GlobalSCAPE® Cryptographic Module, Version 1.0.2
FIPS 140-2 Non-Proprietary Security Policy, Version 1.0.7

Last Revised 1/24/2008

Copyright Notice

Revision History

Document Version	Date	Description
1.0.0	6/5/2007	Initial document.
1.0.1	8/30/2007	Modifications to address comments from validation lab. Added “Design Assurance” section.
1.0.2	9/11/2007	Added guidance for minimum DSA key length and updated Module version number.
1.0.3	11/7/2007	Updated Module version number.
1.0.4	12/14/2007	Modifications to address NIST and CSE comments.
1.0.5	1/16/2008	Modifications to address additional NIST and CSE comments.
1.0.6	1/21/2008	Added clarification concerning DES algorithm.
1.0.7	1/24/2008	Updated company name.

Table of Contents

1 Introduction

1.1 Purpose

This document is the non-proprietary FIPS 140-2 Security Policy for the GlobalSCAPE Cryptographic Module, Version 1.0.2. This Security Policy describes how the module meets the requirements set forth in the Federal Information Processing Standards Publication 140-2 (Ref: 1) for a Level 1 multi-chip standalone module. Additionally, this document specifies how the module may be run in a FIPS 140-2 approved mode.

The FIPS 140-2 standard specifies the security requirements for cryptographic modules. For additional information concerning the FIPS 140-2 standard and the Cryptographic Module Validation Program (CMVP), please refer to the National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program website at http://csrc.nist.gov/groups/STM/cmvp/index.html

For additional information on GlobalSCAPE products, please refer to http://www.globalscape.com

2 Cryptographic Module Specification

2.1 Module Overview

The GlobalSCAPE Cryptographic Module is a software-only cryptographic library, hereafter known as the “Module”. In terms of the FIPS 140-2 standard the Module is classified as a *Multi-chip Standalone Module*. The Module is implemented in the C programming language and is provided as a dynamically linked export library (DLL) executing on a general-purpose computer system. The Module is intended for use by applications through the Module’s Application Programming Interface (API), which is based on the OpenSSL API defined by the OpenSSL Project (Ref: 3).

For the purposes of FIPS 140-2 validation, the Module was tested on an x86 based computer system running the Microsoft Windows 2003 Server operating system in single-user mode. When operated under the requirements of section G.5 of the FIPS 140-2 Implementation Guidance documentation (Ref: 2), the Module maintains compliance when on other versions of the Microsoft Windows operating system.

2.2 Cryptographic Module Boundaries

2.2.1 Physical Cryptographic Boundary

The physical cryptographic boundary of the Module includes the standard enclosure of the general-purpose computer system on which it is loaded.

The following diagram depicts the physical cryptographic boundary of the Module.

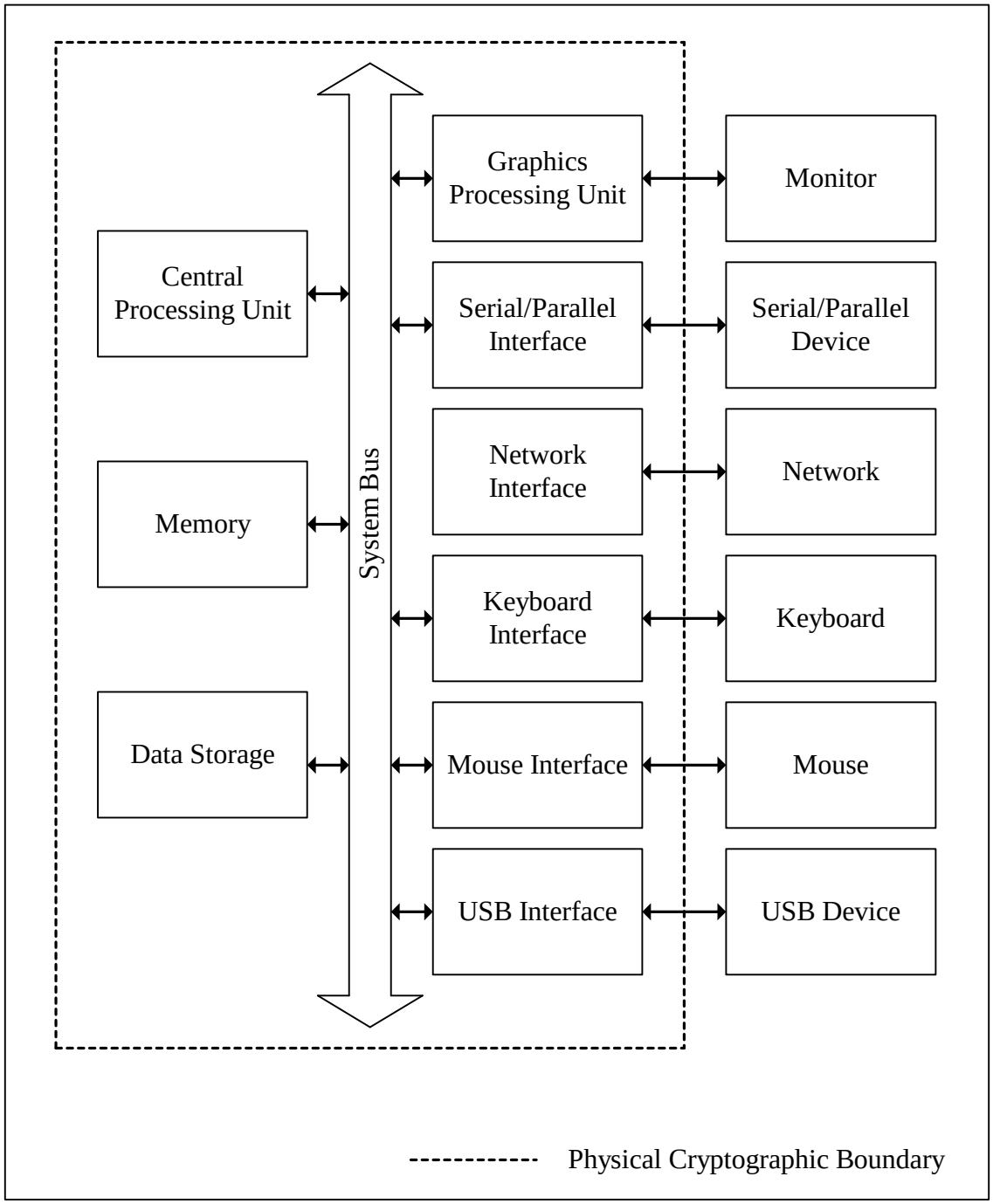

Figure 2.2.1 – Module Physical Cryptographic Boundary

2.2.2 Logical Cryptographic Boundary

The logical cryptographic boundary of the Module contains the software of the Module, which consists of the Module's dynamic link library.

The following diagram depicts the logical cryptographic boundary of the Module.

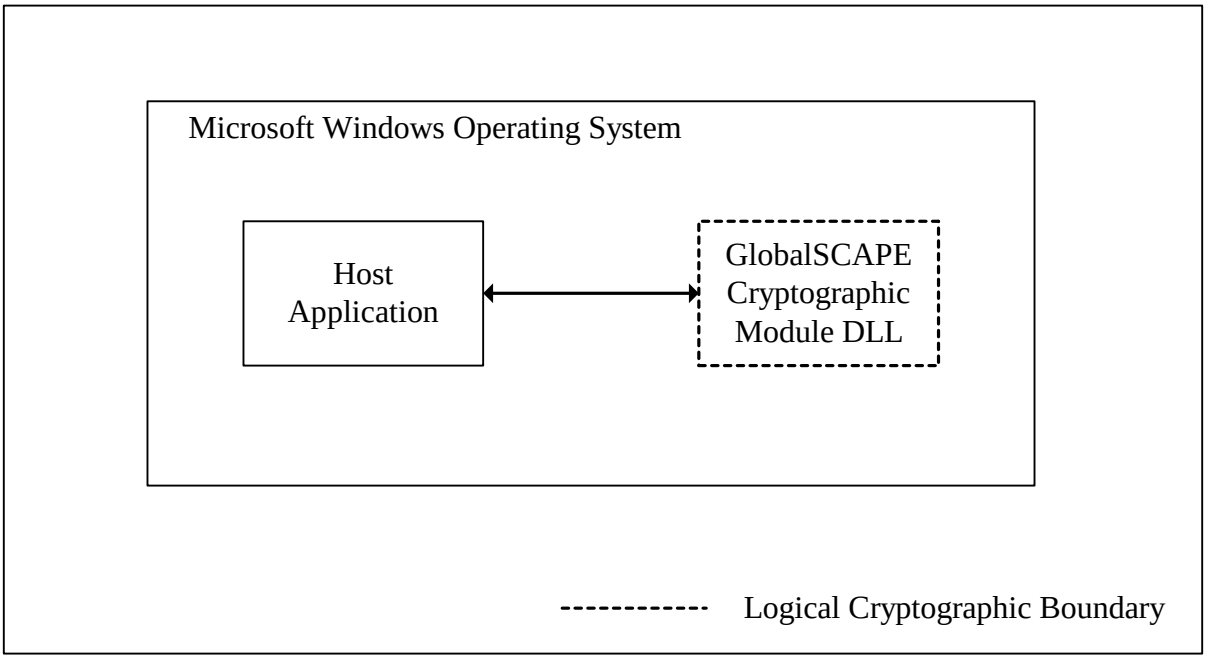

Figure 2.2.2 – Module Logical Cryptographic Boundary

2.3 FIPS 140-2 Security Level Specification

The Module meets the FIPS 140-2 Level 1 security requirements as summarized in the table below.

Security Requirements Section	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 (EMI/EMC)	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 2.3 – Module Security Level per Security Requirements Section

2.4 Approved Mode of Operation

The FIPS approved mode of the Module is recorded using the internal flag *fips_mode*. By default this value is set to FALSE and thus by default the Module is not operating in FIPS approved mode.

To operate in FIP approved mode, the referencing application must initialize the Module by performing a single call to the *GSCM_FIPS_mode_set* API function. When the *GSCM_FIPS_mode_set* function is invoked, the Module will perform an integrity check and a series of power-up self-tests.

The integrity check performs a check of the entire Module by comparing a 160-bit HMAC-SHA-1 hash value of the Module's DLL file (GSCrypto.dll) computed at build time with the on-disk image of the Module. The build-time hash is stored in a file in the same location as the Module's DLL and has the filename "GSCrypto.dll.sha1". The HMAC-SHA-1 instance uses a 136-bit key when computing the hash values.

If the stored build-time hash matches the hash of the on-disk DLL, then the Module will proceed with a set of Known Answer Tests (KATs) and Pair-wise consistency tests.

If any of the tests fail, the Module sets an internal flag, *fips_selftest_fail*. This flag is then used to prevent subsequent access to the cryptographic function calls. If all of the tests pass, the Module will set the *fips_mode* flag and proceed to operate in FIPS approved mode.

For more detailed information on the supported set of cryptographic algorithms, please refer to section *13 Cryptographic Algorithms*.

For more detailed information on the set of self-tests, please refer to section *10 Self-Tests*.

3 Cryptographic Module Ports and Interfaces

As the Module is implemented as a software-only cryptographic library, its logical interfaces are defined by its API. Applications may access the services provided by the Module using this API.

The Module's API maps onto the FIPS 140-2 logical interfaces as follows:

- Data Input Interface – the input parameters to API functions accepting data input
- Data Output Interface – the parameters generated or updated by API functions either through arguments or return values
- Control Input Interface – the API functions that initiate and control the operation of the Module
- Status Output Interface – the API functions that provide status information for the Module as return values

4 Roles, Services and Authentication

4.1 Roles

In terms of the FIPS 140-2 standard, the Module implements two roles: a User role and a Crypto Officer role. No other roles, including the Maintenance role, are implemented.

An entity assumes the User or Crypto Officer role when accessing the services provided by the Module. The particular role is based on the specific services accessed. When accessing the User services, the entity assumes the User role. When accessing any other service, the entity assumes the Crypto Officer role.

Specifically, the supported roles are defined as follows:

- User Role – any entity that can access services provided in the Module.
- Crypto Officer Role – any entity that can access services provided by the Module, install the Module, or initialize the Module.

4.2 Services

All services provided by the Module may be accessed by both the User and Crypto Officer roles with the exception of the Module Initialization service, which may only be performed by the Crypto Officer role.

The following services are provided by the Module:

- Symmetric Encryption and Decryption – a service providing symmetric encryption and decryption.
- Digital Signature – a service using asymmetric cryptography to generate digital signatures.
- Key Establishment – a service providing key agreement functionality.
- Key Zeroization – a service providing key zeroization functionality.
- Message Digest – a service providing cryptographic hash functionality.
- Random Number Generation – a service providing random number generation.
- Show Status – a service that provides status output for the Module and a means to enable the FIPS approved mode of operation of the Module.
- Self Test – a service that performs the self-test functions of the Module to verify the integrity and proper operation of the Module.
- Module Initialization – a service that allows the installation and initialization of the Module.

The FIPS approved mode of operation for the Module may be entered through both the Show Status and Module Initialization services using the *GSCM_fips_mode_set* API function.

The following table summarizes the authorized services for each role.

Role	**Authorized Services**
User	Symmetric Encryption and Decryption Digital Signature Key Establishment Key Zeroization

	Message Digest Random Number Generation Show Status Self Test
Crypto Officer	Symmetric Encryption and Decryption Digital Signature Key Establishment Key Zeroization Message Digest Random Number Generation Show Status Self Test Module Initialization

Table 4.2– Services Authorized for Roles

4.3 Access Control Policy

The Module's Access Control Policy describes the rules governing access to the Critical Security Parameters (CSPs) used by the various services of the Module. As the Module is a cryptographic library, the CSPs are provided to or returned from the services API functions. A service's access to CSPs is limited by the API for that service.

The following table summarizes both the types of cryptographic keys and CSPs used by the Module services and the types of access to those keys and CSPs.

Service	**Cryptographic Keys and CSPs**	**Types of Access**
Symmetric Encryption and Decryption	Symmetric Key	Read Write Execute
Digital Signature	Asymmetric Private Key	Read Write Execute
Key Establishment	Asymmetric Public and Private Keys	Read Write Execute
Key Zeroization	All Keys	Write Execute
Message Digest	None (SHA1, SHA2) HMAC Key (HMAC)	Read Write Execute
Random Number Generation	Seed Key	Read Write Execute
Show Status	None	Not Applicable
Self Test	None	Not Applicable

Service	Cryptographic Keys and CSPs	Types of Access
Module Initialization	None	Not Applicable

Table 4.3 – Access Rights within Services

4.4 Operator Authentication

As allowed by the FIPS 140-2 level 1 requirements, the Module does not implement identification or authentication of users accessing the Module. The Module relies on the underlying operating system's user authentication facilities to perform this functionality.

The following table summarizes the supported roles and their associated authentication details:

Role	Type of Authentication	Authentication Data	Authentication Mechanism	Strength of Authentication
User	Not Required	Not Required	Not Required	Not Required
Crypto Officer	Not Required	Not Required	Not Required	Not Required

Table 4.5 – Roles and Associated Authentication Details

5 Finite State Model

The Finite State Model is provided in a separate document as described in Ref: 4.

6 Physical Security

As the Module is a software-only cryptographic library executed on a general-purpose computer system, the physical security is provided by the host computer system. As such, physical security for cryptographic modules is not applicable to the Module for FIPS 140-2 Level 1 compliance.

7 Operational Environment

7.1 Operating System Requirements

The operational environment of the Module is a Microsoft Windows based operating system running on a general purpose computer system in single-user mode.

7.2 Operational Rules

To run in FIPS approved mode, the following operational rules must be followed:

1. The Module must be initialized by a successful call to the GSCM_FIPS_mode_set function.

2. The Module must only be used by a single operator at a time.
3. The operating system must be configured to only allow login of a single user.
4. The operating system must be configured to disallow remote logins.
5. The operating system must authenticate the user.
6. The operating system authentication mechanisms must be configured to prevent unauthorized modification or replacement of the Module.
7. The operating system must properly isolate and prevent unauthorized access to the Module's memory.
8. The operating system must properly isolate access to an instance of the Module to a single application.

For additional information concerning the correct installation and configuration of the Module, please refer to the Crypto Officer and User Guide (Ref: 12).

8 Cryptographic Key Management

8.1 Key Generation

The Module performs key generation functions in response to API function calls by the referencing application. The Module supports generation of DSA, RSA, and Diffie-Hellman public and private keys in addition to the symmetric keys used by the symmetric algorithms such as Triple-DES and AES.

When operating in FIPS approved mode, only a single random number generator is available for key generation. As described in Table *13.1 Approved Cryptographic Algorithms,* this is an ANSI X9.31 Appendix A.2.4 pseudo-random number generator.

For detailed information on the set of supported cryptographic algorithms please refer to section *13 Cryptographic Algorithms*.

8.2 Key Establishment

As allowed in Annex D to the FIPS 140-2 publication, when in FIPS Approved mode the Module provides methods for asymmetric key establishment. Specifically, the Module provides the RSA key wrapping and Diffie-Hellman key agreement methods.

8.3 Key Entry and Output

The Module's API provides functions for key input and key output. It is the responsibility of the application and the Crypto Officer to ensure the protection of the keys through the various facilities available to them.

8.4 Key Storage

The Module does not perform any long-term or persistent storage of keys or other critical security parameters.

8.5 Key Zeroization

The Module provides functionality via its API to clear the in-memory representation of keys. Note that some of these functions will be called internally and/or automatically during typical cryptographic processing. The following table summarizes the provided API functions:

Function	Description
BN_clear_free	Erases the big num data and releases the memory. Big nums are often used internally to store sensitive data.
BN_CTX_free	Erases the big num context and releases the memory.
BN_rand	Erases the big num using random data.
DH_free	Erases the Diffie Hellman data and releases the memory.
DSA_free	Erases the DSA data and releases the memory.
DSA_SIG_free	Erases the DSA signature data and releases the memory.
EVP_CIPHER_CTX_cleanup	Erases the Cipher Context's data and releases the memory.
EVP_MD_CTX_cleanup	Clears the message digest data and releases the memory.
EVP_PKEY_free	Clears the private key data and releases the memory.
HMAC_CTX_cleanup	Erases the HMAC data and releases the memory.
OPENSSL_cleanse	Used internally to erase contiguous bytes.
RAND_bytes	Can be used to erase data with random bytes.
RAND_free	Clears the random number generator and releases the memory.
RSA_free	Erases the RSA data and releases the memory.

Table 8.5 – Key Zeroization API Functions

9 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

This Module is a cryptographic library running on a general purpose computer system. As such, the EMI/EMC requirements conformance is not applicable to the Module and is instead assumed of the host computer system.

10 Self-Tests

The Module provides a set of self-tests that ensure the integrity and proper operation of the approved and non-approved cryptographic services allowed while in FIPS approved mode. The set of self-tests includes both power-up and conditional tests. The Module does not implement any Critical Function Tests.

Failure of any of the tests causes the Module to enter an error state. The error state is recorded in an internal flag variable *fips_selftest_fail* which may be checked within the Show Status service via *FIPS_selftest_failed* function. While in the error state

cryptographic operations are disabled and remain so until a subsequent successful execution of the set of power-up self-tests has been performed via the *GSCM_FIPS_mode_set* function.

Critical Security Parameters (CSP) are not automatically zeroized upon entering the error state. However, the Key Zeroization service may be used to zeroize CSPs through the API function calls itemized in table *8.5 – Key Zeroization API Functions*.

10.1 Power-Up Tests

The set of power-up tests are conducted both automatically at the time of Module initialization and on-demand. As discussed in section *2.4 Approved Mode of Operation*, the power-up self-tests are conducted automatically when the module is initialized by the referencing application via a call to the *GSCM_FIPS_mode_set* function. The referencing application may optionally initiate an on-demand execution of the power-up self-tests by calling the *FIPS_selftest* function.

The power-up tests consist of an integrity check of the Module, as described in section *2.4 Approved Mode of Operation*, and a set of cryptographic algorithm tests.

The following table summarizes the set of cryptographic algorithm tests performed and their associated algorithms:

Algorithm	**Cryptographic Algorithm Test**
DES (ECB)[1]	Encryption and decryption
Triple-DES (ECB) - 2-key	Encryption and decryption
Triple-DES (ECB) – 3-key	Encryption and decryption
AES	Encryption and decryption with 128-bit key
DSA	Power-up self-test for DSA signature generation and verification using a pair-wise consistency test
RSA	Pair-wise consistency using public key encryption and private key decryption Signature and verification known answer tests
SHA-1	One-way hash
HMAC-SHA-1	One-way hash
HMAC-SHA-224	One-way hash
HMAC-SHA-256	One-way hash
HMAC-SHA-384	One-way hash
HMAC-SHA-512	One-way hash
Random Number Generation	Known Initialization Vector

Table 10.1 – Power-Up Cryptographic Algorithm Tests

[1] DES is a non-FIPS approved algorithm that can be executed while FIPS mode is set. It is provided for backwards compatibility and must not be used in the FIPS approved mode of operation.

10.2 Conditional Tests

Conditional tests are conducted automatically when accessing the applicable cryptographic services. For the Module, the conditional tests include pair-wise consistency tests and a continuous random number generator test. Software/firmware load tests, manual key entry tests, and bypass tests are not implemented as the Module does not perform these functions.

The following table summarizes the conditional tests performed by the module:

Algorithm	Conditional Test
DSA	Pair-wise consistency test using signing and verification
RSA	Pair-wise consistency test using public encryption and private decryption
Random Number Generation	Continuous random number generator test as defined in FIPS publication 140-2 section 4.9.2.

Table 10.2 – Conditional Tests

11 Design Assurance

11.1 Configuration Management

The software development processes for the Module are managed within the Borland StarTeam Software Change and Configuration Management (SCM) tool. The software development processes include version control, configuration management, change control, and defect tracking.

All source code and associated documentation for the Module are stored and managed within the StarTeam Tool. Items stored within StarTeam are protected from unauthorized access and modification using StarTeam's internal user authentication and management mechanisms. Access to items pertaining to the Module is currently limited to the GlobalSCAPE Cryptographic Module development team.

11.2 Versioning

Internally, the StarTeam Tool automatically assigns distinct version numbers to each version of the source code files, documentation, Module binary, and other accompanying resources. These version numbers are used internally to manage the software development life cycle.

Externally, the versions for the various configuration items comprising the Module package are specified using a version number in the format X.X.X (e.g. 1.0.1). The first

number represents the major version, the second number represents the minor version, and the third number represents the build number.

The major number is only updated when the Module package experiences significant modifications. The minor number is updated when the Module package experiences minor modifications. The major and minor numbers will be identical across all configuration items comprising a validated version of the Module package. The build number is modified independently within each configuration item for every change to that specific item.

The external version numbers for configuration items comprising the Module package are specified as follows:

- The Module DLL Binary (GSCrypto.dll) – the version number is specified as a resource to the DLL file and may be viewed in the set of file properties within the Windows Operating System. The version is updated manually for each revision of the binary.
- Associated documentation – documentation, including the FIPS 140-2 Non-Proprietary Security Policy, FIPS 140-2 Finite State Model, and FIPS 140-2 Crypto Officer and User Guide, specify the version number on the document title page. The version number is updated manually for each revision of the document.

12 Mitigation of Other Attacks

The Module does not implement mechanisms for the mitigation of any specific attacks.

13 Cryptographic Algorithms

The Module supports operation in both a FIPS-mode and a non-FIPS mode. FIPS-mode is enabled and disabled using the GSCM_FIPS_mode_set API function.

13.1 Approved Cryptographic Algorithms

When operating in the FIPS Mode, the Module will perform the following FIPS Approved Cryptographic Algorithms:

- Triple-DES
- Advanced Encryption Standard (AES)
- Digital Signature Algorithm (DSA)[2]
- Rivest, Shamir, Adleman (RSA) for Digital Signatures
- Secure Hashing Algorithm (SHA-1 and SHA-2)
- Keyed-Hash Message Authentication Code (HMAC)
- ANSI X9.31 Appendix A.2.4 pseudo-random number generation

[2] Users of the Module are responsible for using a minimum of a 1024 bit key length with the DSA algorithm. Failing to do so will result in operating in a non-FIPS approved mode.

The following table summarizes the set of FIPS approved cryptographic algorithms.

Algorithm Type	Algorithm	Standard	Algorithm Validation Certificate	Use
Symmetric Cipher	Triple-DES – CBC, CFB8, CFB64, ECB, OFB modes	SP800-67 (Ref: 5)	586	Encryption, Decryption
Symmetric Cipher	AES (128, 192, 256 bit keys) – CBC, CFB8, CFB128, ECB, OFB modes	FIPS 197 (Ref: 6)	618	Encryption, Decryption
Asymmetric Algorithm	RSA	ANSI X9.31 (Ref: 10), RSASSA-PKCS1_V1_5 (Ref: 11), RSASSA-PSS (Ref: 11)	287	Signature Generation, Signature Verification
Asymmetric Algorithm	DSA	FIPS 186-2 (Ref: 7)	240	Signature Generation, Signature Verification
Message Digest	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512	FIPS 180-2 (Ref: 8)	666	Hashing
Message Authentication	HMAC-SHA-1 HMAC-SHA-224 HMAC-SHA-256 HMAC-SHA384 HMAC-SHA-512	FIPS 198 (Ref: 9)	320	Integrity
Random Number Generation	ANSI X9.31 Appendix A.2.4	ANSI X9.31 Appendix A.2.4 (Ref: 10)	388	Random Number Generation

Table 13.1 – Approved Cryptographic Algorithms

13.2 Non-Approved Cryptographic Algorithms

When the Module is operating in FIPS Approved mode, a small subset of additional non-FIPS approved algorithms are allowed by the FIPS 140-2 standard and provided by the Module.

The following table summarizes the set of non-approved cryptographic algorithms allowed while in the FIPS Approved mode of operation.

Algorithm Type	Algorithm	Standard	Use
Asymmetric Algorithm	DH (provides 80 to 256 bits of equivalent encryption strength)	ANSI X9.42-2001 (Ref: 13)	Key Agreement
Asymmetric Algorithm	RSA (provides 80 to 150 bits of equivalent encryption strength)	PKCS #1 (Ref: 11)	Key Wrapping

Table 13.2 – Non-Approved Cryptographic Algorithms

A minimum key size of 1024 bits and a maximum key size of 15360 bits must be used with Diffie-Hellman key agreement/key establishment, providing between 80 and 256 bits of equivalent encryption strength

A minimum key size of 1024 bits and a maximum key size of 4096 bits must be used with RSA key wrapping/key establishment, providing between 80 and 150 bits of equivalent encryption strength.

13.3 Additional Algorithms

When not operating in FIPS-mode, the Module provides an additional set of algorithms. Attempts to use these algorithms while in FIPS-mode are not allowed and, with the exception of the DES algorithm, are prevented by the Module. Callers of the Module's API are responsible for not using the DES algorithm while in FIPS-mode as doing so will result in operating in a non-FIPS approved mode.

The following table summarizes the set of additional non-FIPS approved mode algorithms.

Service Type	Algorithm	Use
Message Digest	MD2	Hashing
Message Digest	MD4	Hashing
Message Digest	MD5	Hashing
Message Digest	MDC2	Hashing
Message Digest	RIPEMD160	Hashing
Symmetric Cipher	Blowfish – CBC, ECB, CFB, OFB modes	Encryption, Decryption
Symmetric Cipher	CAST5 – CBC, ECB, CFB, OFB modes	Encryption, Decryption
Symmetric Cipher	DES – CBC, ECB, CFB, OFB modes	Encryption, Decryption
Symmetric Cipher	RC2 – CBC, ECB, CFB, OFB modes	Encryption, Decryption
Symmetric Cipher	RC4	Encryption, Decryption
Symmetric Cipher	RC5 – CBC, ECB, CFB, OFB modes	Encryption, Decryption
Symmetric Cipher	IDEA – CBC, ECB, CFB, OFB modes	Encryption, Decryption

Table 13.3 – Additional Algorithms

14 References

1. National Institute of Standards and Technology. "Security Requirements for Cryptographic Modules." FIPS PUB 140-2, May 25, 2001.
2. National Institute of Standards and Technology. "Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program." Released March 28, 2003. Updated July 26, 2007.
3. OpenSSL Project. "OpenSSL Documents" <http://www.openssl.org/docs/>
4. GlobalSCAPE Cryptographic Module, Version 1.0.0 FIPS 140-2 Finite State Model, Version 1.0.0. Revised 5/31/2007.

5. National Institute of Standards and Technology. ”NIST Special Publication 800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher”, SP800-67, May 2004.
6. National Institute of Standards and Technology. “Specification for the Advanced Encryption Standard (AES).” FIPS PUB 197, November 26, 2001.
7. National Institute of Standards and Technology. “Digital Signature Standard (DSS).” FIPS PUB 186-2, January 27, 2000.
8. National Institute of Standards and Technology. “Secure Hash Standard.” FIPS PUB 180-2, August 1, 2002.
9. National Institute of Standards and Technology. “The Keyed-Hash Message Authentication Code (HMAC).” FIPS PUB 198, March 6, 2002.
10. X9F - Data & Information Security Committee. “Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA).” X9.31-1998, 1998.
11. RSA Laboratories. “PKCS #1 v2.1: RSA Cryptography Standard.” June 14, 2002.
12. GlobalSCAPE Cryptographic Module, Version 1.0.0 FIPS 140-2 Crypto Officer and User Guide, Version 1.0.0. Revised June 6, 2007.
13. American National Standards Institute. “Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography, X9.42-2001, 2000.