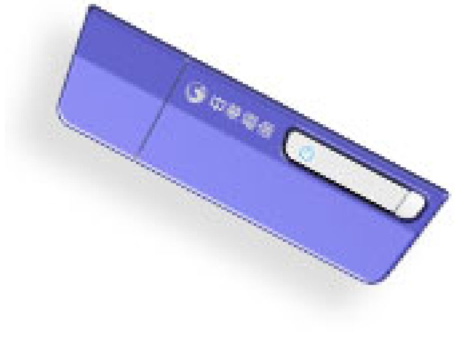
Telecommunication Laboratories
Chunghwa Telecom Co., Ltd.

Chunghwa Telecom Co., Ltd.
HiKey – Flash and HiKey PKI Token
Security Policy

FIPS 140-2 Level 2 Validation



Hardware Versions: 2.0 and 2.1
Software Version: Card OS 3.2 with PKI Applet: 2.1
Firmware Version: 2.0

November 1st, 2011
Version 2.00

1 Introduction

This document is the Security Policy for the Chunghwa Telecom Co., Ltd. HiKey – Flash and HiKey PKI Token. These modules, hereafter called the HiKey Token cryptographic module, or simply, the module, are multi-chip standalone modules that are used to provide user authentication and cryptographic services. The modules are identical in operation with the only difference being flash memory capacity for the HiKey - Flash while the HiKey PKI Token does not contain flash memory.

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

This Security Policy describes the features and design of the Chunghwa Telecom Co., Ltd. HiKey – Flash and HiKey PKI Token cryptographic modules 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 products.

The FIPS 140-2 standard, and information on the CMVP program, can be found at http://csrc.nist.gov/CMVP.

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 module 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 2.

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

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	N/A

1.2 Acronyms and Abbreviations

AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CMVP	Cryptographic Module Validation Program
CSEC	Communications Security Establishment
CSP	Critical Security Parameter
DAP	Data Authentication Pattern
DES	Data Encryption Standard
DRBG	Deterministic Random Bit Generator
ECR	EEPROM Control Register
EF	Elementary File
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
PUB	Publication
RAM	Random Access Memory
RNG	Random Number Generator
RSA	Rivest Shamir and Adleman Public Key Algorithm
SHS	Secure Hash Standard
SRDI	Security Related Data Item
X.509	Digital Certificate Standard RFC 2459

2 Chunghwa Telecom HiKey PKI Token

2.1 Functional Overview

The HiKey – Flash and HiKey PKI Token cryptographic modules contain an implementation of the Global Platform (GP) Version 2.1.1 specification defining a secure infrastructure for post-issuance programmable smart card chips. They are identical in performance with the exception of additional flash storage on the HiKey – Flash module. Global platform compliant modules have a life cycle defined by the open platform specification. Transitions between different life cycle states have well defined sequences of operation. PINS and keys that have been securely loaded at token issuance authenticate the roles of the Crypto Officer and User (Token Holder).

2.2 Cryptographic Module Specification

The HiKey PKI Token and HiKey - Flash modules are multi-chip standalone implementations of a cryptographic module. Figure 1 shows a physical view of the token module.

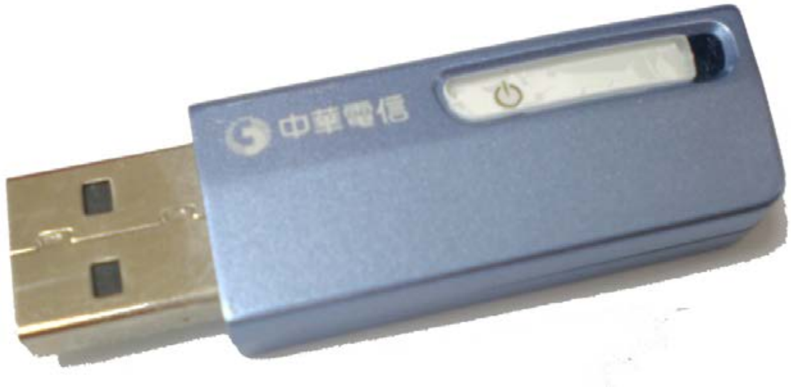


Figure 1. Physical view of the HiKey PKI Token and HiKey - Flash

The HiKey – Flash and HiKey PKI Token modules are USB modules 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 module is encased in a hard opaque tamper evident enclosure required in the FIPS 140-2 physical Level 2 validation for a multi-chip standalone implementation.

The hardware base is the Renesas AE55C1 and AE57C1 smartcard IC that is validated under the Common Criteria at EAL 4+.

The HiKey Token module consists of the following elements:

- Renesas HD65255C1 and HD65257C1 microcomputer, USB controller, and voltage regulator. These are standard, production-quality IC's.

- One HiKey applet is loaded into the EEPROM of the module at manufacturing. The applet version can be determined by a call to the applet command GET VERSION. The module has the following applet:
  - PKI applet version 2.1 – Provides RSA signing and verification services

The applet offers additional commands that the token supports, in addition to those commands provided by the basic resident (ROM-stored) software on the token. The resident ROM-stored software is referred to as the token manager. The PKI Applet provides support for signing and verification commands in support of off-token public key infrastructures. This specific applet version is validated. Loading any other applets on the token invalidates the validation of the token.

- Critical Security Parameters are stored in the EEPROM as part of the module personalization operation.

- The token is encased in 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 module has a limited operational environment consisting of a Java Virtual Machine operating on a Renesas HD65255C1 and HD65257C1 Smartcard Integrated Circuit chips. The module does not support software/firmware updates as this function is performed at the factory. The module does allow applets to be loaded, however loading of any other applets that have not been validated to FIPS 140-2 invalidates this FIPS 140-2 validation.

2.4 Module Interfaces

2.4.1 PHYSICAL INTERFACE DESCRIPTION

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

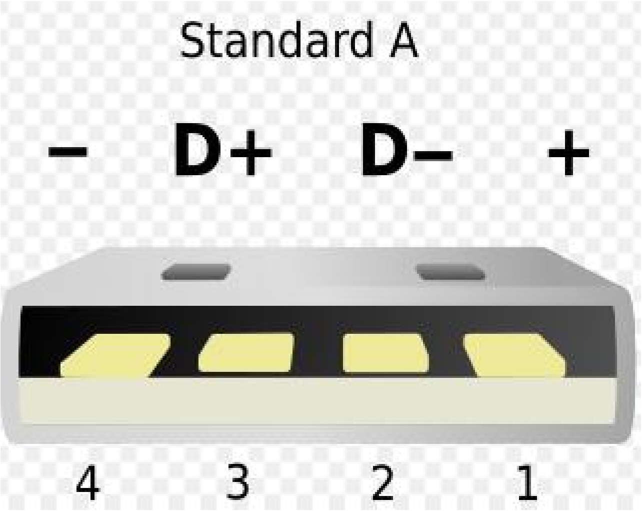


Figure 2. Interfaces

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

Module security and key access command set defined by the following specifications:

- ISO/IEC 7816-4.
- Global Platform – Open Platform – Card Specification v2.1.1 – 25 March 2003.

2.4.5 EMI/EMC

The base cryptographic module has been tested by Advance Data Technology Corporation, and found in compliance with the requirement of the following standards.

- FCC Part 15 : 2005 Subpart B, Class B.(Section 15.31,15.107, and 15.109)
- CISPR 22: 1997,Class B.(Section 5,6,9 and 10)
- ICES-003: 2004,Class B.(Section 4 and 5)
- ANSI C63.4-2003 (Section 7 and 8)

2.4.6 HOST TO TOKEN COMMUNICATIONS PROTOCOL

Host to Token module communications protocol is defined by ISO/IEC 7816-3 & 4. This is based on a standardized, half-duplex character transmission, ISO 7816 protocol. Protocol T=0 and T=1 are supported.

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 Host by the Host USB interface issuing a RESET command to the token which then responds with an "Answer To Reset (ATR)” containing the version numbers of the hard and soft masks contained on the token. 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 Global Platform 2.1.1 Specification and ISO 7816-4.

3 Roles, Services, and Authentication

3.1 Roles

The HiKey PKI token module 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 module defines three distinct roles that are supported by the on-token cryptographic system: the Crypto Officer role, a Token Holder 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.

Through the on-token applet, services are provided to the Token Holder based on his authenticating to his role. The Token Holder authenticates his role to an applet by proving knowledge of a Personal Identification Number (PIN) stored within the applet. Individual applets perform their own authentication of the Token Holder. The Global PIN is always 8 bytes. The applet PIN lengths are as follows:

- The PKI applet PIN length is 8 bytes.

The module ensures the authentication of off-Token entities (Cryptographic Officer and Token Holder) and provides them with cryptographic services according to their role. Operators may not change roles without reauthenticating in the new role. All previous authentications are cleared when the module powers down.

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

3.1.1 Cryptographic Officer Role

The Cryptographic Officer is the token administrator. The crypto officer authenticates his role on the token by demonstrating to the token manager that he possesses the knowledge of the Secure Channel Encryption Key ($\mathbf{K_{ENC}}$), Secure Channel Message Authentication Code Key ($\mathbf{K_{MAC}}$), and Key Encryption Key ($\mathbf{K_{KEK}}$) and the key ID stored within the token manager. By successfully executing a series of commands, the Cryptographic Officer establishes a secure channel to the token manager; establishment of this channel includes mutual authentication of roles between the Cryptographic Officer and the token manager. Once established, authorization (on the token) to access information and services is granted by the token manager. The token manager security domain corresponds to the token issuer security domain.

3.1.2 Token Holder

The Token Holder (User) is responsible for ensuring the ownership of his token and for not communicating his PIN. The Token Holder is authenticated by verification of a PIN selected at issuance. The PIN is provided by each applet.

3.1.3 Unauthenticated

An unauthenticated user is any unauthenticated operator having access to the host application. The operator can only select an applet and 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 or within applets 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 manager applet and using the key set to establish a secure channel. Available commands are:

SELECT: this Global Platform command is used for selecting an application (Card Manager or Applet)

DELETE: this Global Platform command is used to delete a single Load File (package) or an Application (applet instance) in the module.

ERASE ALL: this private command is used to zeroize all EEPROM contents of card.

EXTERNAL AUTHENTICATE: Global Platform command used by the module to authenticate the crypto officer, to establish a Secure Channel. A previous and successful execution of the **INITIALIZE UPDATE** command is required prior to processing this command.

GET DATA: this Global Platform command is used to retrieve a single, tagged data object from the Card Manager. Card Manager data objects are define in the GPv2.1.1 specification.

GET STATUS: this Global Platform command is used to retrieve Card Manager, Executable Load File and Application related life cycle status information according to a given search criteria.

INITIALIZE UPDATE: this Global Platform command is used to initiate a Secure Channel with the Card Manager. Card and host session data are exchanged, and session keys ($\mathbf{K_{enc}}$ & $\mathbf{K_{mac}}$) generated by the card. The Secure Channel is considered open upon completion of a successful **EXTERNAL AUTHENTICATE** command that must immediately follow the **INITIALIZE UPDATE** command.

INSTALL: this Global Platform command is used to install an application or a Security Domain and requires the invocation of several different module functions. The command is used to instruct a Security Domain or the Card Manager as to which installation step it shall perform during an application installation process.

LOAD: this Global Platform command is used to load the byte-codes of a Load File (package).

PIN CHANGE/UNBLOCK: this command is used to set the PIN value, retry limit, or retry counter of the Global PIN. The Crypto Officer establishes the secure channel for this command.

STORE DATA: this Global Platform command stores or replaces one tagged data object modifies the life cycle state of the card or the lifecycle state of an application defined in the GPv2.1.1 specification.

PUT KEY: this Global Platform command is used to add or replace Security Domain key sets. A PUT KEY command with a key-set of all 0xFF will zeroize specified Security Domain key sets.

MANAGE CHANNEL: this command is used to open or close a logical channel

User Services
An operator authenticates in a user role by proving knowledge of a PIN. Available commands are:

GET DATA: this Global Platform command is used to retrieve a single, tagged data object from the Card Manager. Card Manager data objects are define in the GPv2.1.1 specification.

SELECT: this Global Platform command is used for selecting an application (Card Manager or Applet).

MANAGE CHANNEL: this command is used to open or close a logical channel

Unauthenticated Services
Any operator with access to the host application can give some commands that do not require any authentication. These commands are:

SELECT: this Global Platform command is used for selecting an application (Card Manager or Applet).

GET DATA: this Global Platform command is used to retrieve a single, tagged data object from the Card Manager. Card Manager data objects are define in theGPv2.1.1 specification.

SELF TEST: this command will run self-tests on demand.

MANAGE CHANNEL: this command is used to open or close a logical channel

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	*Authorized Services*	*Cryptographic Keys and CSPs*	*Type(s) of Access*
Crypto-Officer	SELECT	None	Execute
	DELETE	None	Execute
	ERASE ALL	Encryption Key, Triple-DES MAC Key	Execute, Write
	EXTERNAL AUTHENTICATE	Encryption Key, Triple-DES MAC Key	Execute
	GET DATA	None	Read
	GET STATUS	None	Read
	INITIALIZE UPDATE	Encryption Key, Triple-DES MAC Key	Execute
	INSTALL	None	Execute
	LOAD	Data Authentication Pattern Key	Execute
	PIN CHANGE/UNBLOCK	Encryption Key, Triple DES MAC Key	Execute,
	STORE DATA	None	Write
	PUT KEY	Encryption Key, Triple-DES MAC Key, Key Encryption Key	Write
	MANAGE CHANNEL	None	Execute
User	GET DATA	None	Read
	SELECT	None	Execute
	MANAGE CHANNEL	None	Execute
Unauthenticated	SELECT	None	Execute
	GET DATA	None	Read
	SELF TEST	None	Execute
	MANAGE CHANNEL	None	Execute

Table 3. Basic Token Service Access Controls

3.2.2 *PKI Applet Services*

The PKI applet provides RSA sign and verify services to authenticated users. The PKI applet Cryptographic Officer services are:

SELECT FILE: Select file from the PKI Applet storing the key or PIN values. This may be an RSA public key file, private file, or certificate file.

READ BINARY: Read binary data from a Transparent EF (elementary file) and read public key from a public key EF.

UPDATE BINARY: Write binary data into a Transparent EF and write public and private key to key file EF.

UNBLOCK PIN: Unblock the user PIN.

PERSONALIZE APPLET: This command is used to initialize the applet.

GENERATE HASH: Create a HASH of the given data.

CHANGE PIN: Change the user PIN

GET APPLET VERSION. Get applet version number. Response is one byte: 0x21.

GET STATUS: Get the life cycle of the PKI applet

User Services
An operator authenticates in a user role by proving knowledge of a PIN. Available commands are:

SELECT FILE: Select file from the PKI Applet storing the key or PIN values. This may be an RSA public key file, private file, or certificate file.

VERIFY PIN: Verify the PIN value presented by user with the PIN value stored inside the PKI Applet.

READ BINARY: Read binary data from a Transparent EF (elementary file) and read public key from a public key EF.

GENERATE KEY PAIR: Create a new RSA key pair.

RSA SIGN: Create a RSA signature.

RSA VERIFY: Verify a RSA signature.

GENERATE HASH: Create a HASH of the given data.

GET APPLET VERSION. Get applet version number. Response is one byte: 0x21.

GET STATUS: Get the life cycle of the PKI applet

Unauthenticated Services
Any operator with access to the host application can give some commands that do not require any authentication. These commands are:

SELECT FILE: Select file from the PKI Applet storing the key or PIN values. This may be an RSA public key file, private file, or certificate file.

READ BINARY: Read binary data from a Transparent EF (elementary file) and read public key from a public key EF.

GET STATUS: Get the life cycle of the PKI applet

GET APPLET VERSION. Get applet version number. Response is one byte: 0x21.

The authenticated token holder has access to PKI applet services. The services, in turn, may use or operate on cryptographic keys or critical security parameters (CSPs). The following table shows the relationship between roles and services, and indicates the type of access provided to various cryptographic keys and CSPs. In cases where the Cryptographic Keys and CSPs are "None", the Type(s) of Access identifies the type of operation.

Role	***Authorized Services***	***Cryptographic Keys and CSPs***	***Type(s) of Access***
Crypto-Officer	SELECT FILE	None	Execute
	READ BINARY	RSA Public Key	Read
	UPDATE BINARY	RSA Public/Private Keys	Write
	UNBLOCK PIN	PIN	Execute
	PERSONALIZE APPLET	None	Execute
	GENERATE HASH	None	Execute
	CHANGE PIN	PIN	Write
	GET APPLET VERSION	None	Read
	GET STATUS	None	Read
User	SELECT FILE	None	Execute
	VERIFY PIN	PIN	Execute
	READ BINARY	RSA Public Key	Read
	GENERATE KEY PAIR	RSA Public/Private Keys	Execute
	RSA SIGN	RSA Private Key	Execute
	RSA VERIFY	RSA Public Key	Execute
	GENERATE HASH	None	Execute
	GET APPLET VERSION	None	Read
	GET STATUS	None	Read
Unauthenticated	SELECT FILE	None	Execute
	READ BINARY	RSA Public Key	Read
	GET STATUS	None	Read
	GET APPLET VERSION	None	Read

Table 4. PKI Applet FIPS Approved Service Access Controls

RSA CRYPTOGRAPHY: RSA encryption/decryption with an RSA public key or private key (Non-FIPS Approved service)

Role	***Non-Authorized Services***	***Cryptographic Keys and CSPs***	***Type(s) of Access***
User	RSA Cryptography	RSA Public/Private Key	Execute

Table 5 PKI Applet Non-FIPS Approved Service Access Controls

3.3 Authentication

3.3.1 Triple DES keys

Each of the three-key Triple DES keys used by the CO has an effective key length of 112 bits (which is 2^{112} possible keys per key or 2^{348}). As all three keys are required, this far exceeds the 1 in a million test requirement.

To try a key against the module, an attacker must send a minimum 13 byte APDU to the token, and get a resulting 2-byte response. Each triple-DES key attempt requires 15 bytes of data to be clocked in or out of the token. The maximum data rate for the module is 38,400Kbps through the port. Ignoring the processing time required on the module 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 = 120bits/attempt
- 120bits/attempt divided by 38,400bits/second = .003125 seconds/attempt
- 60seconds/minute divided by .003125 seconds/attempt = 19,200attempts/minute

As the Triple-DES key space is over 2^{348} 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 module far exceeds the requirement of 1 in 100,000 for multiple attempts in a 60 second interval.

3.3.2 Global PIN and User PIN

The length of the Global PIN and the User PIN is a string of 8 digits. PINs contain the digits 0 to 9 yielding a maximum of 100,000,000 possible PINs. This far exceeds the 1 in a million test.

An 8-bit counter internal to the Access Control applet limits the number of failed PIN attempts an attacker could perform by blocking the token if the counter limit (3 attempts per PIN) is exceeded. This far exceeds the requirement of 1 in 100,000 for multiple attempts in a 60 second interval.

4 FIPS-Approved Mode of Operation

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

1. Pre-Personalize the HiKey PKI Token by the following steps.
   a. initialize the token;
   b. load the transport key (Triple DES) and perform the GET CHALLENGE and EXTERNAL AUTHENTICATE commands. If the key is authenticated, the token is fully initialized;
2. Issue the PUT KEY command to generate a new key set;
3. Select the issuer security domain;
4. Load the applet verification key into the Token Manager;
5. Create a secure session using the INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands
6. Issue the PUT KEY command to create a new Applet Verification Key;
7. Set the Applet Verification Secure Domain to the Personalization State;
8. Configure the applets by:
   a. setting a token holder PIN.

From this point, the module and applets are in FIPS approved mode.

5 Module Cryptographic Functions

The purpose of the HiKey PKI Token module is to provide a FIPS validated module for applets that may in turn provide cryptographic services to end-user applications. 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 module to provide cryptographic services; these include:

- Triple-DES for establishing a secure channel, and encrypting keys input to the module using the PUT KEY command within the secure channel.
- Triple-DES-MAC used for data authentication and applet load test;
- AES MAC used for data authentication (Non-FIPS Mode);
- HMAC-SHA-1 and HMAC-SHA-2 used for integrity-protecting data sent within the secure channel;
- AES for encrypting data stored within the applet;
- RSA Sign and Verify;
- SHA-1 and SHA-2 Hashing;
- NIST SP 800-90 DRBG used for cryptographic key generation;
- Hardware RNG used for seeding the NIST SP 800-90 DRBG ;

Details of cryptographic functions are shown in this table:

Type	Algorithm	FIPS-Approved	Certificate
Public/Private Keys	RSA. Key size: 1024 and 2048 bits.	Yes (PKCS#1)	839
Symmetric Key	Triple-DES (ECB, CBC) 2 key TDES. Triple-DES (ECB, CBC) 3 key TDES.	Yes (NIST SP 800-67)	1100
	AES (ECB, CBC) Key Sizes 128,192,256 bits.	Yes (FIPS 197)	1710
Keyed Hash	Triple-DES MAC	Vendor affirmed	Triple-DES Cert 1100
	AES MAC	NO	
	HMAC-SHA-1	Yes (FIPS 198)	988
	HMAC-SHA-256	Yes	988
	HMAC-SHA-384	Yes	988
	HMAC-SHA-512	Yes	988
	HMAC-MD5	NO	
	HMAC-RIPEMD160	NO	
Digest	SHA-1	Yes (FIPS 180-3)	1493
	SHA-256	Yes	1493
	SHA-384	Yes	1493
	SHA-512	Yes	1493
	MD5	NO	
	RIPEMD160	NO	
RNG	DRBG (NIST SP 800-90)	Yes (NIST SP 800-90)	106
	NDRNG (HARDWARE RNG)	NO	
Asymmetric Encryption	RSA	NO	

Table 6. Module Cryptographic Functions.

5.1 Algorithm Deprecation Statements

As of January 2011 the following algorithms are restricted or deprecated:

- 1024 bit RSA;

Please refer to NIST Special Publication 800-131A for more information.

6 Cryptographic Key Management

The module contains a variety of keys and CSPs defined by the Global Platform specification and by the applet design documents. The module does not input or output plaintext cryptographic key components, plaintext authentication data, or other unprotected CSPs.

6.1 Cryptographic Keys and Critical Security Parameters

The HiKey PKI Token module includes the following keys:

- Initialization Key, $\mathbf{K_{init}}$ Triple DES key (128 bits) used only for the first Token Manager key-set loading.
- Applet Load Key, $\mathbf{K_{ALD}}$ Triple DES MAC key used to create a MAC on an applet loaded on the token to verify its authenticity.
- Crypto Officer Security Domain double-length keys ($\mathbf{K_{ENC}}$, $\mathbf{K_{MAC}}$, & $\mathbf{K_{KEK}}$).
- Hash DRBG Internal State Values (V and C)

A Security Domain key set is structured to contain three types of Triple-DES keys:

a. $\mathbf{K_{ENC,}}$ used to generate Triple-DES session key $\mathbf{K_{enc}}$ for the encrypted mode of the secure channel,
b. $\mathbf{K_{MAC}}$, used to generate Triple-DES session key $\mathbf{K_{mac}}$ for MAC mode of the secure channel authentication,
c. $\mathbf{K_{KEK}}$, used to encrypt keys to be imported into the module.

Security Domains allow a number of distinct identities to be established on the HiKey PKI token module. These identities control access to the various applets stored on the module. A Security Domain represents the identity of the Crypto Officer.

6.2 Public Keys

Public and private keys can be generated on the token using the PKI applet GENERATE KEY PAIR command. Alternatively the keys may be loaded onto the token using the UPDATE BINARY command.

$\mathbf{K_{SIGN}}$ (PKI Key pair)

- RSA Public Sign Key, $\mathbf{K_{PUBSIGN}}$ for signature verification operations.
- RSA Private Key for Sign operations $\mathbf{K_{PRIVSIGN}}$

6.3 Cryptographic Key Generation

RSA key pairs may be generated using the GENERATE KEY PAIR (PKI applet) function along with a key ID. The public key is returned from the function and may be used externally from the module by being included on a digital certificate establishing the relationship between the public-key and the identity of the Token Holder. The private-key, which is retained securely within the PKI container, is used to establish the identity of the Token Holder by verifying a digital signature.

Triple-DES and AES keys are generated according to NIST SP 800-90 hash based DRBG. RSA keys are generated according to the ANSI X9.31 key generation standard.

6.4 Cryptographic Key Entry

Security Domain Keys are input to the Token Manager in encrypted format, using the PUT KEY command within a secure channel. During this process, the keys are double encrypted (using the Triple-DES Session Key $\mathbf{K}_{\mathbf{enc}}$ and the $\mathbf{K}_{\mathbf{KEK}}$ Key).

The public-key is used externally from the module 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 PKI applet container. The private-key, which is retained securely within the PKI container, is used to establish the identity of the Token Holder by forming a digital signature.

6.5 Cryptographic Key Storage

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

The following keys are stored on the module:

- $\mathbf{K}_{\mathbf{ENC}}$ (Triple-DES Encryption Key)
- $\mathbf{K}_{\mathbf{MAC}}$ (Triple-DES MAC Key)
- $\mathbf{K}_{\mathbf{KEK}}$ (Triple-DES Key Encryption Key)
- $\mathbf{K}_{\mathbf{ALD}}$ (Triple-DES MAC Applet Load Key)

Symmetric session keys reside in RAM only and become invalid when a secure channel session ends.

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

6.6 Cryptographic Key Destruction

The module zeroizes secret and private cryptographic keys and CSPs using the ERASE ALL command.

The PUT KEY command can be used to zeroize the crypto-officer key set by specifying a key value of all “F”s for all crypto-officer keys.

7 Self Tests

7.1 Power Up Self Tests

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

- RAM functional test and clearing at Reset
- Firmware integrity check (CRC32)
- Algorithm (known answer) tests for:
  - o Triple-DES (CBC mode encrypt/decrypt)
  - o Triple-DES MAC
  - o AES (CBC mode encrypt/decrypt)
  - o HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512(using the $K_{(MAC)}$ key)
  - o RSA PKCS1 (sign and verify)
  - o DRBG

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

No data of any type is transmitted from the module to the reader while self-tests are being performed. The firmware self-test operations do not implement any capability to output data from the module. The only output is status data indicating self-test success or failure**.** If the self-test operation is successful, the module will execute the first received APDU command, and output the normal execution result of first received APDU command.

Known answer tests for encryption/decryption, signatures, and hashing, functions by encrypting/decrypting, signing, (or hashing) a string for which the calculated output is known and stored within the cryptographic module. An encryption, signature, 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.

Known answer tests for Random Number Generators function by seeding the DRBG with known values and checking that the output matches the pre-calculated value stored within the cryptographic module. 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 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:

- HRNG: A continuous RNG test is performed during each use of the Hardware non-deterministic RNG to verify that it is not generating the same value. The HRNG is used to generate seed values to feed the DRBG.
- DRBG: A continuous DRBG test is also performed during each use of the FIPS140-2 approved DRBG to verify that it is not generating the same value.

Software/Firmware load test

- A Triple-DES CBC MAC is verified each time an applet is loaded onto the HiKey Token module. (Only validated applets may be loaded onto the HiKey PKI Token. Loading of non-validated applets will invalidate the module's FIPS 140-2 Certification.) The HiKey Token does not support firmware upgrades. This has to be performed at the factory.

8 Security Rules

8.1 Secure distribution and delivery procedure

The secure distribution and delivery procedures for the HiKey - Flash and HiKey PKI Token are detailed in the "Chunghwa Telecom Co., Ltd. HiKey - Flash and HiKey PKI Token Delivery and Operation" document.

8.2 Operational Security Rules

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

1. The Crypto Officer must instantiate all token applets to require a PIN for all Sign operations.
2. The Crypto Officer must instantiate all container applets to require External Authenticate or Global Platform secure channel for all write operations.

3. The Crypto Officer must set the PIN Policies for the crypto officer and Token Holder to have a minimum length of eight bytes (characters).
4. The Crypto Officer must set the incorrect PIN counter to three failed attempts before locking the token.
5. The Token Holder must enter a valid PIN.
6. The Token Holder must generate or upload an RSA key pair to configure the PKI applet to generate or verify digital signatures.
7. For key and CSP zeroization purposes, a crypto officer may use the ERASE ALL command

8.3 Physical Security Rules

The physical security of the HiKey – Flash and HiKey PKI Token modules are designed to meet FIPS 140-2 level 2 requirements. A hard opaque plastic enclosure is used to encapsulate the module to meet level 2 requirements. The tokens have tamper evident seals applied to both sides of the token at manufacturing to show tamper evidence if the cover is compromised. Removal of the tamper seals that show tamper evidence will require the crypto-officer to zeroize the module and perform the setup and initialization procedures specified in section 4.

8.4 Mitigation of Attacks Security Policy

The module does not claim to mitigate against any specific attacks.

9 Security Policy Check List Tables

9.1 Roles & Required Authentication

Role	Type of Authentication	Authentication Data
Crypto Officer	Triple-DES authentication	Triple-DES keys (Crypto Officer Security Domain) **K_{ENC}** , **K_{MAC}** & **K_{KEK}**
User	PIN	Global PIN, Token Holder PIN

Table 7. Roles and Required Authentication.

9.2 Strength of Authentication Mechanisms

Authentication Mechanism	PIN Length	Key Bit-lengths	Strength of Mechanism
Triple-DES authentication	8 bytes	128-bits	High (Far exceeds the 1 in a million test)
PIN-based authentication	8 bytes		High (Far exceeds the 1 in a million test)

Table 8. Strength of Authentication Mechanisms.

9.3 Access Rights within Services

Service	*CSP*	*Type of Access (eg. Read, Write, Execute)*
Crypto Officer		
EXTERNAL AUTHENTICATE (Secure Channel)	Triple-DES Crypto Officer Keys	Execute
PUT KEY	Triple-DES Crypto Officer Keys	Write
PIN CHANGE/UNBLOCK	Triple-DES Crypto Officer Keys	Write
User		
Verify PIN	Token Holder PIN	Read
RSA CRYPTO	RSA $\mathbf{K}_{\mathbf{PRIVSIGN}}$	Execute
RSA CIPHER	RSA $\mathbf{K}_{\mathbf{PRIVSIGN}}$	Execute

Table 9. Access Rights Within Services.

10 Cryptographic Module References

1. Application Programming Interface Java Card™ Platform, Version 2.2.1 – October 21, 2003.
2. Global Platform – Open Platform – Card Specification v2.1.1 – 25 March 2003.
3. Appendix One of Financial Information System Design Specification V1.4 – 1 August 2003

11 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, *Data Encryption Standard (DES)*, Federal Information Processing Standards Publication 46-3, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-2, available at URL: http://www.nist.gov/cmvp.

National Institute of Standards and Technology, *Secure Hash Standard (SHS)*, Federal Information Processing Standards Publication 180-3, available at URL: http://www.nist.gov/cmvp.