IBM

IBM® z/VM® Version 7 Release 2 System SSL Cryptographic Module

FIPS 140-2

Non-Proprietary Security Policy

Policy Version 1.2

IBM Systems & Technology Group
System z Development
1701 North Street, Building 256-3
Endicott, NY 13760
USA

IBM Research
Zurich Research Laboratory

June 21, 2022

Table of Contents

1. Cryptographic Module Specifications

1.1 Scope of Document

This document describes the services that the IBM® z/VM® Version 7 Release 2 System SSL Cryptographic Module (““z/VM System SSL library” or “z/VM System SSL” or “System SSL” or “module”) provides to security officers and end users, and the policy governing access to those services. It complements official product documentation, which concentrates on server usage of the functionality, as well as environmental set-up.

Module Description The z/VM System SSL library in its FIPS 140-2 configuration consists of a set of loadable 31-bit components. The deployed version consists of the following components:

Table 1 z/VM System SSL libraries

Core	Auxiliary
GSKCMS31	GSKSUS31
GSKC31F	Side Decks
GSKSSL	Message Catalogs
GSKS31F	
ICSFLIB	
GSKKYMAN	

The z/VM System SSL library consists of the core components that provide FIPS 140-2 approved services, as well as some auxiliary components and files. The auxiliary components provide functionality that is not cryptographically relevant. The files consist of side decks and message catalogs.

The z/VM System SSL logical and physical boundaries are described in Figure 3 in the Operational Environment Section.

Note: Throughout this document, the CP Assist for Cryptographic Functions will also be referenced using the terms CP Assist and CPACF.

1.2 Cryptographic Module Specification

The z/VM System SSL module is classified as a multi-chip standalone Software-hybrid module for **FIPS Pub 140-2** purposes. The actual cryptographic boundary for this FIPS 140-2 module validation includes the System SSL module running in configurations backed by hardware cryptography. The System SSL module consists of software-based cryptographic algorithms, as well as symmetric and hashing algorithms provided by the CP Assist for Cryptographic Function (CPACF). (See Figure 1.)

System SSL validation was performed using the z/VM Version 7 Release 2 operating system with the following platform configuration:

1.IBM System z14™ with CP Assist for Cryptographic Functions DES/TDES Enablement Feature 3863

The module running on the above platforms was validated as meeting all **FIPS Pub 140-2** Level 1 security requirements. The z/VM System SSL module is packaged as a set of DLLs and executables which contains all the code for the module.

In addition to the configurations tested by the laboratory, vendor-affirmed testing was performed using z/VM Version 7 Release 2 on the following platforms:

2.IBM Z z15™ with CP Assist for Cryptographic Functions DES/TDES Enablement Feature 3863

3.IBM LinuxONE III with CP Assist for Cryptographic Functions DES/TDES Enablement Feature 3863

4.IBM LinuxONE Emperor II with CP Assist for Cryptographic Functions DES/TDES Enablement Feature 3863

5.IBM Z z14 ZR1 with CP Assist for Cryptographic Functions DES/TDES Enablement Feature 3863

Security level This document describes the security policy for the z/VM System SSL with Level 1 overall security as defined in **FIPS Pub 140-2** [1].

Table 2 System SSL Module Components

Type	Name
Software (DLLs and executables)	5735FAL00: z/VM Version 7 Release 2 with 7201RSU (GA-level release) and the PTF for APAR PH24751
Documentation	Not applicable
Hardware components	z14 CP Assist for Cryptographic Functions DES/TDES Enablement Feature 3863

1.3 Cryptographic Module Security Level

The module is intended to meet requirements of Security Level 1 overall, with certain categories of security requirements not applicable (Table 3).

Table 3 Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module Specification	**1**
Module Ports and Interfaces	1
Roles, Services and Authentication	1
Finite State Model	1
Physical Security	1
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of other attacks	N/A
Overall	1

1.4 Determining Mode of Operation

The module support two modes of operation. The module enters FIPS approved mode after successful completion of the power up self-tests. Invoking a non-approved service will result in the module implicitly switching to non-approved mode. After completion of the service the module will immediately switch back to the FIPS approved mode.

The application utilizing services must enforce key management compliant with **FIPS Pub 140-2** requirements. This should be indicated in an application-specific way that is directly observable by crypto officers and end-users.

While such application-specific details are outside the scope of the validation, they are mentioned here for completeness.

The user application must comply with the key size requirements specified in the latest revision of the NIST Special Publication 800-131A Revision 2. If the services defined in table 6 and 7 are utilized, the module is then FIPS mode. If the services defined in table 8 are utilized, the module will be considered not in FIPS mode.

2. Ports and Interfaces

As a multi-chip standalone Software-hybrid module, the System SSL physical interfaces are the boundaries of the host running System SSL library code. The underlying logical interfaces of the module are self-controlled. Control inputs which control the mode of the module are provided. Data input and data output are provided in the variables passed through user-supplied buffers. Status output is provided in return codes and through messages.

Table 4: Data input, data output, control input and status output

Interfaces into and out of the Module		
FIPS 140-2 Interface	**Logical Interface**	**Description**
Data Input	API	Input variables are passed on the internal application programming interface (API)
Data Output	API	Output results are passed back through the API
Control Input	API function calls	Setting gsk_fips_state_set
Status Output	API return codes	Status output is provided in return codes
Power	Not applicable	Setting of GSK_DEFAULT_FIPS_STATE environment variable

Cryptographic bypass capability is not supported by System SSL.

2.1 Module Status

The System SSL library communicates any error status synchronously through the use of its return codes. It is the responsibility of the calling application to handle exceptional conditions in a FIPS 140-2 appropriate manner.

System SSL is optimized for library use and does not contain any terminating assertions or exceptions. Any internal error detected by System SSL and not induced by user data will be reflected back to the application with an appropriate return code. The calling application must examine the return code and act in a FIPS 140-2 appropriate manner to such failures and reflect this error in a fashion consistent with this application.

User-induced or internal errors do not reveal any sensitive material to callers.

3. Roles, Services and Authentication

3.1 Roles

The module supports two roles: a cryptographic officer (Officer) role and a User role (Table 5). The module does not support user identification or authentication that would allow the module to distinguish between the two supported roles. Each of the roles is authenticated through the operating system implicitly prior to using any system services.

The Officer role is a purely administrative role that does not involve the use of cryptographic services. The role is not explicitly authenticated but assumed implicitly on implementation of the module's installation and usage sections defined in the security rules section.

The User role has access to all of the module's services. The role is not explicitly authenticated, but assumed implicitly on access of any of the non-Officer services. An operator is implicitly in the User or Officer role based upon the service(s) chosen. If any of the User-specific services are called, then the operator is in the User role; otherwise the operator is in the Officer role.

Table 5 Roles and Authentication Mechanisms

Role	Type of Authentication	Authentication Data	Strength of Mechanism
Officer	None(Automatic)	None	N/A
User	None(Automatic)	None	N/A

3.2 Services

The module provides services (Tables 6, 7 and 8). Services are accessed only by an authorized calling application. Officers perform install and configuration of the module. Users perform cryptographic services.

Certificate management services (CMS) perform both non-cryptographic and cryptographic PKI management activities, as well as general cryptographic operations, such as signature verification. Functions in this group parse and categorize X.509 certificates and transport certificates, and also handle standard encodings (such as PKCS#7). Cryptographic operations, such as signature verification, are delegated to lower-level crypto core functions.

SSL protocol implementation is split into infrastructure and protocol functions. Lower-level functions implement SSL message formatting and other primitives. SSL protocol operations extend SSL primitives with handshake state machines, session caching, and attribute parsing to provide a full SSL/TLS implementation. Both System SSL layers use cryptographic cores indirectly. SSL 3.0 functionality is disallowed by FIPS 140-2: all other compliance checks are implemented at lower levels. Cipher suites are restricted to those built with approved algorithms only.

Format conversions, labeled as “other operations", are other non-cryptographic commands that change the representation of data. Format converters read and write, among others, the following formats:

- Various protocols based on ASN.1/BER encoded data (PKI-related and similar standard formats)
- Conversions between industry-standard object identifiers and internal symbolic constants (mainly intrinsic,

not externalized).

Protocol-level format conversions generally package data without modification, treating output or input of lower-level crypto primitives as opaque data. The purpose of these conversions is to bridge protocols with predefined formats with cryptographic primitives, which are oriented around raw byte streams or blocks, but generally not standard encapsulation methods:

- Base-64 encoding (“ASCII armor"), generating and reading printable representation of binary data, for example, encountered in certificates

- Conversions between ASCII and non-ASCII data (such as EBCDIC), non-cryptographic but potentially modifying security-relevant data.

Format conversion services do not provide cryptographic functionality, but may use other services if the transport mechanism requires them. As an example, if signed data is represented as a standard ASN.1 structure, it implicitly uses one of the sign calls. Similarly, certificate management or processing of PKCS#7 data may involve signature verifications, for example.

Table 6: Approved Services in FIPS-Approved mode of operation
(There are algorithms that have been CAVS tested with key sizes and block chaining modes for which the module does not provide interfaces. Only the algorithms’ key sizes and block chaining modes present in this table are made available by the module.)

Service	Roles		CSP	Modes / Notes	Cert #	Access (Read, write, execute)	Standard
	User	Crypto Officer					
Module installation And Configuration		X	N/A	N/A	N/A	N/A	N/A
Self-Tests	X		N/A	N/A	N/A	N/A	N/A
Zeroization	X		All keys and CSPs	N/A	N/A	Write	N/A
Show Status	X		N/A	N/A	N/A	N/A	N/A
Software							
Symmetric Algorithms							
AES Encryption and Decryption	X		AES Symmetric key (128, 256 bit)	CBC	C1862	Read	FIPS 197 SP 800-38A
AES Authentication	X		AES Symmetric key 128, 192 or 256 bits	GMAC	C1862	Read	SP 800-38D
Triple DES Encryption And Decryption	X		Triple DES Symmetric key (192 bit)	CBC	C1862	Read	SP 800-67
Public Key Algorithms							

DSA Parameter/Key Generation	X		DSA Parameter And Asymmetric keys	L=2048, N=256	C1862	Read Write	FIPS 186-4
DSA Signature Generation	X		DSA Asymmetric Private Key	L=2048, N=256 with SHA(224/256)	C1862	Read	FIPS 186-4
DSA Signature Verification	X		DSA Asymmetric Public Key	L=1024,N=160 with SHA(1/224/256) L=2048,N=256 with SHA (1/224/256)	C1862	Read	FIPS 186-4
RSA Key generation	X		RSA Asymmetric Key	2048 and 3072	C1862	Read Write	FIPS 186-4
RSA signature Generation – PKCS1.5	X		RSA Asymmetric Private Key	2048 and 3072 with SHA[2] (1/224/256/384/512) SHA-1 for use with protocols only.	C1862	Read	FIPS 186-4
RSA Signature Verification – PKCS1.5	X		RSA Asymmetric Public Key	1024, 2048 and 3072 with SHA (1/224/256/384/512)	C1862	Read	FIPS 186-4
Hash Functions							
SHS Message Digest	X		N/A	SHA -1 SHA-224 SHA-256 SHA-384 SHA-512	C1862	N/A	FIPS 180-4
Message Authentication Codes (MACs)							
HMAC Message Authentication	X		Key sizes 112 bits in length and greater[1]	HMAC SHA-1, HMAC SHA-256 HMAC SHA-384	C1862	Read	FIPS 198-1
Component							
TLS Key Derivation	X		TLS V1.0, V1.1, V1.2 premaster	with SHA-256 and SHA-384	CVL C1862	Read Write	SP 800-135

			secret, master secret and derived key				
Other							
DRBG			Entropy input, Seed, Internal State (V and C values)	SHA-512 Hash DRBG	C1862	Read	SP 800-90A
CP Assist for Cryptographic Functions							
Symmetric Algorithms							
AES Encryption and Decryption	X		AES Symmetric key (128, 256 bit)	CBC	C79	Read	FIPS 197 SP 800-38A
Triple DES Encryption And Decryption	X		Triple DES Symmetric key (192 bit)	CBC	C79	Read	SP 800-67
Hash Function							
SHS Message Digest	X		N/A	SHA -1 SHA-224 SHA-256 SHA-384 SHA-512	C79	N/A	FIPS 180-4

Notes:
1. Per FIPS 198-1 and SP 800-107, keys less than 112 bits in length are not approved for HMAC generation.

2. Digital signature generation using SHA-1 is Approved only when used within the TLS protocol, as explained In SP800-52

Table 7: Allowed Services in FIPS-Approved mode of operation

Service	Roles		CSP	Access (Read, write, execute)	Standard	Caveat
	User	Crypto Officer				
RSA Key Wrapping based on PKCS#v1.5	X		RSA Asymmetric Private Key Modulus size from 2048 to 4096 bits	Read	PKCS#v1.5 (Allowed per IG D.9)	key wrapping; key establishment methodology provides between 112 and 149 bits of encryption strength
Message Authentication Codes (MACs)						
Hash Functions						
MD5	X		N/A	N/A	N/A	MD5 (when used in

						the context of TLS protocol)
NDRNG	X		seed	Read	N/A	Seeding for the DRBG

Table 8: Non-approved Services in non-FIPS-Approved mode of operation

Service	Notes
Software	
Public Key Algorithms	
RSA Key Generation, Digital Signature Generation	Key size equal to 4096 or less than 2048 not approved
RSA Key Wrapping	Key bit sizes less than 2048 not approved
RSA Digital Signature Verification	Key bit size 4096 is not approved
DSA Parameter Generation, Key Generation, Digital Signature Generation	Key Parameters L=1024, N=160 not approved
DSA Parameter Generation, Key Generation, Digital Signature Generation/Verification	Key Parameters L=3072
DSA Signature Generation	Using SHA-1
Diffie-Hellman Key agreement or shared secret computation	Using any key size
Message Authentication Codes (MACs)	
HMAC	Key sizes less than 112 bits HMAC using SHA-224/512 HMAC-MD5 usage outside of the TLS protocol
Message Digest	
MD5	MD5 usage outside of the TLS protocol
Component	
TLS	using SHA-1,SHA-224/SHA-384

Note: When any of the services in table 8 are utilized, the module will be in non-FIPS mode.

4. Physical Security

The System SSL installation inherits the physical characteristics of the host running it. The System SSL library has no physical security characteristics of its own.

The CP Assist for Cryptographic Function (CPACF) (see Figure 1) is also a hardware device – part of the CoProcessor Unit (CoP) and offers the full complement of the Triple DES algorithm, Advanced Encryption Standard (AES) algorithm and Secure Hash Algorithm (SHA). Security Level 1 is satisfied by the device (CoP) being included within the physical boundary of the module and the device being made of commercial-grade components.

CPACF Physical Design: Each microprocessor (core) on the 8-core chip has its own dedicated CoP, which implements the crypto instructions and also provides the hardware compression function. The compression unit is integrated with the CP Assist for Cryptographic Function (CPACF), benefiting from combining (sharing) the use of buffers and interfaces.

The CP Assist for Cryptographic Function (CPACF) accelerates the encrypting and decrypting of SSL transactions and VPN-encrypted data transfers and data-storing applications. The assist function uses a special instruction set for symmetrical clear key cryptographic encryption and decryption operations. Five special instructions are used with the cryptographic assist function.

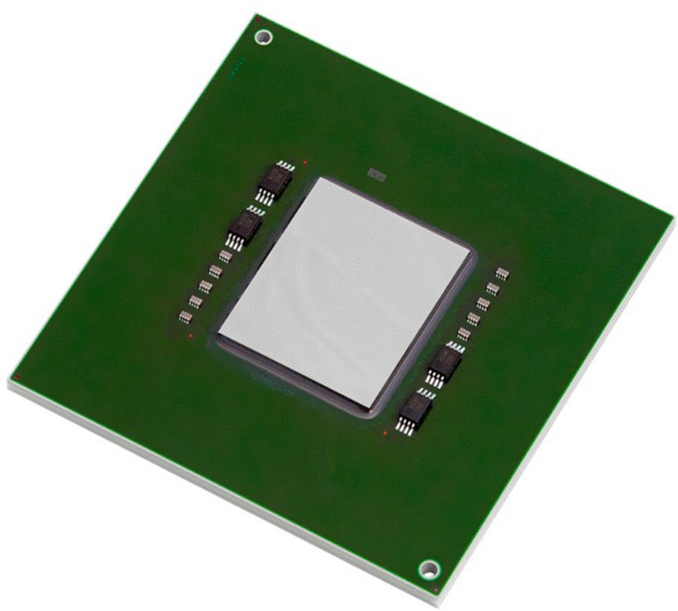


Figure 1: Processor Unit chip with CPACF CoP

Figure 2: IBM z Systems z14 Mainframe Computer.

5. Operational Environment

5.1 Installation and Invocation

System SSL is installed as part of the z/VM Version 7 Release 2 SDO. This is the validated version.

The cryptographic module is invoked via specific IBM programs. All other programs attempting to access the cryptographic module will abend. These authorized programs are the z/VM TLS/SSL Server, the z/VM LDAP server and client utilities, the gskkyman utility (certificate management) and the gsktrace utility (debugging utility).

The System SSL security module is written in C, with certain functionality contained within assembler, such as functions that utilize the CPACF. Extensive internal consistency checks verify both user input and library configuration, terminating early if errors are encountered. Internal errors are externalized and do not terminate execution, since the code has been developed mainly for library use.

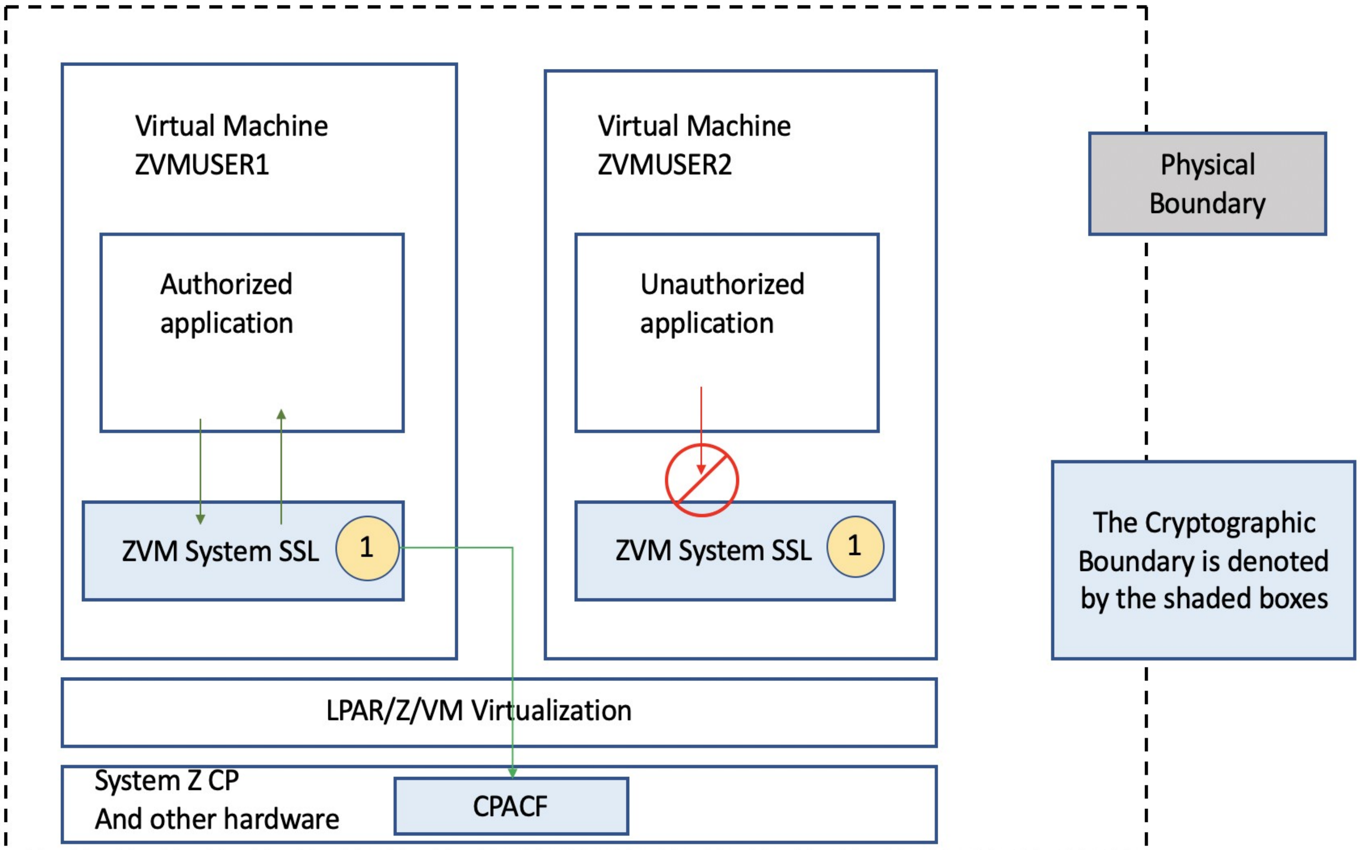
Using z/VM System SSL in a FIPS 140-2 approved manner assumes that the following defined criteria are followed:

- The Operating System enforces authentication method(s) to prevent unauthorized access to Module services.
- All host system components that can contain sensitive cryptographic data (main memory, system bus, disk storage) must be located within a secure environment.
- The application using the module services must consist of one or more processes in which each process is utilizing a separate copy of the executable code.
- The unauthorized reading, writing or modification of the virtual machine which contains the System SSL instance is not allowed.
- An instance of the System SSL Library DLLs must be accessed only by a single process (virtual machine). This means that each process has it own instance of the System SSL Library DLLs.
- The CP Assist for Cryptographic Functions DES/TDES Enablement Feature 3863 must be installed and enabled.

This module will be in FIPS mode when calling services from Table 6 or 7. The module will be in non-FIPS mode when calling any of the services in Table 8.

The System SSL DLLs and gskkyman certificate utility represent the logical boundary of the module. The physical cryptographic boundary for the module is defined as the enclosure of the host on which the cryptographic module is to be executed.

As shown in Figure 3, System SSL Cryptographic Module, the cryptographic module's DLLs are instantiated within an application's virtual machine. Each operating system component that utilizes the System SSL cryptographic module will create a new instance of the z/VM System SSL DLLs.



1 The System SSL DLLs are considered to be within the cryptographic boundary. The System SSL DLLs may issue the System z CPACF machine instructions to perform symmetric encryption and hashing cryptographic functions that are provided by these machine instructions.

Figure 3: System SSL Cryptographic Module

6. Key Management

6.1 Key Storage

The System SSL library retains key material within its virtual machine. In a typical SSL/TLS setting, private keys would be imported from a different key store. Public keys (certificates) would be distributed through other channels, such as out-of-band PKI messages.

The module provides API parameters to applications such that key material can be used in conjunction with cryptographic services. It is the responsibility of applications using library services to ensure that these services are used in a FIPS 140-2 compliant manner. Keys managed or generated by applications or libraries may be passed from applications to the module in the clear, provided that the sending application or library exists within the physical boundary of the host computer.

Key material resides in memory as clear data or in a standard key store format. The most frequently used standard formats, using passphrase-derived keys such as PKCS#12, are classified as clear-key storage according to **FIPS Pub 140-2** guidelines.

6.2 Key Generation

Key generation uses random bytes produced by an approved RNG algorithm (specified in **NIST SP 800-90A**) which is known as Hash_DRBG (DRBG). The DRBG instance is based on SHA-512 and, thus, has a security strength of 256 bits.

The DRBG is seeded by hardware based (CPACF) TRNG.

All asymmetric key generation algorithms use an instance of the software DRBG engine for random numbers. The DRBG instance is seeded with at least 48 bytes (384 bits) of entropy. Seeding for the DRBG instance comes from a true random number generator (TRNG/NDRNG). This NDRNG extracts entropy by sampling bytes from the system clock. Samples are conditioned using an approved, non-keyed, conditioning function (SHA-384). The DRBG is reseeded whenever 4M (4,194,304) cumulative bytes have been requested.

The Key Generation methods implemented in the module for Approved services in FIPS mode is compliant with SP800-133.

DSA key generation is done according to **FIPS Pub 186-4** [3]. When in FIPS mode, RSA key generation implements only the **FIPS Pub 186-4** key generation method. A non-compliant RSA key generation method is also present, which allows for the generation of RSA keys shorter than 1024 bits (**FIPS Pub 186-4** does not permit generation of shorter keys), the module will be in non-FIPS mode with these keys.

6.3 Key Establishment

The module provides support for asymmetric key establishment methods as allowed by Annex D in the **FIPS Pub 140-2**. The supported asymmetric key establishment method is RSA Key Wrapping.

When using RSA Encrypt/decrypt, the allowed modulus lengths must be between 2048 and 3072 bits which provides between 112 and 128 bits of encryption strength. Note that NIST SP 800-131A Revision 2 requires a modulus length of at least 2048 bits, which provides at least 112 bits of encryption strength.

6.4 Key Entry and Key Exit

The module does not support manual key entry or intermediate key generation key output.

The module does not output or input keys outside of the physical boundary, with the exception of the premaster secret that is used for key establishment. The premaster secret is wrapped with RSA.

6.5 Key Protection

To enforce compliance with **FIPS Pub 140-2** key management requirements on the System SSL library itself, code issuing calls must manage keys in a **FIPS Pub 140-2** compliant method. Keys managed or generated by applications may be passed from the application to the module in the clear in the **FIPS Pub 140-2** validated configuration.

The management and allocation of memory is the responsibility of the operating system. It is assumed that a unique process is allocated for each request, and that the operating system and the underlying hardware control access to the virtual machine which contains the process that uses the module. Each instance of the cryptographic module is self-contained within a process; the library relies on such process separation and address separation to maintain confidentiality of secrets. All platforms used during **FIPS Pub 140-2** validation provide per-process protection for user data. Keys stored internally within the address range of System SSL are similarly separated logically (even if they reside in the same virtual machine).

All keys are associated with the User role. It is the responsibility of application program developers to protect keys output from the System SSL module.

6.6 Key Destruction

Applications must destroy persistent key objects and similar sensitive information using **FIPS Pub 140-2** compliant procedures. The System SSL library itself does not destroy externally stored keys and secrets, as it does not own or discard persistent objects. Objects, when released on behalf of a caller, are erased before they are released.

6.7 Key/CSP Table

Table 9: Keys and CSPs

Key/CSP	Generation	Entry/Output	Storage	Zeroization
AES Symmetric Key	The key material is entered via API parameter or derived during TLS handshake.	Input and output through API parameters	application memory	Zeroized when key context is closed
Triple-DES Symmetric Key				
HMAC Key				
RSA Private Key	SP 800 90A DRBG as a seed to the FIPS 186-4 key generation method	Input and output through API parameters	application memory	Zeroized when key context is closed
DSA Private Key	SP 800 90A DRBG as a seed to the FIPS 186-4 key generation method	Input and output through API parameters	application memory	Zeroized when key context is closed
SP 800-90A DRBG seed and entropy input)	Obtained from NDRNG	N/A	application memory	Zeroized when key context is closed
SP 800-90A DRBG internal values (C, V values)	Derived from the seed and entropy input using SP800-90A mechanisms	N/A		
TLS Pre-Master Secret	Generated by TLS client as output from DRBG when using RSA key exchange.	Entry: if received by module as TLS server, wrapped with server's public RSA key; otherwise no entry. Output: if generated by module as TLS client, wrapped with server's public RSA key; otherwise, no output.	application memory	Zeroized when key context is closed
TLS Master Secret	Derived from pre-master using TLS KDF	N/A	application memory	Zeroized when key context is closed

7. EMI/EMC

EMI/EMC properties of System SSL are not meaningful for the library itself. Systems utilizing the System SSL library services have their overall EMI/EMC ratings determined by the host system. The tested platform has FCC Class A ratings.

8. Self-Tests

8.1 System SSL Module

The System SSL library implements a number of self-tests to check proper functioning of the module including power-up self-tests and conditional self-tests. Conditional tests are performed when asymmetric keys are generated and for the DRBG. These tests include pair-wise consistency tests of the generated DSA or RSA keys, and comparing every newly-generated RNG block with the previously-generated one

8.2 Startup Self-Tests

“Power-up" self-tests consist of software integrity test(s) and known-answer tests of algorithm implementations listed below. The module integrity test is automatically performed during loading. If the integrity test fails, the module will terminate the loading process. The module cannot be used in this state. If any of the known answer tests fail, the module is rendered unusable (all cryptographic services return an error return code). Any attempts to use the module will fail.

The integrity of the module is verified by checking a HMAC-SHA-256-based hash value of each module binary prior to being utilized. Initialization will only succeed if all utilized module hash values are verified successfully. Module hash values are generated during the final phase of the build process. The integrity check for CPACF is a CRC32 check.

Algorithm known answer tests (KAT) are invoked automatically upon loading the System SSL module. The initialization function is executed via DEP (default entry point) as specified in FIPS 140-2 Implementation Guidance 9.10.

Prior to the execution of the power-up self-tests, the System SSL module checks whether environment variable GSK_HW_CRYPTO has been set. If not set, AES, TDES, SHA-1 and SHA-2 KAT tests are performed using the CPACF. If GSK_HW_CRYPTO is set, AES, TDES, SHA-1 and SHA-2 CPACF cryptographic algorithms can be disabled for use by the System SSL through bit settings within the specified value. If the cryptographic algorithm has been disabled, the KAT is run against the software implementation within the System SSL module. Only one algorithm implementation is supported for the entire instance of the System SSL module.

Note: Additionally the module also performs Diffie Hellman shared secret computation KAT but it is a non-approved algorithm

The module tests the following cryptographic algorithms:

CPACF: AES encryption/decryption, Triple DES encryption/decryption, SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512.

System SSL module software: AES encryption/decryption, AES-GMAC, Triple DES encryption/decryption, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, RSA (2048-bit key sign/verify, wrapping/unwrapping), DSA (2048-bit prime sign/verify), HMAC-SHA-1, HMAC-SHA256 and HMAC-SHA384, and DRBG.

Self-tests are performed in logical order, verifying library integrity incrementally:

1. Integrity test on library, using HMAC-SHA-256

2. Known-answer tests on algorithms, from integrity-verified binary.

The integrity check process covers all constituent DLLs. DLLs are individually hashed and verified.

8.3 Startup Recovery

If the integrity fails, System SSL will terminate the loading of the module needed for the FIPS 140-2 processing. If any of the known answer tests fail, the module is render unusable (all cryptographic services return an error return code). The System SSL element's calling application must recognize this error and handle it in a FIPS 140-2 appropriate manner, for example, by reinitializing the module instance.

8.4 Conditional Self-Testing

Conditional self-testing includes continuous DRBG testing. Continuous DRBG testing involves comparing every newly-generated RNG block with the previously-generated one. The first output block generated by DRBG is used only for the purpose of initiating the continuous DRBG test. The test fails if the DRBG outputs the same value twice subsequently.

If the DRBG outputs identical, subsequent pseudo-random blocks, it enters an error state and returns the corresponding status. The calling application must recognize this error and handle it in a FIPS 140-2 appropriate manner, for example, by restarting z/VM System SSL.

8.5 Pair-wise Consistency Checks

This test is run whenever the module generates a private-public key-pair. The private key structure always contains either the data of the corresponding public key or information sufficient for computing the corresponding public key.

If the pair-wise consistency check fails, the module enters an error state and returns an error status code. The calling application must recognize this error and handle it in a FIPS 140-2 appropriate manner, for example, by reinitializing the library instance.

8.6 Invoking FIPS 140-2 self-tests on demand

If a user can access System SSL services, the library has passed its integrity and power-up self tests. On-demand self-tests can be invoked by reloading the module.

If a KAT failure is encountered, the module enters an error state and returns an error status code. The calling application must recognize this error and handle it in a FIPS 140-2 appropriate manner, for example, by reinitializing the library instance. The error state is indicated by a return code describing the error. No cryptographic services are available in the error state.

9.Operational Requirements (Officer/User Guidance)

9.1 Module Configuration for FIPS 140-2 Compliance

To verify FIPS 140-2 compliant usage, the following requirements must be observed:

- The Operating System (OS) hosting the library must be set up in accordance with **FIPS Pub 140-2** rules. It must provide sufficient separation between processes to prevent inadvertent access to data of different processes. (This requirement was met for all platforms tested during validation.)

- An instance of the module must not be used by multiple callers simultaneously such that they might interfere with each other. Note that for keys retained in caller-provided storage, this requirement is automatically met if the OS provides sufficient process separation (since the ownership of each memory region, therefore, each object, is uniquely determined.)

- Applications using System SSL services must verify that ownership of keys is not compromised, and keys are not shared between different users of the calling application.

Note that this requirement is not enforced by the System SSL library itself, but by the application providing the keys to System SSL.

- Applications utilizing System SSL services must avoid using key sizes and algorithms that are disallowed after the transition period as stated in the NIST SP 800-131A Revision 2 .

- To be in a FIPS 140-2 compliant state, the System SSL installation must run on a host with commercial grade components and must be physically protected as prudent in an enterprise environment.

- According to IG A.13, the same Triple-DES key shall not be used to encrypt more than 2^{20} 64-bit blocks of data.

    - **Physical assumptions**
        - The module is intended for application in user areas that have physical control and monitoring. It is assumed that the following physical conditions will exist:
            - **LOCATION**
                - The processing resources of the module will be located within controlled access facilities that will prevent unauthorized physical access.
            - **PROTECTION**
                - The module hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.
    - **Personnel assumptions**
        - It is assumed that the following personnel conditions will exist:
            - **MANAGE**
                - There will be one or more competent individuals assigned to manage the module and the security of the information it contains.
            - **NO EVIL ADMINISTRATOR**
                - The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.
            - **CO-OPERATION**
                - Authorized users possess the necessary authorization to access at least some

of the information managed by the module and are expected to act in a cooperative manner in a benign environment.

9.2 Testing/Physical Security Inspection Recommendations

In addition to automatic tests, which are described elsewhere in this document, System SSL users may invoke FIPS 140-2 mode self-tests at any time. These self-tests are initiated through a dedicated function which is invoked automatically at startup. Continuous tests reside within their respective functions and are called implicitly during the function processing. These tests are not observable unless a failure is detected.

Apart from prudent security practice of server applications and those of security-critical embedded systems, no further restrictions are placed on hosts utilizing these services.

10. Mitigation of Other Attacks

The Mitigation of Other attacks security section of FIPS 140-2 is not applicable to the System SSL cryptographic module.

11. Glossary

CP Control Program. A component of z/VM that manages the resources of a single computer so that multiple computing systems appear to exist. Each apparent system, or virtual machine, is the functional equivalent of the real computer, and CP simulates the real machine architecture in the virtual machine.

CPACF CP Assist for Cryptographic Function, clear key on-chip accelerator integrated into mainframe processors. CPACF functionality is restricted to symmetric and hashing operations.

DLL Dynamic Link Library, shared program library instantiated separately from binaries using it. FIPS 140-2 configurations of System SSL DLLs are never statically linked.

DRNG Deterministic Random Number Generator, a deterministic function expanding a “true random” seed to a pseudo-random sequence.

Enclave In the z/VM Language Environment, a collection of routines, one of which is named as the main routine. The enclave contains at least one thread. Multiple enclaves may be contained within a process.

IPL Initial Program Load

KAT Known Answer Test

OS Operating System

Process A collection of resources; both program code and data, consisting of at least one enclave.

SDO Prepackaged version of the z/VM Operating System

Side deck The functions and variables that can be imported by DLL applications.

Thread An execution construct that consists of synchronous invocations and terminations of routines. The thread is the basic run_time path within the z/VM Language Environment program management model, and is dispatched by the operating system with its own run-time stack, instruction counter and registers. A thread may exist concurrently with other threads within a virtual machine.

TRNG True Random Number Generator, a service that extracts cryptographically-useful random bits from non-deterministic (physical) sources. The “random seed” bits are post-processed by a DRNG.

Virtual Device The simulation of a device by CP.

Virtual Machine The virtual processors, virtual storage, and virtual devices that CP allocates to a single user. A virtual machine also includes any expanded storage dedicated to it.

Virtual Processor A representation of a processor that is dispatched by CP on a real processor. It includes the contents of all registers and the state of the processor.

Virtual Storage Storage space that can be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses.

12. References

[1] National Institute of Standards and Technology, Security Requirements for Cryptographic Modules (FIPS 140-2), 2002

[2] American National Standard Institute, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (X9.31), 1998

[3] National Institute of Standards and Technology, Digital Signature Standard (DSS) (FIPS 186-2), 2000

[4] National Institute of Standards and Technology, Digital Signature Standard (DSS) (FIPS 186-4), 2013

[5] National Institute of Standards and Technology, Secure Hash Standard (FIPS 180-4), 2015

[6] National Institute of Standards and Technology, Special Publication 800-131aR2: Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, 2019

[7] National Institute of Standards and Technology, Advanced Encryption Standard (FIPS 197), 2001

13. Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- z14
- z13
- z196
- zEC12
- z/VM
- LinuxONE Emperor