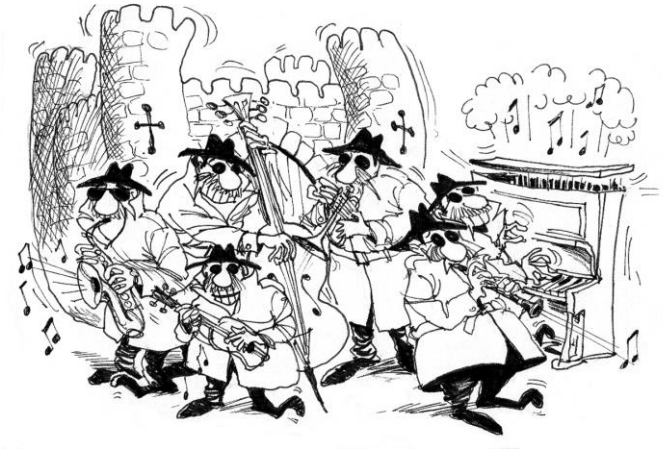
Legion of the Bouncy Castle Inc.

BC-FNA (Bouncy Castle FIPS .NET API)

Non-Proprietary FIPS 140-2 Cryptographic Module Security Policy

Version: 1.0.2 Date: 11/14/22



Legion of the Bouncy Castle Inc.
(ABN 84 166 338 567)
https://www.bouncycastle.org

Table of Contents

List of Tables

List of Figures

1 Introduction

This document defines the Security Policy for the Legion of the Bouncy Castle Inc. FIPS .NET (BC-FNA) Module, hereafter denoted the Module. The Module is a cryptographic library. The Module meets FIPS 140-2 overall Level 1 requirements. The SW version is 1.0.2.

The cryptographic module was tested on the following operational environment on the general-purpose computer (GPC) platforms detailed in Table 1.

Table 1 – Cryptographic Module Tested Environments

Operational Environments			
Hardware Platform	**Operating System**	**.NET Framework Version**	**CLR Version**
Dell XPS 15 7590 with Intel Core i7 9750H	Microsoft Windows 10 Professional (64 bit)	.NET 4.5.2	CLR 4

As per FIPS 140-2 Implementation Guidance G.5, the cryptographic module will remain compliant with the FIPS 140-2 validation when operating on any general purpose computer

(GPC) provided that the GPC uses the specified single-user platform, or another compatible single-user platform such as one of the .NET Runtime Environments listed on any of the following:

Windows 7 with .NET 4.6.1
Windows 8 with .NET 4.6.1
Windows 8.1 with .NET 4.6.1
Windows Server 2012 R2 with .NET 4.6.1
Windows Vista with .NET 4.5.2
Windows 8 with .NET 4.5.2
Windows Server 2012 R2 with .NET 4.5.2
Windows Server 2008 R2 SP1 with .NET 4.5.2

Compliance is maintained for other versions of the respective operational environments where the Module binary is unchanged. No claim can be made as to the correct operation of the Module or the security strengths of the generated keys if any source code is changed and the module binary is reconstructed.

For the avoidance of doubt, it is hereby stated that the CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

The Module is intended for use by US Federal agencies and other markets that require a FIPS 140-2 validated Cryptographic Library. The Module is a software-only embodiment; the cryptographic boundary is the Windows Dynamic Link Library (DLL) file, *bc-fips-1.0.2.dll*.

The FIPS 140-2 security levels for the Module are given in Table 2 as follows:

Table 2 – Security Level of Security Requirements

Security Requirement	Security 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
EMI/EMC	1

Security Requirement	Security Level
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	1

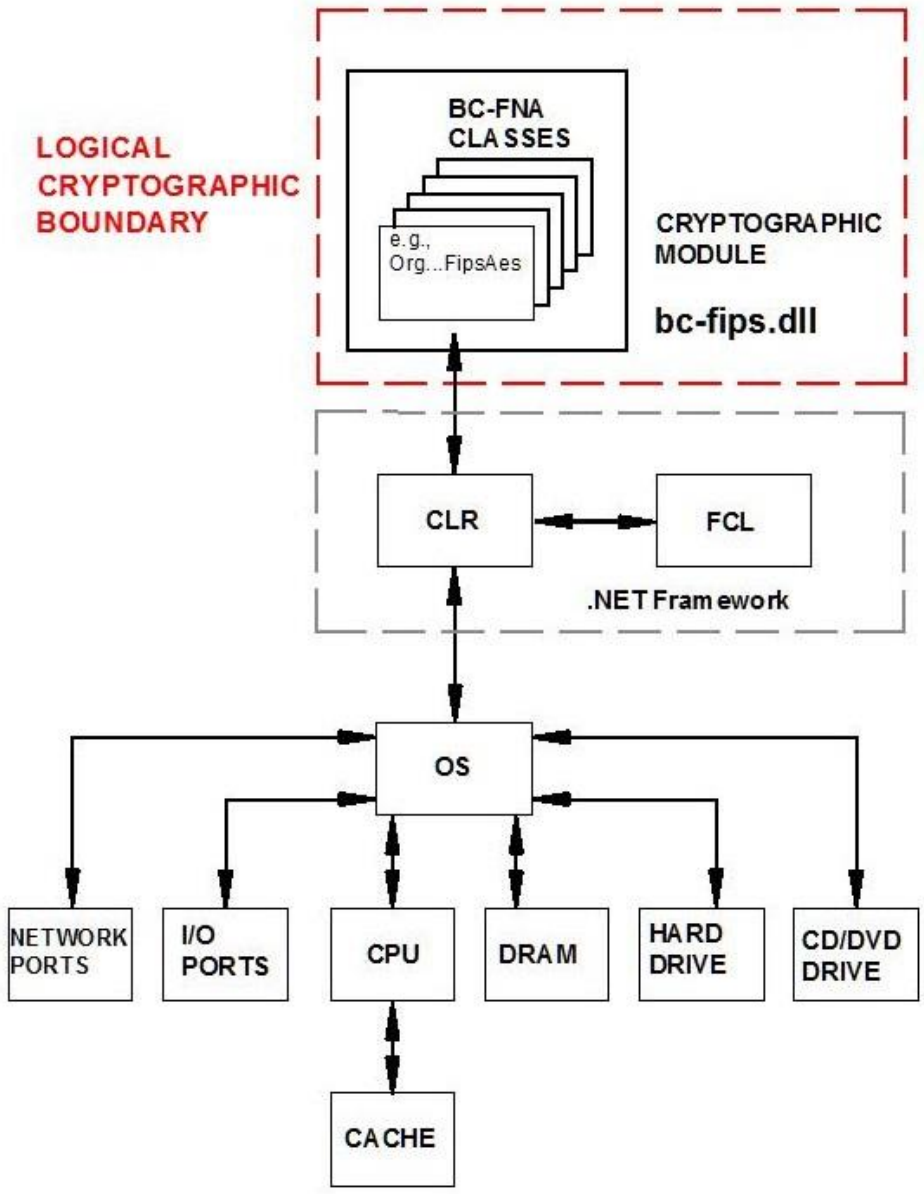
1.1

1.1.1 Logical Cryptographic Boundary

The executable for the BC-FNA Module is*: bc-fips-1.0.2.dll*. This module is the only software component within the Logical Cryptographic Boundary and the only software component that carries out cryptographic functions covered by FIPS 140-2. Figure 1 shows the logical relationship of the cryptographic module to the other software and hardware components of the computer. The BC classes are executed on the .NET Framework Common Language Runtime (CLR) using the classes of the Framework Class Library (FCL) . The CLR is the interface to the computer's Operating System (OS) that is the interface to the various physical components of the computer.

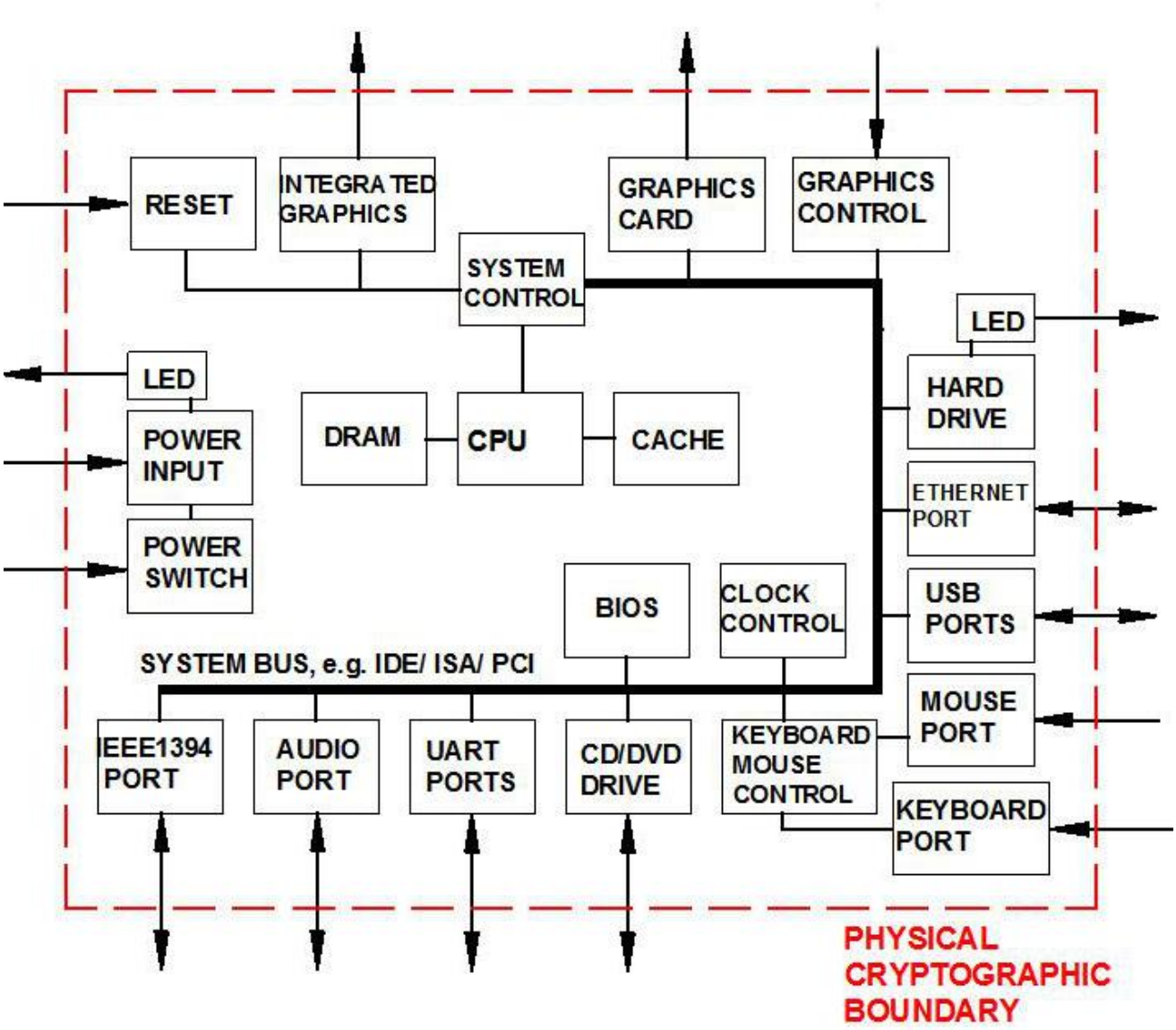
The physical components of the computer are discussed further in Section 6. Abbreviations introduced in Figure 1 that describe physical components are: Central Processing Unit (CPU), Dynamic Random Access Memory (DRAM) and Input Output (I/O).

Figure 1 – Block Diagram of the Software for the BC-FNA Module.



1.1.2 Physical Boundary

The BC-FNA Module runs on a General Purpose Computer (GPC). The Physical Cryptographic Boundary for the module is the case of that computer. Figure 2 shows a block diagram of the physical components of a typical GPC and the ports or interfaces across the Physical Cryptographic Boundary. All the physical components are standard electronic components; there are not any custom integrated circuits or components dedicated to FIPS 140-2 related functions.



Abbreviations introduced in Figure 2 are: Basic I/O System (BIOS), Integrated Device Electronics (IDE), Institute of Electrical and Electronic Engineers (IEEE), Instruction Set Architecture (ISA), Peripheral Component Interconnect (PCI), Universal Asynchronous Receiver/Transmitter (UART) and Universal Serial Bus (USB). Input or output ports are designated by arrows with single heads, while I/O ports are indicated by bidirectional arrows.

Figure 2 – Block Diagram of the Physical Components of a typical GPC.

For FIPS 140-2 purposes, the BC-FNA Module is defined as a “multi-chip standalone module”, therefore, the module’s physical ports or interfaces are defined as those for the hardware of the GPC. These physical ports are separated into the logical interfaces defined by FIPS 140-2, as shown in Table 3.

The BC-FNA module is a software module only, and, therefore, control of the physical ports is outside of the module's scope. The module does provide a set of logical interfaces which are mapped to the following FIPS 140-2 defined logical interfaces: data input, data output, control input, status output, and power. When the module performs self-tests or is in an error state, the module prevents all output on the logical data output interface. Activities in the module are single-threaded, and when in an error state, the module does not return any output data, only an error value.

The mapping of the FIPS 140-2 logical interfaces to the module is described in table 3.

Table 3 – FIPS 140-2 Logical Interfaces

Interface	Module Equivalent
Data Input	API input parameters – plaintext and/or ciphertext data.
Data Output	API output parameters and return values – plaintext and/or ciphertext data.
Control Input	API method calls – method calls, or input parameters, that specify commands and/or control data used to control the operation of the module.
Status Output	API output parameters and return/error codes that provide status information used to indicate the state of the module.
Power	Start up/Shutdown of a process containing the module.

1.2 Modes of Operation

There will be two modes of operation: Approved and Non-approved. The module will be in FIPS-approved mode when the appropriate factory is called. To verify that a module is in the Approved Mode of operation, the user can call a FIPS status method (*CryptoServicesRegistrar.IsInApprovedOnlyMode()*). If the module is configured to allow approved and non-approved mode operation, a call to *CryptoServicesRegistrar.setApprovedMode(true*) will switch the current thread of user control into approved mode.

In FIPS-approved mode, the module will not provide non-approved algorithms, therefore, exceptions will be called if the user tries to access non-approved algorithms in the Approved Mode.

2 Cryptographic Functionality

The Module implements the FIPS Approved and Non-Approved but Allowed cryptographic functions listed in Table 4 and Table 5, below.

Table 4 – Approved and CAVP Validated Cryptographic Functions

Algorithm	Description	Cert #
AES	[FIPS 197, SP 800-38A, and Addendum to SP 800-38A (Oct 2010)] Functions: Encryption, Decryption Modes: ECB, CBC, OFB, CFB8, CFB128, CTR, CBC-CS1, CBC-CS2, CBC-CS3 Key sizes: 128, 192, 256 bits	A1905
CCM	[SP 800-38C] Functions: Authenticated Encryption and Decryption Key sizes: 128, 192, 256 bits	A1905
CKG	SP 800-133	Vendor Affirmed
CMAC	[SP 800-38B] Functions: Authenticated Encryption and Decryption Key sizes: AES with 128, 192, 256 bits and Triple-DES with 2-key[1],[2], 3-key	A1905
DRBG	[SP 800-90A] Functions: Hash DRBG, HMAC DRBG, CTR DRBG Security Strengths: 112, 128, 192, and 256 bits	A1905
DSA[3]	[FIPS 186-4] Functions: PQG Generation, PQG Verification, Key Pair Generation, Signature Generation, Signature Verification Key sizes: 1024, 2048, 3072 bits (1024 only for SigVer)	A1905

[1] 2^{16} block limit enforced by module.
[2] In approved mode of operation, the use of 2-key Triple-DES to generate MACs for anything other than verification purposes is non-compliant.
[3] DSA signature generation with SHA-1 is only for use with protocols.

Algorithm	Description	Cert #
ECDSA	[FIPS 186-4] Functions: Signature Generation Component, Public Key Generation, Signature Generation, Signature Verification, Public Key Validation Curves/Key sizes: P-192[4], P-224, P-256, P-384, P-521, K-163[4], K-233, K-283, K-409, K-571, B-163[4], B-233, B-283, B-409, B-571	A1905
FF1 Format Preserving Encryption	[SP 800-38G] Functions: Encryption, Decryption Key sizes: 128, 192, 256 bits	A1905
GCM/GMAC[5]	[SP 800-38D] Functions: Authenticated Encryption and Decryption Key sizes: 128, 192, 256 bits	A1905
HMAC	[FIPS 198-1] Functions: Generation, Authentication SHA sizes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	A1905
KAS[6]	[SP 800-56A-rev3] Parameter sets/Curves/Key sizes: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192, P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571	A1905
KAS[7]	SP 800-56A rev3 KAS-SSC (Cert. #A1905) with SP 800-135 TLS v1.0/1.1 KDF, TLS 1.2 KDF or X9.63 KDF (CVL Cert. #A1905). Compliant to IG D.8 X1 Option 2, testing the shared secret and separately testing the key derivation function.	Uses A1905

[4] In approved mode of operation, the use of this curve for anything other than signature verification is non-compliant.
[5] Internal IV generation with an approved DRBG as outlined in Section 8.2.2 of NIST SP 800-38D, see Section 8.3 concerning external IVs. IV generation is compliant with IG A.5.
[6] KAS-ECC (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength; anything less than 112 bits of encryption strength is non-compliant). KAS-FFC (key agreement; key establishment methodology provides between 112 and 200 bits of encryption strength; anything less than 112 bits of encryption strength is non-compliant). Keys are not directly established into the module using KAS-ECC and KAS-FFC.
[7] Keys are not directly established into the module using KAS-ECC and KAS-FFC.

Algorithm	Description	Cert #
KAS[8]	SP 800-56A rev3 KAS-SSC (Cert. #A1905) with SP 800-56C, Rev 2 One Step KDA or HKDF KDA (Cert. #A1905) Compliant to IG D.8 X1 Option 2, testing the shared secret and separately testing the key derivation function.	Uses A1905
KAS-SSC	[All of SP 800-56A-rev3 EXCEPT KDF] Parameter sets/Key sizes: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571 [SP 800-56A-rev3 Section 5.7.1.2 ECC CDH Primitive] Parameter sets/Key sizes: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571	A1905
KDF, Existing Application-Specific[9] (CVL)	[SP 800-135] Functions: TLS v1.0/1.1 KDF, TLS 1.2 KDF, X9.63 KDF.	A1905 (CVL)
KDA, One Step	[SP 800-56C, Rev 2] PRFs: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512, HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512, HMAC SHA-512/224, HMAC SHA-512/256, HMAC SHA3-224, HMAC SHA3-256, HMAC SHA3-384, HMAC SHA3-512, KMAC-128, KMAC-256	A1905
KDA, HKDF	[SP 800-56C, Rev 2] PRFs: HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512, HMAC SHA-512/224, HMAC SHA-512/256, HMAC SHA3-224, HMAC SHA3-256, HMAC SHA3-384, HMAC SHA3-512	A1905
KDF, Password-Based	[SP 800-132] Options: PBKDF with Option 1a only. Functions: HMAC-based KDF using SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	A1905

[8] Keys are not directly established into the module using KAS-ECC and KAS-FFC.
[9] These protocols have not been reviewed or tested by the CAVP and CMVP.

Algorithm	Description	Cert #
KTS[10]	AES Cert. #A1905; key establishment methodology provides between 128 and 256 bits of encryption strength	Uses A1905
KTS[11]	Triple-DES Cert. #A1905; key establishment methodology provides 112 bits of encryption strength	Uses A1905
KTS-RSA	Key establishment methodology provides 112 or 128 bits of encryption strength	Uses A1905
Key Wrapping Using Block Ciphers[12]	[SP 800-38F] Modes: AES KW, KWP Key sizes: 128, 192, 256 bits (provides between 128 and 256 bits of strength)	A1905
	[SP 800-38F] Mode: Triple-DES TKW Key size: 3-key (provides 112 bits of strength)	A1905
RSA[13]	[FIPS 186-4, FIPS 186-2, ANSI X9.31-1998 and PKCS #1 v2.1 (PSS and PKCS1.5), SP 800-56B-rev2] Functions: Key Pair Generation, Signature Generation, Signature Verification, Key Transport, RSADP Component Test Key sizes: 2048, 3072, 4096 (1024, 1536 only for SigVer)	A1905
SHA	[FIPS 180-4], Functions: Digital Signature Generation, Digital Signature Verification, non-Digital Signature Applications SHA sizes: SHA-1 (only for sigver), SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	A1905
SHA-3, SHAKE	[FIPS 202] SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256	A1905
SHA-3 Derived Functions	[SP 800-185] Functions: cSHAKE-128, KMAC-128, TupleHash-128, ParallelHash-128, cSHAKE-256, KMAC-256, TupleHash-256, ParallelHash-256	A1905

[10] Keys are not established directly into the module using key unwrapping.
[11] Keys are not established directly into the module using key unwrapping.
[12] Keys are not established directly into the module using key unwrapping.
[13] Keys are not established directly into the module using key transport.

Algorithm	Description	Cert #
Triple-DES (Triple-DES)	[SP 800-67] Functions: Encryption, Decryption Modes: TECB, TCBC, TCFB64, TCFB8, TOFB, CTR Key sizes: 2-key[14], 3-key	A1905

Table 5 – Non-Approved but Allowed Cryptographic Functions

Algorithm	Description
Non-SP 800-56B compliant RSA Key Transport[15]	[IG D.9] RSA May be used by a calling application as part of a key encapsulation scheme. Key sizes: >= 2048 bits Keys are not established into the module using RSA. Key wrapping; key establishment methodology provides 112 or 128 bits of encryption strength.
MD5 within TLS	[IG D.2]

Table 6 – Non-Approved Cryptographic Functions for use in non-FIPS mode only.

AES (non-compliant)	FF3-1
ARC4 (RC4)	NewHope
Camellia	OpenSSL PBKDF
ChaCha	PKCS#12 PBKDF
DSA (non-compliant)	Poly1305
ECDSA (non-compliant)	RSA (non-compliant)
EdDSA	SEED
ElGamal	Serpent
	SPHINCS-256

2.1 Critical Security Parameters

All CSPs used by the Module are described in this section in Table 7. All usage of these CSPs by the Module (including all CSP lifecycle states) is described in the services detailed in Section 3.2.

[14] 2^{16} block limit is enforced by the module, encryption disabled for 2-key.
[15] RSA (key wrapping; key establishment methodology provides at least 112 bits of encryption strength; non-compliant less than 112 bits of encryption strength).

Table 7 – Critical Security Parameters (CSPs)

CSP	Description / Usage
AES Encryption Key	[FIPS-197, SP 800-38C, SP 800-38D, SP 800-38G, Addendum to SP 800-38A (Oct 2010)] AES (128/192/256) encrypt key[16]
AES Decryption Key	[FIPS-197, SP 800-38C, SP 800-38D, SP 800-38G Addendum to SP 800-38A (Oct 2010)] AES (128/192/256) decrypt key
AES Authentication Key	[FIPS-197] AES (128/192/256) CMAC/GMAC key
AES Wrapping Key	[SP 800-38F] AES (128/192/256) key wrapping key
DH Agreement key	[SP 800-56A-rev3] Diffie-Hellman (224 - 512 bits) private key agreement key
DRBG(CTR AES)	V (128 bits) and AES key (128/192/256), entropy input (length dependent on security strength)
DRBG(CTR Triple-DES)	V (64 bits) and Triple-DES key (192 bits), entropy input (length dependent on security strength)
DRBG(Hash)	V (440/888 bits) and C (440/888 bits), entropy input (length dependent on security strength)
DRBG(HMAC)	V (160/224/256/384/512 bits) and Key (160/224/256/384/512 bits), entropy input (length dependent on security strength)
DSA Signing Key	[FIPS 186-4] DSA (2048/3072 bits) signature generation key
EC Agreement Key	[SP 800-56A-rev3] EC (All NIST defined B, K, and P curves) private key agreement key
EC Signing Key	[FIPS 186-4] ECDSA (All NIST defined B, K, and P curves >= 224 bits) signature generation key.
HMAC Authentication Key	[FIPS 198-1] Keyed-Hash key (SHA-1, SHA-2). Key size determined by security strength required (>= 112 bits).
PBKDF Secret	[SP 800-132] Secret value used in construction of Keyed-Hash key for the specified PRF.
RSA Signing Key	[FIPS 186-4] RSA (>= 2048 bits) signature generation key
RSA Key Transport Key	[SP 800-56B-rev2] RSA (>= 2048 bits) key transport (decryption) key
TLS KDF Secret Value	[SP 800-135] Secret value used in construction of Keyed-Hash key for the specified TLS PRF.

[16] The AES-GCM key is generated randomly per IG A.5, and the Initialization Vector (IV) is also generated randomly and at least 96 bits. In the event of power loss the AES-GCM key will be lost and the consuming application must ensure that new AES-GCM keys for encryption or decryption are re-distributed.

CSP	Description / Usage
Triple-DES Authentication Key	[SP 800-67] Triple-DES (128/192 bits) CMAC key
Triple-DES Encryption Key	[SP 800-67] Triple-DES (192 bits) encryption key
Triple-DES Decryption Key	[SP 800-67] Triple-DES (128/192 bits) decryption key
Triple-DES Wrapping Key	[SP 800-38F] Triple-DES (192 bits) key wrapping/unwrapping key, (128 unwrapping only).
X9.63 KDF Secret Value	[SP 800-135] Secret value used in construction of input for the specified X9.63 PRF.
SP 800-56C-rev2 One-Step Derivation Function	[SP 800-56C-rev2] Secret value used in construction of key for underlying PRF.
SP 800-56C-rev2 Hash Derivation Function (HKDF)	[SP 800-56C-rev2] Secret value used in construction of key for underlying PRF.

2.2 Public Keys

Table 8 – Public Keys

CSP	Description / Usage
DH Agreement Key	[SP 800-56A-rev3] Diffie-Hellman (>= 2048) public key agreement key (All SP 800-56A-rev3 parameter sets)
DSA Verification Key	[FIPS 186-4] DSA (1024/2048/3072) signature verification key
EC Agreement Key	[SP 800-56-rev3] EC (All NIST defined B, K, and P curves) public key agreement key
EC Verification Key	[FIPS 186-4] ECDSA (All NIST defined B, K, and P curves) signature verification key
RSA Key Transport Key	[SP 800-56B-rev2] RSA (>= 2048) key transport (encryption) key.
RSA Verification Key	[FIPS 186-4] RSA (1024, 1536, >= 2048) signature verification key

3 Roles, Authentication and Services

3.1 Assumption of Roles

The module supports two distinct operator roles, User and Cryptographic Officer (CO). The cryptographic module implicitly maps the two roles to the services. A user is considered the owner of the thread that instantiates the Module and, therefore, only one concurrent user is allowed.

Table lists all operator roles supported by the Module. The Module does not support a maintenance role and/or bypass capability. The Module leverages the CLR to allow multiple threads (concurrent operations), and the operating system and CLR manage separate memory for each thread. In addition, there is high level thread management implemented by the module. The module does not support authentication.

Table 9 – Roles Description

Role ID	Role Description	Authentication Type
CO	Cryptographic Officer – Initialize the module.	N/A – Authentication not required for Level 1
User	User – use of the complete API.	N/A – Authentication not required for Level 1

3.2 Services

All services implemented by the module are listed in Table 1 and Table 12 below. Each service description also describes all usage of CSPs by the service.

Table 1 lists the services. The second column provides a description of each service and availability to the Cryptographic Officer and User, in columns 3 and 4, respectively. All services available to the User are also available to the Cryptographic Officer. Only one role may be active at a time and the module does not allow concurrent operators, although an operator may perform more than one task concurrently.

Authentication of the Cryptographic Officer and/or User is not supported by the module but is a task performed by the host environment.

Table 10 – Services

Service	Description	CO	U
Initialize Module and Run Self-Tests on Demand	The CLR will call the static constructor for self-tests on module initialization.	X	
Show Status	A user can call Crypto*Status.IsReady()* at any time to determine if the module is ready. *IsInApprovedOnlyMode()* can be called to determine the FIPS mode of operation.		X
Zeroize / Power-off	The module uses the CLR garbage collector on thread termination when objects are reclaimed.		X
Data Encryption	Used to encrypt data.		X
Data Decryption	Used to decrypt data.		X
MAC Calculation	Used to calculate data integrity codes with CMAC.		X
Signature Generation	Used to generate signatures (DSA, ECDSA, RSA).		X
Signature Verification	Used to verify digital signatures.		X
DRBG (SP800-90A) output	Used for random number and IV generation.		X
Key Generation – Based on DRBG (SP800-90A)	Used for key generation.		X
Message Hashing	Used to generate a SHA-1, SHA-2, or SHA-3 message digest, SHAKE output.		X
Keyed Message Hashing	Used to calculate data integrity codes with HMAC.		X
TLS Key Derivation Function	(secret input) (outputs secret) Used to calculate a value suitable to be used for a master secret in TLS from a pre-master secret and additional input.		X
X9.63 Derivation Function	(secret input) (outputs secret) Used to calculate a value suitable to be used for a secret key from a input secret and additional input.		X
SP 800-56C-rev2 One-Step Derivation Function	(secret input) (outputs secret) Used to calculate a value suitable to be used for a secret key from a input secret and additional input.		X
SP 800-56C-rev2 Hash Derivation Function (HKDF)	(secret input) (outputs secret) Used to calculate a value suitable to be used for a secret key from a input secret and additional input.		X

Service	Description	CO	U
Password-Based Key Derivation Function	(secret input) (outputs secret) Used to generate a key using an encoding of a password and an additional function such as a message hash.		X
Key Agreement Schemes	Used to calculate key agreement values (SP 800-56A-rev3).		X
Key Wrapping	Used to encrypt a key value. (RSA, AES, Triple-DES)		X
Key Unwrapping	Used to decrypt a key value. (RSA, AES, Triple-DES)		X
NDRNG Callback	Gathers entropy in a passive manner from a user-provided function		X
Utility	Miscellaneous utility functions, does not access CSPs		X

Services in the module are accessed via the public APIs of the DLL. The ability of a thread to invoke unapproved services depends on whether it has been registered with the module as approved mode only. In approved only mode no unapproved services are accessible.

Table 12 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as:

- G = Generate: The module generates the CSP.
- R = Read: The module reads the CSP. The read access is typically performed before the uses the CSP.
- E = Execute: The module executes using the CSP.
- W = Write: The module writes the CSP. The write access is typically performed after a CSP is imported into the module, when the module generates a CSP, or when the module overwrites an existing CSP.
- Z = Zeroize: The module zeroizes the CSP.

Table 11 - CSP Access Rights within Services

Service	CSPs																							
	AES Encryption Key	AES Decryption Key	AES Authentication Key	AES Wrapping Key	DH Agreement Key	DRBG (CTR AES)	DRBG (CTR Triple-DES)	DRBG (Hash)	DRBG (HMAC)	DSA Signing Key	EC Agreement Key	EC Signing Key	HMAC Authentication Key	PBKDF Secret	RSA Signing Key	RSA Key Transport Key	SP 800-56C Concat. DF Secret	SP 800-56C HKDF Secret	TLS KDF Secret	Triple-DES Authentication Key	Triple-DES Encryption Key	Triple-DES Decryption Key	Triple-DES Wrapping Key	X9.63 KDF Secret
Initialize Module and Run Self-Tests on Demand																								
Show Status																								
Zeroize / Power-off	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
Data Encryption	R																				R			
Data Decryption		R																				R		
MAC Calculation			R										R										R	
Signature Generation										R		R			R									
Signature Verification										R		R			R									
DRBG (SP800-90A) output	G	G	G	G	G	G R	G R	G R	G R	G	G	G	G		G	G				G	G	G	G	
Key Generation – Based on DRBG (SP800-90A)						R	R	R	R															
Message Hashing																								
Keyed Message Hashing													R											
TLS Key Derivation Function																			R					
X9.63 Derivation Function					G						G				G									R
SP 800-56C-rev2 One-Step Derivation Function (KDM)					G						G				G		R							
SP 800-56C-rev2 Hash derivation Function (HKDF)					G						G				G			R						
PBKDF													GR	R										
Key Agreement Schemes	G	G	G	G	R						R		G			R				G	G	G	G	
Key Wrapping/Transport (RSA, AES, Triple-DES)				R									R			R							R	
Key Unwrapping (RSA, AES, Triple-DES)				R									R			R							R	
NDRNG Callback						G	G	G	G															
Utility																								

Note: keys are not established directly into the module using derivation functions or unwrapping schemes.

4 Self-tests

Each time the Module is powered up, it tests that the cryptographic algorithms still operate correctly and that sensitive data have not been damaged. Power-up self–tests are available on demand by power cycling the Module.

On power-up or reset, the Module performs the self-tests that are described in Table 12 below. All KATs must be completed successfully prior to any other use of cryptography by the Module. If one of the KATs fails, the Module enters the Self-Test Failure error state. The module will output a detailed error message when Crypto*Status.IsReady()* is called. The error state can only be cleared by reloading the module and calling Crypto*Status.IsReady()* again to confirm successful completion of the KATs.

Table 12 – Power Up Self-tests

Test Target	Description
Software Integrity	HMAC-SHA512
AES	KATs: Encryption, Decryption Modes: ECB Key sizes: 128 bits
AES-CMAC	KATs: Generation, Verification Key sizes: AES with 128 bits
CCM	KATs: Generation, Verification Key sizes: 128 bits
DH Agreement	KATs: Agreement Test Parameter Sets/Key sizes: ffdhe2048
FFC KAS	KATs: Per IG D.8 Scenario X1 – Primitive “Z” Computation Parameter Sets/Key sizes: ffdhe2048
ECC KAS	KATs: Per IG D.8 Scenario X1 – Primitive “Z” Computation EC Curve: P-256 (Fp), B-233 (F2m)
DRBG	KATs: HASH_DRBG, HMAC_DRBG, CTR_DRBG Security Strengths: 256 bits
DSA	KAT: Signature Generation, Signature Verification Key sizes: 2048 bits
ECDSA	KAT: Signature Generation, Signature Verification Curves/Key sizes: P-256 (Fp), B-233 (F2m)
GCM/GMAC	KATs: Generation, Verification Key sizes: 128 bits
HKDF (KDA)	KATs: Key derivation PRFs: HMAC-SHA-256
HMAC	KATs: Generation, Verification

Test Target	Description
	SHA sizes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512
SP 800-56C-rev2 Key-Derivation Methods (KDA)	KATs: Key derivation Modes: One-Step PRFs: HMAC-SHA-256, SHA-256, KMAC256
RSA	KATs: Signature Generation, Signature Verification Key sizes: 2048 bits
SHS	KATs: Output Verification SHA sizes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512
Triple-DES	KATs: Encryption, Decryption Modes: TECB, Key sizes: 3-Key
Triple-DES-CMAC	KATs: Generation, Verification Key sizes: 3-Key
Extendable-Output functions (XOF)	KATs: Output Verification XOFs:SHAKE128, SHAKE256
KDF, Existing Application-Specific (CVL)	MD5 KAT performed to verify operation of MD5 digest used in TLS 1.0 KDF. TLS 1.0 SHA-1 KDF KAT performed to verify TLS 1.0 KDF, TLS 1.1/1.2 KDF SHA-256-HMAC KAT performed to verify TLS 1.1/1.2 KDF. X9.63 SHA-256 KDF KAT performed to verify X9.63 KDF
KDF, Password-Based (PBKDF)	KATs: Master key derivation PRFs: HMAC-SHA-256
Key Transport Using RSA	KATs: SP 800-56B-rev2 specific KATs per IG D.4 Key sizes: 2048 bits

Table 13 – Conditional Self-tests

Test Target	Description
NDRNG	NDRNG Continuous Test performed when a random value is requested from the entropy source.
DH	DH Pairwise Consistency Test performed on every DH key pair generation.
DRBG	DRBG Continuous Test performed when a random value is requested from the DRBG (on approved and unapproved DRBG).
DSA	DSA Pairwise Consistency Test performed on every DSA key pair generation.
ECDSA	ECDSA Pairwise Consistency Test performed on every EC key pair generation.

Test Target	Description
RSA	RSA Pairwise Consistency Test performed on every RSA key pair generation.
DRBG Health Checks	Performed conditionally on approved and unapproved DRBG, per SP 800-90A Section 11.3.
SP 800-56A Assurances	Performed conditionally per SP 800-56A-rev3 Sections 5.5.2 and/or 5.6.2. Required per IG 9.6 and IG D.8.

5 Physical Security Policy

The module is a software-only module and does not have physical security mechanisms.

6 Operational Environment

The Module is as a modifiable operational environment under the FIPS 140-2 definitions.

The Module runs on a GPC running one of the operating systems specified in the approved operational environment list. Each approved operating system manages processes and threads in a logically separated manner. The Module's user is considered the calling application that instantiates the Module within the process space of the CLR. When the Module is not otherwise configured, it will start by default in the non-FIPS-approved mode.

6.1 Use of External RNG

The module makes use of FipsSecureRandom to seed the DRBG. FipsSecureRandom has three builder methods used to control how entropy is provided. The method FromDefaultEntropy() shall not be used in the Approved mode of operation. In the Approved mode either FromEntropySource(SecureRandom) or FromEntropySource(IEntropySourceProvider) can be used. In either case the user shall ensure an Approved entropy source is provided and will block, or fail, if it is unable to provide the amount of entropy requested.

The module's FipsSecureRandom () function will request entropy as appropriate to the security strength and seeding configuration for the DRBG that is using it. In approved mode the minimum amount of entropy that would be requested is 112 bits with a larger minimum being set if the security strength of the operation requires it.

The module will wait until the FipsSecureRandom() returns the requested amount of entropy before seeding the DRBG.

7 Mitigation of Other Attacks Policy

The Module implements basic protections to mitigate against timing based attacks against its internal implementations. There are two counter-measures used.

The first is Constant Time Comparisons, which protect the digest and integrity algorithms by strictly avoiding “fast fail” comparison of MACs, signatures, and digests so the time taken to compare a MAC, signature, or digest is constant regardless of whether the comparison passes or fails.

The second is made up of Numeric Blinding and decryption/signing verification which both protect the RSA algorithm.

Numeric Blinding prevents timing attacks against RSA decryption and signing by providing a random input into the operation which is subsequently eliminated when the result is produced. The random input makes it impossible for a third party observing the private key operation to attempt a timing attack on the operation as they do not have knowledge of the random input and consequently the time taken for the operation tells them nothing about the private value of the RSA key.

Decryption/signing verification is carried out by calculating a primitive encryption or signature verification operation after a corresponding decryption or signing operation before the result of the decryption or signing operation is returned. The purpose of this is to protect against Lenstra's CRT attack by verifying the correctness the private key calculations involved. Lenstra's CRT attack takes advantage of undetected errors in the use of RSA private keys with CRT values and, if exploitable, can be used to discover the private value of the RSA key.

8 Security Rules and Guidance

8.1 Basic Enforcement

The module design corresponds to the Module security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Level 1 module.

1. The module shall provide two distinct operator roles: User and Cryptographic Officer.
2. The module does not provide authentication.
3. The operator shall be capable of commanding the module to perform the power up self-tests by cycling power or resetting the module.
4. Power up self-tests do not require any operator action.
5. Data output shall be inhibited during key generation, self-tests, zeroization, and error states.
6. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.

7. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
8. The module does not support concurrent operators.
9. The module does not have any external input/output devices used for entry/output of data.
10. The module does not enter or output plaintext CSPs from the module's physical boundary.
11. The module does not output intermediate key values.

8.2 Basic Guidance

Functionality in the module is provided via distinct classes that provide access to the FIPS approved and non-FIPS approved services provided by the module.

When the module is being used in FIPS approved-only mode, classes providing implementations of algorithms which are not FIPS approved, or allowed, are explicitly disabled.

8.3 Enforcement and Guidance for GCM IVs

The module supports two methods of AES GCM IV generation. The first is when AES GCM is used as part of TLS 1.2 cipher suites conformant to IG A.5 Scenario 1, RFC 5288 and SP 800-52 Section 3.3.1. The construction of the 64-bit nonce_explicit part of the IV is generated using the FipsNonceGenerator, where a monotonically increasing counter is used as the basis for the nonce. Rollover of the counter in the FipsNonceGenerator will result in an IllegalStateException indicating the FipsNonceGenerator is exhausted and, as per IG A.5, where used for TLS, rollover will terminate any TLS session in process using the current key and the exception can only be recovered from by using a new handshake and creating a new FipsNonceGenerator.

The GCM IV can also be generated randomly, per IG A.5, Scenario 2. The IV is constructed to be at least 96-bits. The module enforces the use of an approved DRBG in line with Section 8.2.2 of SP 800-38D.

Per IG A.5, in the event module power is lost and restored the consuming application must ensure that any of its AES-GCM keys used for encryption or decryption are re-distributed.

8.4 Enforcement and Guidance for use of the Approved PBKDF

In line with the requirements for SP 800-132, keys generated using the approved PBKDF must only be used for storage applications. Any other use of the approved PBKDF is non-compliant.

In approved mode the module enforces that any password used must encode to at least 14 bytes (112 bits) and that the salt is at least 16 bytes (128 bits) long. The iteration count associated with the PBKDF should be as large as practical.

As the module is a general-purpose software module, it is not possible to anticipate all the levels of use for the PBKDF, however a user of the module should also note that a password should at least contain enough entropy to be unguessable and also contain enough entropy to reflect the security strength required for the key being generated. In the event a password encoding is simply based on ASCII a 14 byte password is unlikely to contain sufficient entropy for most purposes. Users are referred to Appendix A, “Security Considerations” of SP 800-132 for further information on password, salt, and iteration count selection.

8.5 8.5 Rules for setting the N and the S String in cSHAKE

The cSHAKE algorithm offers to input string for customizing the output of the cSHAKE function, the Function-Name input (N) and the Customization String (S).

The Function-Name input (N) is reserved for values specified by NIST and should only be set to the appropriate NIST specified value. Any other use of N is non-conformant.

The Customization String (S) is available to allow users to customize the cSHAKE function as they wish. The length of S is limited to the available size of a byte array in the CLR running the module.

9 References and Definitions

The following standards are referred to in this Security Policy.

Table 14 – References

Abbreviation	Full Specification Name
ANSI X9.31	*X9.31-1998, Digital Signatures using Reversible Public Key Cryptography for the Financial Services Industry (rDSA), September 9, 1998*
FIPS 140-2	*Security Requirements for Cryptographic modules, May 25 2001*
FIPS 180-4	*Secure Hash Standard (SHS)*
FIPS 186-3	*Digital Signature Standard (DSS)*
FIPS 186-4	*Digital Signature Standard (DSS)*
FIPS 197	*Advanced Encryption Standard*
FIPS 198-1	*The Keyed-Hash Message Authentication Code (HMAC)*
FIPS 202	*SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*
IG	*Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program, January 11, 2016.*
PKCS#1 v2.1	*RSA Cryptography Standard*

Abbreviation	Full Specification Name
PKCS#5	*Password-Based Cryptography Standard*
SP 800-38A	*Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode*
SP 800-38B	*Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*
SP 800-38C	*Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*
SP 800-38D	*Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*
SP 800-38F	*Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*
SP 800-38G	*Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption*
SP 800-56A-rev3	*Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography Revision 2.*
SP 800-56B-rev2	*Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography*
SP 800-56C-rev2	*Recommendation for Key Derivation through Extraction-then-Expansion*
SP 800-67	*Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*
SP 800-89	*Recommendation for Obtaining Assurances for Digital Signature Applications*
SP 800-90A	*Recommendation for Random Number Generation Using Deterministic Random Bit Generators*
SP 800-132	*Recommendation for Password-Based Key Derivation*
SP 800-135	*Recommendation for Existing Application –Specific Key Derivation Functions*

Table 15 – Acronyms and Definitions

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
BC	Bouncy Castle
BC-FNA	Bouncy Castle FIPS .NET API
CBC	Cipher-Block Chaining
CCM	Counter with CBC-MAC
CDH	Computational Diffie-Hellman
CFB	Cipher Feedback Mode
CLR	Common Language Runtime
CMAC	Cipher-based Message Authentication Code
CMVP	Crypto Module Validation Program
CO	Cryptographic Officer
CPU	Central Processing Unit

Acronym	Definition
CS	Ciphertext Stealing
CSP	Critical Security Parameter
CTR	Counter-mode
CVL	Component Validation List
DES	Data Encryption Standard
DLL	Dynamic Link Library
DRAM	Dynamic Random Access Memory
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Authority
EC	Elliptic Curve
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Authority
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCL	Framework Class Library
FIPS	Federal Information Processing Standards
GCM	Galois/Counter Mode
GMAC	Galois Message Authentication Code
GPC	General Purpose Computer
HMAC	key-Hashed Message Authentication Code
IG	See References
IV	Initialization Vector
KAS	Key Agreement Scheme
KAT	Known Answer Test
KDF	Key Derivation Function
KW	Key Wrap
KWP	Key Wrap with Padding
MAC	Message Authentication Code
MD5	Message Digest algorithm MD5
N/A	Non Applicable
NDRNG	Non Deterministic Random Number Generator
OCB	Offset Codebook Mode
OFB	Output Feedback
OS	Operating System
PBKDF	Password-Based Key Derivation Function
PKCS	Public Key Cryptography Standards
PRF	Pseudorandom Function

Acronym	Definition
PQG	Diffie-Hellman Parameters P, Q and G
RC	Rivest Cipher, Ron’s Code
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
TCBC	TDEA Cipher-Block Chaining
TCFB	TDEA Cipher Feedback Mode
TDEA	Triple Data Encryption Algorithm
TDES	Triple Data Encryption Standard
TECB	TDEA Electronic Codebook
TOFB	TDEA Output Feedback
TLS	Transport Layer Security
USB	Universal Serial Bus
XOF	Extendable-Output Function