HYPERSECU®

FIPS Security Level 3

For HyperPKI™ HYP2003
Hardware Version 1.0.0

FIPS 140-2 Non-Proprietary Security Policy

06/29/2017

HSTE-NB0042-RV 1.1

HYPERSECU INFORMATION SYSTEMS INC

200-6191 Westminster Hwy, Richmond BC, V7C 4V4 Canada
1-604-279-2000 | hypersecu.com

Table of Contents

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the HyperPKI™ HYP2003 token from Hypersecu Information Systems Inc. This Security Policy describes how the HyperPKI HYP2003 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 HyperPKI™ HYP2003 token is referred to in this document as HYP2003 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 Hypersecu Information Systems Inc website (www.hypersecu.com) contains information on the full line of products from Hypersecu Information Systems Inc.
- 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.

2.1 Overview

Hypersecu Information Systems Inc 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 Hypersecu Information Systems Inc's continued leadership and innovation is demonstrated within this Security Policy, which specifies their First FIPS 140-2 validated cryptographic module. This new module, referred to as the HYP2003 token, is a USB [1] token containing Hypersecu Information Systems Inc's own Hypersecu-FIPS-COS cryptographic operating system. The Hypersecu-FIPS-COS is embedded in an ST23YT66 Integrated Circuit (IC) chip and has been developed to support Hypersecu Information Systems Inc's HYP2003 USB token (Figure 1). The HYP2003 token is designed to provide strong authentication and identification and to support network login, secure online transactions, digital signatures, and sensitive data protection. Hypersecu Information Systems Inc's HYP2003 token guarantees safety of its cryptographic IC chip and other components with its hard, semi-transparent, polycarbonate shell.

Figure 1: HYP2003 token

[1] USB – Universal Serial Bus

The HYP2003 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
5	Physical Security	3
6	Operational Environment	N/A
7	Cryptographic Key Management	3
8	EMI/EMC[2]	3
9	Self-tests	3
10	Design Assurance	3
11	Mitigation of Other Attacks	N/A

2.2 Module Specification

The HYP2003 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 HYP2003 token are defined by the hard, semi-transparent, polycarbonate casing of the USB token. The HYP2003 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 HYP2003 token. All cryptographic functions and firmware are stored within the microcontroller package and executed by an 8/16-bit ST23 CPU (Core Processing Unit). An LED[3] 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 HYP2003 token.

[2] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility
[3] LED – Light Emitting Diode

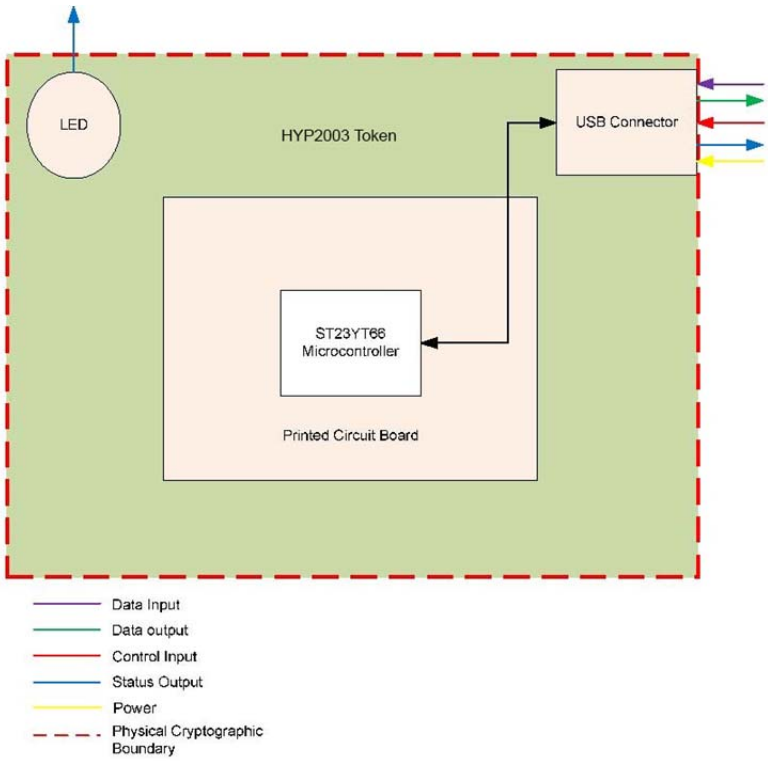

Figure 2: Physical Cryptographic Boundary

The HYP2003 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 HYP2003 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-), VCC[4], 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[5] DC[6] power input from whatever source the USB connector is plugged into.
- 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 Mechanism	Authentication Data	Authentication Mechanism
Identity-based	128-bit AES[7] Key Shared Secret	The AES key is 128 bits in length. The probability that a random 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 Secret	Each Triple-DES key is effectively 56 bits in length, resulting in a 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

[4] VCC – Common Collector Voltage
[5] V -Volt
[6] DC – Direct Current
[7] AES – Advanced Encryption Standard

		less than 1/100,000.
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 HYP2003 token are implemented in accordance with ISO[8]/IEC[9] 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

Header		**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

[8] ISO – International Organization for Standardization
[9] IEC – International Electrotechnical Commission

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)

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 CSP[10].

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 Binary	B0	Allows read access to a binary file. A binary file is a file whose content is a sequential string of bits.	• Offset address of the binary file to read • Length of the data to be read	File data or "Nonexistent" Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.
Update Binary	D6	Allows write access to a binary file.	• Offset address of the binary file to read • Length of the data to be read	Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.
Read Record	B2	Allows read access to a record. A record is a type of data storage structure as defied	• Record number • Read parameter (i.e, all records starting at specified record	Record data or "Nonexistent" Status (e.g. 9000, 6283, 6284,	No CSPs are accessed via this service.

[10] CSP – Critical Security Parameter

		within ISO 7816. Records are stored in files.	number, or just one record)	6A80, 6A81, 6A82, 6A86, 6A87)	
Update Record	DC	Allows write access to a record.	• Record number • Length of record • Record data • Read parameter (i.e, update the record specified by the record number)	Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.
Append Record	E2	Allows a record to be appended	• Record number • Current file • Length of record • Record data • Read parameter (i.e, update the record specified by the record number)	Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.
External Authenticate	82	Authenticates an external entity to the cryptographic module. This service may also be used to both authenticate and initiate a secure session with an external entity. **NOTE:** Prerequisite to this service is the use of Get Challenge service. The key as referenced within the service call exists under the current file.	• Initiate a secure session: Authentication data of external entity (32 bytes) plus the MAC[11] (8 bytes) Or • Authenticate only: Algorithm type (AES, Triple-DES[12] , RSA[13]) Key ID (Key Index) • Length of data in the field • Authentication data (data field)	Status (e.g. 9000) Retry number for the referenced key incremented by one. **NOTE:** If successful, this number is then reset to the maximum.	Initiate a secure session: $INIT\_KEY_{enc}$: R, X $INIT\_KEY_{mac}$: R, X K_{enc}: R, X K_{mac}: R, X KS_{enc}: W KS_{mac} :W **Or** Authenticate Only: Symmetric key: R, X RSA Private Key: R, X
Internal Authenticate	88	Authenticates the cryptographic module to an external entity **NOTE:** In order for this service to be utilized, the external entity must have privileged access to the referenced key.	• Algorithm type (AES, Triple-DES, RSA) • Key ID (Key Index) • Length of data in the field • Random data (data field)	Authentication data Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	Authenticate Only: Symmetric key: R, X RSA Private Key: R, X

[11] MAC – Message Authentication Code
[12] DES – 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. **NOTE:** In order for this service to be utilized, the external entity must have privileged access to the referenced PIN.	• Reference to the PIN/PID[15] • Data to be verified	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	PIN: R, X
Change Reference Data	24	Modify the PIN **NOTE:** In order for this service to be utilized the external entity must have privileged access to the referenced PIN.	• Old PIN • New PIN • Reference to the PIN/PID	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	PIN: R, W, X
Enable Verification Requirement	28	Modifies a PIN's state from invalid to valid. **NOTE:** Utilization of this service requires permission to activate the PIN.	Reference to the PIN/PID	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	No CSPs are accessed via this service.
Disable Verification Requirement	26	Modifies a PINs state from valid to invalid. **NOTE:** Utilization of this service requires permission to invalidate the PIN.	Reference to the PIN/PID	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	No CSPs are accessed via this service.

[14] PIN – Personal Identification Number
[15] PID – Personal Identification number index

Reset Retry Counter	2C	Resets the retry counter of the PIN to its initial value. **NOTE:** Utilization of this service requires permission to modify PIN.	• Reset parameter (resets recount maximum number and remaining count to default) • Restore parameter (restores recount to initial default value) • Reference to PIN/PID	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	No CSPs are accessed via this service.
Generate Asymmetric Key Pair	46	Generates an Asymmetric key pair	• Key parameter information • Algorithm ID • Modulus Length • Private Key • File Identifier (FID)	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	RSA Private Key: W RSA Public Key: W DRBG[16] Seed: R,W, X
Encrypt	2A	Performs an encrypt 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.	Plaintext data	Ciphertext data Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	Symmetric key: R, X RSA Public Key: R, X
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.	Ciphertext	Plaintext	Symmetric key: R, X RSA Private Key: R, X
Verify Digital Signature	2A	Verifies a digital signature using RSA PKCS[17]#1	Data Object of the signed data plus the digital signature	Status of the verification	RSA Private Key: R, X

[16] DRBG – Deterministic Random Bit Generator
[17] PKCS -Public-Key Cryptography Standards

Compute Digital Signature	2A	Computes a digital signature using RSA PKCS#1.	Input data for generating the digital signature	Digital Signature	RSA Public Key: R, X
Verify Cryptographic Checksum	2A	Performs AES or Triple DES checksum verification.	Plaintext data object plus the cryptographic checksum data	Status (e.g. 9000, 6300)	Symmetric Key: R, X
Compute Cryptographic Checksum	2A	Computes an AES or Triple-DES checksum. The length of the checksum is 8 bytes.	The data used to compute the cryptographic checksum	Cryptographic checksum	Symmetric Key: R, X
Create File	E0	Creates a file	File control parameters (data field) Length of data field	Status (e.g. 9000)	No CSPs are accessed via this service.
Delete File	E4	Deletes a file and all files which exist within that file	File ID	Status (e.g. 9000)	No CSPs are accessed via this service.
Terminate Card	FE	Terminates all applications on the card	None	None	No CSPs are accessed via this service.
Install Secret	E3	This service is used to enter AES keys, Triple-DES keys, and PINs. The keys which may be entered are as follows: • K_{enc} • K_{mac} • Internal Auth key • External Auth key • Symmetric Key • PIN	• Encrypted PIN or Key data • “Final” secret or “Not Final” secret flag	Status (e.g. 9000, 6700, 6982, 6986, 6A8, 6A82, 6B00, 6CXX)	K_{enc} : W K_{mac} : W Internal Auth key: W External Auth key: W Symmetric Key: W PIN: W

Update Key	E5	Allows the updating of the INIT_KEYs or secret file keys.	• INIT_KEYs • Secret Key data • New error counter plus the key value	Status (eg. 9000, 6700, 6982, 6986, 6A8, 6A82, 6B00, 6CXX)	Symmetric Key: W $INIT\_KEY_{enc}$: W $INIT\_KEY_{mac}$: W K_{enc} : W K_{mac} : W Internal Auth key: W External Auth key: W
Get File List	34	Allows the reading of the FID list of child files of the current file.	None	FID list or “Nonexistent” Status (eg. 9000, 6700, 6982, 6986, 6A8, 6A82, 6B00, 6CXX)	No CSPs are accessed via this service.
Read Public Key	B4	Allows the output of a public key	• FID of the public key • Public Key component read parameter (Read all component, read E component, or read N component)	Public Key data or “Nonexistent” Status (eg. 9000, 6700, 6982, 6986, 6A8, 6A82, 6B00, 6CXX)	No CSPs are accessed via this service.
Import RSA Key	E7	Allows the input of an RSA key	• Encrypted key data • FID of the RSA key	Status (e.g. 9000, 6700, 6982, 6986, 6A8, 6A82, 6B00, 6CXX)	RSA key pair: W

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 must CSPs.

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 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 Binary	80	Allows read access to a binary file	• Offset address of the binary file to read • Length of the data to be read	File data or "Nonexistent" Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service
Read Record	B2	Allows read access to a record.	• Record number • Read parameter (i.e, all records starting at specified record number, or just one record)	Record data or "Nonexistent" Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.
External Authenticate	82	Authenticates an external entity to the cryptographic module. This service may also be used to both authenticate and initiate a secure session with an external entity. **NOTE:** Prerequisite to this service is the use of Get Challenge service. The key as referenced within the service call exists under the current file.	Initiate a secure session Or Authentication data of external entity (32 bytes) plus the MAC (8 bytes) Authenticate only: Algorithm type (AES, Triple-DES, RSA) Key ID (Key Index) Length of data in the field Authentication data (data field)	Status (e.g. 9000) Retry number for the referenced key incremented by one. **NOTE:** If successful this number is then reset to the maximum.	Initiate a secure session: Kenc: R, X Kmac: R, X KSenc: W KSmac :W Or Authenticate Only: Symmetric key: R, X RSA Private Key: R, X

Internal Authenticate	88	Authenticates the cryptographic module to an external entity. **NOTE:** In order for this service to be utilized the external entity must have privileged access to the referenced key.	• Algorithm type (AES, Triple-DES, RSA) • Key ID (Key Index) • Length of data in the field • Random data (data field)	Authentication data Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	Symmetric key: R,X RSA Private Key: R, X
Verify	20	Provides PIN verification. **NOTE:** In order for this service to be utilized the external entity must have privileged access to the referenced PIN.	• Reference to the PIN • PID • Data to be verified	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	PIN: R, X
Change Reference Data	24	Modifies the PIN. **NOTE:** In order for this service to be utilized the external entity must have privileged access to the referenced PIN.	• Old PIN • New PIN • Reference to the password PID	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	PIN: R, W, X
Reset Retry Counter	2C	Resets the retry counter of the PIN to its initial value. **NOTE:** Utilization of this service requires permission to modify PIN.	• Reset parameter (resets recount maximum number and remaining count to default) • Restore parameter (restores recount to initial default value) • Reference to PIN PID	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	No CSPs are accessed via this service.
Generate Asymmetric Key Pair	46	Generates an asymmetric key pair.	• Key parameter information • Algorithm ID • Modulus Length • Private Key File Identifier (FID)	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	RSA Private Key: W RSA Public Key: W DRBG Seed: R,W, X
Encrypt	2A	Performs an encrypt 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.	Plaintext data	Ciphertext data Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	Symmetric key: R, X RSA Public Key: R, X
Decrypt	2A	Performs a decrypt operation. **NOTE:** The MSE service must have previously been utilized to choose the algorithm and	Ciphertext	Plaintext	Symmetric key: R, X RSA Private Key: R, X

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

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 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 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 and stored by the cryptographic module. In the Put Data service, only the OEM information is allowed to be set.	• Data object tag ('81' which indicates OEM info, followed by up to 32 bits of OEM info. • Length of object data	Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.
Get Data	CA	This service allows data to be retrieved. Data refers to global data, which belongs to the cryptographic module, such as the serial number, OEM information, chip information which includes algorithm support, RAM size.	Data object tag (e.g., '80' which indicates card serial number)	Content of object Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.
Get Challenge	84	Requests a random value that will be used as a challenge within the External Authenticate service.	None	Random value Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	• DRBG Key Value: R, W, X · DRBG 'V' Value: R; W, X
Manage Security Environment (MSE)	22	Prepares the cryptographic module for the subsequent commands, SET, STORE, RESTORE, SEID, and ERASE.	• CRDO[18] • Algorithm Reference • Key Reference • File Reference • Length of CRDOs	Status (e.g. 9000, 6300, 62CX, 6581, 6700, 6982, 6984, 6A81, 6A2, 6A86, 6A88)	No CSPs are accessed via this service.
Select	A4	Allows the selection of a specified file.	• File identifier • Dedicated file Name • File path starting at master file • File path starting at dedicated file	File control information Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.

[18] CRDO – Control Reference Data Object

Manage Channel	70	Allows the assignment; opening, and closing of a logical channel. A logical channel is a logical link between the host system and a file on the smart card.	Number of logical channel to be assigned, opened, or closed (01-03).	Status (e.g. 9000, 6283, 6284, 6A80, 6A81, 6A82, 6A86, 6A87)	No CSPs are accessed via this service.
Hash	2A	Performs a hash using SHA[19]-1 or SHA-256.	Input Data	Hash result or None	No CSPs are accessed via this service.

2.5 Physical Security

The HYP2003 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 HYP2003 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 HYP2003 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.

[19] SHA – Secure Hash Algorithm

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[20], CBC[21] modes using 128-bit key sizes	1473
Triple-DES in ECB, CBC modes using Keying Option 1	991
RSA PKCS#1 v1.5 signature generation– using 2048-bit keys	720
RSA PKCS#1 v1.5 signature verification – using 1024-and 2048bit keys	720
ANSI[22] X9.31 Key Pair Generation	720
SHA-1 and SHA-256	1332
SP[23] 800-90 CTR[24]_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)

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.

[20] ECB – Electronic Codebook
[21] CBC – Cipher-Block Chaining
[22] ANSI – American National Standards Institute
[23] SP – Special Publication
[24] CTR -Counter

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

Key	Key Type	Use	Generation / Input	Output	Storage	Zeroization	Key To Entity
Symmetric Key	AES 128-bit key; Triple-DES 168-bit Key	These keys are used to encrypt/decrypt data, or within a symmetric MAC algorithm to generate authentication data.	**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 in EEPROM[25] in special files used to store symmetric keys and PINs.	Procedurally overwrite keys with arbitrary data using the Update Key service.	Storage: 4-bit key ID Input/Output: This key is associated with the Crypto-Officer role during Input.
Internal Auth Key	AES 128-bit key; Triple-DES 168-bit Key	These keys are used to authenticate the module to an external entity.	**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 in EEPROM in special files used to store symmetric keys and PINs.	Procedurally overwrite keys with arbitrary data using the Update Key service.	Storage: 4-bit key ID Input/Output: This key is associated with the Crypto-Officer role during Input
External Auth Key	AES 128-bit key; Triple-DES 168-bit Key; RSA 2048-bit key	These keys are used to modify the security state of the currently selected DF[26] .	**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 in EEPROM in special files used to store symmetric keys and PINs.	Procedurally overwrite keys with arbitrary data using the Update Key service.	Storage: 4-bit key ID Input/Output: This key is associated with the Crypto-Officer role during Input

[25] EEPROM -Electronically Erasable Programmable Read-Only Memory
[26] DF –Dedicated File

$INIT\_KEY_{enc}$	AES 128-bit key	This key is used to derive a session key which is then used to encrypt/decrypt data over a secure session between an authorized external entity and the module.	**Generation**: This key is not generated within the module. It is a factory-set key which is used only in the initialized state of the module. **Input**: This key is factory-set and cannot be modified or input outside of manufacturing.	N/A: The module does not support the output of this key.	This key is stored under in the reserved file in EEPROM.	Procedurally overwrite key with arbitrary data using the Update Key service.	Storage: 4-bit key ID Input/Output: N/A
$INIT\_KEY_{mac}$	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. It is a factory-set key which is used only in the initialized state of the module. **Input:** This key is factory-set and cannot be modified or input outside of manufacturing.	N/A: The module does not support the output of this key.	This key is stored under in the reserved file in EEPROM.	Procedurally overwrite keys with arbitrary data using the Update Key service.	Storage: 4-bit key ID Input/Output: N/A
K_{enc}	AES 128-bit key	This key is used to derive a session key which is then used to encrypt/decrypt 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
K_{mac}	AES 128-bit key	This key is used to drive a session key which is then used to authenticate an	**Generation:** This key is not generated within the module. **Input:** This key may be input encrypted within a secure	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

		operator or data over a secure session between an authorized external entity and the module.	channel.				
KS_{enc}	AES 128-bit key	This key is used to encrypt /decrypt data over a secure session.	**Generation:** Generated from the $INIT\_KEY_{enc}$ or K_{enc} 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 (03) for which it is being used to secure messaging. Input/Output: N/A, this key is not output
KS_{mac}	AES 128-bit key	This key is used to authenticate data over a secure session.	**Generation:** Generated from the $INIT\_KEY_{mac}$ or K_{mac} 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 (03) for which it is being used to secure messaging. Input/Output: N/A, this key is not output
Personal Identification Number (PIN)	6-16 byte secret	This key is used to modify the security state of the currently selected DF.	**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.	EEPROM in plaintext	Procedurally overwrite keys with arbitrary data using the Update Key service.	Storage: 4-bit key ID
RSA Private Key	2048-bit RSA private key	This key is used to decrypt or verify data.	**Generation:** This key is generated using the Approved SP800-90 DRBG. **Input:** This key may be input encrypted within a secure channel.	N/A: The module does not support the output of this key.	EEPROM in plaintext	Procedurally overwrite keys with arbitrary data using the Import RSA Key service.	Storage: 4-bit File ID **NOTE:** Only one RSA Private key may be stored in an RSA Private Key file.

RSA Public Key	2048-bit RSA public key	This key is used to encrypt or sign data.	**Generation:** This key is generated using the Approved SP800-90 DRBG. **Input:** This key may be input encrypted within a secure channel.	Output in plaintext using the Read Public key command.	EEPROM in plaintext	N/A: this key is a public key and therefore does not have to be zeroized.	Storage: 4-bit File ID **NOTE:** Only one RSA Public key may be stored in an RSA Public Key file.
DRBG 'V' Value	Internal CTR DRBG state value	Used for SP 800-90 CTR_DRBG	Internally Generated	Never	Plaintext in volatile memory	Power Cycle	Associated with an internal module variable
DRBG Key Value	Internal CTR DRBG state value	Used for SP 800-90 CTR_DRBG	Internally Generated	Never	Plaintext in volatile memory	Power Cycle	Associated with an internal module variable

2.8 EMI/EMC

The HYP2003 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 HYP2003 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 HYP2003 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 HYP2003 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.

3.1 Detecting a FIPS Cryptographic Module

The HyperPKI™ HYP2003 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 HYP2003 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 HYP2003 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 HYP2003 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

Appendix A

Table 12 defines the acronyms used in this Security Policy.

Table 12: List of 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	Definition
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
DRBG	Deterministic Random Bit Generator
USB	Universal Serial Bus
V	Volt
VCC	Common Collector Voltage