IBM Cloud Object Storage System's™

FIPS Cryptographic Module

Software Version 2.0

Documentation Version 20240529

May 29, 2024

Non-Propriety FIPS 140-2 Security Policy

Contents

Document Information

Copyright Notice

Intended Purpose and Audience

This security guide is designed to instruct developers on how to create a version of the IBM Cloud Object Store uses an encryption module that is compliant to FIPS 140-2 Level 1 for a software module.

Acknowledgments

IBM serves as the “vendor” for this validation. Project management coordination and technical work for this effort was provided by:

Jason Cox
222 S Riverside Plz Ste 1700
Chicago, IL 60606 USA
+1 312 871-5351
Jason.P.Cox@ibm.com

Mark Seaborn
222 S Riverside Plz Ste 1700
Chicago, IL 60606
+1 312 871-5351
mseaborn@us.ibm.com

with additional technical work by:

Dusty Hendrickson
222 S Riverside Plz Ste 1700
Chicago, IL 60606 USA
dhendrickson@us.ibm.com

References

Reference	Full Specification Name
ANSI X9.31	*Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)*
FIPS 140-2	*Security Requirements for Cryptographic modules, May 25, 2001*
FIPS 180-4	http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf]
FIPS 186-4	*Digital Signature Standard*
FIPS 197	*Advanced Encryption Standard*
FIPS 198-1	*The Keyed-Hash Message Authentication Code (HMAC)*
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 800-56A	*Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*
SP 800-67R1	*Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*
SP 800-89	*Recommendation for Obtaining Assurances for Digital Signature Applications*
SP 800-90B	*Recommendation for Random Number Generation Using Deterministic Random Bit Generators*
SP 800-90Arev1	http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Arev1.pdf]
SP 800-131A	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*

1. Introduction

This document is the non-proprietary security policy for the IBM Cloud Object Storage System's™ FIPS Cryptographic Module (hereinafter referred to as the Module).

The Module is a software library providing a C-language API for use by other processes that require cryptographic functionality. The Module is classified by FIPS 140-2 as a software module, multi-chip standalone module embodiment. The physical cryptographic boundary is the physical perimeter of the general-purpose computer on which the Module is installed. The logical cryptographic boundary of the Module is the libcrypto object module, a single object module file named *libcrypto.so.* The Module only communicates with the calling application (the process that invokes the Module services). The Module's software version for this validation is 2.0.

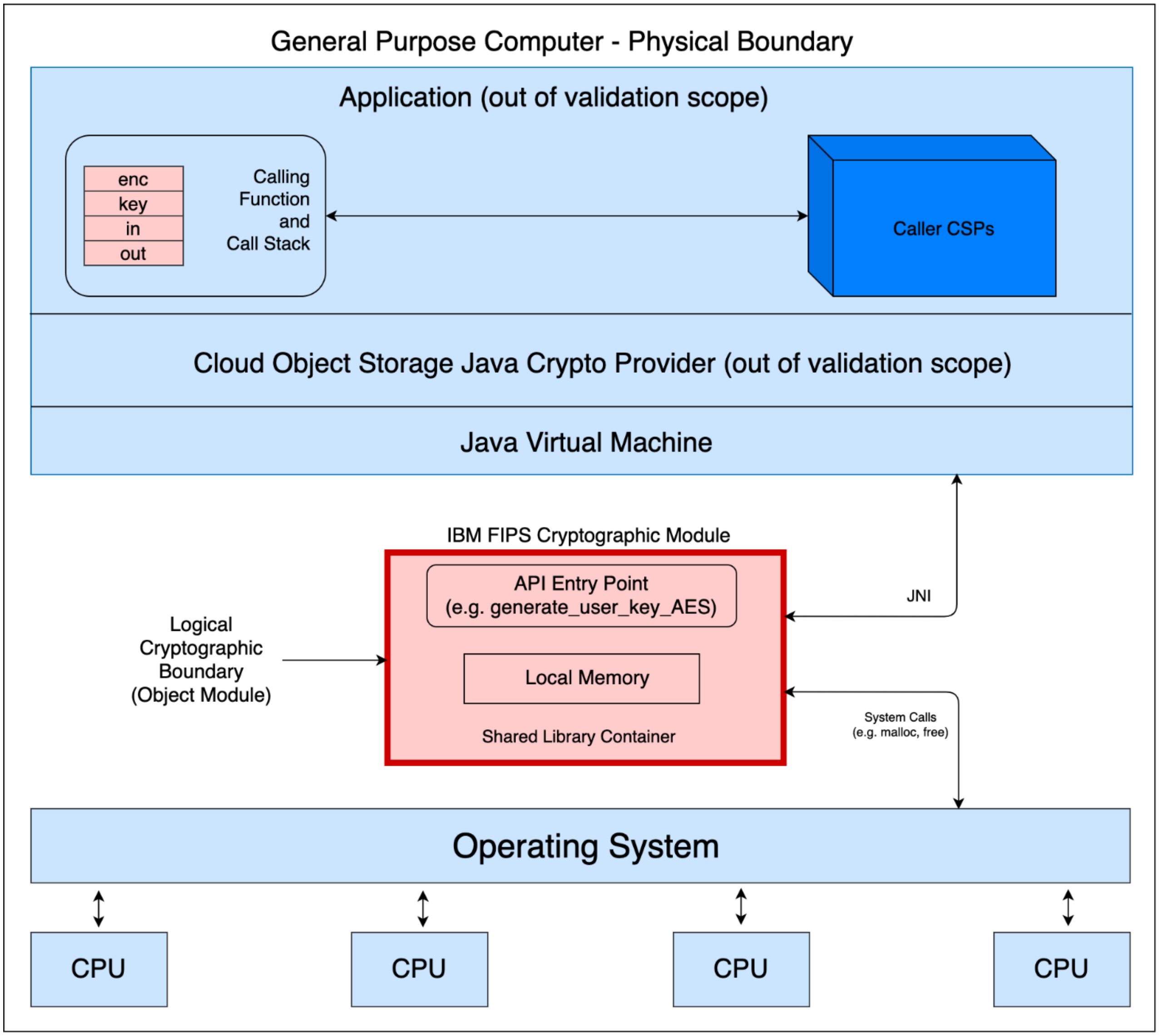
This Security Policy describes the features and design of the module using the terminology contained in the FIPS 140-2 specification. FIPS 140-2, Security Requirements for Cryptographic Modules specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST Cryptographic Module Validation Program (CMVP) validates cryptographic modules to FIPS 140-2 and other cryptography-based standards. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information. The FIPS 140-2 standard and information on the CMVP can be found at https://csrc.nist.gov/projects/cryptographic-module-validation-program.

This Security Policy contains only non-proprietary information. This document may be freely reproduced and distributed whole and intact. All other documentation submitted for FIPS 140-2 conformance testing and validation is "IBM - Proprietary" and is releasable only under appropriate non-disclosure agreements. The module meets the overall requirements applicable to Level 1 security for FIPS 140-2 as shown in Table 1 below.

Table 1. 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
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A
Overall	1

The Module’s block diagram can be found in **Figure 1: Module Block Diagram**. The diagram depicts the Module’s cryptographic boundary as a heavy red boarder labeled *Logical Cryptographic Boundary (Object Module).



General Purpose Computer - Physical Boundary
Application (out of validation scope)
enc
key
in
out
Calling Function and Call Stack
Caller CSPs
Cloud Object Storage Java Crypto Provider (out of validation scope)
Java Virtual Machine
IBM FIPS Cryptographic Module
API Entry Point (e.g. generate_user_key_AES)
Local Memory
Shared Library Container
Logical Cryptographic Boundary (Object Module)
JNI
System Calls (e.g. malloc, free)
Operating System
CPU
CPU
CPU
CPU

Figure 1: Module Block Diagram

2. Ports and Interfaces

The module runs on a General Purpose Computer (GPC). The Physical Cryptographic Boundary for the module is the case of that GPC. All the physical components are standard electronic components; there are not any custom integrated circuits or components dedicated to FIPS 140-2 functionality.

Table 2. Logical interfaces

Logical interface type	Description
Control input	API entry point and corresponding stack parameters
Data input	API entry point data input stack parameters
Status output	API entry point return values and status stack parameters
Data output	API entry point data output stack parameters

The Data Input interface consists of the input parameters of the API functions and data received through the I/O system calls. The Data Output interface consists of the output parameters of the API functions and the data sent through the I/O system calls. The Control Input interface consists of the API function calls. The Status Output interface includes the return values of the API functions and status sent through output parameters.

3. Modes of Operation and Cryptographic Functionality

The Module supports only a FIPS 140-2 Approved mode. Tables 3 lists the approved algorithm, Table 4 lists the algorithm certs used by the entropy source, and Table 5 lists the non-approved but Allowed algorithms, respectively.

Table 3. FIPS Approved Cryptographic Functions

Algorithm/Standard	Cert #	Options	Function
AES [FIPS 197]	A2467	256 bits; CBC, CTR	Encryption/Decryption
CMAC [SP 800-38B] CCM [SP 800-38C] GCM [SP 800-38D]	A2467	256 bits CCM; GCM; CMAC	Authenticated Encryption/Decryption
KDF SSH, KDF TLS [SP 180-135]	CVL A2467	SSHv2 and TLSv1.2	CVL KDF
DRBG [SP 800-90Arev1]	A2467	CTR DRBG (AES-256)	Random Number Generation
KAS-ECC-SSC [SP 800-56Arev3]	A2467	ephemeralUnified, onePassDh, staticUnified; P-224, P-256, P-384, P-521	Key Agreement Scheme Shared Computation Secret; Key establishment methodology provides between 112 and 256 bits of encryption strength
ENT (NP) [SP 800-90B]	N/A	N/A	ENT (NP)
HMAC [FIPS 198]	A2467	SHA-1, SHA-2 (256, 384, 512)	Keyed Hash
DSA [FIPS 186-4]	A2467	PQG Gen, PQG Ver, DSA KeyGen, SigGen (2048/3072 with all SHA-2 sizes), SigVer (2048/3072 with SHA-1, SHA2-256, SHA2-384, SHA2-512)	Asymmetric Key Generation; Digital Signature Generation and Verification
ECDSA [FIPS 186-4]	A2467	KeyGen: CURVES (P-224, P-256, P-384, P-521) SigGen: Curve: P-224, P-256, P-384, P-521 Hash Algorithm: SHA2-256, SHA2-384, SHA2-512 SigVer: Curve: P-224, P-256, P-384, P-521 Hash Algorithm: SHA-1, SHA2-256, SHA2-384, SHA2-512	Asymmetric Key Generation; Digital Signature Generation and Verification
RSA [FIPS 186-4]	A2467	SigGen PKCS 1.5 (2048/3072/4096); SigVer PKCS v1.5 (1024/2048/3072)	Digital Signature Generation and Verification
RSA [FIPS 186-2]	A2467	SigVer9.31, SigVerPKCS1.5, SigVerPSS (1024/1536/2048/3072/4096)	Digital Signature Verification
PBKDF [SP 800-132]	A2467	HMAC Algorithm: SHA-1, SHA2-256, SHA2-384, SHA2-512, SHA3-256, SHA3-384, SHA3-512	Password based KDF
SHS [FIPS 180-4]	A2467	SHA-1, SHA-2 (256, 384, 512); SHA3-(256, 384, 512), SHAKE-128,	Message Digest

Algorithm/Standard	Cert #	Options	Function
SHA-3 [FIPS 202]		SHAKE-256	
CKG (vendor affirmed) [SP 800-133rev2]	N/A	N/A	Cryptographic Key Generation. In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation as per section 6 in SP 800-133rev2. The resulting generated seed used in the asymmetric key generation is the unmodified output from SP800-90Arev1 DRBG.

Table 4. FIPS Approved Algorithm (Vetted Conditioner used by Entropy Source)

Algorithm/Standard	Cert #	Options	Function
SHA2-512 [FIPS 180-4]	A2466	N/A	Prerequisite Algorithm for HASH_DRBG
HASH_DRBG [SP 800-90Arev1]	A2466	Mode: SHA2-512	Vetted Conditioner for Entropy Source

Notes:

1. Not all CAVS tested modes of the algorithms are used in this module.
2. Per the requirements from IG 7.18, SHA2-512 (SHS Cert. #2466) and HASH_DRBG (DRBG Cert. #A2466) implemented by IBM Cloud Object Storage Internal IRQ Entropy Source Module was validated for the vetted conditioning component (SHA2-512 and HASH_DRBG) used by the module's entropy source. No Power-up test was conducted for SHA-256 (SHS Cert. #2466) and HASH_DRBG (DRBG Cert. #A2466).
3. Per FIPS 198-1 and SP 800-107, keys less than 112 bits in length are not approved for HMAC generation.
4. No parts of the TLS and SSH protocols, other than the KDFs, have been tested by the CAVP or CMVP.

Table 5. Non-FIPS Approved but Allowed Cryptographic Functions

Algorithm	Description
RSA (encrypt, decrypt) with key size equal or larger than 2048 bits up to 16384 bits.	Key wrapping; key establishment methodology provides between 112 and 256 bits of encryption strength.

4. Roles, Services, and Authentication

4.1 Roles

The module supports the Crypto Officer (CO) role and User role, which meets all FIPS 140-2 level 1 requirements for Roles and Services. The module does not support a Maintenance role. The User and Crypto Officer roles are implicitly assumed by the entity accessing services implemented by the module. No further authentication is required. The module does not allow concurrent operators.

4.2 Services

All services implemented by the Module are listed below, along with a description of service CSP access.

Table 6. Services and CSP Access

Service	Role	Description/CSP	Access Permission
Initialize	User, CO	Module initialization. **CSPs**: None	N/A
Self-Tests	User, CO	Perform self-tests **CSPs**: None	N/A
Show status	User, CO	Functions that provide module status information. **CSPs**: None	N/A
Zeroize	User, CO	Functions that destroy all CSPs. **CSPs**: All CSPs	read/write/execute
Random Number Generation	User, CO	Used for random number and asymmetric key generation. **CSPs**: CTR_DRBG CSPs	read/write/execute
Asymmetric Key Generation	User, CO	Used to generate asymmetric keys. **CSPs**: DSA SGK, ECDSA SGK	read/write/execute
Symmetric Encrypt/Decrypt	User, CO	Used to encrypt or decrypt data. **CSPs**: AES EDK, AES GCM Key	read/write/execute
Symmetric digest	User, CO	Used to generate or verify data integrity with CMAC. CSPs: AES CMAC Key	read/write/execute
Message Digest	User, CO	Used to generate a SHA-1 or SHA-2 message digest. **CSPs**: None	read/write/execute
Keyed Hash	User, CO	Used to generate or verify data integrity with HMAC. **CSPs**: HMAC Key	read/write/execute
Key Agreement	User, CO	Performs key agreement primitives on behalf of the calling process. **CSPs**: EC Diffie-Hellman Private key, EC Diffie-Hellman Shared Secret	read/write/execute
Key Transport	User, CO	Encrypts or decrypts a key value on behalf of the calling process. **CSPs**: RSA KDK	read/write/execute
Digital Signature	User, CO	Used to generate or verify digital signatures. **CSPs**: RSA SGK, DSA SGK, ECDSA SGK	read/write/execute
Key Derivation	User, CO	Perform key derivation per SSH or TLS **CSPs**: SSH Shared Secret, TLS Pre-Master Secret and TLS Master Shared Secret	read/write/execute

Utility	User, CO	Miscellaneous helper functions. **CSPs**: None	N/A

4.3 Operator Authentication

The module is a software-only cryptographic module. No authentication is required at security level 1, and the authentication is implicit by assumption of the role.

5. Operational Environment

The Module will operate in a modifiable operational environment per the FIPS 140-2 definition. The operating system shall be restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded). The external application that makes calls to the Module is the single user of the Module, even when the application is serving multiple clients.

As per FIPS 140-2 Implementation Guidance G.5, 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.

The GPC(s) used during testing met Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part 15, Subpart B. FIPS 140-2 validation compliance is maintained when the module is operated on other versions of the GPOS running in single user mode, assuming that the requirements outlined in NIST IG G.5 are met.

The module has been tested on the following platforms.

Table 7. Tested Platforms

#	Operating System	Processor	Platform
1	ClevOS 3.16 FIPS	Intel Xeon 6230N with PAA	IBM A10 Series
2	ClevOS 3.16 FIPS	Intel Xeon 6230N without PAA	IBM A10 Series
3	ClevOS 3.16 FIPS	Intel Xeon E5-2620 with PAA	PIO-648R-E1CR36L+-ST031
4	ClevOS 3.16 FIPS	Intel Xeon E5-2620 without PAA	PIO-648R-E1CR36L+-ST031
5	ClevOS 3.16 FIPS	Intel Xeon E5-2620 with PAA	PIO-628U-TR4T+-ST031
6	ClevOS 3.16 FIPS	Intel Xeon E5-2620 without PAA	PIO-628U-TR4T+-ST031

Table 8. Vendor Affirmed Tested Platforms

#	Operating System	Processor	Platform
1	ClevOS 3.16 FIPS	Intel Xeon 4314	IBM 4616-A1D Series
2	ClevOS 3.16 FIPS	Intel Xeon 4314	IBM 4616-M1D Series
3	ClevOS 3.16 FIPS	Intel Xeon 4314	IBM 4616-C1D Series
4	ClevOS 3.16 FIPS	Intel Xeon 4314	IBM 4616-S2D Series
5	ClevOS 3.16 FIPS	Intel Xeon Gold 6438N	IBM 4616-S3D Series
6	ClevOS 3.17 FIPS	Intel Xeon 6126	IBM A10 Series
7	ClevOS 3.17 FIPS	Intel Xeon 6226	IBM A10 Series
8	ClevOS 3.16 FIPS	Intel Xeon 4110	IBM M10 Series
9	ClevOS 3.16 FIPS	Intel Xeon 4210R	IBM M10 Series
10	ClevOS 3.16 FIPS	Intel Xeon 4110	IBM C10 Series
11	ClevOS 3.16 FIPS	Intel Xeon 4210R	IBM C10 Series

12	ClevOS 3.16 FIPS	Intel Xeon 4416+	IBM 4616-A2D Series
13	ClevOS 3.16 FIPS	Intel Xeon 4416+	IBM 4616-M2D Series
14	ClevOS 3.16 FIPS	Intel Xeon 4416+	IBM 4616-C2D Series
15	ClevOS 3.16 FIPS	Intel Xeon 4416+	IBM 4616-S4D Series

6. Physical Security

There are no physical security requirements as this is a software module.

7. Critical Security Parameters and Public Keys

All CSPs used by the Module are described in this section. All access to these CSPs by Module services is described in *Authentication and Services*. The CSP names are generic, corresponding to API parameter data structures.

Table 9. Critical Security Parameters

CSP Name	Description
RSA SGK	RSA (2048/3072/4096 bits) signature generation key
RSA KDK	RSA (2048 to 16384 bits) key decryption (private key transport) key
DSA SGK	DSA (2048/3072 bits) signature generation key
ECDSA SGK	ECDSA (Curves: P-224, P-256, P-384, P-521) signature generation key
EC Diffie-Hellman Private	EC DH private key with Curves: P-224, P-256, P-384, P-521
EC Diffie-Hellman Shared Secret	EC DH Shared secret calculated from KAS-ECC-SSC with Curves: P-224, P-256, P-384, P-521
AES EDK	AES (256 bits) encrypt / decrypt key
AES CMAC Key	AES (256 bits) CMAC generate / verify key
AES GCM Key	AES (256 bits) encrypt / decrypt / generate / verify key
HMAC Key	Keyed hash key (256/384/512 bits)
CTR_DRBG CSPs	V (256 bit) and Key (AES 256 bits), entropy input (length dependent on security strength)
SSH Shared Secret	Secret value used in construction of key for the specified SP 800-135 SSH KDF
TLS Pre-Master Secret	Shared Secret; 48 bytes of pseudorandom data, Internally Generated
TLS Master Shared Secret	Secret value used in construction of key for the specified SP 800-135 TLS KDF

Below is the table listing all public keys used within the module

Table 10. Public Keys

CSP Name	Description
RSA KEK	RSA (2048 to 16384-bit) key encryption (public key transport) key
RSA SVK	RSA (1024 to 3072 bits) signature verification public key
DSA SVK	FIPS 186-4 DSA (2048/3072 bits) signature verification key
ECDSA SVK	FIPS 186-4 ECDSA (Curves: P-224, P-256, P-384, P-521) signature verification key

Key/CSP Storage, Generation, Entry, Output and Zeroization

Storage: RAM, associated to entities by memory location. The module stores DRBG state values for the lifetime of the DRBG instance. The module uses CSPs passed in by the calling application on the stack or registers. The module does not store any CSP persistently (beyond the lifetime of an API call), with the exception of DRBG state values used for the module's default key generation service.

Generation: In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation as per section 6 in SP800-133. The resulting generated seed used in the asymmetric key generation is the unmodified output from SP800-90Arev1 DRBG. The calling application is responsible for storage of generated keys returned by the Module.

The module is a software module that contains an approved DRBG that is seeded exclusively from a known entropy source located within the operational environment inside the module’s physical boundary but outside the logical boundary, which is compliant with FIPS 140-2 IG 7.14 #1 (b). The minimum number of bits of entropy requested per each GET function call is at least 256 bits.

Entry: All CSPs enter the Module’s logical boundary in plaintext as API parameters, associated by memory location. However, none cross the physical boundary.

Output: The Module does not output CSPs, other than as explicit results of key generation services. However, none cross the physical boundary.

Destruction: Zeroization of sensitive data is performed automatically by API function calls for temporarily stored CSPs. In addition, the module provides functions to explicitly destroy CSPs related to random number generation services. The calling application is responsible for parameters passed in and out of the module.

Private and secret keys as well as seeds are provided to the module by the calling application and are destroyed when released by the appropriate API function calls. Keys residing in internally allocated data structures (during the lifetime of an API call) can only be accessed using the module defined API. The operating system protects memory and process space from unauthorized access. Only the calling application that creates or imports keys can use or export such keys. All API functions are executed by the invoking calling application in a non-overlapping sequence such that no two API functions will execute concurrently. An application assumed as the Crypto Officer role or User role has access to all key data generated during the operation of the module.

Use of AES-GCM

In approved mode, users of the module must not utilize GCM with an externally generated IV unless the source of the IV is also FIPS approved for GCM IV generation. The module’s implementation of AES-GCM is used together with an application that executes outside of the module’s cryptographic boundary. The application negotiates the protocol session’s keys and the value of the IV. The module’s AES-GCM implementation complies with IG A.5 scenario 1 and RFC 5288, and supports acceptable GCM cipher suites from SP 800-52 Rev2, Section 3.3.1. AES-GCM is only used in TLS version 1.2. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, that encounters this condition will trigger a handshake to establish a new encryption key. The Module also supports internal IV generation using the module’s Approved DRBG. The IV is at least 96-bits in length per NIST SP 800-38D, Section 8.2.2. Per FIPS 140-2 IG A.5 Scenario 2 and NIST SP 800-38D, the approved DRBG generates outputs such that the (key, IV) pair collision probability is less than $2^{-32.}$

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.

The Module also supports importing of GCM IVs when an IV is not generated within the Module. In the FIPS approved mode, an IV must not be imported for encryption from outside the cryptographic boundary of the Module as this will result in a non-conformance.

Key derivation using SP800-132 PBKDF

The module provides password-based key derivation (PBKDF), compliant with SP800-132. The module supports option 1a from section 5.4 of [SP800-132], in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK). In accordance to [SP800-132] and IG D.6, the following requirements shall be met.

- Derived keys shall only be used in storage applications. The Master Key (MK) shall not be used for other purposes. The length of the MK or DPK shall be of 112 bits or more.
- A portion of the salt, with a length of at least 128 bits, shall be generated randomly using the SP800-90Arev1 DRBG.
- The iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The minimum value shall be 1000.
- Passwords or passphrases, used as an input for the PBKDF, shall not be used as cryptographic keys. The length of the password or passphrase shall be of at least 20 characters, and shall consist of lower-case, upper-case and numeric characters. The probability of guessing the value is estimated to be 1/(62^20) = 10^(-36), which is less than 2^(-112). The calling application shall also observe the rest of the requirements and recommendations specified in [SP800-132].

8. EMI/EMC

The Cryptographic Security Kernel is a software module. Therefore, the only electromagnetic interference produced is that of the host platform on which the module resides and executes. FIPS 140-2 requires that the host systems on which FIPS 140-2 testing is performed meet the Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal Regulations Part 15. However, all systems sold in the United States must meet these applicable FCC requirements.

9. Self-Test

This section describes the power-up and conditional self-tests performed by the module. The module is designed with a default entry point (DEP) which ensures that the self-tests are initiated automatically when the module is loaded. If any of the tests listed below fails to complete successfully, the module enters into a critical error state where all cryptographic operations and output of any data is prohibited. The Module performs the self-tests listed below on invocation of Initialize or Self-Test.

Table 11. Power On Self-Tests

Algorithm	Type	Test Attributes
AES CBC	KAT	Separate encrypt and decrypt
AES CCM	KAT	Authenticated encryption and decryption are tested separately
AES GCM	KAT	Authenticated encryption and decryption are tested separately
AES CMAC	KAT	Authenticated encryption and decryption are tested separately
DRBG	KAT	DRBG Health Tests: Generate, Reseed, Instantiate functions (per Section 11.3 of SP 800-90Arev1)
DSA	PWCT	Sign and verify
ECDSA	PWCT	Sign and verify
HMAC	KAT	HMAC-SHA-1/SHA-256/SHA-384/SHA-512
KAS-ECC-SSC	KAT	SP800-56Arev3 compliant key agreement scheme shared secret computation
RSA	KAT	Signature generation and verification are tested separately
SHA	KAT	SHA-1, SHA-256, SHA-512
SHA3, SHAKE	KAT	SHA3-256, SHA3-384, SHA3-512, SHAKE-128, SHAKE-256
SP800-132 PBKDF	KAT	With SHA-1, SHA2-256, SHA2-384, SHA2-512, SHA3-256, SHA3-384, SHA3-512
SP800-135 KDF	KAT	TLSv1.2_KDF, SSHv2 KDF
Software integrity	KAT	HMAC-SHA-256

(KAT = Known answer test; PWCT = Pairwise consistency test)

In addition, the module's entropy source also conducted following Self-Tests:

1. ENT (NP) SP800-90B Start-Up Health Tests:
   - Repetition Count Test (RCT)
   - Adaptive Proportion Test (APT)

Note: Please refer to SP800-90B, sections 4.4.1 and 4.4.2 for more information about the RCT and APT.

2. ENT (NP) SP800-90B Continuous Health Tests:
   - Repetition Count Test (RCT)
   - Adaptive Proportion Test (APT)

The FIPS_module_mode_set() function performs all power-up self-tests listed above with no operator intervention required, returning a "1" if all power-up self-tests succeed, and a "0" otherwise. If any component of the power- up self-test fails an internal flag is set to prevent subsequent invocation of any cryptographic function calls. The module will only enter the FIPS Approved mode if the module is reloaded and the call to FIPS_module_mode_set() succeeds.

The Module also implements the following conditional tests:

Table 11. Conditional Tests

Conditional self-tests are run during operation of the module. If any of these tests fail, the module will enter an error state where no services can be accessed by the operators.

- DSA Pairwise consistency test (PWCT)
- ECDSA PWCT
- RSA PWCT

10. Mitigation of Other Attacks

The Module is not designed to mitigate attacks which are outside of the scope of FIPS 140-2.

11. Secure Operation

11.1 Crypto Officer/User Guidance

The module is provided directly to IBM solution developers and is not available for direct download to the general public. The module and its host application are to be installed on an operating system specified in Section 5.

Additional Rules of Operation:

1. The writable memory areas of the module (data and stack segments) are accessible only by the application so that the operating system is in "single user" mode, i.e. only the application has access to that instance of the module.
2. The operating system is responsible for multitasking operations so that other processes cannot access the address space of the process containing the module.
3. Only the services defined in Table 6 shall be used in FIPS mode of operation.

11.2 General Guidance

The module is not distributed as a standalone library and is only used in conjunction with the solution.

The end user of the operating system is also responsible for zeroizing CSPs via wipe/secure delete procedures. If the module power is lost and restored, the calling application can reset the IV to the last value used.

Appendix A: Module Installation

For module installation, IBM assumes the role of Crypto Officer and performs the following steps to configure and install the module. This is performed once at OS build time, and the module cannot be removed.

The tested platform uses Debian packaging to build and install the module. The *dpkg-buildpackage* command is used by the build system to create a Debian package for the Module. The *dpkg* utility is used to make and install the module onto the platform. A file named *rules* is created for the Debian packaging process to direct the compilation of the Module's source. The *rules* files and other Debian specific packaging files are placed in a debian directory located at the same level of the directory structure the as the directory containing the uncompressed Module source code from Red Hat. The commands in the *rules* file are executed relative to the top of the directory containing that uncompressed and expanded content.

Source Assurance and Installation

To assure the integrity of the source code for the module, the source code is managed in IBM's github repositories and access control to those repositories is limited to authorized IBM employees. The installation of the module is done during the CICD process that produces ClevOS images. That process generates a SHA-256-HMAC of the Module and stores it on the system image for verification during module start up. This verification is a runtime operation performed each time the System is started/restarted. ClevOS images are digitally signed to assure they have not been tampered with and the image signature can be validated before initial installation and at upgrade time though the IBM Cloud Object Storage's Manager device.

Linking the Runtime Executable Application

Note that applications interfacing with the Module are outside of the cryptographic boundary. When linking the application with the FIPS Object Module two steps are necessary:

- The HMAC-SHA-256 digest of the FIPS Object Module file must be calculated and verified against the installed digest to ensure the integrity of the Module. This operation is completed by the IBM CICD process at image creation.
- A HMAC-SHA-256 digest of the Module must be generated and embedded in the Module for use by the FIPS_module_mode_set() function at runtime initialization. This is done by the System automatically at runtime.

The ClevOS build system uses *make* facilities (via Debian packaging) to embed the digest in OpenSSL at build time. Failure to embed the digest in the executable object will prevent initialization of FIPS mode. At runtime, the FIPS_module_mode_set() function compares the embedded HMAC-SHA-256 digest generated from IBM's CICD process.

Optimization

The platform was tested against both AES-NI acceleration on and off. ClevOS if AES-NI acceleration is available on the platform and AES-NI will always be used by the platform.

Appendix B: Compilers

This appendix lists the specific compilers used to generate the Module for the respective Operational Environments. Note this list does not imply that use of the Module is restricted to only the listed compiler versions, only that the use of other versions has not been confirmed to produce a correct result.

#	Operational Environment	Compiler
3	ClevOS 3.16.5 FIPS Edition	gcc 8.3

Appendix C: Glossary

Abbreviation	Term
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	application program interface
B	B Elliptic Curve
CBC-MAC	Cipher Block Chaining Message Authentication Code
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
CMAC	Cipher-based Message Authentication Code
CSP	Critical Security Parameter
CTR_DRBG	Counter mode Deterministic Random Byte Generator
ECB	Electronic Code Book
DH	Diffie-Hellman scheme
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
Dual_EC_DRBG	Dual Elliptic Curve Deterministic Random Bit Generator
IBM Cloud Object Storage System	The IBM Cloud Object Storage System Appliances are the general purpose computing devices where the Module is installed
EC	Elliptic Curve
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EDK	Encrypt/Decrypt Key
FIPS	Federal Information Processing Standards
GCM	Galois/Counter Mode
HASH_DRBG	Hash - Deterministic Random Bit Generator
HMAC	Keyed-Hash Message Authentication Code
HMAC_DRBG	Keyed-Hash Message Authentication Code Deterministic Random Bit Generator
K	K Elliptic Curve
KAS	Key Agreement Scheme
KAT	Known Answer Test Power on Self-Test
KDK	Key Decryption Key
KEK	Key Encryption Key
NIST	National Institute of Standards and Technology
P	Prime Elliptic Curve
PCT	Pairwise Consistency Test Power on Self-Test
PKCS	Public-Key Cryptography Standards
PKG	Public Key Generation
PKV	Public Key Validation
POST	Power-up Self-Test
RNG	Random Number Generator
RSA	Rivest-Shamir-Adleman Cryptosystem
SGK	Signature Generation Key