expresslogic

NetX Crypto

Version 5.11SP1-FIPS

FIPS 140-2 Non-Proprietary Security Policy

Document Version 1.3

Last update: 2019-03-06

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

Table of Contents

1 Cryptographic Module Specification 6
1.1 Module Overview 6
1.2 FIPS 140-2 Validation 7
1.3 Modes of operation 9
2 Cryptographic Module Ports and Interfaces 10
3 Roles, Services and Authentication 11
3.1 Roles 11
3.2 Services 11
3.3 Algorithms 13
3.4 Operator Authentication 15
4 Physical Security 16
5 Operational Environment 17
5.1 Applicability 17
5.2 Policy 17
6 Cryptographic Key Management 18
6.1 Random Number Generation 18
6.2 Key Generation 18
6.3 Key Derivation 18
6.4 Key Establishment 19
6.5 Key Entry/Output 19
6.6 Key/CSP Storage 19
6.7 Key/CSP Zeroization 19
7 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC) 20
8 Self Tests 21
8.1 Power-Up Tests 21
8.1.1 Integrity Tests 21
8.1.2 Cryptographic algorithm tests 21
8.2 On-Demand self-tests 22
8.3 Conditional Tests 22
9 Guidance 23
9.1 Crypto Officer Guidance 23
9.1.1 Module installation 23
9.1.2 Module configuration 23
9.2 User Guidance 23
9.2.1 API Functions 23

9.2.2 Triple-DES Data Encryption 23
9.2.3 Handling FIPS Related Errors 23
10 Mitigation of Other Attacks 25

List of Tables

Table 1 - Security levels 8
Table 2 - Tested platform 8
Table 3 - Vendor affirmed platforms 9
Table 4 - Ports and interfaces 10
Table 5 – Services in FIPS mode of operation 12
Table 6 - Services in the non-FIPS mode of operation 13
Table 7 – FIPS-approved cryptographic algorithms 13
Table 8 – FIPS-allowed cryptographic algorithms 14
Table 9 - Non-approved cryptographic algorithms 15
Table 10 - Life cycle of keys and other critical security parameters (CSPs) 18
Table 11 - Self-tests 22

List of Figures

Figure 1 - Software block diagram 6
Figure 2 – Hardware block diagram with physical boundary 7

Copyrights and Trademarks

ThreadX, NetX, NetX Secure and NetX Crypto are registered trademarks of Express Logic, Inc.

All other product and company names are trademarks or registered trademarks of their respective holders.

1 Cryptographic Module Specification

This document is the non-proprietary FIPS 140-2 Security Policy for version 5.11SP1-FIPS of the NetX Crypto module. It contains the security rules under which the module must be operated 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 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

NetX Crypto (hereafter referred to as “the module”) is a high-performance real-time implementation of cryptographic algorithms designed to provide data encryption and authentication services. NetX Crypto is designed to plug in for NetX Secure TLS, DTLS, and IPsec modules. Applications may also use NetX Crypto as a standalone module outside network security.

The module is delivered as a software library, providing cryptographic services through a C language Application Program Interface (API). Developers integrate the module into their applications through the IAR Embedded Workbench for ARM, an integrated development environment (IDE).

The module runs on microcontrollers that include ARM Cortex M4 processors and does not use any Processor Algorithm Acceleration (PAA).

The software block diagram in Figure 1 shows the module, its interfaces with the operational environment and the delimitation of its logical boundary.

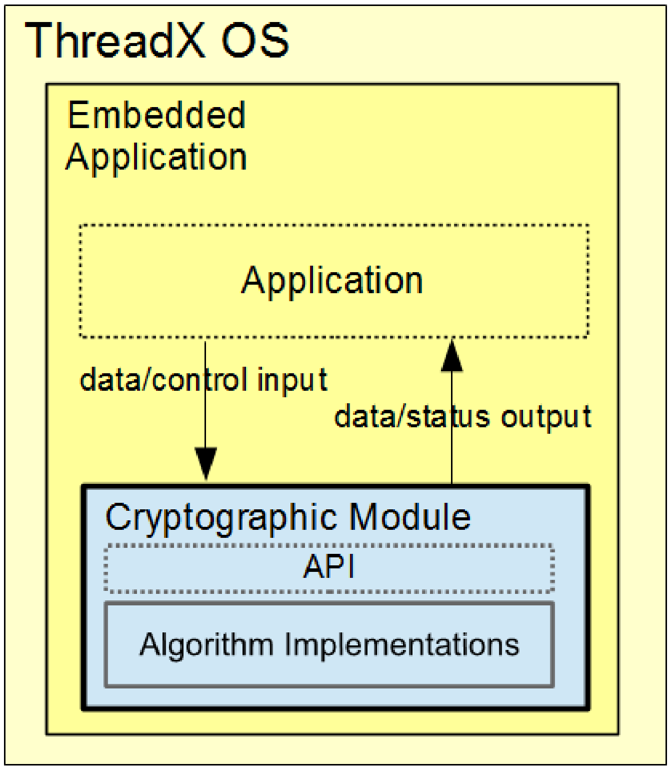

Figure 1 - Software block diagram

The cryptographic logical boundary consists of all object files included in the library nx_crypto_5.11SP1-FIPS.a, including all API functions and the implementation of all services and algorithms supported by the module.

The embedded application, the cryptographic module itself, and the underlying ThreadX operating system run on a microcontroller with an embedded microprocessor. The microcontroller is included in an evaluation board, which contains the microcontroller itself, memory, storage, power interface and several communication interfaces (e.g., RS-232, USB, Ethernet). The whole hardware platform constitutes the physical boundary of the module. Figure 2 depicts the main components of the target hardware platform and the red dotted lines show the physical boundary of the cryptographic module.

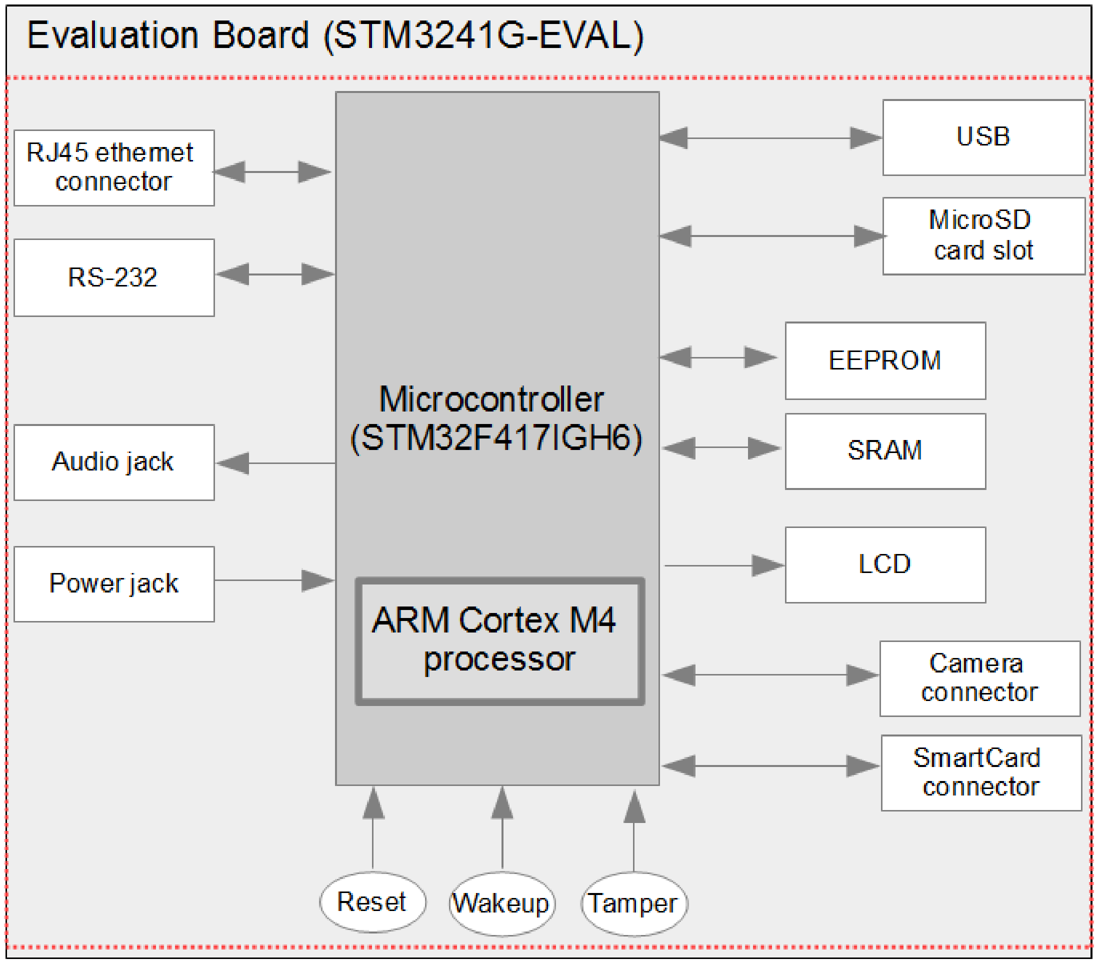

Figure 2 – Hardware block diagram with physical boundary

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. Table 1 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 1 - Security levels

The module has been tested on the platform shown in Table 2.

Evaluation Board[1]	Microcontroller[1]	Processor	Operating System
STM3241G-EVAL	STM32F417IGH6	ARM Cortex M4	ThreadX Cortex-M4/IAR Version 5.8

Table 2 - Tested platform

The module can be also used in the vendor-affirmed platforms shown in Table 3.

Evaluation Board[1]	Microcontroller[1]	Processor	Operating System
STM3240G-EVAL	STM32F407IGH6	ARM Cortex M4	ThreadX Cortex-M4/IAR Version 5.8
STM32429I-EVAL	STM32F429NIH6	ARM Cortex M4	ThreadX Cortex-M4/IAR Version 5.8
STM32439I-EVAL	STM32F439NIH6	ARM Cortex M4	ThreadX Cortex-M4/IAR Version 5.8
STM32F469I-DISCOVERY	STM32F469NIH6	ARM Cortex M4	ThreadX Cortex-M4/IAR Version 5.8

[1] Evaluation boards and microcontrollers are manufactured by STMicroelectronics.

Evaluation Board[1]	Microcontroller[1]	Processor	Operating System
STM32479I-EVAL	STM32F479NIH6	ARM Cortex M4	ThreadX Cortex-M4/IAR Version 5.8
B-L475E-IOT01A	STM32L475	ARM Cortex M4	ThreadX Cortex-M4/IAR Version 5.8

Table 3 - Vendor affirmed platforms

The CMVP makes no statement as to the correct operation of the module on the vendor-affirmed platforms.

1.3 Modes of operation

The module supports two modes of operation.

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

The module enters the FIPS mode of operation after the power-up self-tests (POST) 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.

If the POST fails, the module enters the error state (see Section 8).

Critical security parameters 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 microcontroller where the module runs.

The logical interfaces are the API through which applications request services and receive output data through return values or modified data referenced by pointers. Table 4 summarizes the four logical interfaces.

The logical interfaces of the module interact only with the application running in the microcontroller. Therefore, there is no mapping between logical and physical interfaces. A power interface is not applicable for software-only modules.

Logical Interface	Description
Data Input	API input parameters for data.
Data Output	API output parameters for data.
Control Input	API function calls, API input parameters for control.
Status Output	API return codes.

Table 4 - Ports and interfaces

3 Roles, Services and Authentication

3.1 Roles

The module supports the following roles:

- **User role**: performs all services (in both FIPS mode and non-FIPS mode of operation), except module installation and configuration. This role is assumed by the calling application accessing the module.
- **Crypto Officer role**: performs module installation and configuration.

The User and Crypto Officer roles are implicitly assumed depending on the service requested.

3.2 Services

The module provides services to calling applications that assume the user role, and human users assuming the Crypto Officer role. Table 5 and Table 6 depict all services, which are described with more detail in the user documentation. Details about the algorithms supported by the module are found in Section 3.3.

Table 5 shows the Approved services and the non-Approved but allowed services in FIPS mode of operation, the cryptographic algorithms supported for each service, the roles that can perform each service, and the keys and other Critical Security Parameters (CSPs) involved. The table also specifies how the access permission that the module has for each CSP or key, using the following convention:

- Create: the calling application can create a new CSP.
- Read: the calling application can read the CSP.
- Update: the calling application can write a new value to the CSP.
- Zeroize: the calling application can zeroize the CSP.
- n/a: the calling application does not access any CSP or key during its operation.

Service	Algorithms	Role	Keys/CSPs	Access
Cryptographic Services				
Symmetric encryption and decryption	AES	User	AES key	Read
	Triple-DES	User	Triple-DES key	Read
RSA digital signature generation and verification	RSA (PCKS#1v1.5)	User	RSA public/private key	Read
ECDSA digital signature verification	ECDSA	User	ECDSA public key	Read
Message digest	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	User	None	n/a
Message authentication code (MAC)	HMAC (with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256)	User	HMAC key	Read

Service	Algorithms	Role	Keys/CSPs	Access
Key transport	RSA	User	RSA public/private key	Read
Key derivation	KDF (PRF) in TLS v1.0/1.1, TLS v1.2	User	Shared secret (pre-master secret), derived key	Create, Read, Update
Other FIPS-related Services				
Show status	n/a	User	None	n/a
Zeroization	n/a	User	All CSPs	Zeroize
Self-Tests	AES, Triple-DES, SHS, HMAC, RSA, ECDSA	User	None	n/a
Module installation	n/a	Crypto Officer	None	n/a
Module configuration	n/a	Crypto Officer	None	n/a

Table 5 – Services in FIPS mode of operation

Table 6 lists the services only available in non-FIPS mode of operation.

Service	Algorithms/Key sizes	Role	Keys	Access
Symmetric encryption and decryption	AES-CCM, DES, XCBC, Two-key Triple-DES	User	Symmetric keys	Read
Key Generation	ECDSA	User	Public and private keys.	Create
RSA digital signature generation and verification	RSA (PCKS#1v1.5) using keys listed in Table 9.	User	RSA public/private key	Read
ECDSA Digital Signature Generation	ECDSA	User	ECDSA Private key	Read
Message digest	MD5	User	None	n/a
Message authentication code (MAC)	HMAC-MD5 and HMAC with SHS using keys listed in Table 9.	User	HMAC keys	Read
Key establishment	Diffie-Hellman, EC Diffie-Hellman, EC J-PAKE	User	Diffie-Hellman public/private keys EC Diffie-Hellman public/private keys	Create, Read
Key transport	RSA encrypt/decrypt using keys listed in Table 9.	User	RSA public and private keys	Read
Random number generation	DRBG	User	Entropy input string, Internal state	Read, Update

Table 6 - Services in the non-FIPS mode of operation

3.3 Algorithms

The algorithms implemented in the module approved to be used in FIPS mode of operation are tested and validated by the CAVP. The module provides C language generic implementations for all algorithms. The CAVS certificate number for all algorithms of this implementation is C6.

Table 7 shows the cryptographic algorithms that are approved and non-approved but allowed in FIPS mode of operation, including the algorithm name, supported standards, available modes and key sizes, and usage.

Algorithm	Standard	Mode/Method	Key size	Use
AES	[FIPS197] [SP800-38A]	CBC, CTR	128, 192 and 256 bits	Data Encryption and Decryption
ECDSA	[FIPS186-4]	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	P-192, P-224, P-256, P-384, P-521	Signature Verification
HMAC	[FIPS198-1]	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	112 bits or greater	Message Authentication Code
KDF(PRF) in TLS v1.0/1.1 TLS v1.2	[SP800-135]			Key Derivation
RSA	[FIPS186-4]	PKCS#1v1.5 with SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	2048, 3072, and 4096 bits	Digital Signature Generation
		PKCS#1v1.5 with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	1024, 2048, and 3072 bits 4096 bits (vendor affirmed)	Signature Verification
SHS	[FIPS180-4]	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256		Message Digest
Triple-DES	[SP800-67] [SP800-38A]	CBC	192 bits	Data Encryption and Decryption

Table 7 – FIPS-approved cryptographic algorithms

Table 8 shows the cryptographic algorithms that are not approved but allowed in FIPS mode of operation.

Algorithm	Standard	Mode/Method	Key size	Use
RSA (key encapsulation[2])	[PKCS#1]		Modulus size between 2048 and 15360 bits	Key Establishment; allowed by [FIPS140-2_IG] D.9.
MD5	[SP800-52]			Pseudo-random function (PRF) in TLSv1.0 and TLSv1.1

Table 8 – FIPS-allowed cryptographic algorithms

Table 9 lists the cryptographic algorithms implemented in the module that are not allowed in FIPS mode of operation, including the algorithm name and the reason for being forbidden. Using any of these algorithms will implicitly turn the module in Non-FIPS mode of operation.

Algorithm	Reason
DES, AES-XCBC, EC J-PAKE	Non FIPS-Approved algorithms.
Two-key Triple-DES	Not allowed per [SP800-131A].
AES CCM 128-bit key	Not CAVS tested.
MD5	Non FIPS-Approved algorithms, except MD5 when used as the PRF for TLSv1.0 and TLSv1.1, per [SP800-52].
SHA-1	Not allowed for Digital Signature Generation per [SP800-131A].
HMAC with key size less than 112 bits.	Not allowed key size for Message Authentication Code per [SP800-131A].
RSA with key size less than 2048 bits.	Not allowed key size for Digital Signature Generation and Key Encapsulation per [SP800-131A].
RSA with key size less than 1024 bits.	Not allowed key size for Digital Signature Verification per [SP800-131A].
ECDSA key generation and digital signature generation	Entropy rate of the DRBG seed cannot be assessed (CAVS tested with Cert. #C6).
Diffie-Hellman	Entropy rate of the DRBG seed cannot be assessed.

[2] RSA key encapsulation is also known as RSA key wrapping in FIPS 140-2 related standards and the Implementation Guidance.

EC Diffie-Hellman	Entropy rate of DRBG seed cannot be assessed (CAVS tested with Cert. #C6).
DRBG [SP800-90A]; CTR_DRBG with/without DF, with/without PR	Entropy rate of the DRBG seed cannot be assessed (CAVS tested with Cert. #C6).

Table 9 - Non-approved cryptographic algorithms

3.4 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 the ThreadX Cortex-M4/IAR Version 5.8 operating system executing on the hardware specified in Section 1.2.

5.2 Policy

The application that requests cryptographic services is the single user of the cryptographic module. Concurrent operators are explicitly excluded.

6 Cryptographic Key Management

Table 10 summarizes the secret keys and CSPs that are used by the cryptographic services implemented in the module.

Name	Usage	Generation	Entry and Output
AES keys	Encryption and decryption	Not Applicable. Keys are provided by the calling application.	The key is passed into the module via API input parameters in plaintext. No output.
Triple-DES keys	Encryption and decryption		
HMAC keys	MAC generation and verification		
RSA public key	RSA signature verification RSA key encapsulation		
RSA private key	RSA signature generation RSA key encapsulation		
ECDSA public key	ECDSA signature verification		
Shared secret (pre-master secret)	Establishment of encrypted session	Provided by the calling application.	Passed into the module via API input parameters in plaintext. No output.
Derived key	Establishment of encrypted session	SP 800-135 KDF (TLS v1.0/1.1, TLS v1.2)	No entry. Output via API output parameters in plaintext.

Table 10 - Life cycle of keys and other critical security parameters (CSPs)

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

6.1 Random Number Generation

The module implements a Deterministic Random Bit Generator (DRBG) based on [SP800-90A], but it cannot be used in FIPS mode of operation.

The DRBG is used for the Random Number Generation, the ECDSA key generation and the ECDSA signature generation services. These services can only be run in the non-approved mode of operation.

6.2 Key Generation

The module does not provide any key generation services in FIPS mode of operation.

6.3 Key Derivation

The module supports key derivation for the TLS protocol. The module implements the pseudo-random functions (PRF) for TLS1.0/1.1 and TLS1.2.

6.4 Key Establishment

The module provides RSA-based key encapsulation (using public key encryption and private key decryption primitives) for key transport, as allowed by [FIPS140-2_IG] D.9.

Table 5 specifies the key sizes allowed in FIPS mode of operation. According to “Table 2: Comparable strengths” in [SP800-57], the key sizes for RSA provide the following security strength[3]:

- RSA-based key encapsulation provides between 112 and 256 bits of security strength.

Note: the module also provides Diffie-Hellman and EC Diffie-Hellman services for key agreement, but they cannot be used in FIPS mode of operation.

6.5 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. The module does not enter or output keys in plaintext format outside its physical boundary.

6.6 Key/CSP Storage

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

The module does not perform persistent storage of keys. The keys and CSPs are stored as plaintext in the RAM. The only exception is the HMAC key used for integrity test, which is stored in the module and relies on the operating system for protection.

6.7 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 zeroization functions provided in the module's API, and documented in the API documentation. In addition, the calling application is responsible for parameters passed in and out of the module.

The zeroization mechanism replaces the memory occupied by keys and CSPs with “zeroes” and deallocates the memory with the regular memory deallocation operating system call.

[3] See Section 5.6.1 in [SP 800-57] for a definition of “security strength”.

7 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The cryptographic module is a software-only module, consisting of a software library that is integrated into an embedded application running on microcontrollers. Designers of microcontroller applications integrate the cryptographic module and its operational environment (application, operating system and microcontroller) in their own hardware. It is expected that the EMI/EMC requirements have to be met by the whole hardware platform solution.

8 Self Tests

8.1 Power-Up Tests

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

While the module is executing the power-up self-tests, services are not available, and input and output are inhibited. The module is not available to be used by the calling application until the power-up self-tests are completed successfully.

If any power-up test fails, the module enters the error state and returns the NX_CRYPTO_LIBRARY_STATE_POST_FAILED error code. Subsequent calls to cryptographic services offered by the module will fail, thus no further cryptographic operations are possible. If the power-up tests complete successfully, the module will set its status to NX_CRYPTO_LIBRARY_STATE_OPERATIONAL, enter the FIPS mode and will accept cryptographic operation service requests.

8.1.1 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 binary file and computed at build time. If the HMAC values do not match, the test fails and the module enters the error state.

8.1.2 Cryptographic algorithm tests

The module performs self-tests on all FIPS-Approved cryptographic algorithms supported in the approved mode of operation, using Known Answer Tests (KAT). Table 11 details these self-tests.

Algorithm	Test
AES	• KAT AES-CBC with 256-bit key, encryption • KAT AES-CBC with 256-bit key, decryption
Triple-DES	• KAT Triple-DES-CBC with 192-bit key, encryption • KAT Triple-DES-CBC with 192-bit key, decryption
SHS	• KAT SHA-1 • KAT SHA-224 • KAT SHA-256 • KAT SHA-384 • KAT SHA-512 • KAT SHA-512/224 • KAT SHA-512/256
HMAC	• KAT HMAC-SHA-1 • KAT HMAC-SHA-224 • KAT HMAC-SHA-256 • KAT HMAC-SHA-384 • KAT HMAC-SHA-512

Algorithm	Test
	• KAT HMAC-SHA-512/224 • KAT HMAC-SHA-512/256
ECDSA	• PCT in lieu of KAT[4] for ECDSA signature verification with P-256 and SHA-256
RSA	• KAT RSA PKCS#1v1.5 signature generation with 2048-bit key and using SHA-256 • KAT RSA PKCS#1v1.5 signature verification with 2048-bit key and using SHA-256 • KAT RSA public-key encryption with 2048, 3072 and 4096-bit keys • KAT RSA private-key decryption with 2048, 3072 and 4096-bit keys

Table 11 - Self-tests

The module calculates the result and compares it with the known value. If the answer does not match the known answer, the KAT fails and the module enters the Error state.

8.2 On-Demand self-tests

On-Demand self-tests can be invoked by powering-off and reloading the module, thus forcing the module to run the power-up self-tests.

8.3 Conditional Tests

The module does not perform conditional tests.

[4] Pair-consistency test (PCT) is performed using ECDSA key generation, signature generation and verification; all CAVS tested algorithms with certificate #C6.

9 Guidance

9.1 Crypto Officer Guidance

The module is a software-only cryptographic module delivered on a single CD-ROM compatible disk, or as a compressed image, which contains the following directories:

- *inc*: public API headers needed to build applications.
- *lib*: pre-built NetX Crypto 5.11SP1-FIPS library (nx_crypto_5.11SP1-FIPS.a).
- *docs*: supporting documents.

9.1.1 Module installation

In order to use the module, the crypto officer needs to configure the IAR workspace to include the *inc* directory in the header file search path, and include the *lib* directory in the library path.

Files included under the *inc* directory (nx_crypto*.h) cannot be modified.

9.1.2 Module configuration

In order to run the module as a FIPS validated module, the development environment must be configured to meet the following requirements:

- Application and libraries linking the module need to be built with position independent code.
- The POST routine _nx_crypto_method_self_test() must be inserted into the system initialization table

Please see Section “Build Application Using NetX Crypto” in the NetX Crypto User’s Guide for detailed instructions.

9.2 User Guidance

In order to run in FIPS Approved mode of operation, the module must be operated using the FIPS approved services, with their corresponding FIPS approved or FIPS allowed cryptographic algorithms provided in this Security Policy (Section 3.2). In addition, key sizes must comply with [SP800-131A].

9.2.1 API Functions

Services provided by the module shall be requested using the API functions as specified in the User Guide. Directly invoking functions outside of the API specification, or directly modifying global variables and indicators outside of the API specification is prohibited. Doing so implicitly switches the module out of the FIPS-mode, and the module can only achieve the FIPS-mode again by resetting the module, which automatically invokes the Power-On Self-Test and ensures the module is properly operational.

9.2.2 Triple-DES Data Encryption

Data encryption using the same three-key Triple-DES key shall not exceed 2^{16} Triple-DES (64-bit) blocks, in accordance to [SP800-67] and IG A.13 in [FIPS140-2-IG].

9.2.3 Handling FIPS Related Errors

When the module fails any self-test, the module enters the error state. In this state, cryptographic operations are not allowed and output is inhibited. Any invocation to API functions returns the value NX_CRYPTO_INVALID_LIBRARY (0x20001) to the calling application.

Applications can also obtain the status of the module by calling the nx_crypto_module_state_get() function.

10 Mitigation of Other Attacks

There are no mitigations from other attacks.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CAVS	Cryptographic Algorithm Validation System
CBC	Cipher Block Chaining
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DF	Derivation Function
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
FIPS	Federal Information Processing Standards Publication
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
PAA	Processor Algorithm Acceleration
PCT	Pair-wise Consistency Test
POST	Power-on Self-Tests
PR	Prediction Resistance
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard

Appendix B. References

FIPS140-2	**FIPS PUB 140-2 - Security Requirements For Cryptographic Modules** May 2001 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** February 5, 2019 https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips140-2/FIPS1402IG.pdf
FIPS180-4	**Secure Hash Standard (SHS)** March 2012 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS186-4	**Digital Signature Standard (DSS)** July 2013 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS197	**Advanced Encryption Standard** November 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2** November 2016 https://tools.ietf.org/rfc/rfc8017.txt
SP800-38A	**NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf
SP800-52	**NIST Special Publication 800-52 Revision 1 - Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations** April 2014 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r1.pdf
SP800-56A	**NIST Special Publication 800-56A - Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)** March, 2007 http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf

SP800-67	**NIST Special Publication 800-67 Revision 2 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher** November 2017 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf
SP800-90A	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 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 Lengths** November 2015 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** December 2011 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf