Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library

Module version 513b121d8d789b1e5a7fd22743994650a94b222d108c33b0d82c98ff282bac64

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.1

Last update: 08-26-2024

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

www.atsec.com

1 Table of Contents

1 General

1.1 This Security Policy Document

This Security Policy describes the features and design of the module named Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library using the terminology contained in the FIPS 140-3 specification. The FIPS 140-3 Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CCCS Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-3. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

The Security Policy document is one document in a FIPS 140-3 Submission Package. In addition to this document, the Submission Package contains:

- The validation report prepared by the lab.
- Other supporting documentation and additional references.

This Non-Proprietary Security Policy may be reproduced and distributed, but only whole and intact and including this notice. Other documentation is proprietary to their authors.

1.2 How this Security Policy was Prepared

The vendor has provided the non-proprietary Security Policy of the cryptographic module, which was further consolidated into this document by atsec information security together with other vendor-supplied documentation. In preparing the Security Policy document, the laboratory formatted the vendor-supplied documentation for consolidation without altering the technical statements therein contained. The further refining of the Security Policy document was conducted iteratively throughout the conformance testing, wherein the Security Policy was submitted to the vendor, who would then edit, modify, and add technical contents. The vendor would also supply additional documentation, which the laboratory formatted into the existing Security Policy, and resubmitted to the vendor for their final editing.

This document is the non-proprietary FIPS 140-3 Security Policy for the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library. It has a one-to-one mapping to the [SP800-140B] starting with section B.2.1 named “General” that maps to section 1 in this document and ending with section B.2.12 named “Mitigation of other attacks” that maps to section 12 in this document.

ISO/IEC 24759 Section 6. [Number Below]	FIPS 140-3 Section Title	Security Level
1	General	1
2	Cryptographic Module Specification	1
3	Cryptographic Module Interfaces	1
4	Roles, Services, and Authentication	1

5	Software/Firmware Security	1
6	Operational Environment	1
7	Physical Security	2
8	Non-invasive Security	N/A
9	Sensitive Security Parameter Management	1
10	Self-tests	1
11	Life-cycle Assurance	2
12	Mitigation of Other Attacks	1
Overall		**1**

Table 1 - Security Levels

2 Cryptographic Module Specification

2.1 Module Description

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is used by secure applications. It is part of the common library and provides APIs to the secure applications for cryptography and hashing functions.

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library uses the Arm® v8 instruction set architecture for hash operations for SHA-1, SHA-224 and SHA-256.

2.2 Module Details

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library cryptographic module (hereafter referred to as “the module”) is a hybrid software Single-Chip cryptographic module that consists of components listed in the table below. The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is bound to the on-chip Pseudo Random Number Generator module with version 3.1.0 validated to FIPS 140-3 under Cert. #4778. The bound module resides within the same physical perimeter of the binding module.

Component	Type	Version Number	Operating System
Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library (64 bit)	Hybrid software	513b121d8d789b1e5a7fd2 2743994650a94b222d108c 33b0d82c98ff282bac64	Qualcomm TEE TZ.XF.5.24
ARMv8 processor[1]	Hardware	513b121d8d789b1e5a7fd2 2743994650a94b222d108c 33b0d82c98ff282bac64	N/A
TZ_SW_CRYPTO_FIPS_ENABLE fuse with value of 1[2]			

Table 2 - Components of the Hybrid Software Cryptographic Module

[1] The ARMv8.5-a is the instruction set version used within the Snapdragon 8 Gen 2 Mobile Platform
Snapdragon is a product of Qualcomm Technologies, Inc. and/or its subsidiaries.
Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

[2] The TZ_SW_CCRTPTO_FIPS_ENABLE fuse will enable FIPS compliance for Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library. Disabled by default and blow to enable.

2.3 Tested Operational Environments

The module has been tested on the operational environments indicated in Table 3 with the corresponding module variants and configuration options.

#	Operating System	Hardware Platform	Processor	PAA/Acceleration
1	Qualcomm TEE TZ.XF.5.24	Snapdragon 8 Gen 2 Mobile Platform	Snapdragon 8 Gen 2 Mobile Platform	ARMv8 instruction set architecture (SHA-1, SHA-224 and SHA-256)

Table 3 - Tested operational environments

2.4 Security Functions

Table 4 lists all approved security functions (cryptographic algorithms) of the module, including specific key lengths employed for approved services, and implemented modes or methods of operation of the algorithms.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A2940	AES FIPS 197, SP800-38A, SP800-38C, SP800-38E	CBC, ECB, CTR, CCM, CFB128, XTS, OFB	128, 192, 256 bits (CBC, ECB, CTR, CCM, CFB128, OFB) 128, 256 bits (XTS)	Encryption, Decryption
#A2940	AES SP800-38A Addendum	CBC-CS2	128, 192, 256 bits	Encryption, Decryption
#A2940	SHA-1 (ARMv8) FIPS 180-4	N/A	N/A	Hash
#A2940	SHA-224 (ARMv8) FIPS 180-4	N/A	N/A	Hash
#A2940	SHA-256 (ARMv8) FIPS 180-4	N/A	N/A	Hash
#A2940	SHA-384 (software) FIPS 180-4	N/A	N/A	Hash

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A2940	SHA-512 (software) FIPS 180-4	N/A	N/A	Hash
#A2940	HMAC FIPS 198-1	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	Key sizes are between 112-4096 bits in length 112-256 bits of key strength	Message Authentication
#A2940	ECDSA Key Pair Generation FIPS 186-4 SP800-133rev2 Section 4 without V (CKG)	B.4.2 (Testing Candidates)	112 – 256 bits of security strength P-224, P-256, P-384, P-521	Key Pair Generation
#A2940	ECDSA Signature Generation FIPS 186-4	SHA-224, SHA-256, SHA-384, SHA-512	112 – 256 bits of security strength P-224, P-256, P-384, P-521	Signature Generation
#A2940	ECDSA Signature Verification FIPS 186-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	96 – 256 bits of security strength P-192, P-224, P-256, P-384, P-521 (ECDSA SigVer with P-192 is a legacy algorithm)	Signature Verification
#A2940	ECDSA Signature Generation – Component (CVL) FIPS 186-4	N/A	112 – 256 bits of security strength P-224, P-256, P-384, P-521	Signature Generation Component
#A2940	RSA Key Pair Generation FIPS 186-4 SP800-133rev2 Section 4 without V (CKG)	B.3.3 Probable Prime Generation	112-149 bits of security strength 2048, 3072, 4096 bit modulus	Key Pair Generation

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A2940	RSA Signature Generation (PKCS#1 V1.5) FIPS 186-4	SHA-224, SHA-256, SHA-384- SHA-512	112-149 bits of security strength 2048, 3072, 4096 bit modulus	Signature Generation
#A2940	RSA Signature Verification (PKCS#1 V1.5) FIPS 186-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	80-149 bits of security strength 1024, 2048, 3072, 4096 bit modulus (RSA SigVer with a modulus length of 1024 is a legacy algorithm)	Signature Verification
#A2940	RSA Signature Generation (PSS) FIPS 186-4	SHA-224, SHA-256, SHA-384, SHA-512	112-149 bits of security strength 2048, 3072, 4096 bit modulus	Signature Generation
#A2940	RSA Signature Verification (PSS) FIPS 186-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	80-149 bits of security strength 1024, 2048, 3072, 4096 bit modulus (RSA SigVer with a modulus length of 1024 is a legacy algorithm)	Signature Verification
#A2940	RSA Signature Generation – Primitive (CVL) FIPS 186-4	N/A	112 bits of security strength 2048 bit modulus	Signature Generation Primitive
#A2940	PBKDF2 SP800-132 (Option 1b)	SHA-1, SHA-256, SHA-512	128-256 bits	Key Derivation
Vendor Affirmed	CKG SP800-133rev2 Section 4 without V	RSA	2048, 3072, 4096 bit modulus 112 – 256 bits of security strength	Key Pair Generation

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
		ECDSA	P-224, P-256, P-384, P-521 112 – 256 bits of security strength	
Pseudo Random Number Generator bound module (FIPS 140-3 certificate #4778)				
#A2945 and #A2949	SHA-256 (bound) FIPS 180-4	N/A	N/A	Hash
#A2945	Hash DRBG SP800-90Arev1	SHA-256	N/A	Random Number Generation

Table 4 – Approved Algorithms

Table 5 lists all non-approved security functions not allowed in approved services of the module.

Algorithm/Functions	Use/Function
DES	Encryption, Decryption
Triple DES[3]	Encryption, Decryption
GCM/GMAC[4]	Encryption, Decryption, Message Authentication
HMAC (key sizes below 112 bits)	Message Authentication
RIPEMD-160	Hash
MD5	Hash
SM2	Signature Generation, Signature Verification, Hybrid Encryption, Hybrid Decryption
SM3	Hash

[3] Triple DES is CAVP certified with CAVP Cert. #A2940. However, there are two requirements from FIPS 140-3 IG C.G below that contribute to the non-compliance: 1) FIPS 140-3 requires that only 2^16 encryptions are performed with a given key; 2) the aforementioned requirement must be enforced by the module itself, not by policy.

[4] GCM is CAVP certified with CAVP Cert. #A2940. However, there are two requirements from FIPS 140-3 IG C.H below that contribute to the non-compliance: 1) the IV uniqueness must be enforced by the module; 2) FIPS 140-3 requires that only 2^32 cipher operations are performed with a given key.

SM4	Encryption, Decryption
SHA-1, SHA-224 and SHA-256 (software)	Hash
ECDSA (secp160r1, P-192)	Key Pair Generation, Signature Generation
ECDSA (secp160r1)	Signature Verification
ECDSA (P-192, P-224, P-256, P-384 and P-521)	Signature Verification – Component
Elliptic Curve Integrated Encryption Scheme (ECIES)	Hybrid Encryption, Hybrid Decryption
RSA-OAEP	Key Wrapping
RSA (1024 bit modulus)	Key Pair Generation, Signature Generation
Ed25519	Key Pair Generation, Signature Generation, Signature Verification
ECDH[5]	Shared Secret Computation
HKDF	Key Derivation

Table 5 – Non-Approved Algorithms Not Allowed in Approved Services

NOTE: There are no non-approved algorithms allowed in approved mode, and no non-approved algorithms allowed in the approved mode with no security claimed.

2.5 Description of Modes of Operation

The module implements two modes of operation: (1) the approved mode, in which the approved services are available; and (2) the non-approved mode, in which the non-approved services are available. The current mode of operation of the module can be inferred by the service indicator, which indicates the approved state of the current service being invoked. No configuration is necessary for the module to operate and remain in the approved or non-approved modes. All SSPs are kept separate between the two modes.

After the module successfully passes the pre-operational integrity self-test, the module is in the approved mode. If the operator requests a non-approved service, the module implicitly switches to the non-approved mode of operation. When in the non-approved mode of operation, if the operator requests an approved service, the module implicitly switches to the approved mode of operation.

Table 8 and 9 list the services available in approved and non-approved mode of operation, respectively.

[5] The ECDH has been tested with CAVP certificate #A2940. However, the shared secret generation does not check the key assurance requirements from SP800-56A Rev 3 around trusted third parties during key import. There is a self-test for ECDH but is not listed since it is non-approved.

2.6 Cryptographic Module Boundary

The physical perimeter of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is the physical perimeter of the device that contains it. Consequently, the embodiment of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is a single-chip cryptographic module. Figure 1 shows a block diagram of the module, with the cryptographic boundary indicated in red, and the physical perimeter in black.

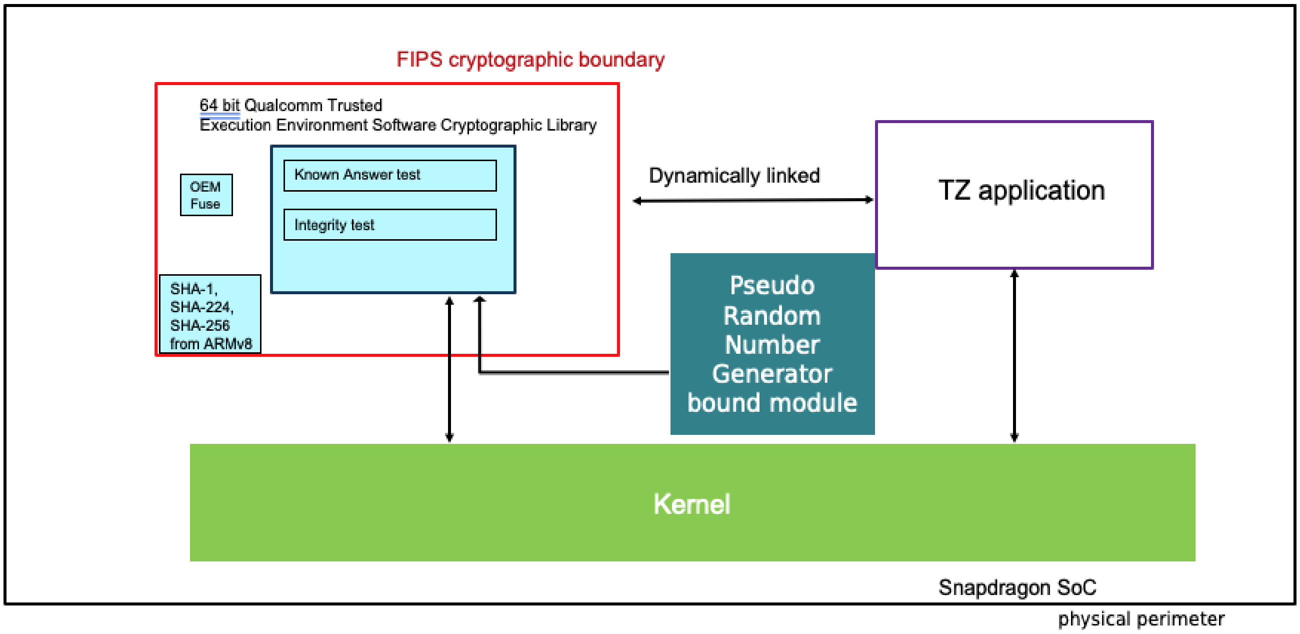

Figure 1 – Block diagram depicting the cryptographic boundary and physical perimeter, and data flow between the components in the Snapdragon SoC

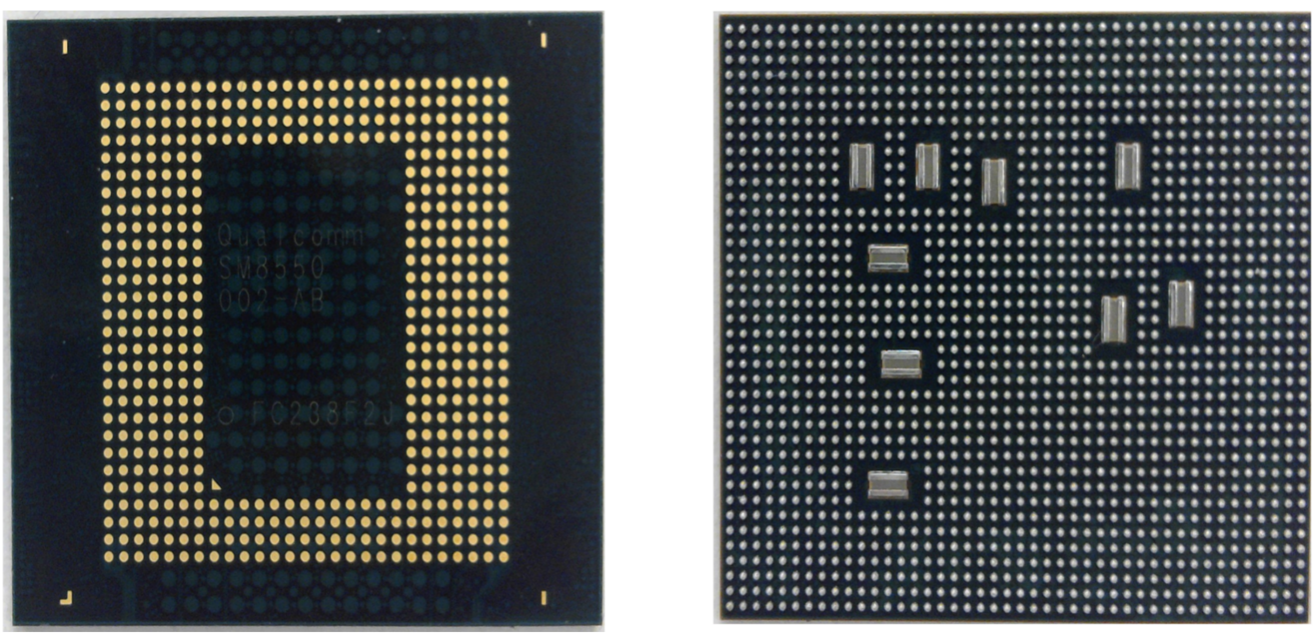

Figure 2: Snapdragon *8 Gen 2 Mobile Platform*

The TOEPP (tested operational environment's physical perimeter) of the module is the entire single chip, the Snapdragon 8 Gen 2 Mobile Platform.

2.7 Rules of Operation

The Crypto Officer interacts with the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library in two distinct ways:

1. Initializing the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library
2. The application services (API's) invoked by users

Once Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library initializes and the self-tests complete successfully, all cryptographic functions are made available. See section 10.3 for error states and error recovery.

Caller-induced or internal errors do not reveal any sensitive material to callers. The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library ensures that there is no means to obtain data from itself by performing key zeroization. There is no means to obtain sensitive information from the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library.

3 Cryptographic Module Ports and Interfaces

Physical port	Logical Interface	Data that passes over port/interface
N/A	Data Input	Input parameters of API calls for data
	Data Output	Output parameters of API calls for data
	Control Input	Function calls, input parameters for control
	Status Output	Return code, status values
Physical power connector	Power Input	Power port or pin for single-chip

Table 6 – Ports and Interfaces

Table 6 summarizes the cryptographic module interfaces. The logical interfaces are logically separated from each other by the API design. All status ports and control ports are directed through the interface of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library's cryptographic boundary, which is its software APIs. The power interface is physically separated from any other interface. The module does not implement a control output interface.

4 Roles, services, and authentication

4.1 Roles

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library supports the Crypto Officer role. The role is implicitly assumed based on the services requested.

Table 7 lists the roles supported by the module with corresponding services with input and output.

Role	Service	Input	Output
From module			
Crypto Officer	Encryption	Key, Plaintext	Ciphertext, Success/Fail
	Decryption	Key, Ciphertext	Plaintext, Success/Fail
	Hash	Input data	Hash value
	Message Authentication	HMAC key, Input data	HMAC value
	Key Pair Generation	Key size	Key pair (public key + private key)
	Signature Generation	Private key, Input data, Hash algorithm	Signature
	Signature Verification	Public key, Input data, Signature, Hash algorithm	Success/Fail
	Signature Generation – Component or Primitive	Private key, Pre-hashed data	Signature
	Password Based Key Derivation	PRF algorithm, Salt, Iteration count, Password	Derived key
	Random Number Generation	Output length	Random bytes
	Get FIPS Info	enum value of MODULE_HMAC	Versioning information Self-test Success/Fail
	Show Status	None	Current status (as return codes and/or log messages)
	Zeroization	None	None
	Hybrid Encryption	Key, Plaintext	Ciphertext, Success/Fail

	Hybrid Decryption	Key, Ciphertext	Plaintext, Success/Fail
	Signature Verification – Component	Public key, Input data, Signature, pre-hashed data	Success/Fail
	Key Wrapping	Key wrapping key, key to be wrapped	Wrapped key
	Shared Secret Computation	Private key, public key from peer	Shared secret

Table 7 – Roles, Service Commands, Input and Output

4.2 Authentication

The module does not support authentication for roles.

4.3 Services

The module provides services to operators that assume the available role. Services are accessed through documented API interfaces from the calling application.

Additional services are provided by the bound Pseudo Random Number Generator module on the Snapdragon 8 Gen 2 Mobile Platform SoC. This Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library utilizes the random number generation service from the bound Pseudo Random Number Generator module.

The next tables define the services that utilize approved, allowed, and non-approved security functions in this module. For the respective tables, the convention below applies when specifying the access permissions (types) that the service has for each SSP.

- **Generate (G)**: The service establishes the SSP by generation, agreement, or derivation.
- **Read I**: The SSP exists in the module and is read by the service and may be output.
- **Write (W)**: The caller provides the SSP to the service to be imported into the module; written; or updated if the SSP already exists in the module.
- **Execute (E) (or use)**: The service uses the SSP in performing a cryptographic operation. Other access types identify the provenance of the SSP.
- **Zeroize (Z)**: The service zeroizes the SSP.
- **N/A**: The service does not access any SSP or key during its operation.

An operator can read the service indicator from a service by invoking the qsee_get_fips_approval_status() function with enum value for QSEE_FIPS_CRYPTO_SVC_TYPE. For details on the enum values please see the product documentation.

4.3.1 Approved Services

Table 8 lists the approved services in this module, the roles that can request the service, the algorithms involved, the Sensitive Security Parameters (SSPs) involved and how they are accessed, and the respective service indicator.

In the service tables, CO specifies the Crypto Officer role.

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights	Indicator
Encryption	Encrypts data using symmetric cryptography	AES	AES key	CO	W, E	0 return value with enum QSEE_FIPS_AES_*
Decryption	Decrypts data using symmetric cryptography	AES	AES key	CO	W, E	0 return value with enum QSEE_FIPS_AES_*
Hash	Computes the hash value of data	SHA-1 (ARMv8) SHA-224 (ARMv8) SHA-256 (ARMv8) SHA-384 (software) SHA-512 (software)	N/A	CO	N/A	0 return value with enum QSEE_FIPS_SHA*
Message Authentication	Computes the HMAC value of data	HMAC	HMAC key	CO	W, E	0 return value with enum QSEE_FIPS_HMAC*
Key Pair Generation	Generates asymmetric key pairs using the bound DRBG	ECDSA Key Pair Generation, CKG	ECDSA private key	CO	G, R	0 return value with enum QSEE_FIPS_ECDSA_KEY_PAIR_GEN_*
			ECDSA public key		G, R	
			Intermediate key generation value		G, E, Z	
		RSA Key Pair Generation, CKG	RSA private key		G, R	0 return value with enum QSEE_FIPS_RSA_KEY_PAIR_GEN_*
			RSA public key		G, R	
			Intermediate key generation value		G, E, Z	
Signature Generation	Generates cryptographic signatures of data	ECDSA Signature Generation	ECDSA private key	CO	W, E	0 return value with enum QSEE_FIPS_ECDSA_SIG_GEN_*
		RSA Signature Generation (PKCS#1 V1.5)	RSA private key			0 return value with enum

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights	Indicator
		RSA Signature Generation (PSS)				QSEE_FIPS_RSA_SIG_GEN_*
Signature Verification	Verifies cryptographic signatures of data	ECDSA Signature Verification	ECDSA public key	CO	W, E	0 return value with enum QSEE_FIPS_ECDSA_SIG_VER_*
		RSA Signature Verification (PKCS#1 V1.5)	RSA public key			0 return value with enum QSEE_FIPS_RSA_SIG_VER_*
		RSA Signature Verification (PSS)				
Signature Generation – Component or Primitive	Generates cryptographic signatures of pre-hashed data	ECDSA Signature Generation Component	ECDSA private key	CO	W, E	0 return value with QSEE_FIPS_ECDSA_SIG_GEN_COMP_*
		RSA Signature Generation Primitive	RSA private key			0 return value with enum QSEE_FIPS_RSA_SIG_GEN_PRIMITIVE_*
Password Based Key Derivation	Derives a secret key	PBKDF2	Password, salt	CO	W, E	0 return value with enum QSEE_FIPS_PBKDF_*
			Derived key		G, R	
Random Number Generation	Generates random bytes	Hash_DRBG provided by the bound module, which uses SHA in bound module	Entropy input W, seed and internal state G[6]	CO	N/A	qsee_prng_getdata returns positive value
Miscellaneous						
Show Status	Show the status of the module	None	N/A	CO	N/A	N/A

[6] The SSPs can only be accessed by the bound module and hence are not listed in table 10

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights	Indicator
Get FIPS Info	Show the versioning information of the module and execute self-tests on demand (pre-operational self-tests and HMAC CAST)	None	N/A	CO	N/A	N/A
Zeroization	Zeroizes all SSPs in the module	None	All SSPs	CO	Z	N/A

Table 8 - Approved Services

4.3.2 Non-approved Services

Table 9 lists the non-approved services that utilize the non-approved security functions listed in Table 5.

Service	Description	Algorithms Accessed	Role	Indicator
Encryption	Encrypts data using symmetric cryptography	DES, Triple DES, GCM, SM4	CO	N/A
Decryption	Decrypts data using symmetric cryptography	DES, Triple DES, GCM, SM4	CO	N/A
Hybrid Encryption	Encrypts data using hybrid cryptography	SM2, ECIES	CO	N/A
Hybrid Decryption	Decrypts data using hybrid cryptography	SM2, ECIES	CO	N/A
Hash	Computes the hash value of data	RIPEMD-160, MD5, SM3, SHA-1, SHA-224 and SHA-256 (software)	CO	N/A
Message Authentication	Computes the MAC value of data	GMAC, HMAC (key sizes below 112 bits)	CO	N/A
Key Pair Generation	Generates asymmetric key pairs	ECDSA (secp160r1, P-192) RSA (1024-bit modulus) Ed25519	CO	N/A
Signature Generation	Generates cryptographic signatures of data	ECDSA (secp160r1, P-192) RSA (1024-bit modulus) Ed25519 SM2	CO	N/A
Signature Verification	Verifies cryptographic signatures of data	ECDSA (secp160r1) Ed25519 SM2	CO	N/A

Service	Description	Algorithms Accessed	Role	Indicator
Signature Verification Component	Verifies cryptographic signatures of pre-hashed data	ECDSA	CO	N/A
Key Wrapping	Wraps a key using asymmetric cryptography	RSA OAEP	CO	N/A
Shared Secret Computation	Computes a shared secret	ECDH	CO	N/A
Key Derivation	Derive a key	HKDF	CO	N/A

Table 9 - Non-Approved Services

5 Software/Firmware security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC-SHA-256 value calculated at run time with the HMAC-SHA-256 value stored in the module that was computed at build time. If the comparison verification fails, the module transitions to the error state (Section 10.3). The HMAC-SHA-256 algorithm goes through its cryptographic algorithm self-test before the integrity test is performed (Table 11).

5.2 On-Demand Integrity Test

The software integrity test is performed as part of the pre-operational self-tests. The pre-operational self-tests can be invoked when Get_FIPS_Info service is called.

5.3 Executable Code

The module consists of code that will perform algorithmic services for trusted applications. The code is compiled into a shared library.

6 Operational Environment

6.1 Applicability

The procurement, build and configuring procedure are controlled. The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is installed into a commercial off-the-shelf (COTS) mobile device by the customer.

The software components of this module are executed in the Qualcomm Trusted Execution Environment (TEE). Therefore, the operational environment is considered limited.

6.2 Tested Operational Environments

Please see Section 2.3 for the tested operational environment.

6.3 Specifications of the Operational Environment

There are no security rules, settings or restrictions to the configuration of the operational environment.

- The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library does not have the capability of loading software or firmware from an external source.
- The module does not support concurrent operators.

7 Physical Security

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is a hybrid software module implemented as part of the Snapdragon 8 Gen 2 Mobile Platform SoC, which is the physical perimeter of the single-chip hybrid software module. The single-chip conforms to the Level 2 requirements for physical security.

At the time of manufacturing, the die of the Snapdragon 8 Gen 2 Mobile Platform SoC is embedded within a printed circuit board (PCB), which prevents visibility into the internal circuity of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library. The layering process which is used to embed the die into the PCB also prevents tampering of the physical components without leaving tamper evidence.

The Snapdragon 8 Gen 2 Mobile Platform SoC is further protected by being enclosed in commercial off the shelf mobile device utilizing production grade, commercially available components and said mobile device enclosure completely surrounds the Snapdragon 8 Gen 2 Mobile Platform SoC.

There are no steps required to ensure that physical security is maintained.

8 Non-invasive Security

The module does not support any non-invasive security techniques; therefore, this section is not applicable.

9 Sensitive Security Parameter Management

Table 10 summarizes the Sensitive Security Parameters (SSPs) that are used by the cryptographic services implemented in the module in the approved services (Table 8).

SSP	Strength	Security Function and Cert. #	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
AES key	128, 192, 256 bits	AES #A2940	N/A	Input in plaintext via API input parameters. No output.	N/A	RAM	When the module is powered off	Encryption, Decryption Related SSPs: N/A
HMAC key	112-256 bits	HMAC #A2940						Message Authentication Related SSPs: N/A
ECDSA private key	112-256 bits (P-224, P-256, P-384, P-521)	ECDSA #A2940	FIPS 186-4 compliant method described in Appendix B.4.2. random values obtained using the SP800-90Arev1 DRBG provided by the bound module.	Input in plaintext via API input parameters. Output in plaintext via API output parameters.	N/A	RAM		Signature Generation and Signature Generation component Related SSPs: paired with ECDSA public key, generated from Intermediate key generation value
ECDSA public key	96-256 bits (P-192, P-224, P-256, P-384, P-521)							Signature Verification Related SSPs: paired with ECDSA private key, generated from Intermediate key generation value

SSP	Strength	Security Function and Cert. #	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
RSA private key	112-149 bits (2048, 3072, 4096 bit modulus)	RSA #A2940	FIPS 186-4 compliant method described in Appendix B.3.3. random values obtained using the SP800-90Arev1 DRBG provided by the bound module.	Input in plaintext via API input parameters. Output in plaintext via API output parameters.	N/A	RAM		Signature Generation and Signature Generation primitive Related SSPs: paired with RSA public key, generated from Intermediate key generation value
RSA public key	80-149 bits (1024, 2048, 3072, 4096 bit modulus)							Signature Verification Related SSPs: paired with RSA private key, generated from Intermediate key generation value
Password, Salt	N/A	PBKDF2 #A2940	N/A	Input in plaintext via API input parameters. No output.	N/A	RAM	When the module is powered off	Password Based Key Derivation Related SSPs: used to derive Derived key
Derived key	128 – 256 bits		Password Based Key Derivation	No input. Output in plaintext via API output parameters.	N/A			Password Based Key Derivation Related SSPs: derived from Password, salt

SSP	Strength	Security Function and Cert. #	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
Intermediate key generation value	112-256 bits	ECDSA #A2940, RSA #A2940	During ECDSA key generation and RSA key generation	No input No output	N/A	RAM	Automatically	Key Pair Generation Related SSPs: used to generate ECDSA public key, ECDSA private key, RSA public key, RSA private key

Table 10 - SSPs

9.1 SSP Establishment/SSP Derivation

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library implements Password-Based Key Derivation version 2 (PBKDF2) as defined in [SP800-132]. The PBKDF2 function is provided as a service and returns the key derived from the provided password to the caller. The supported option is 1a from Section 5.4 of SP 800-132, whereby the Master Key (MK) is used directly as the Data Protection Key (DPK). The keys derived from passwords, as shown in SP 800-132, may only be used for storage applications.

9.2 SSP Generation

The SSP generation methods implemented in the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library for approved services are compliant with SP 800-133Rev2. EC key pair generation is performed according to Appendix B.4.2 of FIPS 186-4 (Testing Candidates) and corresponds to keys used for ECDSA operations. RSA key pair generation is performed according to Appendix B.3.3 of FIPS 186-4 (Probable Prime Generation). The seeds (i.e., the random values) used in asymmetric key pair generation are directly obtained from the SP 800-90Arev1 Hash DRBG provided by the bound Qualcomm® Pseudo Random Number Generator module, compliant with SP 800-133r2 section 4 without the use of V (as specified in additional comment #2 to IG D.H).

- The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library does not generate symmetric keys.
- Intermediate key generation values are not output from the module during or after processing the service.

9.3 SSP Entry and Output

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library only supports manual distribution and electronic entry for SSPs. The SSPs are provided to the module via API input parameters in plaintext form and output via API output parameters in plaintext form. The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library does not enter or output SSPs in plaintext format outside its physical perimeter.

9.4 SSP Storage

All SSPs are output from and input to the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library through the calling process and are destroyed from memory when released. The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library does not persistently store SSPs. The SSPs are stored temporarily in plaintext in the RAM.

9.5 SSP Zeroization

The module’s functions deallocate and zeroize temporary SSP values in volatile memory used during the function’s execution. The zeroization consists of writing zeroes to the memory location used by the SSP before deallocating the area. The module does not overwrite SSPs with another SSP.

The zeroization service for the SSPs in volatile memory consists of powering off the module, which will remove power from the volatile memory. This action will cause the value of the SSPs in volatile memory to be overwritten by random values the next time the module is powered on. The successful act of powering off the module serves as the implicit indicator of zeroization.

10 Self-tests

All the self-tests are listed in Table 11, with the respective condition under which those tests are performed. The self-tests for the DRBG and SHA used from the bound module are implemented by the bound module.

Algorithm	Parameters	Condition for test	Type	Test
HMAC	SHA-1, SHA-256, SHA-512	Power up	Cryptographic Algorithm Self-Test	KAT HMAC computation
HMAC-SHA-256	SHA-256	Power up (after HMAC CASTs)	Pre-Operational Self-Test	Software integrity test
AES 256 key size	CCM	Before first use	Cryptographic Algorithm Self-Test	KAT encryption
				KAT decryption
	ECB	Before first use	Cryptographic Algorithm Self-Test	KAT decryption
RSA	PKCS#1 V1.5 with SHA-256 and 2048 bit modulus	Before first use	Cryptographic Algorithm Self-Test	KAT signature generation
				KAT signature verification
ECDSA	P-256 with SHA-256	Before first use	Cryptographic Algorithm Self-Test	KAT signature generation
				KAT signature verification
PBKDF2	SHA-1, SHA-256, SHA-512	Before first use	Cryptographic Algorithm Self-Test	KAT key derivation
RSA	PKCS#1 V1.5 with SHA-256	Key pair generation	Pair-wise Consistency Test	PCT signature generation/verification
ECDSA	SHA-256	Key pair generation	Pair-wise Consistency Test	PCT signature generation/verification

Table 11 - Self-tests

10.1 Pre-Operational Self-Tests

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library performs pre-operational self-tests when loaded into memory, without operator intervention. The pre-operational self-tests ensure that the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is not corrupted. The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library transitions to the operational state only after the pre-operational self-tests are passed successfully.

The types of pre-operational self-tests are described in the next sub-sections.

10.1.1Software Integrity Test

Section 5.1 describes the integrity test and the details if the integrity tests are defined in Table 11.

10.2 Conditional Self-Tests

10.2.1Cryptographic Algorithm Self-Tests

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library performs self-tests on all approved cryptographic algorithms as part of the approved services using the tests shown in Table 11. Data output through the data output interface is inhibited during the self-tests. The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library transitions to the operational state only after the cryptographic algorithm self-tests are passed successfully. The known answer test for DRBG is performed by the bound module.

10.2.2Pair-wise Consistency Tests

Pair-wise consistency tests are run whenever the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library generates an asymmetric (RSA or ECDSA) key pair using a SHA-256 hash.

10.2.3Periodic/On-Demand Self-Tests

The Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library performs on-demand self-tests initiated by calling the Get_FIPS_Info service. All self-tests in Table 11 marked as “Power up” are then executed. An operator can perform the pair-wise consistency tests on demand by requesting the Key Pair Generation service for RSA or ECDSA.

10.3 Error States

If the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library fails any of the self-tests, the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library enters the error state. In the error state, the data output interface is inhibited, and the module accepts no more inputs or requests. To recover from the error state, re-initialization is possible by successful execution of the pre-operational self-tests and cryptographic algorithm self-tests, which can be triggered by a power-off/power-on cycle.

Table 12 lists the error state and the status indicator (through calling the qsee_get_fips_info() function with the info_type parameter set to QSEE_FIPS_SELFTEST_STATUS) values that explains the error that has occurred.

Error State	Error Condition	Status Indicator
Error	Cryptographic Algorithm Self-Test, or	The module has halted and is

Error State	Error Condition	Status Indicator
	Software Integrity Test	unable to boot.
Error	Pair-wise Consistency Test	The module returns ICryptoSelfTest_CRYPTO_SELFTEST_FAILED_xxx and enters "Error" state and no further operations is allowed.

Table 12 - Error states

11 Life-cycle assurance

11.1 Configuration Management

Perforce Visual Client (P4V), a version control system from Perforce, is used to manage the revision control of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library software code. The Perforce Visual Client provides version control, branching and merging of code lines, and concurrent development.

Git, an open-source version control system, is also used to manage the revision control of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library unified crypto software code. Git provides version control, branching and merging of code lines, and concurrent development.

11.2 Delivery and Operation

The Snapdragon 8 Gen 2 Mobile Platform SoC is delivered from the vendor via a trusted delivery courier.

On the reception of the SoC, the operator shall first check all sides of the box to verify that it has not been tampered during the shipment. Then, after opening the box the operator shall verify that the moisture barrier bag is still sealed and does not present any trace of tampering. Finally, after retrieving the SoC, the operator shall perform a visual inspection of the external SoC package of the module, it should appear similar to the pictures in Section 2.6.

If one of these verifications fail, the operator shall contact their Qualcomm representative which released the delivery before operating the module.

Once the product is received by the customer, configured as defined in section 11.5, and powered up, the test defined in section 10 will be executed.

11.3 Maintenance Requirements

There are no maintenance requirements.

11.4 End of Life

As stated in Section 9.4, the module does not possess persistent storage of SSPs. The SSP values only exist in volatile memory and these values vanish when the module is powered off. The procedure for secure sanitization of the module at the end of life is simply to power it off, which is the action of zeroization of the SSPs (as specified in Section 9.5). As a result of this sanitization via power-off, all SSPs are removed from the module, so that the module may either be distributed to other operators or disposed.

11.5 Crypto Officer Guidance

To enable FIPS for the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library, the fuse must be set according to Table 2. The fuse enablement is mandatory to run as a FIPS validated module. This step needs to be performed only once during initial configuration.

The information required for the Crypto Officer to verify the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is provided by the qsee_get_fips_info() function in qsee_fips_services.h. To verify that a Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is FIPS certified, the Crypto Officer should verify the following:

- The HMAC of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library is on a list of HMACs of certified crypto modules.
  - This can be done by invoking the qsee_get_fips_info() function with the info_type parameter set to QSEE_FIPS_MODULE_HMAC (0). The buffer parameter should point to a buffer which is at least 32 bytes long, and the buffer_len parameter should be at least 32.
  - The result buffer should contain the HMAC-SHA-256 of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library.
  - To get the HMAC of the 64-bit Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library, this should be run from a 64-bit Trusted Application.
- The FIPS enablement fuse is blown.
  - This can be done by invoking the qsee_get_fips_info() function with the info_type parameter set to QSEE_FIPS_FUSE_STATUS (1). The buffer parameter should point to a 4-byte buffer (sizeof(uint32)) and the buffer_len parameter should equal 4.
  - The result buffer should contain the value QSEE_FIPS_FUSE_BLOWN (1).
- The pre-operational self-tests have passed.
  - This can be done by calling qsee_get_fips_info() with the info_type parameter set to QSEE_FIPS_SELFTEST_STATUS (2). The buffer parameter should point to a 4-byte buffer (sizeof(uint32)) and the buffer_len parameter should equal 4.
    - The result buffer contains an integer with below representation. Users should interpret the value correspondingly. When a self test for an algorithm is not run, the passed bit and the failed bit will be both 0. After a self test is run, it will be either passed or failed.
      - CRYPTO_SELFTEST_PASSED_AES (BIT0)
      - CRYPTO_SELFTEST_PASEED_ECC (BIT1)
      - CRYPTO_SELFTEST_PASSED_ECDH (BIT2)
      - CRYPTO_SELFTEST_PASSED_HMAC (BIT3)
      - CRYPTO_SELFTEST_PASSED_KDF (BIT4)
      - CRYPTO_SELFTEST_PASSED_RSA(BIT5)
      - CRYPTO_SELFTEST_PASSED_SHA(BIT6)
      - CRYPTO_SELFTEST_PASSED_TDES (BIT7)
      - CRYPTO_SELFTEST_FAILED_AES (BIT13)
      - CRYPTO_SELFTEST_FAILED_ECC (BIT14)
      - CRYPTO_SELFTEST_FAILED_ECDH (BIT15)
      - CRYPTO_SELFTEST_FAILED_HMAC (BIT16)
      - CRYPTO_SELFTEST_FAILED_KDF (BIT17)
      - CRYPTO_SELFTEST_FAILED_RSA (BIT18)
      - CRYPTO_SELFTEST_FAILED_SHA (BIT19)
      - CRYPTO_SELFTEST_FAILED_TDES (BIT20)
      - CRYPTO_SELFTEST_INTEGRITY_CHECK_PASSED (BIT29)

If one or more of the self-tests failed, the corresponding crypto service will not be exposed to the users.The operation of the Qualcomm® Trusted Execution Environment (TEE) Software

Cryptographic Library does not need FIPS 140-3 specific guidance. The FIPS 140-3 functional requirements are always invoked.

To use the cryptographic services of the Qualcomm® Trusted Execution Environment (TEE) Software Cryptographic Library, please refer to 80-NH537-4: Qualcomm Trusted Execution Environment Version 5.0 User Guide.

NOTES:

- In compliance with [SP 800-38E], the AES algorithm in XTS mode shall only be used for the cryptographic protection of data on storage devices, and the length of a single data unit encrypted with the AES-XTS shall not exceed 2^20 AES blocks. In compliance with IG C.I, the module performs a check to ensure that the two AES-XTS keys are different.
- The module supports option 1a from section 5.4 of [SP800-132] PBKDF, in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK). In compliance with [SP800-132] and IG D.N, the following requirements are met.
    - Keys derived from passwords 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 128 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 is 1000.
    - Passwords or passphrases, used as an input for the PBKDF2, shall not be used as cryptographic keys.
    - The length of the password or passphrase is at least 8 characters. For all numeric values, the probability of guessing the value is estimated to be 10^{-8}, which is greater than 2^{-112}.
- In compliance with IG C.F, the module supports 2048, 3072, 4096-bit RSA modulus lengths for RSA signature generation. All the RSA signature algorithm implementations have been tested for all implemented RSA modulus lengths. The minimum number of Miller-Rabin tests used in primality testing for key generation is consistent with Table B.1 of FIPS 186-4. The module supports RSA signature verification with 1024, 2048, 3072, 4096-bit modulus lengths, all of which have been CAVP tested.

12 Mitigation of other attacks

The elliptic curve implementation uses the Montgomery Ladder, as well as blinding of base points and private key multiplication. The RSA implementation uses base and modulus blinding to mitigate timing-based side-channel attacks. Blinding countermeasures add randomness to private key operations, making determination of secrets from observations more difficult for the attacker.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMT	Cryptographic Module Testing
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DF	Derivation Function
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
FIPS	Federal Information Processing Standards Publication
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
NDF	No Derivation Function
NIST	National Institute of Science and Technology
OFB	Output Feedback
O/S	Operating System
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS140-3 **FIPS PUB 140-3 - Security Requirements For Cryptographic Modules**
March 2019
https://doi.org/10.6028/NIST.FIPS.140-3

FIPS140-3_IG **Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program**
November 2023
https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf

FIPS180-4 **Secure Hash Standard (SHS)**
August 2015
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

FIPS186-4 **Digital Signature Standard (DSS)**
July 2013
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

FIPS197 **Advanced Encryption Standard**
November 2001
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS198-1 **The Keyed Hash Message Authentication Code (HMAC)**
July 2008
http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

PKCS#1 **Public Key Cryptography Standards (PKCS) #1: RSA Cryptography**
Specifications Version 2.1
February 2003
http://www.ietf.org/rfc/rfc3447.txt

SP800-38A **NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques**
December 2001
http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

SP800-38C **NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality**
May 2004
http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf

SP800-38E **NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices**
January 2010
http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf

SP800-57 **NIST Special Publication 800-57 Part 1 Revision 5 - Recommendation for Key Management Part 1: General**
May 2020
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf

SP800-90A **NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators**
June 2015
http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf

SP800-131A **NIST Special Publication 800-131A Revision 2- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths**
March 2019
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf

SP800-132 **NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation - Part 1: Storage Applications**
December 2010
http://csrc.nist.gov/publications/nistpubs/800-132/nist-sp800-132.pdf

SP800-133 **NIST Special Publication 800-133rev2 - Recommendation for Cryptographic**
Key Generation
June 2020
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf

SP800-140B **NIST Special Publication 800-140B - CMVP Security Policy Requirements**
March 2020
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140B.pdf