KOOLSPAN®

KoolSpan TrustChip Developer Kit (TDK) Cryptographic Library Version 3.0 Security Policy

FIPS 140-2 Level 1 Validation

March 31, 2011

Version 1.12

Table of Contents

List of Tables

List of Figures

1 Introduction

This document is the Security Policy for the KoolSpan TrustChip Developer Kit (TDK) Cryptographic Library Version 3.0. This Security Policy specifies the security rules under which the module shall operate to meet the requirements of FIPS 140-2 Level 1. It describes how the module functions to meet the FIPS requirements, and the actions that operators must take to maintain the security of the module.

This Security Policy describes the features and design of the KoolSpan TrustChip Developer Kit Cryptographic Library 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 sensitive or designated information.

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

This Security Policy contains only non-proprietary information. This document may be freely reproduced and distributed as published. All other documentation submitted for FIPS 140-2 conformance testing and validation is “KoolSpan - Proprietary” and is releasable only under appropriate non-disclosure agreements.

The KoolSpan TDK Cryptographic Library (the cryptographic module) meets the overall requirements applicable to Level 1 security for FIPS 140-2 as shown in Table 1.

Table 1: Cryptographic Module Security Requirements.

Security Requirements Section	***Level***
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles and Services and Authentication	1
Finite State Machine Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A
Cryptographic Module Security Policy	1

1.1 Acronyms and Abbreviations

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CMVP	Cryptographic Module Validation Program
CSE	Communications Security Establishment
CSP	Critical Security Parameter
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
HMAC	Keyed-Hashing for Message Authentication
KAT	Known Answer Test
NIST	National Institute of Standards and Technology
OS	Operating System
PUB	Publication
RAM	Random Access Memory
RNG	Random Number Generator
SHA	Secure Hash Algorithm

2 KoolSpan TrustChip Developer Kit Cryptographic Library

2.1 Functional Overview

The KoolSpan TrustChip Developer Kit (TDK) Cryptographic Library is a set of object modules that provide cryptographic security functions (C APIs) for application developers to integrate cryptographic services into a library, application or system. The TDK Cryptographic Library object modules are only distributed after being linked into the KoolSpan TrustChip Developer Kit executable .dll, .so, or .dylib files..

The TDK Cryptographic Library provides security functions for encryption, decryption, hashing, getting the status of the integrity test, and running the self-tests. The TDK Cryptographic Library is used by the TDK to provide a solution for peer authentication and cryptographic services using a TrustChip. A TrustChip is a micro SD (Secure Digital) form factor hardware platform developed by KoolSpan which provides flash memory storage and a cryptographic processor.

2.2 Module Description

For the purposes of the FIPS 140-2 validation, the KoolSpan TDK Cryptographic Library object modules are linked into the KoolSpan TDK .dll, .so, or .dylib file by KoolSpan. This dynamically linked library is then linked in a run-time environment to some other library, application or system, running on a Microsoft, Linux, or MAC OS operating system. In FIPS 140-2 terminology, the KoolSpan TDK Cryptographic Library is a multi-chip standalone cryptographic module.

The TDK Cryptographic Library provides the following cryptographic services:

- Encryption of data
- Decryption of data
- Generation of hash values

The module was tested for validation with FIPS 140-2 using the following platforms, configured in single user mode (multiple concurrent operators are not allowed):

- Microsoft Windows XP, Intel Core 2 Duo
- MAC OS X 10.5, Intel Core 2 Duo
- Fedora 10, Intel Core 2 Duo
- Windows Mobile 6.1, ARM 32-bit
- Linux 2.6, ARM 7 (Android)

The cryptographic boundary includes the additional general-purpose computing platforms on which the operating systems are supported. Compliance is maintained for all of the above operating system platforms on which the binary executable executes unchanged.

The module has been confirmed by the vendor to be operational without modification or recompilation on the additional platforms listed below. These and other platforms on which the module can operate without modification or recompilation are compliant with the validation per FIPS 140-2 Implementation Guidance G.5

- Microsoft Windows XP, Intel Core 2 Quad
- Microsoft Windows Server 2003 R2, Intel Xeon
- Microsoft Windows Server 2003 R2, Intel Pentium 4
- Microsoft Windows Server 2003 R2, Intel Core 2 Duo
- Microsoft Windows Server 2003 R2, Intel Core 2 Quad
- Microsoft Windows Server 2008 Standard, Core 2 Quad
- Microsoft Windows Server 2008 Standard x64, Intel Xeon
- MAC OS X 10.6, Intel Core 2 Duo
- Fedora 10, Intel Xeon
- Windows Mobile 5, ARM 32-bit
- Windows Mobile 6, ARM 32-bit

The module does not have a bypass or maintenance mode.

2.2.1 File Names List

The following are the names of the KoolSpan TDK distribution for each of the major delivery platforms:

- Microsoft Windows: ***libtdk.dll***
- MAC OS X: ***libtdk.dylib***
- Linux/Android: ***libtdk.so***
- Windows Mobile: ***libtdk.dll***

The TDK Cryptographic Library consists of the following object modules that are linked into the TDK .dll, .so, and .dylib executables:

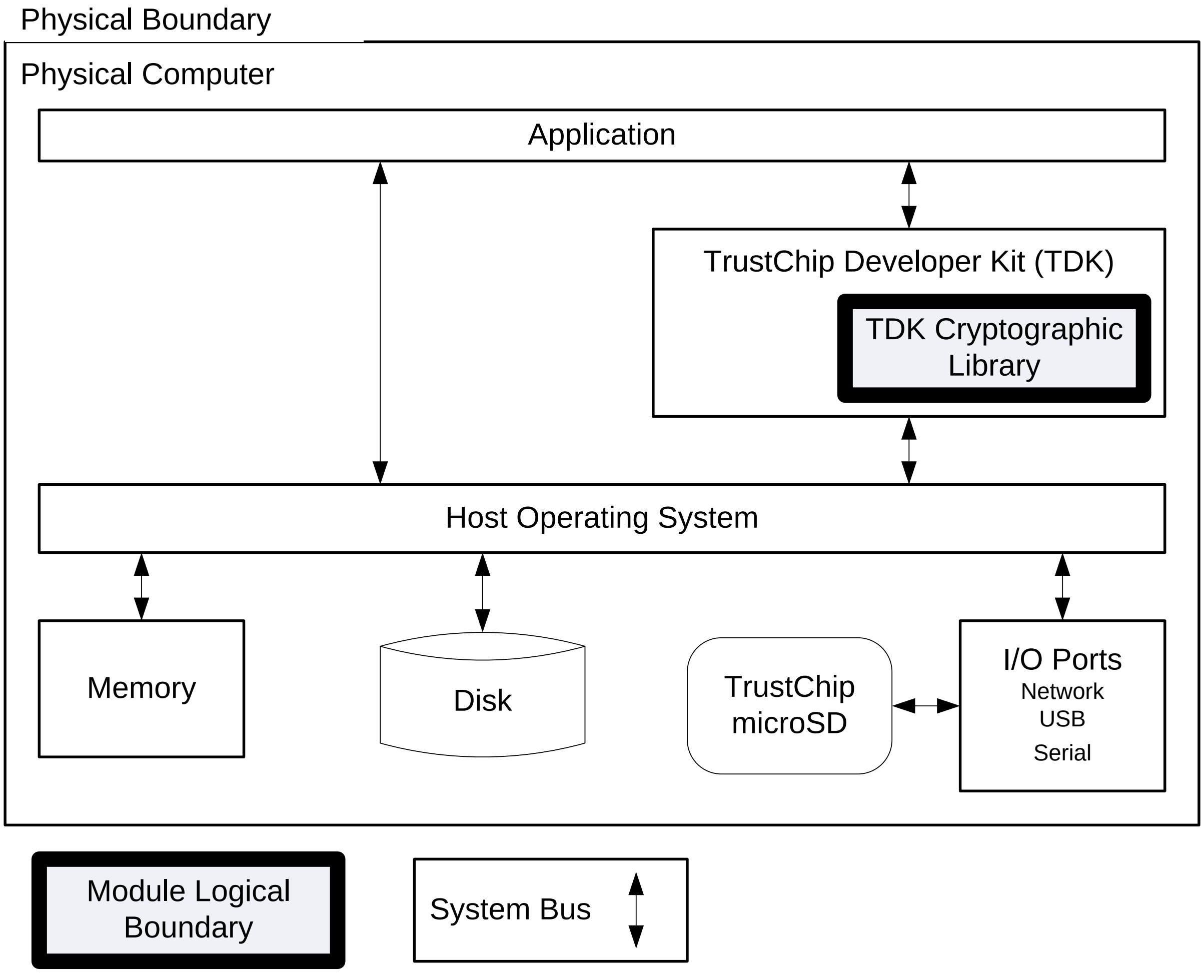
a. Windows XP, Server 2003, Server 2008, and Mobile
   - i. AAfirst.obj

b. MAC OS X 10.5, Fedora 10, Linux
   - i. aescrypt.o
   - ii. aeskey.o
   - iii. aes_modes.o
   - iv. aestab.o
   - v. crypto.o
   - vi. gcm.o
   - vii. gf128mul.o
   - viii. kat.o
   - ix. kscipher.o
   - x. sha2.o

2.2.2 Module Block Diagram

Figure 1 shows a module block diagram of the cryptographic module that illustrates the cryptographic physical and logical boundaries of the module.

Figure 1: TDK Cryptographic Library Module Block Diagram



The logical boundary of the cryptographic module is the binary (object code) which makes up the TDK Cryptographic Library, and is encapsulated by the KoolSpan TDK dynamically linked library. The physical boundary of the cryptographic module is the enclosure of the physical computer on which the toolkit resides and executes.

2.2.3 Configuring Single User Mode

The cryptographic module must be installed on one of the general-purpose operating systems supported by the TDK Cryptographic Library that have been configured to run in single-user mode.

2.2.3.1 Microsoft Windows

The Windows Mobile 5.0, 6.0, and 6.1 operating systems are single user operating systems and as such, no steps are required to configure single user mode for these operating systems.

For Microsoft Windows XP and Windows 2003 / 2008 Server, to configure single user mode, disable the guest user accounts and the following services:

- Server services
- Terminal services
- Remote registry services
- Remote desktop services
- Remote assistance

For detailed instructions on performing these tasks, please refer to the Microsoft support web site.

2.2.3.2 Linux (non-Android)

To configure single user mode for the Linux Fedora 10 operating system, perform the following tasks:

1. Remove all users except root and the daemon users from the /etc/passwd and /etc/shadow files. Ensure that the password field for the daemon users is either a star ‘*’ or ‘!!’ to prevent login by the daemon users.
2. Disable Network Information Service and other name services for users and groups so that only /etc/passwd and /etc/shadow files are used for login and authentication.
3. Disable the following daemons: remote login, remote command execution, file transfer.

For detailed instructions on performing these tasks, please refer to the guidance provided with the Linux Fedora 10 operating system.

2.2.3.3 Mac OS

To configure single user mode for the Mac OS operating system, perform the following tasks:

1. Shut down the computer if it is on.
2. Press the power button to start the computer.
3. Immediately press and hold the Command (Apple) key and the "s" key. (Command-S).

For detailed instructions on performing these tasks, please refer to the Apple support web site.

2.2.3.4 Linux (Android)

The Linux operating system underlying the Android platform is a single user operating system and as such, no steps are required to configure single user mode for this operating system.

2.3 Module Ports and Interfaces

The logical interfaces to the TDK Cryptographic Library are the C application programming interfaces (APIs) exported by the binary toolkit files.

The physical interfaces are the standard I/O ports found a general purpose computer for connecting external devices such as network adapters, monitors, and keyboards. These physical interfaces are outside the boundary of the cryptographic module and are not included in the validation.

Table 2: FIPS 140-2 Logical Interfaces.

Logical Interface	Description
Data input	Data input consists of ciphertext and plaintext data entering the cryptographic module as input to the API functions from the TDK. This data is input from end users or crypto officers via the character input device (e.g., keypad or keyboard) or Ethernet interface for the purpose of encryption, decryption, or hashing.
Data output	Data output consists of ciphertext and plaintext data exiting the cryptographic module and returned to the TDK from the API functions. This data is output to end users or crypto officers via the display output device (e.g., screen or monitor) or Ethernet interface.
Control input	Control input enters the module via parameters passed as input to the API functions from the TDK. Control input commands consist of module commands such as getting or setting operational parameters. The control input is provided by the end users or crypto officers using the character input device (e.g., keypad or keyboard) or Ethernet interface.
Status output	Status output consists of module status returned from status requests and other module outputs indicating module conditions. The status output is provided as return values (including error/exit codes) passed to the TDK from the API functions. This data is output to end users or crypto officers via the display output device (e.g., screen or monitor) or Ethernet interface.

3 Security Functions

The TDK Cryptographic Library implements the Approved security functions described in Table 3.

Table 3: Module Approved Security Functions.

Approved Security Function	Certificate
Symmetric Key Encryption (and Decryption)	
AES (FIPS PUB 197) GCM Mode CBC Mode	1108
Hashing	
SHA (FIPS PUB 180-2) SHA-256 SHA-384 SHA-512	1031
MAC	
HMAC SHA-256 (FIPS PUB 198)	641

The module does not contain or implement any non-Approved algorithms.

4 FIPS Approved Mode of Operation

The module’s Approved mode of operation is restricted to performing only FIPS-Approved cryptographic algorithms and security functions. The module automatically enters FIPS Approved mode on power up as soon as it successfully completes the power-on self test.

The module does not have a non-Approved mode.

5 Cryptographic Keys and CSPs

Key management (random number generation, key generation, key establishment, key storage, key zeroization) is the responsibility of the library, application, or system integrating the TDK Cryptographic Library.

Keys are entered into the TDK Cryptographic Library in plaintext form through the C APIs. All local copies of key data in the module reside in internally allocated volatile memory. These ephemeral keys are only in the TDK Cryptographic Library for the duration of the execution of the API. Once the API is completed, the memory containing the keys is zeroized and released. The underlying operating system is responsible for protecting the memory and process space of the TDK Cryptographic Library from unauthorized access.

Zeroization is carried out by overwriting the storage area or memory location with zeros.

6 Roles, Services, and Authentication

6.1 Roles and Services

The module supports a Crypto Officer role and a User role.

The Crypto Officer and User roles are implicitly assumed by the entity accessing services implemented by the module. The Crypto Officer role is implicitly entered when installing the module or performing system administration functions on the host operating system. The Crypto Officer and User may be different people or they may be the same person performing role-specific module operations.

Table 4: Roles and Services

Role	Services
Crypto Officer	Installation/deinstallation of the cryptographic module onto the computer system. All services provided by the TDK Cryptographic Library API.
User	All services provided by the TDK Cryptographic Library API.

The module does not support a maintenance mode.

6.2 Authentication

The module does not support operator identification or authentication mechanisms. The roles are implicitly assumed by the entity accessing services implemented by the module.

Only a single user in a specific role may access the module services at any given time.

6.3 Services

The module supports services that are available to operators in the Crypto Officer or User role. All of the services provided by the module are described in the *KoolSpan TrustChip™ Developer Kit (TDK) Cryptographic Library API Reference*. Table 5 shows the cryptographic services available to the various roles.

<u>Input/Output key for Table 5</u>

K	Keys and crypto parameters
P	Plaintext (unencrypted data)
C	Cryptext (encrypted data)
R	Return codes for errors or test status
H	Hash results

Table 5: Authorized Services

Service	***API***	***Crypto Officer***	***User***	***Inputs***	***Outputs***
Installation Deinstallation	N/A	●		N/A	N/A
Data encryption using AES-GCM	ksEncryptAes256Gcm	●	●	K,P	C,R
Data decryption using AES-GCM	ksDecryptAes256Gcm	●	●	K,C	P,R
Data encryption using AES-CBC	ksEncryptAes256Cbc	●	●	K,P	C,R
Data decryption using AES-CBC	ksDecryptAes256Cbc	●	●	K,C	P,R
SHA Hashing	ksSHA256Hash ksSHA384Hash ksSHA512Hash	●	●	P	H,R
HMAC Hash	ksSHA256HMAC	●	●	P	H,R
Run self test	ksCryptoGetIntegrityStatus (a parameter forces running the self-tests)	●	●	N/A	R
Show status	ksCryptoGetIntegrityStatus	●	●	N/A	R
Get version	ksCryptoGetVersion	●	●	N/A	R
Zeroization	ksZeroMemory	●	●	N/A	N/A

7 Access Control

Table 6 shows services that use or affect cryptographic keys or CSPs. For each service, the key or CSP is indicated along with the type of access.

Type of Access	**Description**
R	The item is **read** or referenced by the service.
Z	The item is **zeriozed** by the service.
E	The item is **executed** by the service. (The item is used as part of a cryptographic service.)

Table 6: Access Control

Key or CSP	***Service***	***Access Control***
AES Symmetric Key	Data encryption and decryption	E, Z
HMAC Key (Not a CSP)	Integrity Test	R, E
None	Installation, Deinstallation, Get version, SHA hashing, Show status, Run self test	E

8 Physical Security

The TDK Cryptographic Library is a software cryptographic module and does not enforce any physical security since it has no direct physical embodiment.

The module is software that is intended to run on standard computer that conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital devices, Class A. The module was tested on standard computers having a FCC DoC (Declaration of Conformity) meeting these requirements.

9 Self Tests

As shown in Table 7, the module performs power-on self tests (POST) to verify the integrity and correct operational functioning of the cryptographic module. If the system fails a self test, it reports status error codes in the return value of all Service APIs listed in Table 5 except ksCryptoGetVersion. The self test error codes are documented in the *KoolSpan TrustChip™ Developer Kit (TDK) Cryptographic Library API Reference* and indicate exactly which failure occurred. If the system fails a self test, the module transitions to an error state, blocking all data output via the data output interface and preventing use of any cryptographic keys, CSPs, cryptographic algorithms, and security functions.

While the module is performing any power on self test, the module design and implementation prevents it from entering a state where data output via the data output interface is possible.

Table 7: Self Tests

Self Test	***Description***
Mandatory power-up tests performed at power-up and on demand:	
Cryptographic Algorithm Known Answer Tests	Each cryptographic algorithm (AES GCM, AES CBC, HMAC SHA-256, SHA-256, SHA-384, and SHA-512) performed by the module, is tested using a “known answer” test to verify the operation of the function.
Software Integrity Test	The module uses HMAC SHA-256 to test its integrity. The file covered by the software integrity test is the TDK cryptographic library (a single file composed of the TDK cryptographic library object modules in their compiled and linked form.) The module does not perform the self-test on the other subcomponents of the TDK.
Critical Function tests performed at power-up:	
None.	No security-relevant critical function tests are performed.
Conditional tests performed, as needed, during operation:	
None.	No conditional tests are performed during operation.

The known answer tests (KAT) function by encrypting (or hashing) a string for which the calculated output is known and stored within the cryptographic module. An encryption or hashing test passes when the freshly calculated output matches the expected (stored) value. A test fails when the calculated output does not match the expected value. The test then decrypts the ciphertext string. A decryption test passes when the freshly calculated output matches the plaintext value. A test fails when the calculated output does not match the plaintext value.

10 Mitigation of Attacks

The cryptographic module is not designed to mitigate specific attacks.

11 Design Assurance

Configuration Management – Source code, associated documentation, and other product files are managed using a configuration management system. Each modification automatically uses a unique version identifier.

Delivery and Operation – Delivery and first time operation are controlled. The TDK Cryptographic Library object modules are only distributed after being linked into the KoolSpan TrustChip Developer Kit executable .dll, .so, or .dylib files. Organizations purchasing the product must either register to receive a unique authorization key, or utilize a KoolSpan TrustChip hardware token to use the TDK.

Development – The module design follows a High Level Design specification that functionally defines the module, ports and interfaces and the purpose of each. Details are provided in the *KoolSpan TrustChip™ Developer Kit (TDK) Cryptographic Library API Reference*.

12 Guidance

12.1 Installation Instructions

The Crypto Officer installs the KoolSpan TDK Cryptographic Library by installing the TrustChip Developer Kit (TDK) executable .dll, .so, or .dylib file. The Crypto Officer should follow the TDK installation instructions.

12.2 KoolSpan Integration Guidance

KoolSpan will distribute the module only as an object module linked into the TDK executable which ensures it is only accessed in a manner consistent with this document. At the time the module is first compiled into binary code, an HMAC SHA256 hash is calculated on the resulting binary using the ksComputeHMACSHA256 function implemented in the module. This known hash is stored and later used by the module to self-test.

Integrating code will automatically call the power-up self-tests prior to calling any of the cryptographic APIs. The APIs will produce an error if the power-up self-tests have not executed successfully. An external integrator of the KoolSpan TDK can initiate the module and call the power-up self-tests automatically by creating a TDK Context. The power-up self-tests may also be re-called on demand by executing ksCryptoGetIntegrityStatus() from the KoolSpan TDK.

12.3 Operational Guidance

The Crypto Officer or User must observe the application, watching for error messages from the TDK Cryptographic Library. When all self-tests pass without error, the module has initialized and is in the Approved mode of operation. If a self test fails then the module will return an error code and the module will release all memory including all keys stored in RAM/volatile memory. It is the responsibility of the operator of the general purpose computer that is running the user application, or the application itself, to decide how to respond to the generated error. See Table 2 for a description of the Status Outputs.

The corrective action for any self test failure is to restart the application. If any tests continue to fail, reinstall the TDK Cryptographic Library or contact the manufacturer.

Refer to the *KoolSpan TrustChip™ Developer Kit (TDK) Cryptographic Library API Reference* for a description of all of the services provided by the cryptographic module and instructions on using the TDK Cryptographic Library.

The TDK Cryptographic Library is designed for use on a host configured in single user mode as described in section 2.2.3.

13 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, *Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197, 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.

National Institute of Standards and Technology, *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*, available at URL: http://www.nist.gov/cmvp.