | | internal |
| | | | | | module |
| | | | | | variable |
| | I for SP J for SP J for SP J for SP | Input: This key may be input encrypted within a secure channel. key is used crypt or data. Generation: This key is generated using the Approved SP800-90 DRBG. Input: This key may be input encrypted within a secure channel. I for SP Internally Generated I for SP Generated I for SP Generated | Input: This key may be input encrypted within a secure channel. key is used compared using the the Approved SP800-90 DRBG. Input: This key may be input encrypted within a secure channel. If or SP Internally Senerated If or SP Generated Input: This Never Generated Internally Never Generated Internally Never Generated | Input: This key may be input encrypted within a secure channel. Rey is used corrected using the Approved SP800-90 DRBG. Input: This key may be input encrypted within a secure channel. Input: This key may be input encrypted within a secure channel. If for SP Internally Senerated Sen | Input: This key may be input encrypted within a secure channel. key is used crypt or This key is generated using the Approved SP800-90 key DRBG. Input: This key may be input encrypted within a secure channel. Input: This key may be input encrypted within a secure channel. If or SP Internally Never Plaintext in volatile memory I for SP Internally Never Plaintext in Power Cycle of Generated volatile Power Cycle of Generated volatile |

2.8 EMI/EMC

The ST3 ACE Token conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B (i.e., for home use).

2.9 Self-Tests

Self-tests are performed by the ST3 ACE Token when running in a FIPS-Approved mode of operation. The module will run power-up self-tests when first powered up. The module will run conditional self-tests before a random number is generated or when signing and verifying data.

The module supports only one error condition, referred to as the FIPS Error State. Any failure of a FIPS self-test will cause the module to enter the FIPS error state, which does not allow for any data output and/or cryptographic service usage. If an operator attempts to utilize any module services, the service will not be invoked and status output will be provided via the return value of the APDU. The status output provided in the APDU response packet will be '6F 00'. In order to transition out of the FIPS error state, the module must be power-cycled.

2.9.1 Power-Up Self-Tests

The ST3 ACE Token performs the following self-tests at power-up:

- Cryptographic Known Answer Tests (KATs)
 - AES Encrypt KAT
 - AES Decrypt KAT
 - Triple-DES Encrypt KAT
 - Triple-DES Decrypt KAT
 - SHA-1 KAT
 - SHA-256 KAT
 - RSA signature generation/verification KAT
 - DRBG KAT

2.9.2 Conditional Self-Tests

The module performs the following conditional self-tests:

- Continuous Random Number Generator test for both the NDRNG and the SP800-90 DRBG.
- RSA pairwise consistency test for sign/verify and encrypt/decrypt

2.10 Mitigation of Other Attacks

This section is not applicable. The module is not intended to mitigate any attacks beyond the FIPS 140-2 Level 3 requirements for this validation.

The ST3 ACE Token meets Level 3 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-approved mode of operation.

3 SECURE OPERATIONS

3.1 Detecting a FIPS Cryptographic Module

The SECUREMETRIC ST3 ACE Token is shipped as a FIPS token that is already operating in a FIPS-approved mode of operation. It is not possible to change the configuration of the token to operate outside of its shipped configuration. To determine if the token is a FIPS token, the Cryptographic Officer should check for a laser-etched "FIPS" on the token casing, located at the top of the token near the USB connector. Please refer to Figure 3 for the location of the "FIPS" label.



Figure 3 -"FIPS" Label Location

Another way to determine whether the ST3 ACE Token is a FIPS token is by executing the supplied "FIPS-Mode-Detect" tool. After inserting the module into an available USB slot, start up the tool and hit the "Detect" button. If the tool reports "FIPS", that means the module is configured to operate as a FIPS token. See Figure 4 for a screen shot of the "FIPS-Mode-Detect" tool.



Figure 4 -"FIPS-Mode-Detect" Tool

3.2 Initial Setup

The module is delivered with a pair of AES Keys (INIT_KEY_{enc} and INIT_KEY_{mac}) to allow authentication and secure initialization of the module. All communications to initialize the module will require a secure session using this key pair which will encrypt and authenticate all data input.

For additional information regarding module initialization, please refer to the ST3 ACE Token User Manual.

3.2.1 Zeroization

In the case that zeroization is required, the Crypto-Officer shall obtain possession of the module and then maintain sole physical possession of the cryptographic module until all keys have been zeroized. The Crypto-Officer performs zeroization by procedurally overwriting all of the keys with arbitrary data using the Update Key service.

3.3 Non-Approved Mode

The ST3 ACE Token ships as a FIPS module and is meant to always operate in FIPS-Approve mode of operation. The module provides access to non-Approved security functions which use non-Approved algorithms and key sizes. Use of these services transitions the module to the non-Approved mode through the duration of the service being performed. Table 11 lists the non-Approved services and associated algorithms and key sizes.

Table 11 - Non-Approved Services

| Non Approved Service | Algorithm |
|-----------------------|--------------------|
| Signature Generation | RSA 1024-bit SHA-1 |
| Encryption/Decryption | Triple-DES (2-key) |
| Key Establishment | RSA 1024-bit |

4 ACRONYMS

Table 12 defined the acronyms used in this Security Policy.

Table 12- Acronyms

| Acronym | Definition |
|---------|---|
| AES | Advanced Encryption System |
| APDU | Application Protocol Data Unit |
| ANSI | American National Standards Institute |
| API | Application Programming Interface |
| CBC | Cipher Block Chaining |
| CLA | Class Byte |
| CMVP | Cryptographic Module Validation Program |
| COS | Chip Operating System |
| CPU | Core Processing Unit |
| CRC | Cyclic Redundancy Check |
| CRDO | Control Reference Data Objects |
| CSEC | Communications Security Establishment Canada |
| CSP | Critical Security Parameter |
| CTR | Counter |
| DC | Direct Current |
| DES | Digital Encryption Standard |
| DF | Dedicated File |
| DSA | Digital Signature Algorithm |
| DRBG | Deterministic Random Bit Generator |
| ECB | Electronic Codebook |
| EEPROM | Electronically Erasable Programmable Read-Only Memory |
| EMC | Electromagnetic Compatibility |
| EMI | Electromagnetic Interference |
| FID | File Identification |
| FIPS | Federal Information Processing Standard |
| HMAC | (Keyed-) Hash Message Authentication Code |
| IC | Integrated Circuit |
| IEC | International Electrotechnical Commission |
| INS | Instruction Byte |
| ISO | International Organization for Standardization |

| Acronym | D efinition |
|---------|---|
| KAT | Known Answer Test |
| LED | Light Emitting Diode |
| MAC | Message Authentication Code |
| MSE | Manage Security Environment |
| NDRNG | Non-Deterministic Random Number Generator |
| NIST | National Institute of Standards and Technology |
| NVLAP | National Voluntary Laboratory Accreditation Program |
| OEM | Original Equipment Manufacturer |
| PCB | Printed Circuit Board |
| PID | Personal Identification number index |
| PIN | Personal Identification Number |
| PKCS | Public Key Cryptography Standards |
| RAM | Random Access Memory |
| RNG | Random Number Generator |
| RSA | Rivest Shamir and Adleman |
| SHA | Secure Hash Algorithm |
| SP | Special Publication |
| TCP | Transmission Control Protocol |
| USB | Universal Serial Bus |
| V | Volt |
| VCC | Common Collector Voltage |

Prepared by:



SecureMetric Technology Sdn. Bhd.

KUALA LUMPUR (HQ) 2-2, Incubator 2, Technology Park Malaysia, Lebuhraya Sg Besi - Puchong, Bukit Jalil, 57000 Kuala Lumpur, Malaysia T +603 8996 8225 F +603 8996 7225

JAKARTA SINGAPORE HANOI HO CHI MINH CITY MANILA













