Sterling Commerce, Inc.

Sterling Crypto-C

Software Version: 1.5

FIPS 140-2
Non-Proprietary Security Policy

Level 1 Validation

Document Version 1.0

Prepared for:

Sterling Commerce, Inc.

4600 Lakehurst Court
Dublin, Ohio 43016-2000
Phone: (469) 524-2681
Fax: (972) 953-2690
http://www.sterlingcommerce.com

Prepared by:

Corsec Security, Inc.

10340 Democracy Lane, Suite 201
Fairfax, VA 22030-2518
Phone: (703) 267-6050
Fax: (703) 267-6810
http://www.corsec.com

Revision History

Version	Modification Date	Modified By	Description of Changes
1.0	2008-01-09	Xiaoyu Ruan	Release version

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Purpose

This document is a non-proprietary Cryptographic Module Security Policy for the Sterling Crypto-C from Sterling Commerce, Inc. This Security Policy describes how Sterling Crypto-C meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of Sterling Crypto-C.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 – *Security Requirements for Cryptographic Modules*) details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) website at: http://csrc.nist.gov/groups/STM/index.html.

In this document, Sterling Crypto-C is referred to as "the module". The application represents Sterling Commerce's software products linked with the cryptographic libraries provided by Sterling Crypto-C.

1.2 References

This document deals only with the operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The Sterling Commerce website (http://www.sterlingcommerce.com) contains information on the full line of products from Sterling Commerce.
- The CMVP website (http://csrc.nist.gov/groups/STM/index.html) contains contact information for answers to technical or sales-related questions for the module.

1.3 Document Organization

The Security Policy document is one document in a FIPS 140-2 submission package. In addition to this document, the Submission Package contains:

- Vendor Evidence
- Finite State Machine
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation have been produced by Corsec Security, Inc. under contract to Sterling Commerce. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Sterling Commerce and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Sterling Commerce.

2 Sterling Crypto-C

2.1 Overview

Sterling Commerce is the world's leading provider of multi-enterprise collaboration solutions for the Global 5000. Their software and services help companies operate more profitably by giving them visibility and control over the processes they share with business and supply chain partners.

Sterling Crypto-C is a cryptographic module implemented as two software dynamic link libraries (DLLs) on Windows or two Shared Objects (SO's) on Solaris, IBM AIX, and HP-UX. Sterling Crypto-C provides applications with an Application Programming Interface (API) of security-relevant functions and services such as Advanced Encryption Standard (AES), Triple Data Encryption Standard (TDES), Secure Hash Algorithm (SHA), Keyed-Hash Message Authentication Code (HMAC), Digital Signature Algorithm (DSA), Rivest, Shamir, and Adleman (RSA), etc. See Section 2.6 of this document for a complete list of supported algorithms.

Sterling Crypto-C is a user space shared library. It does not modify or become part of the Operating System (OS) kernel.

Sterling Crypto-C for the purpose of this FIPS validation has been tested and validated on the following platforms:

1. Windows Server 2003 on Intel Pentium,
2. Sun Solaris 10 on UltraSPARC II,
3. IBM AIX 5L 5.3 on PowerPC POWER5,
4. HP-UX 11i v2 on HP 9000/80 with PA-RISC[1],
5. HP-UX 11i v2 on HP Integrity with Intel Itanium 2 (formally known as IA-64).

Table 1 depicts platforms and corresponding file names of the module on these platforms.

Table 1 – Binary Forms of the Module

Operating System	Binary File Names
Windows Server 2003	libeay32.dll, ssleay32.dll
Sun Solaris 10	libcrypto.so, libssl.so
IBM AIX 5L 5.3	libcrypto.so, libssl.so
HP-UX 11i v2 (on PA-RISC)	libcrypto.sl, libssl.sl
HP-UX 11i v2 (on HP Integrity)	libcrypto.so, libssl.so

When operating in the FIPS mode of operation, Sterling Crypto-C is validated at FIPS 140-2 section levels shown in Table 2. Note that in Table 2, EMI and EMC mean Electromagnetic Interference and Electromagnetic Compatibility, respectively, and N/A indicates "Not Applicable".

[1] Reduced Instruction Set Computer

Table 2 – Security Level per FIPS 140-2 Section

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self-Tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

2.2 Module Interfaces

Sterling Crypto-C is a software module that meets overall level 1 of FIPS 140-2 requirements. The logical cryptographic boundary of the module consists of Sterling Crypto-C libraries, which run on several OS platforms. The module is composed of two binary files compiled on the OS. Table 1 summarizes the platforms and the binary files.

Figure 1 shows the logical cryptographic boundary of the module. In the FIPS mode of operation, the module provides a set of cryptographic services (API calls) in eight areas such as Transport Layer Security (TLS), Random Number Generator (RNG), etc.

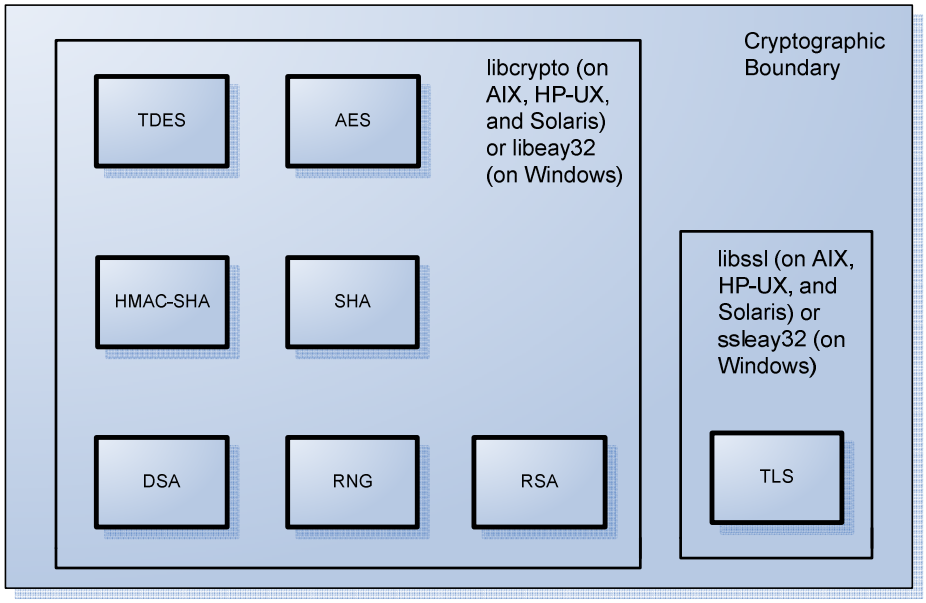

Figure 1 – Logical Cryptographic Boundary

The module's interactions with surrounding components, including the Central Processing Unit (CPU), hard-disk, memory, application, and the OS are demonstrated in Figure 2.

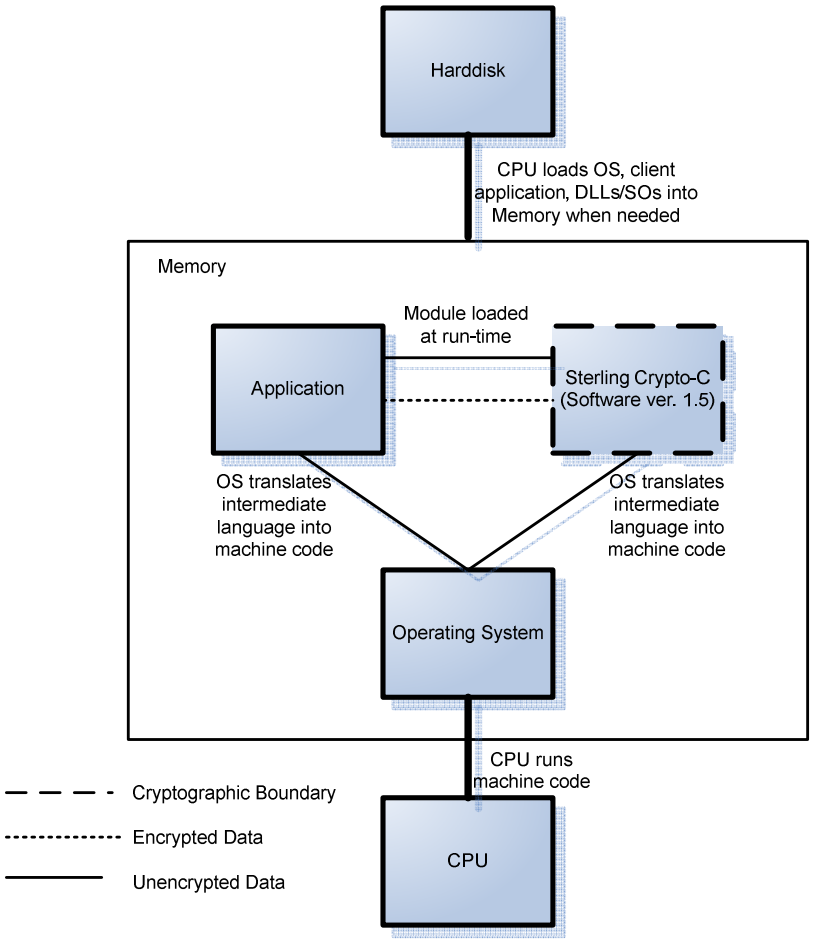

Figure 2 – Logical Cryptographic Boundary and Interactions with Surrounding Components

The module is validated for use on the platforms listed in the first column of Table 1. In addition to the binaries, the physical device consists of the integrated circuits of the motherboard, the CPU, Random Access Memory (RAM), Read-Only Memory (ROM), computer case, keyboard, mouse, video interfaces, expansion cards, and other hardware components included in the computer such as hard disk, floppy disk, Compact Disc ROM (CD-ROM) drive, power supply, and fans. The physical cryptographic boundary of the module is the hard opaque metal and plastic enclosure of the server running the module. The block diagram for a server is shown in Figure 3. Note that in this figure, I/O means Input/Output, BIOS stands for Basic Input/Output System, PCI stands for Peripheral Component Interconnect, ISA stands for Instruction Set Architecture, and IDE represents Integrated Drive Electronics.

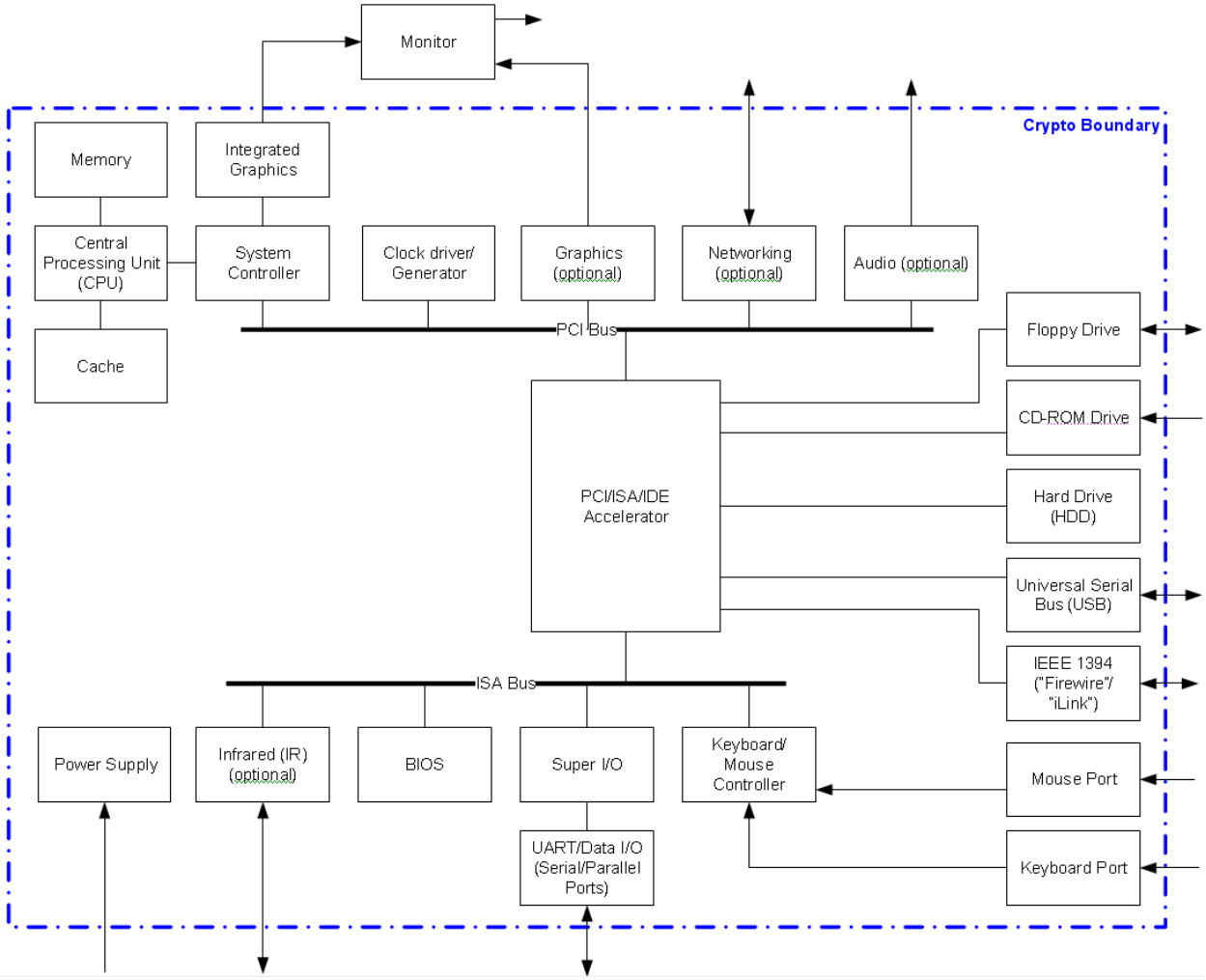

Figure 3 – Physical Block Diagram of a Server

All of these physical ports are separated into logical interfaces defined by FIPS 140-2, as described in Table 3.

Table 3 – Logical Interface, Physical Port, and Module Mapping

Logical Interface	Physical Port Mapping	Module Mapping
Data Input	Hard Disk, keyboard, mouse, CD-ROM, floppy disk, and serial/ Universal Serial Bus (USB)/parallel/network ports	Arguments for API calls that contain data to be used or processed by the module
Data Output	Hard Disk, floppy disk, monitor, and serial/USB/parallel/network ports	Arguments for API calls that contain module response data to be used or processed by the caller
Control Input	Hard Disk, keyboard, CD-ROM, floppy disk, mouse, and serial/USB/parallel/network port	API function calls
Status Output	Hard disk, floppy disk, monitor, and serial/USB/parallel/network ports	Arguments for API calls, function return value, error message

Logical Interface	Physical Port Mapping	Module Mapping
Power	Power supply	N/A

2.3 Roles and Services

The operators of the module can assume two roles as required by FIPS 140-2: a Crypto Officer role and a User role. The operator of the module assumes either of the roles based on the operations performed without any authentication. Table 4 gives a brief description of the roles and their privileges.

Table 4 – Module Roles and Services

Role Name	Role Services
Crypto Officer	1. Installing/uninstalling the module on the specific platform. 2. Initiating power-up self-tests.
User	Calling cryptographic functions provided by the module.

The following subsections detail both of the roles and their responsibilities.

2.3.1 Crypto Officer Role

The Crypto Officer role has the ability to install and uninstall the module and run power-up self-tests. Descriptions of the services available to the Crypto Officer role are provided to Table 5, where CSP refers to Critical Security Parameter. Services available to the User role listed in Table 6 are also available to the Crypto Officer role.

Table 5 – Crypto Officer Services

Service	Description	Input	Output	CSP and Type of Access
Install module	Installs and configures the module according to the operating system	Command	Success or failure	None
Uninstall module	Remove the module from the operating system	Command	Success or failure	None
Run power-up self-tests	Power-up self-tests include: (1) software integrity test; (2) Known Answer Tests (KATs) for TDES, AES, RSA, HMAC, RNG; (3) pair-wise consistency test for DSA keys	Call to *FIPS_mode_set(1)* or *FIPS_selftest()*	Pass or failure	None

2.3.2 User Role

The User role accesses the module's cryptographic services that include encryption, decryption, and authentication functionality. Descriptions of the services available to the User role are provided in Table 6.

Table 6 – User Services

Service	Description	Input	Output	CSP and Type of Access
TDES encryption	Encrypt plaintext using TDES	Command, plaintext, keys	Status, ciphertext	TDES symmetric keys - READ

Service	Description	Input	Output	CSP and Type of Access
TDES decryption	Decrypt TDES-encrypted ciphertext	Command, ciphertext, key	Status, plaintext	TDES symmetric key - READ
TDES key generation	Generate TDES symmetric keys	Command, key length	Status, symmetric keys	TDES symmetric keys - READ/WRITE
AES encryption	Encrypt plaintext using AES	Command, plaintext, key	Status, ciphertext	AES symmetric key - READ
AES decryption	Decrypt AES-encrypted ciphertext	Command, ciphertext, key	Status, plaintext	AES symmetric key - READ
AES key generation	Generate an AES symmetric key and set the key schedule	Command, key length	Status, symmetric key	AES symmetric key - READ/WRITE
RSA encryption	Encrypt symmetric key using RSA	Command, plaintext symmetric key, RSA public key	Status, encrypted symmetric key	RSA public key - READ Symmetric key - READ
RSA decryption	Decrypt RSA-encrypted symmetric key	Command, encrypted symmetric key, RSA private key	Status, decrypted symmetric key	RSA private key - READ Symmetric key - WRITE
RSA signature generation	Sign data using RSA	Command, data to be signed, RSA private key	Status, digital signature	RSA private key - READ
RSA signature verification	Verify an RSA signature	Command, data and signature, RSA public key	Status, acceptance/denial	RSA public key - READ
RSA key-pair generation	Generate a RSA key-pair	Command, key length	Status, RSA private key and public key	RSA private key and public key - WRITE
DSA signature generation	Sign data using DSA	Command, data to be signed	Status, digital signature	DSA private key - READ
DSA signature verification	Verify an DSA signature	Command, data and signature	Status, acceptance/denial	DSA public key - READ
DSA key-pair generation	Generate an DSA key-pair	Command, key length	Status, DSA private key and public key	DSA private key and public key - WRITE
SHA digest generation	Generate a SHA digest	Command, message to be hashed	Status, message digest	None
HMAC-SHA key generation	Generate a HMAC-SHA symmetric key	Command, key length	Status, HMAC-SHA symmetric key	HMAC-SHA key - WRITE
HMAC-SHA digest generation	Generate a HMAC-SHA digest	Command, message to be hashed, key	Status, HMAC-SHA message digest	HMAC-SHA key - READ
Random number generation	Generate a random number	Command, seed, length	Status, random number	Seed - READ/WRITE
TLS session establishment	Establish a new TLS session	Command	Status, session ID	Diffie-Hellman keys - READ

Service	Description	Input	Output	CSP and Type of Access
TLS master secret generation	Generate a TLS master secret	Command	Status, TLS master secret	TLS master secret - WRITE
Encrypted TDES key import	Import an encrypted TDES key in a TLS session	Command	Status	TDES symmetric key - WRITE
Encrypted AES key import	Import an encrypted AES key in a TLS session	Command	Status	AES symmetric key - WRITE
Encrypted TLS master secret import	Import an encrypted TLS master secret in a TLS session	Command	Status	TLS master secret - WRITE

2.4 Physical Security

Sterling Crypto-C is a multi-chip standalone module. The physical security requirements do not apply to this module, since it is purely a software module and does not implement any physical security mechanisms.

Although the module consists entirely of software, the FIPS 140-2 platform is a server that has been tested for and meets applicable Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B of FCC Part 15.

2.5 Operational Environment

The module was tested and validated on general-purpose Microsoft Windows Server 2003, Sun Solaris 10, IBM AIX 5L 5.3, and HP-UX 11i v2 operating systems.

The module must be configured in single user mode as per the instructions provided in Section 3.1.1 of this document. Recommended configuration changes for the supported operating systems can also be found in section 3.1.1.

2.6 Cryptographic Key Management

The module implements the following FIPS-approved algorithms in the FIPS mode of operation.

- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (certificate #655)
- HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 (certificate #312)
- RSA PKCS[2] #1 for signature generation/verification: 1024, 2048, and 4096 bits (certificate #280)
- DSA for key generation and signature generation/verification: 1024 bits (certificate #235)
- ANSI[3] X9.31 RNG with 2-key TDES (certificate #403)
- TDES: 112 and 168 bits, in Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), and Output Feedback (OFB) modes (certificate #578)
- AES: 128, 192, and 256 bits, in ECB, CBC, CFB, OFB, and counter modes (certificate #605)

[2] Public Key Cryptography Standard

[3] American National Standards Institute

The module implements the following non-Approved security functions in the FIPS mode of operation.

- Diffie-Hellman (key agreement, key establishment) methodology provides between 80 and 256 bits of encryption strength.
- RSA (key wrapping, key establishment) methodology provides between 80 and 150 bits of encryption strength.

The module implements the following non-Approved algorithms in the non-FIPS mode of operation: DES, RC2, RC4, Blowfish, CAST, MD2, MD4, MD5, RIPEMD, and HMAC MD5.

The module supports the following CSPs in the FIPS mode of operation:

Table 7 – List of Cryptographic Keys and CSPs

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
TDES keys	Symmetric key	1. Generated by ANSI X9.31 RNG. 2. Derived from TLS master secret. 3. Input in plaintext form.	Via TLS sessions in encrypted form.	Plaintext in volatile memory	Zeroized after use	Encrypt plaintext, decrypt ciphertext
AES key	Symmetric key	1. Generated by ANSI X9.31 RNG. 2. Derived from TLS master secret. 3. Input in plaintext form.	Via TLS sessions in encrypted form.	Plaintext in volatile memory	Zeroized after use	Encrypt plaintext, decrypt ciphertext
RSA private key	Private key	Generated by ANSI X9.31 RNG	In plaintext	Plaintext in volatile memory	Zeroized after use	Decrypt secret keys, sign messages
RSA public key	Public key	1. Generated by ANSI X9.31 RNG 2. Input in plaintext form.	In plaintext	Plaintext in volatile memory	Zeroized after use	Encrypt secret keys, verify signatures
DSA private key	Private key	Generated by ANSI X9.31 RNG	In plaintext	Plaintext in volatile memory	Zeroized after use	Sign messages
DSA public key	Public key	1. Generated by ANSI X9.31 RNG 2. Input in plaintext form.	In plaintext	Plaintext in volatile memory	Zeroized after use	Verify signatures
Diffie-Hellman public keys *p*,*g*	Public keys	1. Generated by ANSI X9.31 RNG 2. Input in plaintext form.	In plaintext	Plaintext in volatile memory	Zeroized after use	Establish symmetric keys
Diffie-Hellman private keys *a*,*b*	Private key	Generated by ANSI X9.31 RNG	Never output	Plaintext in volatile memory	Zeroized after use	Establish symmetric keys
ANSI X9.31 RNG Date/Time value	Date/Time value for ANSI X9.31 RNG	Generated internally	Never output	Plaintext in volatile memory	Zeroized when new Date/Time value is fed	Generate random numbers

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
TLS master secret	TLS master secret	1. Generated by ANSI X9.31 RNG 2. Input via TLS sessions in encrypted form	Via TLS session in ciphertext	Plaintext in volatile memory	Zeroized when TLS session is over	Derive keys in TLS sessions

2.6.1 Key Generation

The module uses an ANSI X9.31 RNG with 2-key TDES to generate cryptographic keys. This RNG is a FIPS 140-2 approved RNG as specified in Annex C to FIPS PUB 140-2.

2.6.2 Key Input/Output

Keys can be input to and output from the module in either plaintext or encrypted form.

2.6.3 Key Storage

All CSPs are stored in volatile memory in plaintext.

2.6.4 Key Zeroization

All CSPs are stored in volatile memory in plaintext. All CSPs are zeroized when they are no longer used. i.e., CSPs are zeroized when the sessions in which they are used are closed. The zeroization of the keys is carried out by overwriting the memory location with zeros. See Section 3.1.3 of this document for details.

2.7 Self-Tests

Sterling Crypto-C performs the following power-up self-tests when *FIPS_mode_set(1)* is called. A call to *FIPS_mode_set(1)* is necessary to set the module in the FIPS mode of operation. See Section 3.1.2 of this document for details. When needed, the power-up self-tests can also be initiated by a call to *FIPS_selftest()*.

- Software integrity test using HMAC-SHA-1.
- TDES KAT with 3 independent keys (56 bits each) in ECB mode.
- TDES KAT with 2 independent keys (56 bits each) in ECB mode.
- AES KAT with a 128-bit key in ECB mode.
- HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 KATs.
- RSA KAT with 1024-bit keys for signature generation/verification and encryption/decryption.
- ANSI X9.31 RNG KAT.
- Pair-wise consistency test for DSA.

In the FIPS mode, the following three conditional self-test are performed by the module.

- Pair-wise consistency test for RSA keys. This test is performed when a new RSA key-pair is generated.
- Pair-wise consistency test for DSA keys. This test is performed when a new DSA key-pair is generated.
- Continuous RNG Test. This test is performed when a new random number is generated. The new random number generated by the RNG is compared with the previous random number generated by the RNG. If the two numbers are equal, then the test fails.

If the self-tests fail, an exception will be thrown on the failure. The application is then alerted that the self-tests failed, and the module will not load and will enter an error state. When in the error state, execution of the module is halted and data output from the module is inhibited.

2.8 Design Assurance

Sterling Commerce uses the Concurrent Versions System (CVS) version 1.1.22 for configuration management of source code and documentation. See the CVS project website http://www.nongnu.org/cvs/ for more information.

Additionally, Microsoft Visual SourceSafe (VSS) version 6.0 is used to provide configuration management for the Sterling Crypto-C's FIPS documentation. This software provides access control, versioning, and logging.

2.9 Mitigation of Other Attacks

This section is not applicable. No claim is made that the module mitigates against any attacks beyond the FIPS 140-2 level 1 requirements for this validation.

3 Secure Operation

Sterling Crypto-C meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in the FIPS mode of operation.

3.1 Crypto Officer Guidance

The Crypto Officer is responsible for installing, uninstalling, configuring, and managing the module and running the power-up self-tests.

3.1.1 Single User Configuration

The user of the module is a software application. FIPS 140-2 mandates that a cryptographic module be limited to a single user at a time. To meet this requirement, a single instantiation of an application must only access a single instantiation of Sterling Crypto-C.

For enhanced security, it is recommended that the Crypto Officer configure the OS to disallow remote login. See the following instructions for details.

Windows Server 2003:

To configure Windows Server 2003 to disallow remote login, the Crypto Officer should ensure that all remote guest accounts are disabled in order to ensure that only one human operator can log into the Windows OS at a time. The services that need to be turned off for Windows are

- Fast-user switching (irrelevant if server is a domain member)
- Terminal services
- Remote registry service
- Secondary logon service
- Telnet service
- Remote desktop and remote assistance service

Once the Windows OS has been configured to disable remote login, the Crypto Officer can use the system "Administrator" account to install software, uninstall software, and administer the module.

Sun Solaris:

The specific procedure to configure Solaris to disable remote login is described below:

1. Login as the "root" user.

2. Edit the system files /etc/passwd and /etc/shadow and remove all the users except "root" and the pseudo-users (daemon users). Make sure the password fields in /etc/shadow for the pseudo-users are either a star (*) or double exclamation mark (!!). This prevents login as the pseudo-users. Also make sure the shell for daemon users is /dev/null, or something else that is not exploitable.

3. Edit the system file /etc/nsswitch.conf and make "files" the only option for "passwd", "group", and "shadow". This disables Network Information Service (NIS) and other name services for users and groups.

4. Edit the system file /etc/inet/inetd.conf, and comment out all unnecessary services (by prepending a hash ('#') sign to the beginning of each unnecessary service line).

   sadmind - Solstice network administration agent server
   rpc.ttdbserverd - Sun tool-talk server

kcms_server - Kodak Color Management System server
fs.auto - Sun font server
cachefsd - Network File System (NFS) cache service
rquotad - remote disk quota server
rpc.metad - Disksuite remote metaset service
rpc.metamhd - Disksuite remote multihost service
rpc.metamedd - Disksuite component service
ocfserv - Smartcard service
dtspcd - Part of the Common Desktop Environment (CDE) package
rpc.cmsd - remote calendar server
in.comsat - biff, mail notification server
in.talkd - talk server
gssd - Remote Procedure Call (RPC) application authentication
in.tnamed - deprecated name server
rpc.smserverd - removable media device sensor service (disabling requires manual CD mounting)
dcs - remote dynamic configuration server
ftpd - File Transfer Protocol server
ktkt_warnd - Kerberos warning server
chargen - deprecated network service
daytime - deprecated network time
time - legacy time service
discard - deprecated network service
echo - network 'echo' service
ufsd - part of RPC
in.uucpd - unix-to-unix copy server

5. Disable service startup scripts within /etc/rc2.d. Many additional services (not bound to inetd) are started by default. To disable startup scripts, files can be renamed to make sure they do not begin with a capital 'S' (which denotes Startup). Disable startup scripts that are not pertinent to the setup.

   nscd - NIS-related
   snmpdx - Simple Network Management Protocol (SNMP) services
   cachefs.daemon - NFS-caching
   rpc - Remote Procedure Call services
   sendmail – Sendmail
   lp - line printer daemon
   pppd - Point-to-point Protocol services
   uucp - Unix-to-Unix copy daemon
   ldap - Lightweight Directory Access Protocol (LDAP) services

6. Reboot the system for the changes to take effect.

Once the operating system has been configured to disable remote login, the Crypto Officer can use the system "root" account to install/uninstall software and administer the module.

IBM AIX:

The specific procedure to configure IBM AIX to disable remote login is described below:

1. Log in as the "root" user.

2. Edit the system file /etc/passwd and remove all the users except "root" and the pseudo-users. Make sure that for the pseudo-users, either the password fields are a star (*) or the login shell fields are empty. This prevents login as the pseudo-users.

3. Remove all lines that begin with a plus sign (+) or minus sign (-) from /etc/passwd and /etc/group. This disables NIS and other name services for users and groups.

4. Edit the system file /etc/inetd.conf. Remove or comment out the lines for remote login, remote command execution, and file transfer daemons such as telnetd, rlogind, krlogind, rshd, krshd, rexecd, ftpd, and tftpd.

5. Reboot the system for the changes to take effect.

Once the operating system has been configured to disable remote login, the Crypto Officer can use the system “root” account to install/uninstall software and administer the module.

HP-UX:

The specific procedure to configure HP-UX to disable remote login is described below:

1. Log in as the “root” user.

2. Edit the system file /etc/passwd and remove all the users except “root” and the pseudo-users. Make sure the password fields for the pseudo-users are a star (*). This prevents login as the pseudo-users.

3. Edit the system file /etc/nsswitch.conf. Make sure that "files" is the only option for “passwd” and “group”. This disables NIS and other name services for users and groups.

4. Edit the system file /etc/inetd.conf. Remove or comment out the lines for remote login, remote command execution, and file transfer daemons such as telnetd, rlogind, remshd, rexecd, ftpd, and tftpd.

5. Reboot the system for the changes to take effect.

Once the operating system has been configured to disable remote login, the Crypto Officer can use the system “root” account to install/uninstall software and administer the module.

3.1.2 Initialization

The Sterling Crypto-C software module itself is not an end-user product. It is provided to the end-users as part of the application. The module is installed during installation of the application. The installation procedure is described in the installation manual for the application.

A single initialization call to *FIPS_mode_set(1)* is required to initialize the module for operation in the FIPS mode. The module is not in the FIPS mode until *FIPS_mode_set(1)* is successfully called. On AIX, Solaris, and HP-UX, before calling *FIPS_mode_set(1)*, an environment variable, *SCI_CRYPTOC_LIBPATH*, must be set to the location of the libraries and the corresponding HMAC file. This variable should not need to be set on Windows. If it is, it will override the location determined using *GetModuleFilePath()*.

The function call to *FIPS_mode_set(1)* must be made by the application to place the module in the FIPS mode of operation. Each time the application is executed, it must call *FIPS_mode_set(1)* if the application has been configured for the FIPS mode of operation. The Crypto Officer must include this specific call when he programs the application that uses the Sterling Crypto-C cryptographic module.

Notice that DES is available in the FIPS mode of operation and must not be used in the FIPS mode. Use of DES will result in the module operating in a non-FIPS state.

3.1.3 Zeroization

All CSPs are stored in volatile memory in plaintext. When no longer needed, CSPs contained within the module are deleted by overwriting the storage locations of CSPs with zeros. The Crypto Officer may manually invoke the zeroization by rebooting the computer on which the module is running.

3.1.4 Management

The Crypto Officer does not perform any management of the module after installation and configuration. The management tasks are conducted by the application.

3.2 User Guidance

The module's cryptographic functionality and security services are provided via the application. End-users are not supposed to utilize the module without an associated application. Only the algorithms listed in Section 2.6 should be invoked by the application. End-user instructions and guidance are provided in the user manual and technical support documents of the application software. Although end-users do not have privileges to modify configurations of the module, they should make sure that the FIPS mode of operation is enforced in the application and thereby proper cryptographic protection is provided.

4 Acronyms

Table 8 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
BIOS	Basic Input/Output System
CBC	Cipher Block Chaining
CD	Compact Disc
CDE	Common Desktop Environment
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CPU	Central Processing Unit
CSP	Critical Security Parameter
CVS	Concurrent Versions System
DLL	Dynamic Link Library
ECB	Electronic Codebook
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
HMAC	(Keyed-) Hash MAC
IDE	Integrated Drive Electronics
ISA	Instruction Set Architecture
KAT	Known Answer Test
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
N/A	Not Applicable
NFS	Network File System
NIS	Network Information System
NIST	National Institute of Standards and Technology
OFB	Output Feedback
OS	Operating System
PCI	Peripheral Component Interconnect

Acronym	Definition
PKCS	Public Key Cryptography Standard
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RNG	Random Number Generator
ROM	Read Only Memory
RPC	Remote Procedure Call
RSA	Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SO	Shared Object
TDES	Triple Data Encryption Standard
TLS	Transport Layer Security
USB	Universal Serial Bus
VSS	Visual SourceSafe