NetBrain OpenSSL Cryptographic Module

Software version 1.0

FIPS 140-2 Non-Proprietary Security Policy

Document version 1.1

Last update: 2017-11-14

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

Table of Contents

1. Cryptographic Module Specification

This document is the non-proprietary FIPS 140-2 Security Policy for version 1.0 of NetBrain OpenSSL Cryptographic Module. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-2 (Federal Information Processing Standards Publication 140-2) for a Security Level 1 software module.

The following sections describe the cryptographic module and how it conforms to the FIPS 140-2 specification in each of the required areas.

1.1. Module Overview

The NetBrain network management products provide network automation platform which gives the user visibility of the live network within a single snapshot for network management, network troubleshooting, path analysis, and data reporting. The NetBrain OpenSSL Cryptographic Module (hereafter also referred to as “the module”) is a set of software libraries implementing general purpose cryptographic algorithms. Through a C language Application Program Interface (API), the module provides cryptographic services to NetBrain network management products.

The software block diagram below shows the module, its interfaces with the operational environment and the delimitation of its logical boundary:

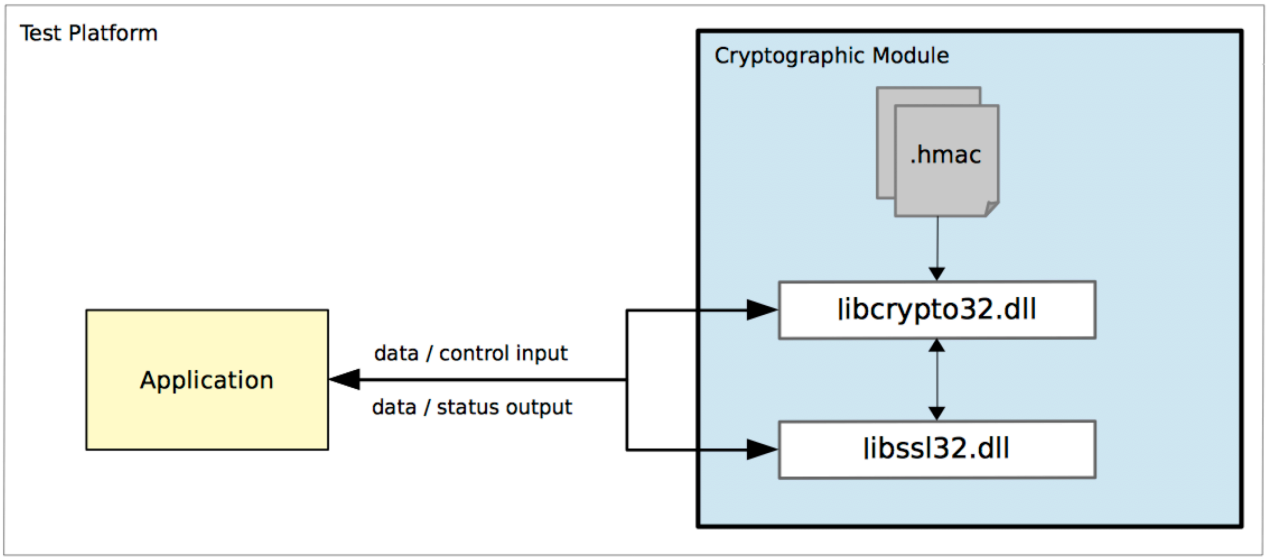

Figure 1 - Software Block Diagram

The module’s logical boundary consists of the shared libraries and the integrity check files used for integrity tests. The following table lists the files that comprise the module:

Component	Description
libcrypto32.dll	Shared library for cryptographic implementations
libssl32.dll	Shared library for SSL/TLS network protocols
.libcrypto32.dll.hmac	Integrity check file for libcrypto shared library with HMAC value “15318675ed1f8a1c11ce5f0546d60d08d9a31d1a6dafd9a184dea48f503840b1"
.libssl32.dll.hmac	Integrity check file for libssl shared library with HMAC value “9b01f4b4713ec2f1ee198aa1a7d219c1f70c2c9a72aa292ba6f31391c4b905de”

Table 1 – Cryptographic Module Components

The module is aimed to run in a general-purpose computer. The physical boundary of the module is the surface of the case of the target platform, as shown with dotted lines in the diagram below:

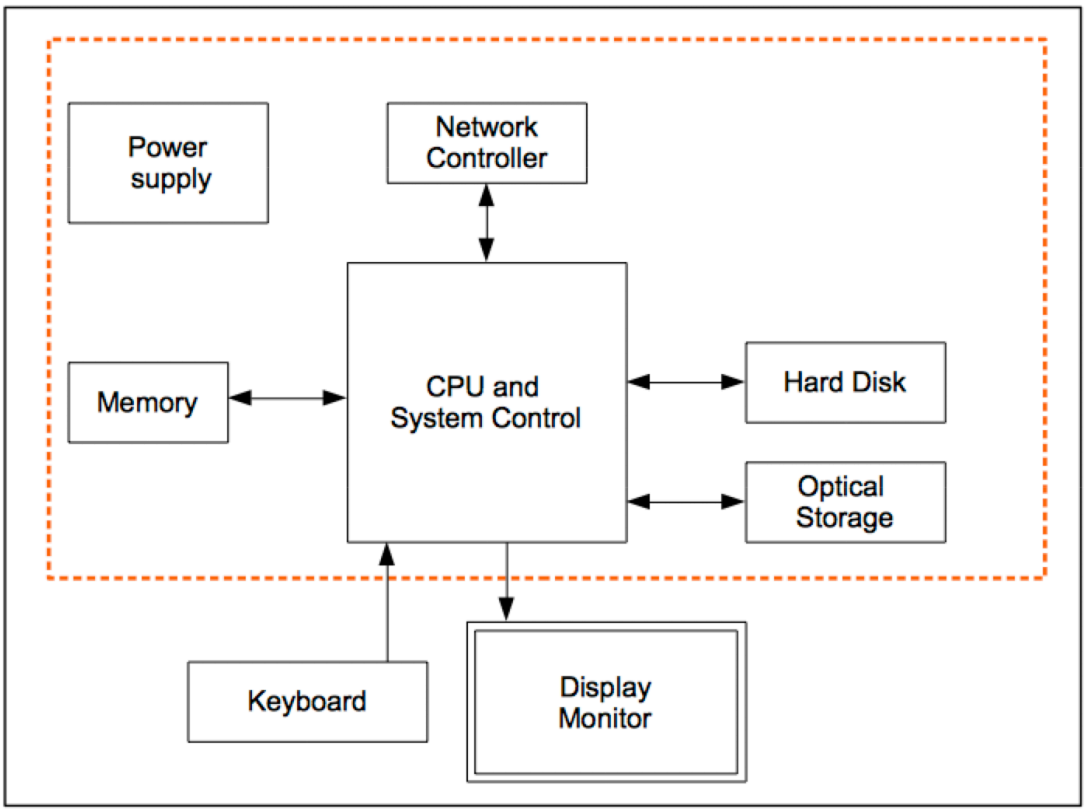

Figure 2 - Cryptographic Module Physical Boundary

1.2. FIPS 140-2 Validation

For the purpose of the FIPS 140-2 validation, the module is a software-only, multi-chip standalone cryptographic module validated at overall security level 1. The table below shows the security level claimed for each of the eleven sections that comprise the FIPS 140-2 standard:

FIPS 140-2 Section		Security 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
Overall Level		1

Table 2 - Security Levels

The module has been tested on the following multichip standalone platform:

Test Platform	Processor	Operating System
Dell PowerEdge R330	Intel Xeon E3 family	Windows Server 2012 R2 Standard

Table 3 - Tested Platform

1.3. Modes of operation

The module supports two modes of operation:

- FIPS mode (the FIPS Approved mode of operation): only approved or allowed security functions with sufficient security strength can be used.
- non-FIPS mode (the non-Approved mode of operation): only non-approved security functions can be used.

The module enters FIPS mode after power-up tests succeed. Once the module is operational, the mode of operation is implicitly assumed depending on the security function invoked and the security strength of the cryptographic keys.

Critical Security Parameters (CSPs) used or stored in FIPS mode are not used in non-FIPS mode, and vice versa.

2. Cryptographic Module Ports and Interfaces

As a software-only module, the module does not have physical ports. For the purpose of the FIPS 140-2 validation, the physical ports are interpreted to be the physical ports of the hardware platform on which it runs.

The logical interfaces are the API through which applications request services. The following table summarizes the four logical interfaces:

Logical Interface	Description
Data Input	API input parameters for data
Data Output	API output parameters for data
Control Input	API function calls
Status Output	API return codes and error messages

Table 4 - Ports and Interfaces

3. Roles, Services and Authentication

3.1. Roles

The module supports the following roles:

- User role: performs all services as listed in Table 5 and Table 6, except module initialization.
- Crypto Officer role: performs module initialization as listed in Table 5.

The User role and Crypto Officer role are implicitly assumed by the entity accessing the module services.

3.2. Services

The module provides services to users that assume one of the available roles. All services are described in detail in the user documentation.

The following table shows the services available in FIPS mode of operation. For each service, it lists the algorithm(s) involved, the CSP(s) and how they are accessed. For each of the Approved algorithms, the CAVS certificate number is listed. For each of the non-Approved but allowed algorithms, the “Allowed” note is added.

Service	Algorithms	CSP	Access	CAVP Cert. #s / Note
Symmetric encryption and decryption	AES (ECB, CBC, CFB1, CFB8, CFB128, OFB, CTR, CCM and GCM modes)	128/192/256-bit AES key	Read	AES Cert. #4690
	3-key Triple-DES (ECB, CBC, CFB1, CFB8, CFB64, OFB and CTR modes)	192-bit 3-key Triple-DES key	Read	Triple-DES Cert. #2493
RSA key generation	FIPS186-4 Appendix B.3.3 RSA key pair generation	RSA public and private keys with 2048/3072-bit modulus size	Write	RSA Cert. #2559
		RSA public and private keys with modulus size > 3072 bits	Write	Allowed
RSA signature generation	RSA X9.31 signature generation with SHA-256, SHA-384 and SHA-512	RSA private key with 2048/3072/4096-bit modulus size	Read	RSA Cert. #2559
		RSA private key with modulus size > 4096-bit	Read	Allowed

Service	Algorithms	CSP	Access	CAVP Cert. #s / Note
	RSA PKCS#1 v1.5 signature generation with SHA-224, SHA-256, SHA-384 and SHA-512	RSA private key with 2048/3072/4096-bit modulus size	Read	RSA Cert. #2559
		RSA private key with modulus size > 4096-bit	Read	Allowed
	RSA PSS signature generation with SHA-224, SHA-256, SHA-384 and SHA-512	RSA private key with 2048/3072/4096-bit modulus size	Read	RSA Cert. #2559
		RSA private key with modulus size > 4096-bit	Read	Allowed
RSA signature verification	RSA X9.31 signature verification with SHA-1, SHA-256, SHA-384 and SHA-512	RSA public key with 1024/2048/3072-bit modulus size	Read	RSA Cert. #2559
		RSA public key with modulus size > 3072-bit	Read	Allowed
	RSA PKCS#1 v1.5 signature verification with SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	RSA public key with 1024/2048/3072-bit modulus size	Read	RSA Cert. #2559
		RSA public key with modulus size > 3072-bit	Read	Allowed
	RSA PSS signature verification with SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	RSA public key with 1024/2048/3072-bit modulus size	Read	RSA Cert. #2559
		RSA public key with modulus size > 3072-bit	Read	Allowed
DSA key generation	FIPS 186-4 DSA key pair generation	DSA public and private keys with L=2048, N=224; L=2048, N=256; L=3072, N=256	Write	DSA Cert. #1242
DSA domain parameter generation	DSA PQG generation with SHA-224, SHA-256, SHA-384 and SHA-512	DSA domain parameters with L=2048, N=224; L=2048, N=256; L=3072, N=256	Write	DSA Cert. #1242

Service	Algorithms	CSP	Access	CAVP Cert. #s / Note
DSA domain parameter verification	DSA PQG verification with SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	DSA domain parameters with L=1024, N=160; L=2048, N=224; L=2048, N=256; L=3072, N=256	Read	DSA Cert. #1242
DSA signature generation	DSA signature generation with SHA-224, SHA-256, SHA-384 and SHA-512	DSA private key with L=2048, N=224; L=2048, N=256; L=3072, N=256	Read	DSA Cert. #1242
DSA signature verification	DSA signature verification with SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	DSA public key with L=1024, N=160; L=2048, N=224; L=2048, N=256; L=3072, N=256	Read	DSA Cert. #1242
ECDSA key generation	FIPS186-4 Appendix B.4.2 ECDSA key pair generation	ECDSA public and private keys with P-256, P-384 and P-521 curves	Write	ECDSA Cert. #1158
ECDSA public key validation	ECDSA public key validation (PKV)	ECDSA public key with P-256, P-384 and P-521 curves	Read	ECDSA Cert. #1158
ECDSA signature generation	ECDSA signature generation with SHA-224, SHA-256, SHA-384 and SHA-512	ECDSA private key with P-256, P-384 and P-521 curves	Read	ECDSA Cert. #1158
ECDSA signature verification	ECDSA signature verification with SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	ECDSA public key with P-256, P-384 and P-521 curves	Read	ECDSA Cert. #1158
Message digest	SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	N/A	N/A	SHS Cert. #3840
Message authentication	HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512	At least 112-bit HMAC key	Read	HMAC Cert. #3105
	CMAC using AES	128/192/256-bit AES key	Read	AES Cert. #4690

Service	Algorithms	CSP	Access	CAVP Cert. #s / Note
	CMAC using 3-key Triple-DES	192-bit 3-key Triple-DES keys	Read	Triple-DES Cert. #2493
Random number generation	NDRNG (used to seed the DRBG)	N/A	N/A	Allowed
	NIST SP800-90A Hash_DRBG with SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	Entropy input string, internal state	Read, Write	DRBG Cert. #1591
	NIST SP800-90A HMAC_DRBG with SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512			
	NIST SP800-90A CTR_DRBG with AES-128, AES-192 and AES-256			
AES key wrapping	NIST SP 800-38F AES Key Wrapping using KW mode	128/192/256-bit AES key	Read	AES Cert. #4690
RSA key wrapping	RSA key encapsulation with encryption and decryption primitives	RSA public and private keys with size >= 2048-bit	Read	Allowed
Diffie-Hellman key agreement	NIST SP 800-56A KAS FFC except KDF	Diffie-Hellman public and private keys with 2048-bit size	Read, Write	CVL Cert. #1335
		Diffie-Hellman public and private keys with size > 2048-bit	Read, Write	Allowed
EC Diffie-Hellman key agreement	NIST SP 800-56A KAS ECC except KDF	EC Diffie-Hellman public and private keys with P-256, P-384 and P-521 curves	Read, Write	CVL Cert. #1335
TLS network protocol	NIST SP 800-135v1 key derivation in TLS v1.0, v1.1 and v.1.2	TLS pre-master secret and master secret	Read	CVL Certs. #1336
	AES, Triple-DES, RSA, DSA, ECDSA, HMAC, Diffie-Hellman and EC Diffie-Hellman	See the CSPs listed above	Read, Write	See the certs. listed above
	MD5 (used in the PRF for TLS v1.0 and v1.1)	N/A	N/A	Allowed

Service	Algorithms	CSP	Access	CAVP Cert. #s / Note
Show status	N/A	N/A	N/A	N/A
Self-tests	AES, Triple-DES, RSA, DSA, ECDSA, SHS, HMAC, DRBG, Diffie-Hellman and EC Diffile-Hellman	N/A	N/A	N/A
Zeroization	N/A	All CSPs	Zeroize	N/A
Module initialization	N/A	N/A	N/A	N/A

Table 5 - Services in FIPS mode of operation

Notice for the TLS protocol, no parts of this protocol, other than the key derivation function (KDF), have been tested by the CAVP.

The table below lists the services only available in non-FIPS mode of operation. These services invoke non-Approved algorithms or algorithms using non-compliant key sizes.

Service	Algorithms	Key Parameters	Access
Symmetric encryption and decryption	2-key Triple-DES	2-key Triple-DES key	Read
RSA key generation	RSA key generation with non-compliant key size	RSA public and private key with modulus size < 2048-bit	Write
RSA signature generation	RSA X9.31/PKCS#1/v1.5/PSS signature generation with non-compliant key size	RSA private key with modulus size < 2048-bit	Read
	RSA X9.31/PKCS#1/v1.5/PSS signature generation with SHA-1	RSA private key	Read
RSA signature verification	RSA X9.31/PKCS#1/v1.5/PSS signature verification with non-compliant key size	RSA public key with modulus size < 1024-bit	Read
DSA key generation	DSA key generation with non-compliant key size	DSA public and private key with L and N pair not listed in Table 5	Write
DSA domain parameter generation	DSA PQG generation with non-compliant key size	DSA domain parameters with L and N pair not listed in Table 5	Write
DSA domain parameter verification	DSA PQG verification with non-compliant key size	DSA domain parameters with L and N pair not listed in Table 5	Read

Service	Algorithms	Key Parameters	Access
DSA signature generation	DSA signature generation with non-compliant key size	DSA private key with L and N pair not listed in Table 5	Read
	DSA signature generation with SHA-1	DSA private key	Read
DSA signature verification	DSA signature verification with non-compliant key size	DSA private key with L and N pair not listed in Table 5	Read
ECDSA signature generation	ECDSA signature generation with SHA-1	ECDSA private key	Read
Message authentication	HMAC with non-compliant key size	HMAC key with size < 112-bit	Read
	CMAC using 2-key Triple-DES	2-key Triple-DES key	Read
RSA key wrapping	RSA key encapsulation with non-compliant key size	RSA public and private key with size < 2048-bit	Read
Diffie-Hellman key agreement	Diffie-Hellman with non-compliant key size	Diffie-Hellman public and private key with size < 2048-bit	Read, Write
Message Digest	MD5	N/A	N/A

Table 6 - Services in non-FIPS mode of operation

3.3. Operator Authentication

The module does not implement user authentication. The role of the user is implicitly assumed based on the service requested.

4. Physical Security

The module is comprised of software only and therefore this security policy does not make any claims on physical security.

5. Operational Environment

5.1. Applicability

The module operates in a modifiable operational environment per FIPS 140-2 security level 1 specifications. The module runs on a general-purpose computer as specified in Table 3.

5.2. Policy

The operating system is restricted to a single operator; concurrent operators are explicitly excluded.

The application that requests cryptographic services is the single user of the module.

6. Cryptographic Key Management

The following table summarizes the keys and CSPs that are used by the cryptographic services implemented in the module:

Name	Generation	Entry and Output	Zeroization
AES keys	N/A	The key is passed into the module via API input parameters in plaintext.	EVP_CIPHER_CTX_cleanup()
Triple-DES keys			EVP_CIPHER_CTX_cleanup()
HMAC keys			HMAC_CTX_cleanup()
RSA public-private keys	The public-private keys are generated using FIPS 186-4 key generation method, and the random value used in the key generation is generated using SP800-90A DRBG	The key is passed into the module via API input parameters in plaintext. The key is passed out of the module via API output parameters in plaintext.	RSA_free()
DSA public-private keys			DSA_free()
ECDSA public-private keys			EC_KEY_free()
Diffie-Hellman public-private keys			DH_free()
EC Diffie-Hellman public-private keys			EC_KEY_free()
TLS pre-master secret and master secret	Established during TLS handshake	N/A	SSL_free() and SSL_clear()
Entropy input string	Obtained from NDRNG	N/A	FIPS_drbg_free()
DRBG internal state	Derived from the entropy string	N/A	FIPS_drbg_free()

Table 7 - Life cycle of Keys and CSPs

The following sections describe how keys and CSPs are managed during its life cycle.

6.1. Random Number Generation

The Module provides an SP800-90A-compliant DRBG for the generation of random values used in asymmetric keys and for providing the random number generation service to the user.

The module uses a NDRNG as the entropy source. The NDRNG is based on the Windows RNG and CPU Jitter RNG. It provides sufficient entropy for seeding the DRBG. The Windows RNG is provided by the OS, so it is within the module's physical boundary but outside its logical boundary. The CPU Jitter RNG is implemented by the module, so it is within the module's logical boundary.

6.2. Key Generation

For generating HMAC keys and symmetric keys, the module does not provide any dedicated key generation service. However, the random number generation service can be called by the user to obtain random numbers which can be used as key material for symmetric algorithms or HMAC. The key material of HMAC keys and symmetric keys may also be generated during the Diffie-Hellman or EC Diffie-Hellman key agreement.

For generating RSA, DSA and ECDSA keys, the module implements asymmetric key generation services compliant with FIPS186-4, and using DRBG compliant with NIST SP800-90A. In accordance with FIPS140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per SP800-133 (vendor affirmed).

6.3. Key Establishment

The module provides Diffie-Hellman key agreement, EC Diffie-Hellman key agreement, AES key wrapping, and RSA key wrapping.

- Diffie-Hellman key agreement provides between 112 and 256 bits of encryption strength; non-compliant less than 112 bits of encryption strength.
- EC Diffie-Hellman key agreement provides between 128 and 256 bits of encryption strength.
- AES key wrapping provides between 128 and 256 bits of encryption strength.
- RSA key wrapping provides between 112 and 256 bits of encryption strength; non-compliant less than 112 bits of encryption strength.

6.4. Key Entry / Output

The module does not support manual key entry or intermediate key generation key output. The keys are provided to the module via API input parameters in plaintext form and output via API output parameters in plaintext form. This is allowed by FIPS140-2 IG 7.7, according to the "CM Software to/from App Software via GPC INT Path" entry which refers to keys communicated within the physical boundary of the GPC.

6.5. Key / CSP Storage

Symmetric keys, HMAC keys, public and private keys are provided to the module by the calling application, and are destroyed when released by the appropriate API function calls.

The module does not perform persistent storage of keys. The only exception is the HMAC-SHA-256 key used for integrity test, which is stored in the module and relies on the operating system for protection.

6.6. Key / CSP Zeroization

The memory occupied by keys is allocated by regular memory allocation operating system calls. The application is responsible for calling the appropriate destruction functions provided in the module's API. The zeroization functions overwrite the memory occupied by keys with “zeros” and deallocates the memory with the regular memory deallocation operating system call.

7. Self-Tests

7.1. Power-Up Tests

The module performs power-up tests automatically without any operator intervention when it is loaded into memory. Power-up tests ensure that the module is not corrupted and that the cryptographic algorithms work as expected.

While the module is executing the power-up tests, services are not available, and input and output are inhibited. The module does not return control to the calling application until the power-up tests are completed. On successful completion of the power-up tests, the module enters operational mode and cryptographic services are available. If the module fails any of the power-up tests, it will return an error code and enter into the Error state to prohibit any further cryptographic operations. The module must be re-loaded in order to clear the error condition.

7.1.1. Integrity Tests

The integrity of the module is verified by comparing an HMAC-SHA-256 value calculated at run time with the HMAC value stored in the module that was computed at build time.

7.1.2. Cryptographic Algorithm Tests

The module performs self-tests on all FIPS-Approved cryptographic algorithms supported in the approved mode of operation, using the Known Answer Test (KAT) and Pair-wise Consistency Test (PCT) as shown in the following table:

Algorithm	Test
DRBG	• KATs of CTR_DRBG, Hash_DRBG and HMAC_DRBG
SHS	• KATs of SHA-1, SHA-256 and SHA-512
HMAC	• KATs of HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512
CMAC	• KATs of AES CMAC and Triple-DES CMAC
AES	• KAT of AES ECB mode; encryption and decryption are tested separately
Triple-DES	• KAT of Triple-DES ECB mode; encryption and decryption are tested separately
RSA	• KATs of RSA signature scheme with 2048-bit key; signature generation and verification are tested separately • KATs of RSA key wrapping scheme with 2048-bit key; encryption and decryption are tested separately
ECDSA	• PCT of ECDSA signature generation and verification with P-256 curve

Algorithm	Test
DSA	• PCT of DSA signature generation and verification with 2048-bit key
Diffie-Hellman	• KAT of Primitive “Z” Computation with 2048-bit key
EC Diffie-Hellman	• KAT of primitive “Z” computation with P-256 curve

Table 8 - Self-Tests

7.2. On-Demand Self-Tests

On-Demand self-tests can be invoked by powering-off and reloading the module which cause the module to run the power-up tests again. During the execution of the on-demand self-tests, services are not available and no data output or input is possible.

7.3. Conditional Tests

The module performs conditional tests on the cryptographic algorithms, using the Pair-wise Consistency Test (PCT) and Continuous Random Number Generator Test (CRNGT), as shown in the following table.

Algorithm	Test
RSA key generation	• PCT of RSA signature generation and verification
DSA key generation	• PCT of DSA signature generation and verification
ECDSA key generation	• PCT of ECDSA signature generation and verification
DRBG	• CRNGT and health-tests as defined in section 11.3 of NIST SP800-90A
NDRNG	• CRNGT

Table 9 - Conditional Tests

8. Guidance

8.1. Delivery

The module is distributed as part of the NetBrain products and is not available for direct download to the general public. The module is to be installed together with the NetBrain products on the operational environment specified in Section 5.

8.2. Crypto Officer Guidance

To configure the operating environment to support FIPS, the crypto officer must make sure the registry value

HKLM\System\CurrentControlSet\Control\Lsa\FIPSAlgorithmPolicy\Enabled

is set to “1” in Windows OS. Once it is configured, it can’t be modified. The crypto officer must reboot the NetBrain product for all FIPS-compliant changes to take effect.

The Crypto Officer should also make sure that the FIPS validated module is installed by verifying the HMAC values of the module’s integrity check files as listed in Table 1.

8.3. User Guidance

In order to run in FIPS mode, the module must be operated using the FIPS Approved services as listed in Table 5 of this Security Policy. Any use of non-approved services will put the module in the non-FIPS mode implicitly.

8.3.1. TLS and Diffie-Hellman

The TLS protocol implementation provides both server and client sides. In order to operate in FIPS mode, digital certificates used for server and client authentication shall comply with the restrictions of key size and message digest algorithms imposed by NIST SP800-131A. In addition, as required also by NIST SP800-131A, Diffie-Hellman with keys smaller than 2048 bits must not be used. Therefore, the crypto officer must ensure that:

- in case the module is used as a TLS server, the Diffie-Hellman parameters of the aforementioned API call must be 2048 bits or larger;
- in case the module is used as a TLS client, the TLS server must be configured to only offer Diffie-Hellman keys of 2048 bits or larger.

8.3.2. AES GCM IV

In case the module’s power is lost and then restored, the key used for the AES GCM encryption or decryption shall be redistributed.

The AES GCM IV generation is in compliance with RFC5288 and shall only be used for the TLS protocol v1.2 to be compliant with FIPS140-2 IG A.5, provision 1 (“TLS protocol IV generation”).

8.3.3. Triple-DES encryption

Data encryption using the same three-key Triple-DES key shall not exceed 2^{28} Triple-DES blocks in accordance to SP800-67 and FIPS140-2 IG A.13.

9. Mitigation of Other Attacks

The module does not implement security mechanisms to mitigate other attacks.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CAVS	Cryptographic Algorithm Validation Scheme
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
KW	Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
NDRNG	Non-Deterministic Random Number Generator
OFB	Output Feedback
OS	Operating System
PCT	Pair-wise Consistency Test
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
TLS	Transport Layer Security

Appendix B. References

FIPS140-2	FIPS PUB 140-2 - Security Requirements for Cryptographic Modules http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
FIPS140-2_IG	Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf
FIPS186-4	Digital Signature Standard (DSS) http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
RFC5288	AES Galois Counter Mode (GCM) Cipher Suites for TLS https://tools.ietf.org/html/rfc5288
SP800-38F	NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP800-56A	NIST Special Publication 800-56A - Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised) http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf
SP800-67	NIST Special Publication 800-67 Revision 1 – Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-67r1.pdf
SP800-90A	NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-131A	NIST Special Publication 800-131A Revision 1- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Length http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf
SP800-135	NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf