Turbo Cryptographic Card (TCC)

Security Policy

SP-14094-0

Revision B

December 14, 1998

Originator: Jim Dawson

Approval: _____

Proj. Eng. Mgr.

Scope of Document

This document contains the security policy requirements for the *Cylink* Turbo Crypto Card (TCC) module. The module shall be referred to as the TCC in this document.

Applicable Documents

- FIPS 140-1 *Security Requirements for Cryptographic Modules*
- DTR *Derived Test Requirements for FIPS 140-1, Security Requirements for Cryptographic Modules (DTR)*
- FIPS 46-2 *Data Encryption Standard (DES)*
- FIPS 81 *DES Modes of Operation*
- FIPS 180-1 *Secure Hash Standard (SHA-1)*
- FIPS 186 *Digital Signature Standard (DSS)*
- X9.42 *Public Key Cryptography for the Financial Services Industry (DH)*
- X9.52 *Triple Data Encryption Algorithm Modes of Operation*
- IS-14094-0 *Internal Specification Turbo Crypto Card*
- ES-14825-001 *External Specification Cryptographic Toolkit (Library) Rev. A*

Security Level

The TCC shall meet the overall requirements applicable to security Level 1 of FIPS 140-1. Table **1** defines the specific security level for each of the 11 requirement categories:

Table 1. Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module	1
Module Interfaces	1
Roles and Services	1
Finite State Machine	1
Physical Security	1
EFP/EFT	N/A
Software Security	1
Operating System Security	N/A
Key Management	1
Cryptographic Algorithms	1
EMI/EMC	3

Self Test	1

Roles and Services

The TCC module supports two roles as required by FIPS 140-1. These roles are the User Role and the Crypto-Officer Role. There is no role identification or authentication required for FIPS security level 1. The services provided to these two roles are the same and are described below.

The *User Role* shall provide all of the services necessary to support cryptographic key management, and for the secure transport of data over an insecure network. See *Internal Specification Turbo Crypto Card* for detailed information on the formats and protocols for using these services. The services provided by the TCC module include:

- Get Status: This service returns the TCC's current operational status.
- Get Methods: This service returns a list of the encryption methods supported by the TCC.
- Get Version: This service returns the TCC's software version number.
- Force Tamper: This service clears all SRDIs that are stored in the Hidden Memory Area (HMA).
- Read Time: This service returns the current time from the TCC's real time clock.
- Write FLASH: This service writes data into the TCC flash parameter area. (Note: this area is for host parameter permanent storage - not used for SRDI data).
- Read FLASH: This service reads data from the TCC flash parameter area. (Note: this area is for host parameter permanent storage - not used for SRDI data).
- HMA Allocate: This service allocates a block of memory from the Hidden Memory Area (HMA) of the requested size and storage type. The storage type defines the memory block's level of access and restricts its usage. Details concerning storage type are found in the Key Management section below.
- HMA Read: This service allows the user to read from the contents of a particular HMA block if this block was allocated with the one of the storage types allowing read access. Details concerning storage type are found in the Key Management section below.

- HMA Write: This service allows the user to write to the contents of a particular HMA block if this block was allocated with the one of the storage types allowing write access. Details concerning storage type are found in the Key Management section below.

- HMA Copy: This service allows the user to copy data from one HMA block to another if these blocks were allocated with the storage types allowing copy access. Details concerning storage type are found in the Key Management section below.

- HMA Free: This service de-allocates a given HMA block. The data in the block is zeroized prior to the release of the memory area.

- Calculate HMA Block Syndrome: This service computes the "syndrome" of an HMA data block. This is done by repeated folding of the data in half and exclusive - ORing the halves until the data block has been reduced to 64 bits in length.

- Read Random: This service generates a random number and outputs the value to the host.

- Generate Random: This service generates a random number and stores the value in the indicated HMA block.

- Secure Hash Initialize: This service resets the SHA-1 algorithm. The internal registers are reset to the initial values given in FIPS 180-1.

- Secure Hash Update: This service updates the state of the SHA-1 register. It is used to update the SHA-1 algorithm using an intermediate block as input.

- Secure Hash Finish: This service takes the last input data block and computes the final SHA-1 result. Padding of this last block is performed if necessary.

- Secure Hash: This service combines the operation of the previous SHA-1 functions.

- Generate DSS Public/Private Number: This service generates the public and private keys used in generating DSS signatures.

- Generate DSS Signature: This service computes a DSS signature. This service requires the SHA-1 hash value of the message as input.

- Verify DSS Signature: This service verifies a DSS signature. This service requires the SHA-1 hash value of the message as input.

- Generate Diffie-Hellman (DH) Key Pair: This service generates the local private random component, computes and outputs the public component. The local private component is stored in HMA.

- Generate DH Shared Number: This service inputs the remote public component, and uses this value in conjunction with the local private component to compute a random secret number shared with the remote TCC. The secret shared number is stored in HMA.

- Generate DH Key: This service uses the DH secret shared number (stored in HMA) and the desired encryption method to compute an encryption key, which is stored in HMA.

- DES Encrypt: This service encrypts a block of data of the requested size without using accelerator hardware. The user specifies the algorithm type and an HMA block containing the encryption key to be used.

- DES Decrypt: This service decrypts a block of data of the requested size without using accelerator hardware. The user specifies the algorithm type and an HMA block containing the decryption key to be used.

- Packet Encryption: This service encrypts a packet of data using accelerator hardware. The user specifies the algorithm type and an HMA block containing the encryption key to be used.

- Packet Decryption: This service decrypts a packet of data using accelerator hardware. The user specifies the algorithm type and an HMA block containing the decryption key to be used.

Security Rules

This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-1 Level 1 module[1].

Cryptographic Module

1. *The TCC module shall be implemented as a "Multi-chip Embedded" module as defined in FIPS 140-1.*

2. *The TCC module shall confirm that the on-board firmware load is valid using a standard checksum before allowing access to services.*

[1] Rules are contained in the number paragraphs and are shown in italics. Other information is included for background purposes only.

Module Interfaces

1. *The TCC module shall contain the following interfaces:*

   a) *Data Input Interface*
   b) *Data Output Interface*
   c) *Control Input Interface*
   d) *Status Output Interface*
   e) *Power Interface*

2. *The Data Input, Data Output, Control Input, Status Output, and Power interfaces shall be implemented as part of the PCI bus interface.*

   (Note: As a consequence, these interfaces shall be considered as a combined interface. This is permissible by FIPS 140-1 for security levels 1 and 2. If no SRDIs in plaintext form are input or output over the PCI bus this architecture may be used at level 3.)

Roles and Services

1. *The TCC module shall implement the User Role and the Crypto Officer Role as specified in the section "Roles and Services" above.*

Physical Security

1. *The TCC module shall provide a tamper switch connector that causes all non-volatile cryptographic data to be electrically erased and generates a non-maskable interrupt to allow power-on tamper detection.*

2. *All chips used on the TCC module shall include the use of industry standard passivation techniques.*

EFP/EFT

This section is not applicable.

Software Security

1. *The software shall be implemented using a high-level language whenever practical.*

Operating System Security

This section is not applicable.

Key Management

1. *The TCC module shall provide for the storage of data such as keys, etc. internal to the module in memory to be known as the Hidden Memory Area (HMA).*

2. *HMA blocks shall be accessed using a "handle" which represents the storage location of the block within the TCC's memory.*

3. *Each HMA block shall have a storage type with one or more of the following attributes:*

   Read: When set this HMA memory block may be read by the host.
   Write: When set this HMA memory block may be written to by the host.
   Key: When set designates this HMA memory block as a location for the storage of secret or private keys. It cannot be read or written by the host.
   DSS: When set designates this HMA memory block is used as secret part of the DSA Signature process. It cannot be read or written by the host.
   DH: When set designates this HMA memory block is used as secret part of the Diffie-Hellman Key Agreement process. It cannot be read or written by the host.

4. *Keys allocated in HMA data blocks with the "Key" attribute bit set shall not be allowed to also have either a "Read" or "Write" attribute setting.*

5. *Services requiring a Diffie-Hellman secret key may only use keys allocated in HMA data blocks with the "Key" and "DH" attribute setting.*

6. *Services requiring a DSA secret key may only use keys allocated in HMA data blocks with the "Key" and "DSS" attribute setting.*

7. *Services that copy HMA blocks shall only copy keys that have an equal level of access as defined by their attribute settings.*

8. *Services that read from or write to HMA blocks shall not have access to HMA blocks without the corresponding "Read" or "Write" attribute setting.*

Note: As a result of the above rules, private and secret keys may not be loaded into or output from the TCC card. Private and secret keys can only be generated and stored internally to the TCC card. Services requiring the use of any of these private or secret keys can only use those that have been generated on the card.

Cryptographic Algorithms

1. *The TCC module shall employ the Data Encryption Standard encryption algorithm as defined in FIPS 46-2. This encryption algorithm shall employ the following modes as defined in FIPS 81:*

   a) 8-bit Cipher Feedback (CFB) mode
   b) 64-bit Cipher Feedback (CFB) mode

c) Cipher Block Chaining (CBC) mode
d) Electronic Code Book (ECB) mode

2. *The TCC module shall employ the Triple Data Encryption Standard encryption algorithm as defined in X9.52This encryption algorithm shall employ the following modes as defined in X9.52:*

   a) 8-bit Cipher Feedback (CFB) mode
   b) 64-bit Cipher Feedback (CFB) mode
   c) Cipher Block Chaining (CBC) mode
   d) Output Feedback (OFB) mode
   e) Electronic Code Book (ECB) mode

3. *The TCC module shall also support the Data Encryption Standard encryption algorithm implemented with the key shortened to 40 bits. The same modes that are supported in the standard implementation shall also be supported in this shortened key version.*

   The shortened key version of DES is included within the product to provide compatibility with exportable versions of the product.

4. *The TCC module shall support a Cylink proprietary encryption algorithm known as Pipelined TDES.*

5. *The TCC module shall support a Cylink proprietary encryption algorithm known as SAFER.*

6. *The TCC module shall support a Cylink proprietary encryption algorithm known as TPASS.*

7. *The TCC module shall support the proposed X9.42 DH algorithm for Diffie-Hellman key exchange.*

8. *The TCC module shall employ the Secure Hash Algorithm as defined in FIPS 180-1 for the computation of message digests.*

9. *The TCC module shall employ the Digital Signature Algorithm as defined in FIPS 186 for the computation and verification of digital signatures.*

10. *The TCC module shall employ an SHA based random number generation algorithm as specified in FIPS 186 Appendix 3 using the value in RVAL (generated and tested during system initialization and test states) for the seed key.*

<u>EMI/EMC</u>

1. *The TCC module shall meet the requirements of FCC Part 15, Subpart J, Class B.*

Self Test

1. *Each time power is applied to the TCC module, the TCC module shall confirm that the ROM memory used to store the operational code is unaltered using a standard checksum technique. The TCC module shall also confirm that the RAM memory used for operational storage is operational using a standard memory testing technique.*

2. *Each time power is applied to the TCC module, the TCC module shall perform the following power on self-tests prior to providing any cryptographic services.*

   a) *A known answer test for encryption hardware using DES in both the encrypt and decrypt state (CFB8, CFB64, CBC64, ECB).*

   b) *A known answer test for encryption hardware using TDES in both the encrypt and decrypt state (if the CY1047 ASIC is present) (CFB64).*

   c) *A known answer test for encryption hardware using SAFER in both the encrypt and decrypt state (if the CY1056 ASIC is present) (CFB8, CFB64, CBC64, ECB).*

   d) *A known answer test for the modular arithmetic integrated circuit using modulo addition, multiplication, and exponentiation.*

   e) *A known answer test for encryption software using DES in both the encrypt and decrypt state (CFB8).*

   f) *A known answer test for the Secure Hash Algorithm software.*

   g) *A known answer test for the Digital Signature Algorithm software.*

   h) *A known answer test for the Diffie-Hellman key generation software.*

   i) *A set of statistical tests of the Random Number generation software (i.e. the Monobit, Poker, Runs, and Long Run tests defined in FIPS140-1 section 4.11.1).*

3. *The TCC module shall perform the following power on conditional self-tests.*

   a) *A continuous random number generator test to be used each time that the random number generator function is invoked (as described in FIPS140-1 section 4.11.2).*

   b) *A pair-wise consistency test to be used each time that a DSS key is generated (as described in FIPS140-1 section 4.11.2).*

Definition of Security Relevant Data Items

The following security relevant data items (SRDIs) are identified:

a) Manufacturing Certificate: This is the certificate that is produced and signed by the Cylink Certification Authority. Contained within this certificate is the following information:

(i) Certificate Type
(ii) Product Type
(iii) Customer ID Number
(iv) Product Serial Number
(v) Product Software Version
(vi) Local Public DSS Signature Key
(vii) Cylink Certificate Authority (CCA) Public DSS Signature Key
(viii)Signature of Certificate based on CCA DSS Signature Key

b) Local Network Certificate: This is the certificate that is produced and signed by the Secure Manager. Contained within this certificate is the following information:

(i) Certificate Type
(ii) Product Type
(iii) Software Version
(iv) Local Public DSS Signature Key
(v) Certificate Revision Number
(vi) Certificate Activation Time
(vii) Certificate Switch Time
(viii)Certificate Renewal Time
(ix) Certificate Expiration Time
(x) Signature of Certificate based on the Secure Manager's Signature Key

c) Remote Network Certificate: This is the certificate that is produced and signed by the Secure Manager. Contained within this certificate is the following information:

(i) Product Type
(ii) Customer ID
(iii) Software Version
(iv) Crypto Method Vector
(v) Node ID
(vi) Units DSS Public Number
(vii) Revision Number
(viii)Node's Group Definitions
(ix) Signature of Certificate based on the Secure Manager's Signature Key
(x)

d) DSA Public Parameters: This is the public value used in the calculation and verification of DSA signatures. They are represented by the variables *p*, *q*, and *g* in FIPS 186.

e) Secure Manager DSA Public Key: This is the public component of the Secure Manager's DSS Signature Key.

f) Local DSA Public Key: This is the local public key used in the calculation of DSA signatures. This value is represented by *y* in FIPS 186.

g) Local DSA Private Key: This is the private component of the public/private DSA key pair. The corresponding public component is contained within the Local Network certificate.

h) Message Signature: This is a FIPS 186 message signature.

i) Message Digest: This is a FIPS 180-1 message digest.

j) Diffie-Hellman (DH) Modulus (p): This is the public modulus used in the Diffie-Hellman Key Agreement protocol.

k) DH Base (α): This is the public base used in the Diffie-Hellman Key Agreement protocol.

l) DH Local Secret (x): This is the locally generated random secret used in the Diffie-Hellman Key Agreement protocol.

m) DH Local Public Number (y_l): This is the locally computed public number used in the Diffie-Hellman Key Agreement protocol. This value is computed as follows:

$$y_l = (\alpha)^x \bmod p$$

n) DH Remote Public Number (y_r): This is the public number computed by the remote TCC that is used in the Diffie-Hellman Key Agreement protocol.

o) DH Shared Secret (s): This is the computed secret that is produced by the Diffie-Hellman Key Agreement protocol. This value is computed as follows:

$$s = (y_r)^x \bmod p$$

p) DES/TDES Key: This is the key used in DES/TDES encryption/decryption for packet data and for data encrypted/decrypted using remote procedure call encryption/decryption functions. This key is accessed by the use of a handle that points the appropriate HMA data block.

q) DES/TDES Initialization Vector: This is the initialization vector used in DES/TDES encryption/decryption for packet data and for data encrypted/decrypted using remote procedure call encryption/decryption functions for those DES modes requiring one. This initialization vector is accessed by the use of a handle that points the appropriate HMA data block.

r) CEPA Key: This is the key used in CEPA encryption/decryption for packet data and for data encrypted/decrypted using remote procedure call encryption/decryption functions. This key is accessed by the use of a handle that points the appropriate HMA data block.

s) CEPA Initialization Vector: This is the initialization vector used in CEPA encryption/decryption for packet data and for data encrypted/decrypted using remote procedure call encryption/decryption functions for those CEPA modes requiring one. This initialization vector is accessed by the use of a handle that points the appropriate HMA data block.

t) Hidden Memory Block (HMA): This is segment of memory internal to the TCC that can be allocated for storage of a variety of different types of information. When no longer needed this segment can be released back to the system. It is referenced by an assigned handle. When allocated HMA blocks have the attributes of "read," "write," and "key." These attributes must be set at the time when the HMA block is allocated.

u) HMA Block Syndrome: This is a value that is computed by folding the data in an HMA block onto itself a multiple number of times. See *Internal Specification Turbo Crypto Card* for an explanation of how this value is derived.

v) Random Number: This is a random value that is generated by the TCC. It may be used for a variety of purposes. If it is to be used as a key it must be placed in an HMA block that has the "key" attribute set.

w) Time: This SRDI represents the number of seconds since the last tamper event.

Definitions of SRDI Modes of Access

The table below defines the relationship between access to SRDIs and the different module services. The modes of access are shown as codes in the table and are defined as follows:

a) **A** - This service allocates memory (HMA) for storage of SRDIs.

b) **F** - This services releases (Free) the allocated SRDI memory (HMA).

c) **G** - This service generates the SRDI internal to the TCC.

d) **G_O** - This service generates the SRDI and then outputs the SRDI.

e) **I** - The SRDI is input into the TCC by this service.

f) **I_V** - This SRDI is input by this service and its value is verified.

g) **O** - This service outputs the SRDI.

h) **R** - The SRDI is read from HMA memory and used by the service.

i) **U** - The SRDI is updated by this service.

j) **Z** - The SRDI is erased by the service.

Service to SRDI Access Operation Relationship

User / Crypto Officer Service	Security Relevant Data Items																							
	Manufacturing Certificate	Local Network Certificate	Remote Network Certificate	DSA Public Parameters	Secure Manager DSA Public Key	Local DSA Public Key	Local DSA Private Key	Message Signature	Message SHA-1 Digest	Diffie-Hellman (DH) Modulus	Diffie-Hellman Base	Diffie-Hellman Local Secret	Diffie-Hellman Local Public	Diffie-Hellman Remote Public	Diffie-Hellman Shared Secret	DES Key	DES Initialization Vector	CEPA Key	CEPA Initialization Vector	Hidden Memory Block	HMA Syndrome	Random Number	Time
Get Status																								
Get Methods																								
Get Version																								
Force Tamper	Z	Z	Z				Z					Z			Z	Z		Z		Z			Z
Read Time																								O
Write Flash																								
Read Flash																								
HMA Allocate							A					A			A	A		A		A			
HMA Read																				O				
HMA Write																				I				
HMA Copy	U	U	U				U					U			U	U		U		U			
HMA Free							A					A			A	A		A		A			
Calculate HMA Block Syndrome																					Go			
Read Random																						Go		
Generate Random																				U		G		
Secure Hash Initialize									Z															
Secure Hash Update									U															
Secure Hash Finish									Go															
Secure Hash									Go															
Generate DSS Public/Private Number				I		Go	G															G		
Generate DSS Signature				I		I			G															
Verify DSS Signature				I	I																			
Generate Diffie-Hellman Key Pair										I	I	G	Go									G		
Generate DH Shared Number										I		R		I	G									
Generate DH Key															R	U		U						
DES Encrypt															R	I	R	I						
DES Decrypt															R	I	R	I						
Packet Encryption															R	I	R	I						
Packet Decryption															R	I	R	I						