THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Northrop Grumman M5 Network Security
SCS Linux Kernel Cryptographic Services

FIPS Security Policy
Version 2.42

www.northropgrumman.com/m5/

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

SCS Linux Kernel Cryptographic Services – Security Policy

Version 2.42

20 March 2014

Modification History

Version 1.0	15 May 2012	Initial Version
Version 1.1	14 Dec 2012	Revised version
Version 2.0	7 Feb 2013	Major revisions
Version 2.1	13 may 2013	Post VOR
Version 2.2	26 June 2013	Revised after input from BAE Detica
Version 2.3	6 August 2013	Revised after input from BAE Detica
Version 2.3.2	17 August 2013	Revised after input from BAE Detica
Version 2.3.3	22 October2013	Revised after input from BAE Detica
Version 2.3.4	11 November 2013	Added cross reference to software list
Version 2.4	27 February 2014	Changes required by VOR CP-12025-VOR-003 18 February 2014
Version 2.4.1	8 March 2014	Minor changes Version
Version 2.4.2	20 March 2014	remove AES GCM from approved security functions

Table of Contents

Contents

1. References

The following documents are relevant to the security policy:

a. SCS Linux Kernel Cryptographic Services - FIPS User Guide, version 2.4, dated 11 November 2013
b. The SCS Linux Kernel Cryptographic Services Software List V1.1, dated 11 November 2013

2. Introduction

This document is the Security Policy for the M5 Networks SCS Kernel Cryptographic Services module version ***kernel-PAE-2.6.32.14-127.scs.fips.fc12.i686***. This document was prepared as part of the Federal Information Processing standard (FIPS) 140-2 Level 1 validation process.

Northrop Grumman M5 Network Security is a fully owned subsidiary of Northrop Grumman Corporation.

FIPS 140-2, Security Requirements for Cryptographic Modules, describes the requirements for cryptographic modules. For more information about the FIPS 140-2 standard and the cryptographic module validation process see http://csrc.nist.gov/cryptval/.

The claimed validation levels for each section of the FIPS 140-2 standard are as follows

FIPS 140-2 Section Title	Validation Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
Electromagnetic Interference / Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 1 Claimed validation levels for each section of the FIPS 140-2 standard

.

The SCS family of products is as follows.

SCS-100

The **SCS-100's** small and lightweight form factor is designed to allow a single user access to highly secure, classified and unclassified networks simultaneously from anywhere in the world.

To ensure highly reliable communications, the SCS-100 establishes simultaneous connections via Wi-Fi, 3G/4G, BGAN satellite, Ethernet and ADSL. The user can direct the SCS-100 to select the best performing, the cheapest or load balance multiple communications bearers.

An intuitive touch screen interface, with pre-saved scenarios, allows the end user to easily configure the system on-site without the need to travel with technical specialists

SCS-200

The **SCS-200** is a break-through in highly secure mobile environments, allowing mobile users to connect to secure networks using any IP network for transport. Its small and lightweight form factor is designed to allow one to four users access to highly secure, classified and unclassified networks simultaneously from anywhere in the world. To ensure highly reliable communications, the SCS-200 supports multiple concurrent networks via Wi-Fi, 3G, Satellite, Ethernet and ADSL. Two intuitive touch screen interfaces allow the end user to easily configure the system on-site without the need for expensive technical specialists.

3. Kernel Module Specification

3.1 Version

The SCS Linux Kernel Cryptographic Services module evaluated version is ***kernel-PAE-2.6.32.14-127.scs.fips.fc12.i686***

3.2 Overview

The SCS Linux Kernel Cryptographic Services module is a software only, cryptographic module, running on a multi-chip standalone platform. It has been evaluated to FIPS level 1 certification.

3.2.1 Hardware Block Diagram

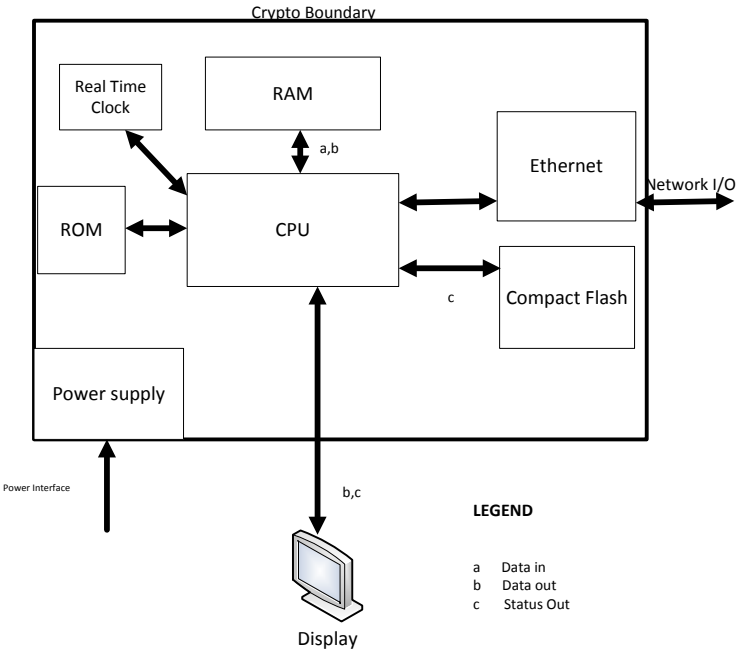

Figure 1: Hardware Security boundaries

3.2.2 Software Block Diagram

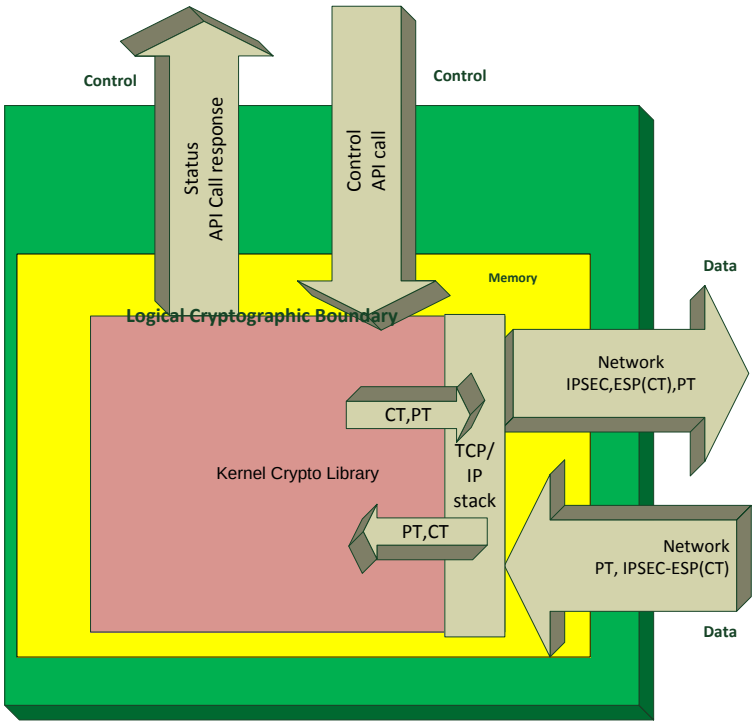

Figure 2: Security boundaries

4 Detailed Kernel Module specification

The SCS Linux Kernel Cryptographic Services (hereafter referred to as the Kernel Module) is a software library supporting FIPS-approved cryptographic algorithms. This module provides a C-language application program interface (API) for use by other processes that require cryptographic functionality. The data structures used by the API logically separate the control and data components.

A HMAC SHA-256 based integrity check 'fipscheck' also forms part of the cryptographic module. It is invoked automatically at boot time and uses the OpenSSL FIPS Object Module V2.0 (cert #1747) to perform the HMAC SHA-256 operation.

The SCS Linux Kernel Module (Version kernel-PAE-2.6.32.14-127.scs.fips.fc12.i686) is based on the Fedora 12 (Linux 2.6.32 kernel) on Intel X86 based hardware. The hardware provides at least 2 Ethernet interfaces and optionally WiFi, 3G/4G and ADSL network interfaces.

For FIPS 140-2 purposes the Kernel Module is classified as a multi-chip standalone module. The hardware Kernel module boundary is indicated in figure1. The Kernel module provides an API to allow other Kernel based consumers to utilise its services. The physical cryptographic boundary of the Kernel Module is the enclosure of the computer system on which it is executing i.e. the SCS-100 and SCS-200. If the contents of the virtual file /proc/sys/crypto/fips_enabled equals 1 the Kernel module is operating in a FIPS approved mode. The contents of this file cannot be changed, it reflects the value of the internal kernel parameter 'fips_enabled' which is set by the presence of the fips=1 flag as a kernel argument at boot time.

The Kernel module has been tested in the following systems

- The M5 Network security model SCS-100
- The M5 Network security model SCS-200

5. Roles and Services

5.1 Description

The Kernel Module meets all FIPS 140-2 level 1 requirements for Roles and Services, implementing both Crypto-User and Crypto-Officer roles. As allowed by FIPS 140-2, the Kernel Module does not support user authentication for those roles. Only one role may be active at a time and the Kernel Module does not allow concurrent operators.

The User and Crypto Officer roles are implicitly assumed by the entity accessing services implemented by the Kernel Module. The Crypto Officer can install and initialize the Kernel Module. The Crypto Officer role is implicitly entered when installing the Kernel Module or performing system administration functions on the host operating system.

- User Role
  - symmetric encryption and decryption
  - encode /decode
  - random number generation
  - Keyed Hash (HMAC)
  - Hash
  - zeroise
- Crypto-Officer Role:
  - Integrity check
  - Show status
  - Initiate self-tests

5.1.1 Approved allowed services

Service	Role	CSP / Alg / Mode	API Call	Access
Symmetric encryption/decryption	User	Key Triple DES ECB,CBC	All API functions with the prefix of crypto_cipher_, crypto_ablkcipher_ and crypto_blkcipher_ crypto_free_ablkcipher crypto_has_ablkcipher ablkcipher_request_set_tfm ablkcipher_request_free ablkcipher_request_set_callback ablkcipher_request_set_crypt crypto_free_blkcipher crypto_has_blkcipher	Read / write / execute
Encode/decode	User	Key AES 128/192/256 ECB, CBC, CTR, RFC3686, CCM	All API functions with the prefix of crypto_cipher_, crypto_ablkcipher_ and crypto_blkcipher_ crypto_free_ablkcipher crypto_has_ablkcipher ablkcipher_request_set_tfm ablkcipher_request_free ablkcipher_request_set_callback ablkcipher_request_set_crypt crypto_free_blkcipher crypto_has_blkcipher	Read / write / execute
Keyed Hash (HMAC)	User	HMAC key HMAC-SHA-1 HMAC SHA-224, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512	All API functions with the prefix of crypto_hmac_ crypto_free_hash	Read / write / execute
Hash	User	NA	All API functions with the prefix of crypto_has	Read / write /

		SHA-1 SHA-224, SHA-256, SHA-384, SHA-512	crypto_free_hash	execute
Random number generation	User	Seed, key ANSI X9.31, AES-128	All API functions with the prefix of crypto_rng_ crypto_alloc_rng crypto_free_rng NB if the RNG is initialised with an identical key and seed, the seed is XOR'd with a fixed 16 byte value.	Read / write / execute
Zeroise	User	Seed, key	All API functions with the prefix of crypto_free_	Read / write
Integrity Check	Crypto Officer	HMAC Key HMAC SHA-256	fipscheck userspace program. It uses the OpenSSL FIPS Kernel Module to check: • The Kernel Module binary and library • The Kernel binary (containing the self-check code for the 'Linux Kernel Cryptographic services Module'). This is run at boot time.	Read / execute
Self-test	Crypto Officer		See section 12.2	execute
Show Status	Crypto Officer		Global status variable fips_enabled See section 12.2	read

Table 2: Approved services Kernel Module

5.1.2 Non Approved services

Service	Role	CSP / Alg / Mode	API Call	Access
Symmetric encryption/decryption	User	Key DES CTR, ECB,CBC	All API functions with the prefix of crypto_cipher_, crypto_ablkcipher_ and crypto_blkcipher_ crypto_free_ablkcipher crypto_has_ablkcipher ablkcipher_request_set_tfm ablkcipher_request_free ablkcipher_request_set_callback ablkcipher_request_set_crypt crypto_free_blkcipher crypto_has_blkcipher	Read / write / execute
Symmetric encryption/decryption	User	Key Triple DES	All API functions with the prefix of crypto_cipher_, crypto_ablkcipher_ and crypto_blkcipher_	Read / write / execute

		CTR (non-compliant)	crypto_free_ablkcipher crypto_has_ablkcipher ablkcipher_request_set_tfm ablkcipher_request_free ablkcipher_request_set_callback ablkcipher_request_set_crypt crypto_free_blkcipher crypto_has_blkcipher	
Encode/decode	User	Key AES 128/192/256 GCM (Non-compliant)	All API functions with the prefix of crypto_cipher_, crypto_ablkcipher_ and crypto_blkcipher_ crypto_free_ablkcipher crypto_has_ablkcipher ablkcipher_request_set_tfm ablkcipher_request_free ablkcipher_request_set_callback ablkcipher_request_set_crypt crypto_free_blkcipher crypto_has_blkcipher	Read / write / execute

Note that while these functions are provided by the API they should not be used.

6 Ports and Interfaces

6.1 Description

The physical ports of the Kernel Module are the same as the computer system on which it is executing. The logical interface is a C-language application program interface (API).

The Data Input interface consists of the input parameters of the API functions. The Data Output interface consists of the output parameters of the API functions. The Control Input interface consists of the actual API functions and the Kernel command line. The Status Output interface includes the return values of the API functions.

FIPS Interface	Module Interface
Data Input	API input parameters
Data Output	API output parameters
Control Input	API function calls Kernel command line
Status Output	API return codes Kernel log

Table 3 Kernel Module Ports and interfaces

7 Self-tests

The Module performs both power-up self-tests at module initialization and continuous condition tests during operation. If a self test fails, a kernel panic will be triggered. This results in a simple message on the screen and no further operation without a restart.

7.1 Algorithm Self-tests

Whenever a crypto algorithm is loaded into memory the algorithm is tested by the core code of the cryptographic service module. If any of the tests fail a kernel panic is triggered. This can only be recovered by the user by resetting the system. The Kernel Module self-tests are listed in Table 4.

Algorithm	Test
AES (128,192,256) ECB, CBC, CTR, RFC3686, CCM	KAT (encrypt and decrypt)
Triple-DES ECB,CBC	KAT(encrypt and decrypt)
ANSI X9.31 Cryptographic PRNG (AES128)	KAT
HMAC-SHA-1	KAT
HMAC-SHA-224	KAT
HMAC-SHA-256	KAT
HMAC-SHA-384	KAT
HMAC-SHA-512	KAT
SHA-1	KAT
SHA-224	KAT
SHA-256	KAT
SHA-384	KAT
SHA-512	KAT

Table 4: Algorithm self-tests (NB: KAT= Known Answer Test)

7.2 Software Integrity Self-tests

During boot a software integrity test is performed on the main kernel binary. Once the main kernel binary is checked, each algorithm module has its integrity verified before being loaded into the running kernel, thus triggering its known answer tests.

The integrity tests are a HMAC SHA-256 initiated with the tool fipscheck. If the fipscheck program returns an error then the boot process will trigger a kernel panic. The only method to proceed is to reboot the device.

The HMAC SHA-256 is performed using the FIPS certified OpenSSL FIPS Object Module V2 (cert #1747) built in accordance with the instructions in the security policy.

When the OpenSSL FIPS Object Module is initialised, it will run its own self checks. These include a HMAC SHA-1 integrity check of the software and a known answer test of the HMAC SHA-256 algorithm.

If the OpenSSL checks pass then the fipscheck program start its own integrity checks with a HMAC SHA-256. First it checks its library file, then it checks its own binary, finally it checks the kernel binary or module it was invoked to check. If any of these checks fail the program will return an error. That error will in turn trigger a kernel panic and the only way to proceed is to reboot the device.

Subject	Algorithm
fipscheck library	HMAC SHA 256 (from OpenSSL Library)
fipscheck program	HMAC SHA 256 (from OpenSSL Library)
Linux Kernel binary	HMAC SHA 256 (from OpenSSL Library)

Table 5: Software verification self-tests

7.3 Software Continuous Self-tests

Each 128 bit block of generated random data is compared against the previous block to ensure that the generator is not repeating. On error the Kernel will panic and the only way to proceed is to restart the Kernel module.

7.4 Power-on Self-tests

Whenever the computer boots the power-on self-test will be run without any further intervention from the user. The start-up process will first determine the kernel binary file. Next it will run the software integrity self-tests described at 7.2 on the kernel binary. If those tests do not pass a 'kernel panic' will be triggered. If the software integrity self-tests pass, then the start-up process will then load each algorithm module in turn. As each module is loaded, its associated algorithm self test will be performed. If an algorithm self test fails a 'kernel panic' will be triggered.

8 Mitigation of Other Attacks

The Kernel Module does not contain additional security mechanisms beyond the requirements for FIPS 140-2 level 1 cryptographic modules.

9 Physical Security

The Kernel Module is comprised of software only and thus does not claim any physical security.

10 Operational Environment

No debugging tools should be used in the operating environment while in FIPS mode.

The Kernel Module requires that the operating system is restricted to a single operator mode and that the kernel component making the calls to the cryptographic module is the only user of that module.

11. Cryptographic Key Management

11.1 Key Generation

The module performs no key generation, however it does provide a Cryptographic PRNG that implements ANSI X9.31 Appendix A.2.4 using AES-128 that users can utilise for key generation.

11.2 Key Storage

Keys are not stored by the Kernel Module. The operating system and memory management features of the X86 CPUs protect keys in memory from unauthorised access. Setting of keys is a distinct API call, separate to encrypt/decrypt operations ensuring that keys and data cannot be mixed.

11.3 Key Zeroisation

When cryptographic objects are freed with the API calls, the memory locations are first overwritten with zeros before returning to the calling function. Note that the integrity check key used to verify the kernel binary is external to the Kernel Module and stored within the fipscheck binary. It cannot be zeroised in this fashion.

11.4 Key Usage

Key usage is done by the calling program. There is no manual entry capability as key entry is done solely by the API.

12. Guidance

12.1 User Guidance

The Kernel Module implements the Triple DES CTR (Non-compliant) and AES GCM (Non-compliant) algorithms but these should not be used.

DES algorithm to support the 3DES cryptographic operations only. It is not to be used to support single DES encryption.

All memory allocation and de-allocation of the crypto data structures shall be performed using the methods provided by the Kernel Module API. The content of these data structures should not be made available to user space using copy_to_user() or any other method.

To use the two-key Triple-DES algorithm to encrypt data or wrap keys in an Approved mode of operation, the module operator shall ensure that the same two-key Triple-DES key is not used for encrypting data (or wrapping keys) with more than 2^{20} plaintext data (or plaintext keys).

12.2 Crypto Officer Guidance

The system operates in FIPS mode if the kernel is started with the command line flag 'fips=1'. The presence of this flag at kernel start sets an internal variable 'fips_enabled' to the value 1. This variable cannot be changed. The current setting can be observed by reading a virtual file provide for this purpose with the command:

```
cat /proc/sys/crypto/fips_enabled
```

The crypto officer can change the value of this flag to be used in future boots but cannot change the current status. The full procedure for changing the flag and preparing the system to operate in FIPS mode is documented in the User Guide (Reference A). While the key step is adding the 'fips=1' flag to the command line, failure to follow all the steps will likely result in a device that fails the self-tests and does not boot.

The crypto officer can also trigger a re-run of the Kernel Module algorithm self-tests listed in the table at 3 by running the command:

```
modprobe tcrypt
modprobe –r tcrypt
```

If the tests fail then a 'kernel panic' will be triggered, and all operation will cease. A listing of expected output after a successful test can be found in Reference A.

13 Cryptographic Algorithms

The Kernel Module supports the following FIPS approved or allowed algorithms:

Algorithm Validation	Certificate	Usage
AES	#2604	encrypt/decrypt
3DES	#1569	encrypt/decrypt
RNG (ANSI X9.31 Appendix A.2.4 using AES)	#1232	random number generation

SHA-1	#2188	hashing
SHA-224	#2188	hashing
SHA-256	#2188	hashing
SHA-384	#2188	hashing
SHA-512	#2188	hashing
HMAC-SHA-1	#1612	message integrity
HMAC-SHA224	#1612	message integrity
HMAC-SHA256	#1612	message integrity
HMAC-SHA384	#1612	message integrity
HMAC-SHA512	#1612	message integrity
Open SSL HMAC-SHA256	#1126	message integrity

Table 6: Kernel Module approved Cryptographic algorithms

The Kernel Module supports the following non-FIPS approved algorithms:

Algorithm	Usage	Keys/CSPs
DES	Symmetric encryption	DES key 56 bits
3DES CTR (Non Compliant)	Symmetric encryption	2 or 3 DES keys 56 Bits
AES GCM (Non Compliant)	Symmetric encryption	128/192/256 bits

Table 7: Kernel Module non-approved Cryptographic algorithms

The Kernel Module supports the following non cryptographic functions:

Function	Usage
Deflate	Compression routine
Null Compression	Dummy Compression routine

Table 8: Non cryptographic functions supported

14 Electromagnetic Interference / Electromagnetic Compatibility (EMI/EMC) compliance

The SCS-100 and SCS-200 devices have been tested and certified as compliant with the requirements of CISPR22: 2008 Ed 6 (Accepted by the FCC as equivalent to 47 CFR, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A. EMC Test report Number M121137 dated 3rd December 2012 for the SCS-100 and EMC report Number M130318-1 dated the 5th of June 2013 for the SCS-200 refer.