FIPS 140-2 Security Policy for

HiKey PKI Token

Hardware Version: HiKey3.0-BK
Firmware Version: HiKey COS V3.1

Chunghwa Telecom Co., Ltd.

September 14th, 2016

1 Introduction

This document is the Security Policy for the Chunghwa Telecom Co., Ltd. HiKey PKI Token which is embedded RS45C chip inside. This token, hereafter called the HiKey PKI Token is used to provide user authentication and cryptographic services.

This Security Policy specifies the security rules under which the token must operate to meet the requirements of FIPS 140-2 Level 2. It describes how the token functions to meet the FIPS requirements, and the actions that operators must take to maintain the security of the token.

This Security Policy describes the features and design of the Chunghwa Telecom Co., Ltd. HiKey PKI Token using the terminology contained in the FIPS 140-2 specification. *FIPS 140-2, Security Requirements for Cryptographic Modules* specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST Cryptographic Module Validation Program (CMVP) validates cryptographic modules to FIPS 140-2 and other cryptography-based standards. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of unclassified but sensitive information. Many other governments, private organizations, and financial institutions also recognize FIPS-validated modules.

The FIPS 140-2 standard, and information on the CMVP, can be found at http://csrc.nist.gov/groups/STM/cmvp/index.html.

This Security Policy contains only non-proprietary information. All other documentation submitted for FIPS 140-2 conformance testing and validation is deemed proprietary and is releasable only under appropriate non-disclosure agreements.

1.1 Security Levels

The HiKey PKI Token meets the overall requirements applicable to Level 2 security of FIPS 140-2. The individual security requirements specific for FIPS 140-2 meet the level specification indicated in the Table 1.

Security Requirements Section	*Level*
Cryptographic Module Specification	2
Cryptographic Module Ports and Interfaces	2
Roles, Services, and Authentication	3
Finite State Model	2
Physical Security	2
Operational Environment	N/A
Cryptographic Key Management	2
EMI/EMC	3
Self-Tests	2
Design Assurance	3
Mitigation of other attacks	2

Table 1 - Security Requirements Specific to FIPS 140-2.

1.2 Acronyms and Abbreviations

APDU	Application Protocol Data Unit
CCID	Circuit Card Interface Device
CBC	Cipher Block Chaining
CDF	Children Dedicated File
CE	Confomite Europeene
CMVP	Cryptographic Module Validation Program
CSE	Communications Security Establishment
CSP	Critical Security Parameter
DES	Data Encryption Standard
ECDSA	Elliptic Curve Digital Signature Algorithm
EDC	Error Detection Code
EEPROM	Electrically Erasable Programmable Read Only Memory
EF	Elementary File
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
KAT	Known Answer Test
LED	Light Emitting Diode
MAC	Message Authentication Code
MF	Master File
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
PDF	Parent Dedicated File
PKI	Public Key Infrastructure
PUB	Publication
RAM	Random Access Memory
RNG	Random Number Generator
RSA	Rivest Shamir and Adleman Public Key Algorithm
SHA	Secure Hash Algorithm
SRDI	Security Related Data Item
TDES	Triple-DES

2 Chunghwa Telecom HiKey PKI Token

2.1 Functional Overview

The HiKey PKI Token contains an implementation of native operation environment (limited operational environment), the security policy is based on the file system created. PINs and keys that have been securely loaded at HiKey PKI Token issuance authenticate the roles of the Crypto Officer and User (HiKey PKI Token Holder).

2.2 Cryptographic Module Specification

The HiKey PKI Token is a multiple-chip implementation of a cryptographic module. Figure 1 shows a physical view of the module.

Figure 1. Physical view of the HiKey PKI Token.

The HiKey PKI Token is an USB token that provides cryptographic services and may be used as a replacement for a standard smart card offering the same services. The “cryptographic boundary” for the module with respect to the FIPS 140-2 validation is the “token enclosure”. The token is protected by a hard opaque tamper evident enclosure required in the FIPS 140-2 physical Level 2 validation for a single -chip implementation.

The hardware base is the Renesas RS45C IC that is validated under the Common Criteria at EAL5+.

The HiKey PKI Token consists of the following elements:

- Renesas RS45C microcomputer, USB controller, CCID Smart Card Reader chip and voltage regulator. These are standard production-quality IC’s.
- System firmware is installed in Read Only Memory (ROM) and the installation is as part of the chip manufacturing process.
- Critical Security Parameters are stored in the EEPROM and the parameter set up is as part of the token personalization operation.
- The token is protected by hard opaque plastic enclosure that contains a tamper evident seals. Removal of the enclosure will show tamper evidence.

2.3 Operational Environment

The HiKey PKI Token has a limited operational environment consisting of a native system operating on a Renesas RS45C Integrated Circuit chip. The system firmware is installed into ROM at factory and the token does not support firmware upgrade.

2.4 Module Interfaces

2.4.1 PHYSICAL INTERFACE DESCRIPTION

The HiKey PKI Token supports four pins that lead to the PCB board.

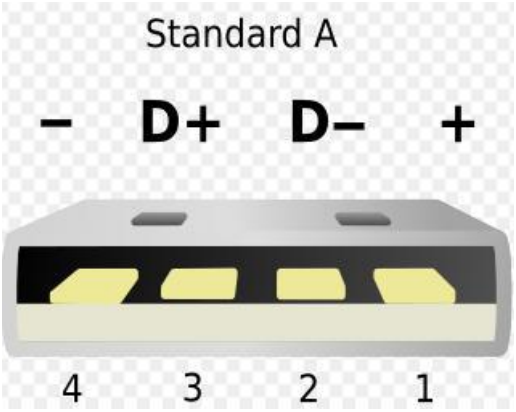


Figure 2. Interfaces.

2.4.2 SPECIFIC FUNCTIONS OF USB CONTACTS

PIN	*Function*	*FIPS 140-2 Logical Interface*
USB 1	V^{BUS} supply voltage 4.75V – 5.25V	Power Interface
USB 2	Data -	Data Input, Data Output, Control Input, Status Output
USB 3	Data +	Data Input, Data Output, Control Input, Status Output
USB 4	Ground	N/A

Table 2. Functional Specifications of PINs.

2.4.3 USB 1 SUPPLY CURRENT

- Maximum Value: 200mA at 5.0V
- Typical Value: 150mA at 5.0V

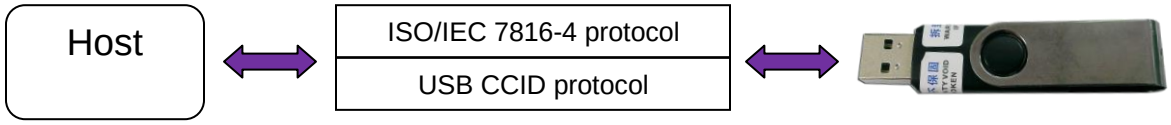
2.4.4 MODULE SECURITY AND KEY ACCESS COMMAND SET

HiKey PKI Token security and key access command set defined by the following specifications:

- Chunghwa Telecom HiKey PKI Token User's Manual v3.1.
- Chunghwa Telecom HiKey PKI Token Operator Guidance v1.0.

2.4.5 HOST TO TOKEN COMMUNICATIONS PROTOCOL

There is a USB CCID smart card reader IC in HiKey PKI Token, so the underlying communication between Host and HiKey PKI Token is in accordance with the USB CCID protocol. TheHost will treat HiKey PKI Token as a CCID smart card reader with smart card inserted. In logic view, the Host communicates with RS45C chip in HiKey PKI Token by ISO/IEC 7816-3 & 4 protocol (APDU command protocol).



2.4.6 DATA PATH

Input Data Path: There are two kinds of Input Data Paths, one is clear-text data path and the other is cipher-text data path. For the APDU command where sensitive data is entered, the sensitive data shall be encrypted according to file security attribute.

Output Data Path: The output data path is the same as the input data path. It may be clear-text or cipher-text data according to file security attribute.

2.4.7 LOGICAL INTERFACE DESCRIPTION

The I/O PIN (USB PIN 2 and 3) of the token (refer to Table 2) provides the following logical interfaces:

- Data In (I/O bidirectional line)
- Data Out (I/O bidirectional line)
- Control In (I/O bidirectional line)
- Status Out (I/O bidirectional line) and LED

The APDU command protocol and synchronization timing controls, provided in part by way of the platform CLK clock input, manage the separation of logical interfaces that use the same physical port.

Electrical (physical) contact and data link layer contact is established between the token and the Host by the Host USB interface issuing a RESET signal to the token which then responds with an "Answer To Reset (ATR)". From this point on, the token functions as a “slave” processor to implement and respond to the Host “master” commands. The token adheres to a well defined set of state transitions. Within each state, a specific set of commands are accessible.

The details of these commands are defined in the Chunghwa Telecom HiKey PKI Token User’s Manual V3.1 and ISO 7816-4.

2.4.8 EMI/EMC

The HiKey PKI Token has been tested by SPORTON International Inc., and found in compliance with the requirement of the following standards:

- FCC Part 15 Subpart B, Class B
- CE EN 55022 / 55024

3 Roles, Services and Authentication

3.1 Roles

The token uses identity-based access control. Access control rules provide services to operators who identify themselves by demonstrating knowledge of a cryptographic key set, or PIN.

The token defines three distinct roles that are supported by the on-token cryptographic system: the Crypto Officer role, a User role, and an unauthenticated role.

- Crypto Officer is a role authenticated by demonstrating knowledge of a key set and key ID.
- Token Holder is a User role authenticated by possession of the token and knowledge of the Token Holder PIN.
- The unauthenticated role is assumed by any unauthenticated operator who has access to the host application.

The token ensures the authentication of off-Token entities (Cryptographic Officer and User) and provides them with cryptographic services according to their role. Operators may not change roles without re-authenticating in the new role. All previous authentications are cleared when the token is powered down.

The token does not allow multiple concurrent operators or support a maintenance role.

3.1.1 Cryptographic Officer Role

The Crypto Officer establishes his/her identity to the on-token security controller on the HiKey PKI Token through the verification of a Triple-DES key set stored in token. Through mutual authentication (Based on the key ID specified in file attribute) between the Crypto Officer and the token, he/she could process this file (It could be MF, PDF, CDF or EF) by commands defined according to each file attribute (ex. Under MF, CREATE FILE command could be used)

The Key ID is a unique identifier created when a Crypto-Officer generates a Triple-DES or RSA key. The Key ID is stored in the module. The Crypto-Officer must demonstrate knowledge of a key's associated key ID before the module will permit a Crypto-Officer service to be performed using that key.

3.1.2 User Role

The User role (Token holder) is responsible for the physical security of his/her token and confidentiality of their PIN. The User Role operator is authenticated by verification of a PIN. After successful authentication of User, he/she could generate RSA or ECDSA key, read non-security relevant data from file.

3.1.3 Unauthenticated Role

It is assumed by any unauthenticated operator who has access to the token. The operator can only read non-security relevant token information.

3.2 Module Services

3.2.1 *Basic Module Services*

Crypto Officer Administrative Services

A crypto officer can make changes on the token using commands that are available after the crypto officer role is authenticated. The crypto officer authenticates to his role by proving knowledge of a crypto officer key set associated with the token and using the key set to establish a secure message.

Roles, Basic Token Services, and Access Controls for Cryptographic Keys and CSPs

Each role has access to specific basic token services. The basic token services, in turn, may use or operate on cryptographic keys or critical security parameters (CSPs). The following table shows the relationship between roles, services and indicates the type of access provided to various cryptographic keys and CSPs.

Role	***Services***	***Cryptographic Keys and CSPs accessed***	***Type(s) of Access***
Crypto-Officer	SELECT FILE	None	Execute
	CREATE FILE	None	Write
	READ BINARY	None	Read
	UPDATE BINARY	None	Write
	ERASE BINARY	None	Write
	READ RECORD	None / $\mathbf{K_{PUBVER}}$ RSA Key / $\mathbf{K_{ECDSA\text{-}PUBVER}}$ ECDSA Key	Read
	UPDATE RECORD	None	Write
	APPEND RECORD	None	Write
	READ VALUE	None	Read
	ADD VALUE	None	Write
	SUBSTRACT VALUE	None	Write
	NEW VALUE	None	Write
	GET CHALLENGE	DRBG Internal State values (V and C)	Execute
	EXTERNAL AUTHENTICATE	$\mathbf{K_{EXTAUTH}}$ TDES Key	Execute
	INTERNAL AUTHENTICATE	$\mathbf{K_{INTAUTH}}$ TDES Key	Execute
	LOAD KEY	PIN, $\mathbf{K_{PUBVER}}$ RSA Key/ $\mathbf{K_{PRIVSIGN}}$ RSA Key, $\mathbf{K_{ENC}}$ TDES Key, $\mathbf{K_{MAC}}$ TDES Key and $\mathbf{K_{UNLOCK}}$ TDES Key	Write
	CHANGE KEY	PIN, $\mathbf{K_{PUBVER}}$ RSA Key/$\mathbf{K_{PRIVSIGN}}$ RSA Key, $\mathbf{K_{MAC}}$ TDES Key, $\mathbf{K_{ENC}}$ TDES Key and $\mathbf{K_{UNLOCK}}$ TDES Key	Read/Write
	UNLOCK KEY	PIN, $\mathbf{K_{UNLOCK}}$ TDES key	Write
	GENERATE RSA KEY PAIR	$\mathbf{K_{PUBVER}}$ RSA Key/$\mathbf{K_{PRIVSIGN}}$ RSA Key	Write
	GENERATE HASH	None	
	RSA CRYPTOGRAPHY	$\mathbf{K_{PUBVER}}$ RSA Key/$\mathbf{K_{PRIVSIGN}}$ RSA Key	Execute
	RSA PKCS1 SIGN	$\mathbf{K_{PRIVSIGN}}$ RSA Key	Execute
	RSA PKCS1 VERIFY	$\mathbf{K_{PUBVER}}$ RSA Key	Execute
	GET RSA RESULT	None	Read
	LOAD ECDSA DOMAIN PARAMETER	None	Write
	LOAD ECDSA KEY	$\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$ ECDSA Key / $\mathbf{K_{ECDSA\text{-}PUBVER}}$ ECDSA Key	Write
	CHANGE ECDSA KEY	$\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$ ECDSA Key / $\mathbf{K_{ECDSA\text{-}PUBVER}}$ ECDSA Key	Write
	GENERATE ECDSA KEY PAIR	$\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$ ECDSA Key / $\mathbf{K_{ECDSA\text{-}PUBVER}}$ ECDSA Key	Write
	ECDSA SIGN	$\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$ ECDSA Key	Execute
	ECDSA VERIFY	$\mathbf{K_{ECDSA\text{-}PUBVER}}$ ECDSA Key	Execute
	KNOWN ANSWER TEST	None	Execute
	GET DATA	None	Read
	FREE MEM	None	Read
	ERASE ALL	$\mathbf{K_{TRAN}}$, PIN, $\mathbf{K_{PUBVER}}$ RSA Key/ $\mathbf{K_{PRIVSIGN}}$ RSA Key, $\mathbf{K_{ECDSA\text{-}PUBVER}}$ / $\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$ ECDSA Key, $\mathbf{K_{ENC}}$ TDES Key, $\mathbf{K_{MAC}}$ TDES Key, $\mathbf{K_{INTAUTH}}$ TDES Key, $\mathbf{K_{EXTAUTH}}$ TDES Key and	Write

		K_{UNLOCK} TDES key	
	GET COS INFO	None	Read
	Obtain FIPS Approved mode of operation indicator: SELECT FILE 0x0001 followed by READ BINARY	None	Read
User	SELECT FILE	None	Read
	READ BINARY	None	Read
	UPDATE BINARY	None	Write
	ERASE BINARY	None	Write
	READ RECORD	None / **K_{PUBVER}** RSA Key / **$K_{ECDSA-PUBVER}$** ECDSA Key	Read
	UPDATE RECORD	None	Write
	APPEND RECORD	None	Write
	READ VALUE	None	Read
	ADD VALUE	None	Write
	SUBSTRACT VALUE	None	Write
	VERIFY	PIN	Execute
	GET CHALLENGE	DRBG Internal State values (V and C)	Execute
	GENERATE HASH	None	Execute
	CHANGE KEY	PIN	Write
	GENERATE RSA KEY PAIR	**K_{PUBVER}** RSA Key/**$K_{PRIVSIGN}$** RSA Key	Write
	RSA CRYPTOGRAPHY	**K_{PUBVER}** RSA Key/**$K_{PRIVSIGN}$** RSA Key	Execute
	RSA PKCS1 SIGN	**$K_{PRIVSIGN}$** RSA Key	Execute
	RSA PKCS1 VERIFY	**K_{PUBVER}** RSA Key	Execute
	GET RSA RESULT	None	Read
	GENERATE ECDSA KEY PAIR	**$K_{ECDSA-PRIVSIGN}$** ECDSA Key / **$K_{ECDSA-PUBVER}$** ECDSA Key	Write
	ECDSA SIGN	**$K_{ECDSA-PRIVSIGN}$** ECDSA Key	Execute
	ECDSA VERIFY	**$K_{ECDSA-PUBVER}$** ECDSA Key	Execute
	KNOWN ANSWER TEST	None	Execute
	GET DATA	None	Read
	FREE MEM	None	Read
	GET COS INFO	None	Read
	Obtain FIPS Approved mode of operation indicator: SELECT FILE 0x0001 followed by READ BINARY	None	Read
Unauthenticated	SELECT FILE	None	Execute
	GET CHALLENGE	DRBG Internal State values (V and C)	Execute
	GET DATA	None	Read
	FREE MEM	None	Read
	GET COS INFO	None	Read
	GENERATE HASH	None	Read
	KNOWN ANSWER TEST	None	Execute

Table 3. Basic Token Services

There is a corresponding command APDU and response APDU for each service, please refer to User's Manual for more information.

3.3 Authentication

HiKey PKI Token provides three authentication mechanisms: external authentication, internal authentication and PIN verification. External authentication is used to authenticate Crypto Officer, internal authentication is used to authenticate token and PIN verification is used to authenticate the User role.

There may be several Crypto Officer and User roles in each token. Each PDF, CDF and EF may belong to individual Crypto Officer and the key ID for these PDF, CDF and EF may be different. Each EF may belong to individual User and the key under these EFs may be different. The file attribute for MF, PDF, CDF and EF will specify the presence of a key or PIN for authentication is required or not, and which key or PIN will be used for authentication.

3.3.1 Mechanisms:

To authenticate a Crypto Officer, he/she should apply SELECT FILE command in order to select the MF, PDF or CDF belonging to them. Then the Crypto Officer should issue the GET CHALLENGE command in order to request an 8-byte random number from the token. The Crypto Officer should then issue the EXTERNAL AUTHENTICATE command along with the key ID of the External Authentication key which is specified in the MF, PDF or CDF file attribute to send back the encrypted random number and accomplish the authentication process.

To authenticate a token, a Crypto Officer applies the INTERNAL AUTHENTICATION command which sends a random number and the key ID of the Internal Authentication Key to the token. The token then encrypts the random number using the Internal Authentication Key (using the key ID specified by the request). The token will then return the encrypted string for verification outside of the module.

To authenticate a User, he/she should apply SELECT FILE command to select the EF belong to him/her first. Then the User applies VERIFY command to accomplish the authentication process.

3.3.2 Strength:

The following table provides rough estimates of the strengths of the token's authentication mechanisms.

Authentication Type	**Strength**
External Authentication	This mechanism applies Triple-DES 3-key, the probability of randomly guessing the correct sequences is 1 in 2^{168}.
Internal Authentication	This mechanism applies Triple-DES 3-key, the probability of randomly guessing the correct sequences is 1 in 2^{168}.
PINs	The minimum length of PIN is 8 bytes and the value is not limited to digital number. The maximum length of the PIN is specified by the application when the password is created. The maximum length of the PIN is ultimately limited by

	the maximum length of the password field, which is 253 bytes.

Table 4. Strength

3.3.3 Strength for multiple attempts:

External Authentication:

To attempt a brute force attack on the token, an attacker has to send APDU commands in a serial fashion to the token, and wait for the corresponding APDU response to each APDU command. Each APDU command must be responded to before the next APDU command can be sent. An attacker must send a minimum 13 byte APDU to the HiKey PKI Token, and get a resulting 2-byte response. As there is a single I/O port on the token, each attack is bottlenecked through this port. Each attack, that is, each attempt to authorize with a different Triple-DES key requires 15 bytes of data to be clocked in or out of the HiKey PKI Token. The maximum data rate for the token is 38,400 bps through this single port. Ignoring the processing time required on the token to process the Triple-DES key, we can compute the maximum number of attempts which could occur within a 60 second interval:

- 15 bytes of I/O at 8bits/byte = 120 bits/attempt
- 120 bits/attempt divided by 38,400 bits/second = 0.003125 seconds/attempt
- 60 seconds/minute divided by 0.003125 seconds/attempt = 19,200 attempts/minute

As the Triple-DES key space is over 2^{168} for 3-key possible values, it follows that 19,200 attempts in a 60 second interval will not significantly traverse the space of possible values. As a result, the token far exceeds the requirement of 1 in 100,000 for multiple attempts in a 60 second interval.

Internal Authentication:

To try a key against the token, an attacker must send a minimum 14 byte APDU to the HiKey PKI Token, and get a resulting 10-byte response. As there is a single I/O port on the token, each Triple-DES key attempt requires 24 bytes of data to be clocked in or out of the HiKey PKI Token. The maximum data rate for the token is 38,400 bps through this single port. Ignoring the processing time required on the token to process the Triple-DES key, we can compute the maximum number of attempts which could occur within a 60 second interval:

- 24 bytes of I/O at 8bits/byte = 192 bits/attempt
- 192 bits/attempt divided by 38,400 bits/second =0.005 seconds/attempt
- 60 seconds/minute divided by 0.005 seconds/attempt = 12,000 attempts/minute

As the Triple-DES key space is over 2^{168} for 3-key possible values, it follows that 12,000 attempts in a 60 second interval will not significantly traverse the space of possible values. As a result, the token far exceeds the requirement of 1 in 100,000 for multiple attempts in a 60 second interval.

PIN Verification:

The minimum length of the User PIN is a string of 8-byte. PINs can contain any visible character from ' '(space) to '~' in ASCII characters, there is 95 possible characters for each byte, so it yielding a maximum of 95^8 possible PINs. This far exceeds the 1 in a million test.

To try a PIN against the token, the attacker must send a 13-byte APDU to the HiKey PKI Token, and get a resulting 2-byte response. As there is a single I/O port on the token, this means that each PIN attempt requires 15 bytes of data to be clocked in or out of the HiKey PKI Token. The maximum data rate for the token is 38,400bps through this single port. If we ignore the processing time required on the token to check the PIN, we can compute the maximum number of PIN attempts which could occur within a 60 second interval:

- 15 bytes of I/O at 8bits/byte = 120 bits/attempt

- 120 bits/attempt divided by 38,400bits/second = .003125 seconds/attempt
- 60seconds/minute divided by .003125 seconds/attempt = 19,200 attempts/minute

As the minimum length PIN results in 95^8 possible values, it follows that 19,200 attempts in a 60 second interval will not significantly traverse the space of possible PIN values. As a result, the token far exceeds the requirement of 1 in 100,000 for multiple attempts in a 60 second interval.

Moreover, an 8-bit counter internal security attribute in each PIN file further limits the number of failed PIN attempts an attacker could perform by blocking the HiKey PKI Token if the counter limit (3 attempts per PIN) is exceeded.

4 FIPS Approved Mode of Operation

The token only supports FIPS Approved mode of operation.

The token is shipped in a manufactured state, which is a partially initialized state.
During the initialization ("personalization") process of the token, the Crypto-Officer executes the required steps (please see below) to put the module in FIPS Approved mode, which is the only operational mode of the token.

The following procedures have to be performed to put the token in the FIPS mode of operation:

1. Pre-Personalization the HiKey PKI Token by:
   a. initialize the HiKey PKI Token;
   b. load the transport key (Triple-DES) and perform the GET CHALLENGE and EXTERNAL AUTHENTICATE commands. If the key is authenticated, the HiKey PKI Token is fully initialized;
2. Issue CREATE FILE command to create the basic file system(MF, Serial Number EF and DES Key EF)
3. Issue LOAD KEY command to load default Triple-DES keys.
4. Issue CREATE FILE command to create the other necessary files on the token, configure the file system. When access security sensitive data or keys, secure messaging must be turned on.

The Crypto-Officer and the User roles can execute the following two commands to obtain the FIPS Approved mode of operation indicator. This is accomplished by querying the status of a special Elementary File (EF) with file ID 0x0001, named Serial Number:

SELECT FILE 0x0001

READ BINARY

If the token has been initialized / personalized, the READ BINARY command will display a unique non-zero value, which means the token is in FIPS Approved mode.

5 Module Cryptographic Functions

The purpose of the HiKey PKI Token is to provide a FIPS validated module that may in turn provide cryptographic services to end-user. Cryptographic keys and CSPs (PINs) represent the roles involved in controlling the token. A variety of FIPS 140-2 validated algorithms are used in the HiKey PKI Token to provide cryptographic services; these include:

- Triple-DES for internal/external authentication, PIN verification, key wrapping and unwrapping, data encryption and message authentication code within the secure message.
- RSA Key Pair Generation

- RSA PKCS #1 Signature Generation and Verification.
- ECDSA Key Pair Generation
- ECDSA Signature Generation and Verification
- SHA-1, SHA-256, SHA-384 and SHA-512 Hashing.
- DRBG used for asymmetric cryptographic key generation and random number generation.

Details of cryptographic functions are shown in this table:

Type (key name)	**Algorithm**	**Key size (bits) / Curve Size**	**Effective Security strength (bits)**	**FIPS Approved**	**Certificate**
RSA Key Pair Generation ($\mathbf{K_{PUBVER}}$, $\mathbf{K_{PRIVSIGN}}$)	RSA	2048	112	Yes (FIPS 186-4)	#2041
RSA Signature Generation and Verification ($\mathbf{K_{PUBVER}}$, $\mathbf{K_{PRIVSIGN}}$)	RSA PKCS#1 v1.5	2048	112	Yes (FIPS 186-4)	#2041
ECDSA Key Pair Generation ($\mathbf{K_{ECDSA\text{-}PUBVER}}$, $\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$)	ECDSA	P-224 P-256 P-384	112 128 192	Yes (FIPS 186-4)	#878
ECDSA Signature Generation and Verification ($\mathbf{K_{ECDSA\text{-}PUBVER}}$, $\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$)	ECDSA	P-224 P-256 P-384	112 128 192	Yes (FIPS 186-4)	#878
ECDSA SigGen Component Validation List (CVL)	ECDSA (Signature Generation of hash sized messages)	P-224 P-256 P-384	112 128 192	Yes (FIPS 186-4)	#833
Symmetric Key ($\mathbf{K_{TRAN}}$, $\mathbf{K_{UNLOCK}}$)	Triple-DES (ECB, CBC)	168	112	Yes (NIST SP 800-67)	#2184
Key Wrapping ($\mathbf{K_{ENC}}$, $\mathbf{K_{MAC}}$)	Triple DEA Key Wrap	168	112	Yes (NIST SP 800-38F)	#2184
CMAC ($\mathbf{K_{INTAUTH}}$, $\mathbf{K_{EXTAUTH}}$)	Triple-DES MAC (Generation/Verification)	168	112	Yes (NIST SP 800-38B)	#2184
Digest	SHA-1		80	Yes (FIPS 180-4)	#3284
	SHA-256		128		
	SHA-384		192		
	SHA-512		256		
RNG	DRBG (NIST SP 800-90 Hash_DRBG)		128	Yes (NIST SP 800-90A)	#1172

	NDRNG (HARDWARE RNG)			No, only utilized to seed the module's Approved DRBG	ISO/IEC 15408 EAL5+

Table 5. Token Cryptographic Functions.

Note:

1. *SHA-1 is used for general hash functions and as part of the NIST SP 800-90A Hash_DRBG and HMAC algorithms. The module does not support SHA-1 for Signature Generation operations with ECDSA or RSA. For additional information on the transition rules for SHA-1 please refer to NIST Special Publication 800-131A and Section 5.6.2 of SP 800-57.*
2. *Triple-DES (Cert. #2184, key wrapping; key establishment methodology provides 112 bits of encryption strength);*
3. *RSA (key wrapping; key establishment methodology provides 112 bits of encryption strength).*

6 Cryptographic Key Management

The token contains a variety of keys and CSPs and does not input or output plaintext cryptographic key components, plaintext authentication data, or other unprotected CSPs.

6.1 HiKey PKI Token Manufacturing Keys

HiKey PKI Token Manufacturing Authentication Key, $\mathbf{K_{TRAN}}$ Triple-DES key used only for the HiKey PKI Token Manufacturing and ERASE ALL command (The token zeroizes) by token vendor's (Chunghwa Telecom Co., Ltd). HiKey PKI Token Manufacturing Authentication Key is owned by the token vendor (Chunghwa Telecom Co., Ltd). When the locked token can't be unlocked or the token might be reused by another application, all the data, keys and values in the token must erase before it can be reused.

6.2 Secure messaging Keys

All secret, private keys and PINs that enter the module are wrapped using Triple-DES Key Wrap.

- Command/Response Encryption Key , $\mathbf{K_{ENC}}$ Triple-DES key used for data encryption/ decryption in CBC mode (to protect confidentiality of command data and response data when using secure messaging)
- Command/Response MAC Key, $\mathbf{K_{MAC}}$ Triple-DES key for data authentication (to protect data authenticity of command data and response data when using secure messaging)

6.3 Authentication Keys

- External Authentication Key, $\mathbf{K_{EXTAUTH}}$ Triple-DES key is used to authenticate the host by token.
- Internal Authentication Key, $\mathbf{K_{INTAUTH}}$ Triple-DES key is used to authenticate the token by host system
- Unlock Triple-DES key, $\mathbf{K_{UNLOCK}}$ is treated as a kind of External Authentication Key used to Unlock the locked Triple-DES key or PIN.

6.4 PKI Key Pairs

RSA public and private keys can be generated on the HiKey PKI Token using the GENERATE RSA KEY PAIR command. Alternatively the RSA key pairs may be loaded onto the HiKey PKI Token using the LOAD KEY command and be changed using the CHANGE KEY command.

RSA PKI Key pair

- RSA Public Verify Key, $\mathbf{K_{PUBVER}}$, used for RSA signature verification operations.
- RSA Private Sign Key, $\mathbf{K_{PRIVSIGN}}$, used for RSA signature generation operations.

ECDSA public and private keys can be generated on the card using the GENERATE ECDSA KEY PAIR command. Alternatively the ECDSA key pairs may be loaded onto the card using the LOAD ECDSA KEY command and be changed using the CHANGE ECDSA KEY command.

ECDSA PKI Key pair

- ECDSA Public Verify Key, $\mathbf{K_{ECDSA\text{-}PUBVER}}$ for ECDSA signature verification operations.
- ECDSA Private Sign Key, $\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$, used for ECDSA signature generation operations.

6.5 NIST SP 800-90A DRBG V and C values

The values V and C are internal values of the Hash-based DRBG that are local variables and kept plaintext in the memory during the random number generation process and their occupied memory will be recycled by the system once the process of the random number generation finishes. So both values never leave the token. The values V and C are only derived from DRBG instantiation process and only used in the DRBG mechanism and never accessed by non-DRBG functions, they will be zeroized before DRBG returns a random number. The values V and C always re-generate when the random number generation is invoked every time. They are always new and never be used next time. So the token use Hash-DRBG mechanism which does not need to support reseed function.

6.6 Token Holder PIN

A Token Holder (User role) must enter a valid PIN as part of the authentication process. The minimum length of PIN is 8 bytes and the value is not limited to digital number. PIN are stored in plaintext format in EEPROM.

6.7 Cryptographic Key Generation

Key pairs may be generated on the token using the GENERATE RSA KEY PAIR function for RSA or GENERATE ECDSA KEY PAIR function for ECDSA along with a key ID. The public key can be read using the READ RECORD command and may be used externally from the token by being included on a digital certificate establishing the relationship between the public-key and the identity of the Token Holder (User role). The private-key, which is retained securely within the Key File, is used to establish the identity of the Token Holder (User role) by generating a digital signature.

The generated RSA keys will be RSA CRT Keys, the private data ‘P’, ‘Q’, ‘dP’, ‘dQ’ and ‘QInv’ will be stored for signing operation. These values are defined in PKCS #1 version 1.5. This will get better performance on RSA (PKCS #1 v1.5) signature generation.

All asymmetric key pairs are generated according to FIPS 186-4 using the NIST SP 800-90A DRBG. A seed is produced by the on board hardware RNG and that is used as entropy-input to the DRBG instantiation process to generate internal values, *V* and *C*, which are parameters to generate the random number. The token use Hash-DRBG mechanism and does not use optional data: personalization string and additional input.

6.8 Cryptographic Key Entry

The HiKey PKI Token Manufacturing Authentication Key is input to token EEPROM by the vendor during token manufacturing.

Triple-DES Keys are input to the DES Key File in encrypted format using the LOAD KEY command with secure messaging. During this process, the keys are encrypted (using the Command/Response Encryption Key)

The public-key (RSA and ECDSA) is used externally from the token by being included on a digital certificate establishing the relationship between the public-key and the User. The certificate containing the public key may be stored on the token in a RSA / ECDSA public key file. The private-key, which is retained securely within the RSA / ECDSA private key file, is used to establish the identity of the Token Holder (User role) by forming a digital signature.

6.9 Cryptographic Key Storage

All secret and private keys are stored in plaintext format in EEPROM. The token uses the key ID to associate each key with the correct entity.

The following keys are stored on the token:

- $\mathbf{K_{TRAN}}$(Triple-DES HiKey PKI Token Manufacturing Authentication Key)
- $\mathbf{K_{ENC}}$ (Triple-DES Command/Response Encryption Key)
- $\mathbf{K_{MAC}}$ (Triple-DES Command/Response MAC verification Key)
- $\mathbf{K_{INTAUTH}}$ (Triple-DES Internal Authentication Key)
- $\mathbf{K_{EXTAUTH}}$ (Triple-DES external Authentication Key)
- $\mathbf{K_{UNLOCK}}$ (Triple-DES unlock Authentication Key)
- $\mathbf{K_{PUBVER}}$ (RSA Public Key for RSA signature verification operations)
- $\mathbf{K_{PRIVSIGN}}$ (RSA Private Key for RSA signature generation operations)
- $\mathbf{K_{ECDSA\text{-}PUBVER}}$ (ECDSA Public Key for ECDSA signature verification operations)
- $\mathbf{K_{ECDSA\text{-}PRIVSIGN}}$ (ECDSA Private Key for ECDSA signature generation operations)

All keys and the Token Holder PIN are stored in plaintext format in EEPROM.

The firmware of token inhibits the read capability of key files (except RSA / ECDSA public key and ECDSA Domain Parameters).

6.10 Cryptographic Key Destruction

The token zeroizes all secret and private cryptographic keys and CSPs using the ERASE ALL command which can only used by Chunghwa Telecom.

7 Self Tests

7.1 Power Up Self Tests

The HiKey PKI Token performs the required set of self-tests at power-up. When the token is inserted into the host PC and power is applied to the token (contact) interface, a “Reset” signal is sent from the token. The token responds (as specified by ISO/IEC 7816) with an Answer To Reset (ATR) packet of information. When the first APDU command comes into the token, the token performs a series of power-on self tests. These tests include:

- Firmware integrity check CRC32 (integrity test)
- Algorithm known answer tests for:
  - Triple-DES encrypt
  - Triple-DES decrypt
  - Triple-DES CMAC
  - Triple-DES Wrapping and Unwrapping
  - RSA sign
  - RSA verify
  - RSA encrypt
  - RSA decrypt
  - ECDSA (256 bit) sign
  - ECDSA (256 bit) verify
  - DRBG (Hash_DRBG)
  - SHA1

- SHA256
- SHA384
- SHA512

If any of these tests fail, the token will respond with a status indication of self-test error. Then, the token will go into an Error state. While in the error state, the token does not perform any operations and does not output any data.

The implementation of self-tests does not output data from the token. There is only result of self-tests output via status output interface.

After power up the token and receive the first APDU command, the token will execute Power up Self Tests automatically. If the Power up Self Tests passes, then the token process the first received APDU command, and output the normal execution result of first received APDU command. If the result of Power up Self Tests is failed, the token returns the error indicator and enters into error state. The token will never process any received APDU command afterward if the token is still in error state.

Power up self-tests could be also initiated by “KNOWN ANSWER TEST” APDU command.

Known answer tests for encryption/decryption or hashing, function by encrypting (or hashing) a string for which the calculated output is known and stored within the token. An encryption or hashing test passes when the calculated output matches the expected (stored) value. The test fails when the calculated output does not match the expected value. The test then decrypts the cipher-text string. A decryption test passes when the calculated output matches the plaintext value. A test fails when the calculated output does not match the plaintext value.

Known answer tests for Random Number Generators function by seeding the RNG with known values and checking that the output matches the pre-calculated value stored within the token. The test passes when the generated output matches the pre-calculated value. A test fails when the generated output does not match the pre-calculated value.

7.2 Conditional Tests

RSA / ECDSA Key generation:

- A pair wise consistency check is performed during key generation which consists of a sign/verify operation.

  The pair wise consistency check for sign/verify calculates and verifies a digital signature. If the digital signature cannot be verified, the test fails.

Random Number Generator:

NDRNG:

- Continuous RNG Test for the non-Approved NDRNG (HRNG, Hardware-based RNG): A continuous RNG test is performed during each use of the Hardware non-deterministic RNG to ensure that it is not generating the same value as previous one. The NDRNG is used to generate seed values to feed the DRNG.

DRBG:

- Continuous RNG Test for the SP 800-90A: A continuous RNG test is also performed during each use of the FIPS140-2 Approved Hash-based DRBG to ensure that it is not generating the same value as previous one.
- The generated bytes will compare with the bytes of the known answer testing to ensure that it is not generating the same value as the known answer testing.
- Some additional methods are used to check the strength of the randomization.

7.3 Error State when Self-Tests fail

The token will go into an Error state when self-tests fail. When the token is in this state, it will not process any APDU command anymore. To exit the error state, re-power up the token is necessary.

7.4 Other Critical Functions

- NIST SP 800-90A Section 11.3 Health Checks
  - DRBG Instantiate Critical Function Test;
  - DRBG Generate Critical Function Test; and
  - DRBG Uninstantiate Critical Function Test.
- The token does not support reseed function since the instantiate function is always performed when the random number generation is invoked. Therefore the reseed function testing specified in SP 800-90A is not required. The token has only one DRBG mechanism, so the instantiate function is never shared by any other function. The internal values that are generated by the instantiate function are local variables that will be zeroized before DRBG returns the random number and their occupied memory will be recycled by system when the process of the random number generation finish.

7.5 Bypass capability

N/A

8 Security Rules

8.1 Operational Security Rules

The following specific actions are required on the part of the Crypto Officer along with a restriction within the token usage environment to ensure the token operates in FIPS Approved mode.

1. The Crypto Officer must set all file security attribute to require a PIN for all Sign operations.
2. The Crypto Officer must set all file security attribute to require External Authenticate for all write operations.
3. The Crypto Officer must set all security attribute of key file to require External Authenticate and Key Encryption for all key update operations.
4. The Crypto Officer must set the PIN Policies for the Crypto Officer and Token Holder (User role) to have a minimum length of eight bytes (characters).
5. The Crypto Officer must set the maximum failure attempts before locking the corresponding authentication keys or PIN to against attack.
6. The Token Holder must enter a valid PIN to begin usage his/her own token.

8.2 Physical Security Rules

The physical security of the HiKey PKI Token is designed to meet FIPS 140-2 level 2 requirements. A hard opaque plastic enclosure is used to encapsulate the token to meet level 2 requirements. The tokens have two (2) tamper evident seals (as shown in Figure 1) applied to both sides of the token at manufacturing to show tamper evidence if the cover is compromised. The Crypto-Officer is responsible for the annual inspection of the integrity of the tamper-evident labels. Removal of the tamper evident seals indicate tamper evidence and require the crypto-officer to zeroize the token and perform the setup and initialization procedures specified in section 4.

8.3 Mitigation of Attacks Security Policy

The HiKey PKI Token has been designed to mitigate the following attacks:

High Frequency
High Voltage
High Temperature
Low Frequency
Low Voltage
Low Temperature
Timing Analysis, SPA, DPA, DFA Attacks
Power Analysis Attacks
Illegal Access
Illegal Instruction
Fault Attacks
EWE Interrupt
Power On Reset Function
RNG Failure

9 Security Policy Check List Tables

9.1 Roles & Required Authentication

Table 6. Roles and Required Authentication.

Role	*Type of Authentication*	*Authentication Data*
Crypto Officer	Triple-DES authentication	Triple-DES keys
User	PIN	PIN

9.2 Strength of Authentication Mechanisms

Table 7. Strength of Authentication Mechanisms.

Authentication Mechanism	*Strength of Mechanism*
PINs	The minimum length of PIN is 8 bytes and the value is not limited to digital number. Assuming that the PIN was only integers between 0-9, the probability of randomly guessing the correct sequences is 1 in 10^8.
Internal Authentication	This mechanism applies Triple-DES 3-key, the probability of randomly guessing the correct sequences is 1 in 2^{168}.
External Authentication	This mechanism applies Triple-DES 3-key, the probability of randomly guessing the correct sequences is 1 in 2^{168}.

9.3 Mitigation of Other Attacks

Table 8. Mitigation of Other Attacks

Other Attacks	*Mitigation Mechanism*	*Specific Limitations*
High Frequency	Countermeasures against high frequency	None
High Voltage	Countermeasures against high voltage	None
High Temperature	Countermeasures against high temperature	None
Low Frequency	Countermeasures against low frequency	None
Low Voltage	Countermeasures against low voltage	None
Low Temperature	Countermeasures against low temperature	None
Timing Analysis, SPA, DPA, DFA Attacks	Countermeasures against Timing Analysis, SPA, DPA, DFA Attacks	None
Power Analysis Attacks	Countermeasures against Power Analysis Attacks	None
Illegal Access	Countermeasures against illegal access	None
Illegal Instruction	Countermeasures against illegal instruction	None
Fault Attacks	Countermeasures against Fault Attacks	None
EWE Interrupt	Countermeasures using EWE interrupt	None
Power On Reset Function	Countermeasures against Power On Reset Function attack	None
RNG Failure	Countermeasures against RNG Failure attack	None

- Low/High Frequency: a hardware clock watchdog is used to detect tampering with the clock frequency and reduce the risk of a power analysis attack.
- Low/High Voltage: an integrated voltage sensor is used to detect voltage which is outside of the operating voltage range, and shutdown the token in such instances.

- Low/High Temperature: an integrated temperature sensor is used to detect extreme temperatures at which the token is not intended to operate, and shut down the token in such instances.
- Timing Analysis, SPA, DPA, DFA Attacks: A high performance cryptographic coprocessor implementing multifunctional Advanced Cryptographic Library (ACL) is available containing secure RSA calculations, various hash functions and key generation with highest protection against all currently known attacks such as SPA (Simple Power Analysis), DPA (Differential Power Analysis), DFA (Differential Fault Analysis), timing attacks and other possible hardware or software attacks.
- Power Analysis Attacks: The token has a random current generation function which will randomly disturb the current consumption of the device while in operation. Also, the token has a random bus cycle function which inserts arbitrary dummy bus cycles as counter measures to mitigate current consumption analysis.
- Illegal Access : Each authentication key or PIN stored on the token must set the maximum failure attempts before locking the corresponding authentication key or PIN to against illegal access.
- Illegal Instruction: Each role has access to specific basic token services, i.e. legal instructions. The token will return an error code when receive an illegal instruction.
- EWE Interrupt: Every time the token writes to EEPROM, it generates a non-maskable interrupt (the EWE interrupt.) When this interrupt occurs, execution is passed to a user-definable address held in the EVE vector. A user can therefore add code at this location to carry out a variety of checks, for example to confirm the integrity of data, or the context in which certain areas of EEPROM are being written.
- Fault Attacks: the token is fabricated using a MONOS (Metal Oxide Nitride Oxide Silicon) EEPROM structure. MONOS advantages compared to standard EEPROM structures are high resistance to radiation disturbance, high reliability and endurance.
- Power On Reset: All previous authentications are cleared when the token is powered down or powered on reset.
- RNG Failure: A Continuous RNG Test (CRNGT) is implemented to ensure that the DRBG generates a different value on each invocation.

Cryptographic Module References

1. HiKey PKI Token V3.1 User's Manual.

10 Standard FIPS References

National Institute of Standards and Technology, FIPS PUB 140-2: Security Requirements for Cryptographic *Modules*, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology, *FIPS 140-2 Annex A: Approved Security Functions*, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology, *FIPS 140-2 Annex B: Approved Protection Profiles*, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology, *FIPS 140-2 Annex C: Approved Random Number Generators*, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology, *FIPS 140-2 Annex D: Approved Key Establishment Techniques*, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology and Communications Security Establishment, *Derived Test Requirements (DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules*, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication 800-67 Revision 1.

National Institute of Standards and Technology, *DES Modes of Operation*, Federal Information Processing Standards Publication 81, available at URL: http://www.nist.gov/cmvp.

RFC 2313 – Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 1.5.

National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-4, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology, *Secure Hash Standard (SHS)*, Federal Information Processing Standards Publication 180-4, available at URL: http://www.nist.gov/cmvp.

NIST Special Publication 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators.

NIST Special Publication 800-131A, Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths.

NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication.

NIST Special Publication 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping.