अंतरिम परीक्षण निर्देशिका
टीईसी ९१०११:२०२६

PROVISIONAL TEST GUIDE

TEC 91011:2026

for

क्वांटम-सुरक्षित क्रिप्टोग्राफिक प्रणालियाँ

Quantum Safe and Classical Cryptographic Systems

(TEC 91010:2026)

ISO 9001:2015

दूरसंचार अभियांत्रिकी केंद्र
खुर्शीदलाल भवन, जनपथ, नई दिल्ली-110001,भारत
TELECOMMUNICATION ENGINEERING CENTRE
KHURSHID LAL BHAWAN, JANPATH, NEW DELHI-110001, INDIA
www.tec.gov.in

Release 1: _____, 2026

FOREWORD

Telecommunication Engineering Centre (TEC) is the technical arm of Department of Telecommunications (DOT), Government of India. Its activities include:

- Framing of TEC Standards for Generic Requirements for a Product/Equipment, Standards for Interface Requirements for a Product/Equipment, Standards for Service Requirements & Standard document of TEC for Telecom Products and Services
- Formulation of Essential Requirements (ERs) under Mandatory Testing and Certification of Telecom Equipment (MTCTE)
- Field evaluation of Telecom Products and Systems
- Designation of Conformity Assessment Bodies (CABs)/Testing facilities
- Testing & Certification of Telecom products
- Adoption of Standards
- Support to DoT on technical/technology issues

For the purpose of testing, four Regional Telecom Engineering Centers (RTECs) have been established which are located at New Delhi, Bangalore, Mumbai, and Kolkata.

ABSTRACT

This Test Guide of testing pertains to detailed provisional test schedule and test procedure for evaluating conformance/functionality/requirements/ performance of standard on Quantum-Safe and Classical Cryptographic Systems as per standard no. TEC 91010:2026.

CONTENTS

A. HISTORY SHEET

Sl. No.	*Standard/document No.*	*Title*	*Remarks*
1.	TEC No. 91011:2026	Test Guide for Quantum-Safe Cryptographic Systems	First Issue ....., 2026

B. INTRODUCTION

This document describes the test schedule and procedures for evaluating the requirements of functionality / performance / operational / interface / interoperability / quality / safety / Electromagnetic compatibility, security services and performance under the Standard on Quantum Safe and Classical Cryptographic Systems against the Generic requirements as per the TEC GR No.: 91010:2026.

The manufacturer shall offer the system for type evaluation along with the following documents:

i. System specifications of the equipment containing features, facilities, and physical description,
ii. Installation, System and Operation & Maintenance manual of the equipment,
iii. Hardware, Software, and firmware details of the equipment,
iv. Bill of material,
v. Block schematic diagram and physical configuration of the equipment,
vi. Test Results as per the TEC Test Guide for the GR.

All the necessary set-ups & measuring instruments duly calibrated by an authorised lab shall be provided by the manufacturer for testing. The manufacturer shall provide proper operating environment required for testing.

Note: Though every care has been taken to cover all the parameters of the GR correctly in this Test Guide, yet to avoid any inadvertent error/ misprint, the testing officer shall ensure that all the parameters of the GR have been tested & verified in accordance with the provisions of the GR, same will be reflected on the report.

C. General Information

Sn	General Information	Details (to be filled by testing team)
1.	Name and Address of the Applicant	
2.	Date of Registration of Application	
3.	Name and No. of TEC Standard /Applicant's Spec. against which the approval sought	
4.	Details of Equipment	
	Type of the Equipment	Model No. and Serial No.
(i)	TCP/IP protocols implemented at which encryption adapted for Quantum-Safe and Classical	X.509/SSH/TLS/IPSec/S-MIME

	Cryptographic Systems				
(ii)	Types of Crypto Module	HCM ( )	SCM ( )	FCM ( )	HyCM ( )
(iii)	Type of the security/Assurance level product opted for	Security/ Assurance Level for 1 ( ) 2A/2B/2C ( ) 3 ( ) 4 ( )			
(iv)	Details of Equipment	Quantum safe (PQC only) ( ) or Hybrid (Classical + PQC) ( )			
(vii)	Details of hardware/software modules in the product				
	Module Name	Hardware Version	Software version	Serial No.	
1.					
2.					
3.					
(viii)	Interface available in product	Ethernet/USB/Optical/PCI/UART/SPI/I2C/Bluetooth/WiFi			
(ix)	Throughput of device	At Server End - _____ At User End - _____			
(x)	Latency of device	At Server End - _____ At User End - _____			
(x)	No. of concurrent connection supported				

(xi)	Symmetric Key encryption Algorithms supported	AES-256/AES-192/ChaCha-20-Poly1305 Modes of operation XTS/ CBC/ CFB / OFB / CTR / GCM /
(xii)	Hash functions supported	SHA-256( ), SHA-384( ), SHA-512 ( ), SHA3-256( ), SHA3-384( ), SHA3-512 ( ) HMAC-SHA-256 ( ), HMAC-SHA-384 ( ), HMAC-SHA-512( ), HMAC-SHA3-256 ( ), HMAC-SHA3-384 ( ), HMAC-SHA3-512( ) SHAKE128 ( ), SHAKE256 ( ) KMAC128( ), KMAC256( )
(xiii)	Asymmetric Algorithms supported	Classical Asymmetric Cryptographic Algorithms Key Establishment / Key Exchange • DH-2048, DH-3072, • ECDH-P256, ECDH-P384, ECDH-P521 Digital Signature Algorithms • DSA-2048, DSA-3072, • ECDSA-P256, ECDSA-P384, ECDSA-P521, • RSA-2048, RSA-3072, RSA-4096 Quantum-Resistant Asymmetric Cryptographic Algorithms Key Encapsulation Mechanism (KEM) • ML-KEM-512, ML-KEM-768, ML-KEM-1024 Digital Signature Algorithms • ML-DSA-44, ML-DSA-65, ML-DSA-87

		• SLH-DSA-SHA2-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-256s, SLH-DSA-SHA2-256f
5.	PQC Algorithm Type supported (other than above)	Hash Based/Code based/Lattice based
6.	Device capabilities (as per vendor product)	• Random Number Generator PRNG/TRNG/QRNG • Key Management Module ( ) • KMIP Support ( ) • Cryptography Interfaces and APIs PKCS#11/CNG/JCE/Others • QKD key delivery interface ( ) • Lightweight Cryptography AEAD/FN-DSA/ASCON • Cryptography-as-a-service (Cloud) ( )
7.	Any other relevant information:-	

D. Testing team

S. no.	Name	Designation	Organization	Signature
1.				
2.				

E. List of the test instruments:

S. no.	Name of the test instrument	Make/Model *(to be filled by team)*	Validity of calibration *(to be filled by testing team)*
1.			_ _/_ _/_ _ _
2.			
3.			
4.			
5.			
6.			
7.			
8.			

F. Equipment Configuration Offered:

a) ..... Configuration

S. No.	Item	Details	Remarks

Relevant information like No. of cards, ports, slots, interfaces, size etc. may be filled as applicable for the product

b) ..... Configuration

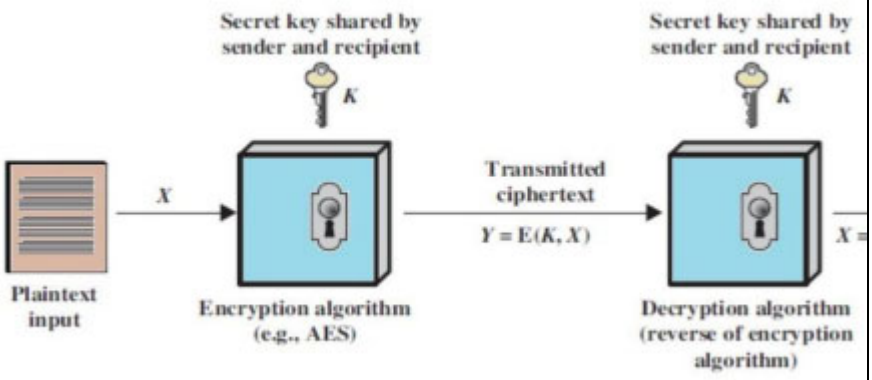
S. No.	Item	Details	Remarks

Relevant information like No. of cards, ports, slots, interfaces, size etc. may be filled as applicable for the product

G. Equipment/System Manuals:

*Availability of Maintenance manuals, Installation manual, Repair manual & User Manual etc. (**Y/N**)*

H. Clause-wise Test Type and Test No.:

Clause No.	Clause	Type of Test/Test No. etc.
	CHAPTER-1	
1.1	Introduction to Cryptographic Systems	
	Cryptography is the practice of securing communication and protecting data from unauthorized access by converting plaintext into ciphertext using mathematical algorithms, making it unintelligible to anyone without the proper key. It plays a critical role in securing our digital infrastructure. The typical cryptographic system is shown in Figure 1. The original message is usually termed plaintext and the scrambled message is called the ciphertext. The encryption  algorithm converts the plaintext to the ciphertext and the decryption algorithm performs a reverse process to get back the original message. Figure 1: Block Diagram of a typical Cryptographic System Our most crucial communication protocols rely on three core cryptographic primitives: public key encryption, digital signatures and key exchange. These primitives are	This is for information purpose.

	implemented using state-of-the-art of cryptographic algorithms, e.g., AES, Diffie-Hellman Key Exchange(DHKE), the RSA (Rivest-Shamir-Adleman) algorithm, and elliptic curve cryptography(ECC). The security of the public key cryptographic primitives as mentioned above depends on the difficulty of a number of theoretical problems, such as Integer Factorisation and the Discrete Log problem. In 1994, Peter Shor showed that Quantum computers, a new technology leveraging the physical properties of matter and energy to perform calculations, can efficiently solve factorisation and discrete log problems, thereby rendering all public key cryptosystems based on such assumptions insecure. Thus, a sufficiently powerful quantum computer will peril many forms of modern communication, from Key exchange to encryption to digital authentication. As a result, RSA and DHKE are no longer secure in a post-quantum era. Further, for data encryption, symmetric algorithms such as AES are widely used. Grover's algorithm offers a quadratic speed-up for brute-force key search compared to classical search, therefore, it can affect AES (and other symmetric encryption algorithms) by reducing the effective security of an n-bit key n/2-bit security on a sufficiently powerful quantum computer—so, for example, AES-128 would offer roughly ~64-bit quantum security, which is why doubling the symmetric key length is generally considered necessary to maintain the same security margin against quantum adversaries but not solely sufficient to prevent attack from quantum computer.	

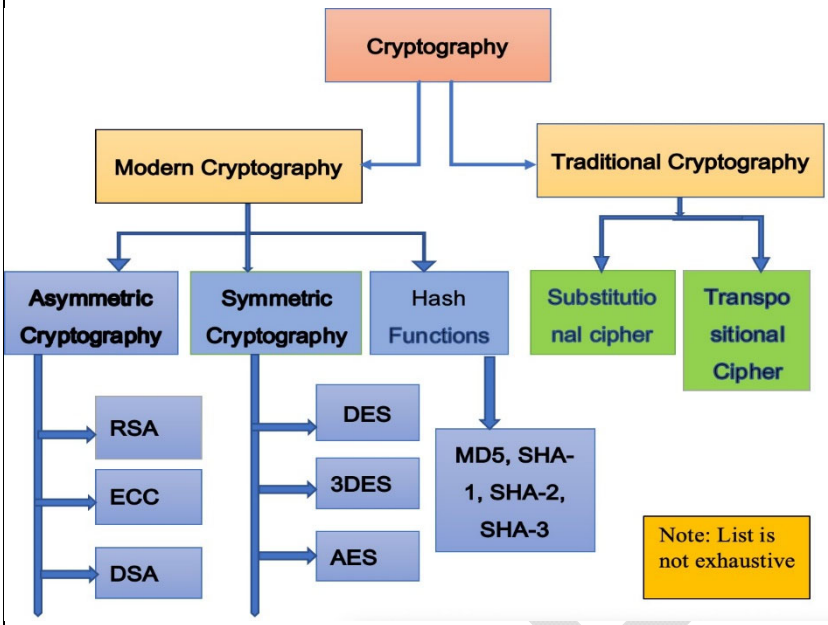
Table 1: Impact of Quantum Computing on common cryptographic algorithms

Sl. No.	Cryptographic Algorithm	Type	Purpose	Impact of the la scale quant computer
1.	AES	Symmetric Key	Encryption	Larger key sizes needed
2.	SHA-2, SHA-3	----------		SHA-2, SHA-3
3.	RSA	Public Key	Signatures key establishment	No longer secur
4.	ECDSA, ECDH	Public Key	Signatures, key exchange	No longer secur

Quantum-safe cryptographic systems, also known as post-quantum cryptography, are designed to be resistant to attacks from both classical and quantum computers. These systems use algorithms that are believed to be secure even against quantum computers. Quantum-safe cryptography is becoming increasingly important as quantum computers continue to evolve and become more powerful.

It is, therefore, critical to begin planning the replacement of hardware, software, and services that can interoperate with existing communications protocols and networks. Most

	quantum-resistant algorithms have larger Key sizes than the ones they will substitute, which is a big challenge. Quantum safe algorithms may change various Internet protocols, such as the Transport Layer Security (TLS) protocol or the Internet Key Exchange (IKE). Implementing quantum-safe algorithms requires identifying hardware and software modules, operating systems, communication protocols, cryptographic libraries, and applications employed in data centres on premises or in the cloud and distributed computing, storage, and network infrastructures. From a compliance and risk-management perspective, transitioning to quantum-safe cryptography helps address long-term confidentiality concerns such as "harvest now, decrypt later," where encrypted sensitive data captured today could potentially be decrypted in the future if cryptographically relevant quantum capabilities emerge.	
1.2	Classification of cryptographic algorithms Cryptographic algorithms are broadly classified into two categories, traditional and modern, based on the type used during the encryption and decryption process (refer to figure 2).	This is for information purpose.
1.3	Traditional Cryptography Traditional cryptography refers to cryptographic methods and techniques developed before the advent of computers.	This is for information purpose.
1.4	Modern Cryptography Modern cryptography is based on publicly known mathematical algorithms that operate on binary bit sequences and utilise secret keys. There are three types of modern cryptography: i. Symmetric (Secret Key) cryptography ii. Asymmetric (Public Key) cryptography	This is for information purpose.

	iii. Cryptographic Hash Functions  Figure 2: Block Diagram of classification of classical cryptography	
1.4.1	Symmetric key cryptography Encryption and decryption keys are identical in this scheme and should be known only to the communicating parties. Symmetric key cryptography is much faster than Asymmetric key cryptography, is far less resource-intensive than asymmetric encryption and is an incredibly efficient way to protect large volumes of data. Examples are Advanced Triple-Data Encryption Standard (DES), i.e., 3DES, Advanced Encryption System (AES), etc.	This is for information purpose.
1.4.2	Asymmetric Key Cryptography In this scheme, two keys are used, i.e., public key (for encryption) and private key (for decryption). The private key is kept secret as it is used for decryption, while the public key is not. For a secure public key cryptosystem, it is impossible to determine the private key's value by knowing the corresponding public key.	This is for information purpose.

	Most public communication networks use a combination of asymmetric and symmetric key cryptography schemes. An asymmetric/ Public Key Cryptography scheme is used for key distribution. At the same time, the data flow is secured using a symmetric technique because of its better performance in the encryption/decryption process.	
1.4.3	Hash Function A Hash function is a cryptographic algorithm that takes an input message of any size and outputs a short fingerprint of fixed length. Typically, it does not require any key along with the input message, and the output is usually called hash-value or hash-digest. These algorithms are typically used to ensure the authenticity or integrity of data. Hash functions can also use keys, referred to as Keyed-hash functions, under such usage. Many operating systems/applications store passwords using hash functions.	This is for information purpose.
1.5	Types of configuration of cryptographic system	
1.5.1	A cryptographic module shall be a set of hardware, software, firmware or some combination thereof that at a minimum, implements a defined cryptographic service employing an approved cryptographic algorithm, security function or process and contained within a defined cryptographic boundary.	This is for information purpose.
1.5.2	The cryptographic systems can be classified based on the hardware, software and or firmware used in modular form within the cryptographic boundary. These modules may be part of any interdependent or standalone system.	This is for information purpose.
1.5.3	The cryptographic module/system can be defined as one of the following types:	This is for information purpose.

	i. Hardware module: It is a module whose cryptographic boundary is specified at a hardware perimeter. Firmware and/or software, which may also include an operating system, may be included within the hardware cryptographic boundary. ii. Software module: It is a module whose cryptographic boundary delimits the exclusive software component(s) (may be one or multiple software components) that execute(s) in an adjustable operational environment. The computing platform and operating system of the working environment in which the software performs are external to the defined software module boundary. iii. Firmware module: It is a module whose cryptographic boundary delimits the exclusive firmware component(s) that execute(s) in a limited or non-modifiable operational environment. The computing platform and operating system of the operational environment in which the firmware executes in are external to the defined firmware module boundary but explicitly bound to the firmware module. iv. Hybrid Software module: It is a module whose cryptographic boundary delimits the composite of a software component and a disjoint hardware component (i.e. the software component is not contained within the hardware module boundary). The computing platform and operating system of the operational environment in which the software executes are external to the defined hybrid software module boundary. v. Hybrid Firmware module: It is a module whose cryptographic boundary delimits the composite of a	

	firmware component and a disjoint hardware component (i.e. the firmware component is not contained within the hardware module boundary). The computing platform and operating system of the operational environment in which the firmware executes in are external to the defined hybrid firmware module boundary but explicitly bound to the hybrid firmware module.	
1.6	Classification of Quantum-safe cryptography configuration The Quantum-safe cryptography module can be classified in a similar manner to classical cryptography modules. However, the algorithms will be different, especially for public key infrastructure like public key encryption schemes, key exchange mechanisms, digital signature schemes and hash functions. These algorithms need to resist attacks by quantum computers, and at the same time, they should still be secure against classical computer attacks. For symmetric key cryptography, doubling the key size can provide some protection against quantum computing attacks, but this is not a complete solution. New search algorithms are being developed for asymmetric key cryptography to resist quantum computing attacks.	This is for information purpose.
1.6.1	Symmetric Cryptography Symmetric key cryptography is vulnerable to quantum attacks. It is mostly threatened by Grover's algorithm. Unlike the asymmetric encryption algorithms (e.g. RSA, etc.) which could be completely broken by the Quantum computer; for symmetric algorithms like AES, the best known Grover's algorithm for attacking these encryption algorithms only weakens them. Grover's algorithm decreases the effective key	This is for information purpose.

	length of a symmetric encryption algorithm by half, so AES-128 has an effective key space of 2^{64} and AES-256 has an effective key space of 2^{128}.	
1.6.2	Quantum-safe asymmetric cryptography Today's most important uses of public key cryptography are for digital signatures and key establishment. Constructing a large-scale quantum computer would render many of these public key cryptosystems insecure. In particular, this includes those based on the difficulty of integer factorisation, such as RSA and those based on the hardness of the discrete logarithm problems. Quantum-safe Cryptography mainly refers to developing new asymmetric cryptography techniques that use a different class of hard mathematical problems. There are a few popular Quantum-safe cryptographic approaches that have emerged, such as Lattice-based, Code-based, multivariate-based and hash based cryptography. These mathematically hard problems are believed to be secure against classical as well as quantum computers.	This is for information purpose.
1.6.3	Quantum-safe Hash and Signature functions Cryptographic hash functions are widely used to provide data integrity and to build digital signature schemes. SHA-2 family hash algorithms (e.g., SHA-256 and SHA-512) are generally considered quantum-safe in the sense that no known quantum algorithm (including Shor's) breaks them outright; instead, the main quantum impact is limited to generic speed-ups such as Grover's algorithm. While Grover's algorithm can reduce the complexity of brute-force search, this impact can typically be addressed by selecting appropriately strong hash functions and security parameters.	This is for information purpose.

	Hash-based signature schemes are an important class of quantum-safe digital signatures. Basic constructions such as Lamport–Diffie and Winternitz are one-time signature schemes, meaning each private signing key must be used only once. To enable practical use for multiple signatures, these one-time keys are combined using Merkle tree constructions, allowing a single public key to authenticate a large number of signatures, bounded by the size of the tree. A well-known example is the eXtended Merkle Signature Scheme (XMSS), which is a stateful hash-based signature scheme; it requires careful state management to ensure that one-time keys are never reused. Overall, quantum-safe hash and signature mechanisms provide a robust alternative for digital signatures in a post-quantum transition, particularly for long-term integrity and authenticity requirements.	
1.7	Elements or Subsystems and Applications of a cryptographic systems A cryptographic system relies upon two basic components, i.e., an algorithm (or cryptographic methodology) and a cryptography key. Cryptographic subsystems in classical cryptography are the same as in Quantum-safe cryptographic systems except that different algorithms are implemented on hardware (Key sharing methods are different in Quantum Key Distribution (QKD) and Quantum-safe Cryptography). It also consists of software/firmware modules, operating systems, communication protocols, cryptography libraries, and applications deployed in data centres on-premises or in cloud, distributed computing, storage and network infrastructure.	This is for information purpose.

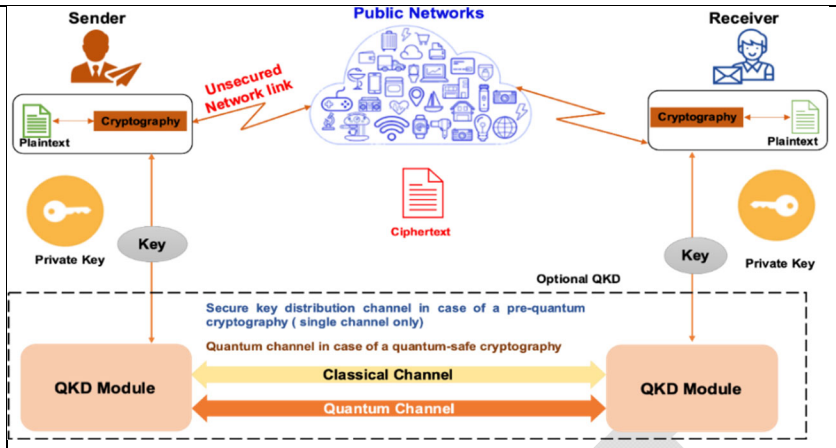


Figure 3: Block Diagram of a Symmetric cryptographic

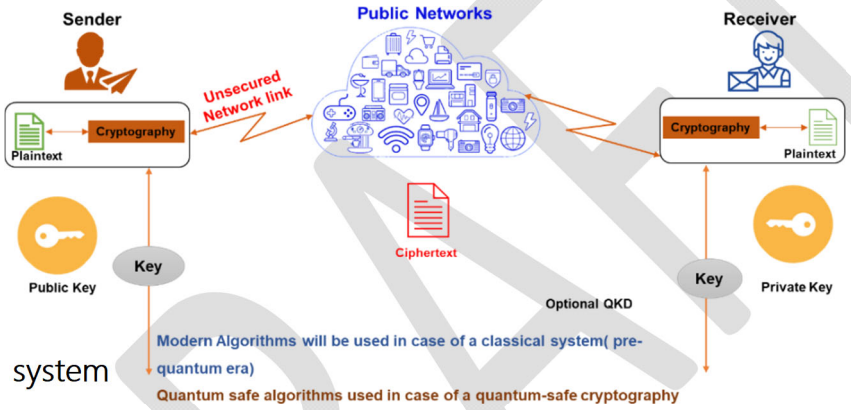


Figure 4: Block Diagram of Asymmetric cryptographic system

QKD, where deployed, may act as an external key source; however, PQC-based mechanisms shall be treated as the primary quantum-safe approach. Note: Encryption algorithms are the same, but in symmetric cryptographic systems, the key is transported through QKD (if used) is an optional external key source, whereas in the case of Asymmetric cryptography systems, the key is shared using Quantum-safe Cryptography key sharing algorithms. QKD is one of the key sources, as shown in Figure 3.

	CHAPTER-2 Functional Requirements	
2.1	Core Cryptographic Modules	
2.1.1	Cryptographic Processing Module The Cryptographic Processing Module shall implement the core cryptographic functions required by the system, including one or more functions like encryption, decryption, digital signature generation and verification, hashing, and message authentication. The module shall support hybrid (classical + PQC) or PQC algorithms. Below are the functional requirements of the Cryptographic Processing module applicable as per function supported by module.	Declaration of the algorithms supported with parameter sets from the Manufacturer.
2.1.2	The cryptographic processing module shall implement cryptographic processing function including symmetric and asymmetric encryption/decryption, hashing, digital signatures, key agreement and key encapsulation mechanism to ensure confidentiality, integrity, authenticity and non-repudiation of data.	Test case 1 Test case 2 Test case 3
2.1.3	The cryptographic processing module shall ensure correctness of all cryptographic operations such that outputs (e.g., decrypted plaintext, verified signatures, derived shared secrets, hash values) match expected results as defined in the applicable cryptographic standards and test vectors.	Test case 1 Test case 2 Test case 3
2.1.4	The cryptographic processing module shall support only approved cryptographic algorithms, including symmetric, asymmetric, and post-quantum algorithms, and shall reject deprecated or insecure algorithms (e.g., RC4, MD5, DES, SHA-1, ECB mode) and unsupported configurations.	Test case 1 Test case 2 Test case 3

2.1.5	The cryptographic processing module shall support cryptographic keys of approved types and sizes and shall validate all keys to ensure correctness, randomness, and compliance with applicable standards, rejecting invalid or weak keys.	Test case 1 Test case 2 Test case 3
2.1.6	The cryptographic processing module shall implement approved modes of operation and shall avoid legacy or insecure modes unless used with appropriate integrity protection mechanisms. Authenticated encryption modes (e.g., GCM) shall be supported for combined confidentiality and integrity.	Test case 1
2.1.7	The cryptographic processing module shall correctly generate, manage, and use cryptographic parameters such as Initialization Vectors (IVs), nonces, salts, and domain separation values, ensuring randomness, uniqueness, and compliance with the selected algorithm requirements.	Test case 1
2.1.8	The cryptographic processing module shall validate all inputs, including plaintext, ciphertext, messages, keys, signatures, and parameters, and shall handle boundary and edge cases (e.g., empty, null, oversized inputs) securely without causing unintended behavior or security compromise.	Test case 1 Test case 2 Test case 3
2.1.9	The cryptographic processing module shall ensure integrity protection by detecting any tampering or unauthorized modification of ciphertext, messages, or signatures and shall fail securely upon such detection.	Test case 1 Test case 2 Test case 3
2.1.10	The cryptographic processing module shall correctly implement padding schemes, encoding formats, and message structures as defined in applicable standards and shall reject malformed or improperly formatted inputs.	Test case 1 Test case 2 Test case 3
2.1.11	The cryptographic processing module shall support hashing functions compliant with approved standards and shall	Test case 2

	correctly process inputs of varying sizes, including incremental processing, and shall produce outputs matching known answer test vectors.	
2.1.12	The cryptographic processing module may support MAC algorithms (e.g., HMAC, CMAC, KMAC) and shall ensure correct key handling, message processing, and output generation consistent with the underlying hash or block cipher functions.	Test case 2
2.1.13	The cryptographic processing module shall support digital signature generation and verification using approved algorithms and shall ensure that signatures are bound to the message and key, rejecting forged, altered, or invalid signatures.	Test case 2
2.1.14	The cryptographic processing module shall support secure key agreement and key encapsulation mechanisms and shall ensure that derived shared secrets or recovered keys are correct and consistent across communicating entities.	Test case 2
2.1.15	The cryptographic processing module shall ensure non-deterministic behavior where applicable, such that repeated operations (e.g., encryption or signature generation) using the same input and key produce different outputs when required by the algorithm.	Test case 1 Test case 2 Test case 3
2.1.16	The cryptographic processing module shall ensure interoperability such that cryptographic outputs (e.g., ciphertexts, signatures, keys, hashes) generated by one compliant implementation can be successfully processed by another implementation conforming to the same standards.	Test case 4
2.1.17	The cryptographic processing module shall implement secure error handling and shall reject invalid inputs, malformed data, unsupported parameters, and unauthorized operations without exposing sensitive information.	Test case 1 Test case 2 Test case 3

2.1.18	The cryptographic processing module shall implement protections against side-channel attacks, including timing and memory-based leakage, as applicable to the implementation environment.	Test case 5
2.1.19	The cryptographic processing module shall securely erase cryptographic keys, intermediate values, and sensitive data from memory after use to prevent residual data exposure.	Test case 5
2.1.20	The cryptographic processing module shall operate within defined performance limits for all supported cryptographic operations under specified operating conditions and shall be suitable for the intended deployment environment.	Test case 5
2.1.21	The module shall support secure key agreement classical mechanisms (e.g., Diffie–Hellman, Elliptic Curve Diffie–Hellman Ephemeral) and post quantum key encapsulation and digital signature mechanisms (e.g., ML-KEM, ML-DSA, SLH-DSA, FN-DSA, HQC) including post-quantum hybrid schemes (e.g., X25519 + ML-KEM), ensuring correct operations.	Details should be submitted by the manufacturer.
2.1.22	The cryptographic module shall provide mechanisms for cryptographic agility, including the ability to enable, disable, or restrict the use of algorithms, modes, and key sizes through configuration or policy control. The module shall support deprecation of insecure or obsolete algorithms without requiring system redesign.	Test case 1 Test case 2 Test case 3
2.1.23	The module shall enforce minimum key sizes as per applicable security strength requirements (e.g., AES ≥ 128-bit, RSA ≥ 2048-bit, ECC ≥ 224-bit or equivalent).	Test case 1 Test case 2 Test case 3
2.2	Hybrid Cryptographic Schemes (Classical + PQC) i. The cryptographic module shall ensure that hybrid constructions correctly combine classical and post-quantum components using approved composition	Test case 6

	methods (e.g., concatenation followed by key derivation function). ii. The cryptographic module shall ensure that the overall security of the hybrid scheme is not weaker than the strongest individual component and that no downgrade or fallback to weaker algorithms is permitted. iii. The cryptographic module shall validate that both classical and PQC components independently satisfy their respective algorithmic correctness and security requirements. iv. The cryptographic module shall ensure interoperability of hybrid schemes across compliant implementations and shall correctly process hybrid keys, ciphertexts, and signatures. v. The cryptographic module shall provide documented procedures for validation, verification, and testing of hybrid cryptographic operations.	
2.3	Post-Quantum Cryptographic Algorithm Verification	
2.3.1	General Algorithm Validation i. The cryptographic module shall validate post-quantum cryptographic algorithms against officially published parameter sets, ensuring that all domain parameters, key sizes, and internal constants conform to the applicable algorithm specifications. ii. The cryptographic module shall demonstrate conformance to official Known Answer Tests (KAT), test vectors, and reference implementations for all supported PQC algorithms.	Test case 7

	iii. The cryptographic module shall ensure that implementations do not use reduced or non-standard parameter sets that weaken the intended security level. iv. The cryptographic module shall ensure that all probabilistic components of PQC algorithms (e.g., noise sampling, randomness generation, rejection sampling) follow the statistical distributions and properties defined in the respective algorithm specifications.	
2.3.2	Code- Based Cryptographic Algorithms i. The cryptographic module shall ensure that code-based schemes correctly implement encoding and decoding operations based on the underlying error-correcting codes (e.g., Goppa codes, quasi-cyclic codes) as defined in the algorithm specification. ii. The cryptographic module shall ensure that error vectors are generated with the correct weight and distribution and that decoding operations correctly recover the original message or shared secret. iii. The cryptographic module shall validate that ciphertexts and public keys conform to the required algebraic structure and size constraints defined in the scheme.	Test case 8
2.3.3	Lattice-Based Cryptographic Algorithms (LWE / R-LWE) i. The cryptographic module shall ensure that polynomial arithmetic (e.g., modular reduction, Number Theoretic Transform (NTT), inverse NTT) is implemented correctly and consistently across all operations. ii. The cryptographic module shall validate that ciphertexts and keys satisfy the underlying LWE/R-LWE	Test case 9

	mathematical relations and that decryption correctness holds within defined error bounds. iii. The cryptographic module shall ensure that parameter sets (e.g., modulus, dimension, noise bounds) meet the required security level and are not weakened in implementation. iv. The cryptographic module shall ensure correct implementation of noise sampling mechanisms (e.g., discrete Gaussian or centered binomial distribution) and shall verify that sampled values conform to the required statistical distribution. v. The cryptographic module shall ensure that decryption failure probability remains within the bounds specified by the algorithm and that implementation does not introduce increased failure rates.	
2.3.4	Hash-Based Cryptographic Algorithms i. The cryptographic module shall ensure correct implementation of hash-based constructions, including one-time signatures and tree-based structures (e.g., Merkle trees, XMSS, SPHINCS+), as defined in the applicable specifications. ii. The cryptographic module shall ensure that internal state management (e.g., leaf index, tree traversal) is correctly maintained and that state reuse or duplication is prevented. iii. The cryptographic module shall ensure correct use of domain separation and hash function inputs to avoid cross-protocol or cross-instance collisions.	Test case 10
2.3.5	Post-Quantum Digital Signature Algorithms i. The cryptographic module shall ensure that signature generation correctly incorporates randomness or	Test case 11

	deterministic processes as defined in the algorithm specification and shall prevent reuse of secret-dependent values across signatures. ii. The cryptographic module shall ensure that signature verification correctly validates all mathematical conditions defined in the algorithm, including bounds checking and rejection conditions. iii. The cryptographic module shall ensure that generated signatures conform to specified encoding formats and size constraints as defined in the applicable standard. *Note-* *1. The testing of classical cryptographic algorithms is out of the scope of this GR and for classical algorithms, the vendor may submit already available test certificate or declaration.* 2. *For post quantum algorithmic validation, Test certificates issued under Cryptographic Algorithm Validation Program (CAVP) or equivalent validation programs shall be acceptable.*	
2.4	Random Number Generator Module In cryptography, randomness is found everywhere, from the generation of keys to encryption systems, even how cryptosystems are attacked. Without randomness, all crypto operations would be predictable and hence, insecure. A good random number generator consists of two parts: a source of entropy and a cryptographic algorithm. Cryptographic algorithms require Keys. A Random Number Generator (RNG), also called a Random Bit Generator (RBG), is needed in the key generation process to create a random (strong) key as well as for other cryptographic purposes such as initialisation vectors and nonces. Typically, a True Random Number Generator	Declaration to be provided by manufacturer regarding the type of RNG mechanism supported. Test case 12

	(TRNG) provides a source of randomness or "entropy" to seed a Pseudo-Random Number Generation (PRNG), also called a Deterministic Random Bit Generator (DRBG).			
	Random bit generation	ISO/IEC 18031 OR TEC GR QRNG TEC 91020:2024 Test Guide of GR QRNG TEC 91021:2024 OR NIST SP 800-90 Series	Developers mu demonstrate that th entropy source is sufficien random through combination of desi and/or test processes a continuous checks duri operation for any fault th could have catastroph consequences f generating secu cryptographic keys.	
2.5	Key Management Module Deals with generating keys, verifying keys, exchanging keys, storing keys, and at the end of their lifetime, destroying keys. The bigger the key, the more secure the algorithm will be. The only negative of having an extremely long key is that the longer the key, the more the CPU is used to decrypt and encrypt data. Below are the functional requirements for key management module –			Declaration to be provided by manufacturer regarding the KMIP and Key Management module support. Test case 13
2.5.1	The key management module shall support secure generation of cryptographic keys, including asymmetric and/or symmetric and where applicable post-quantum keys, using approved mechanisms ensuring sufficient entropy, correct key sizes, and compliance with defined algorithm specifications.			
2.5.2	The key management module shall ensure secure storage and protection of cryptographic keys within a defined boundary,			

	maintaining confidentiality and integrity, and shall support secure key wrapping and unwrapping mechanisms using approved cryptographic methods.	
2.5.3	If the key management module supports distribution or exchange of cryptographic keys, such operations shall be performed through approved mechanisms. Keys transmitted via in-band or out-of-band methods shall be protected against unauthorized access, disclosure, and modification. Keys designated as non-exportable shall not be distributed outside their cryptographic boundary.	
2.5.4	The key management module shall enforce controlled usage of cryptographic keys in accordance with defined policies and attributes, ensuring that keys are used only for their intended purposes such as encryption, decryption, signing, or verification.	
2.5.5	The key management module shall support complete key lifecycle management including generation, activation, suspension, revocation, expiration, archival, and secure destruction, ensuring that invalid or expired keys are not used for any cryptographic operation.	
2.5.6	The key management module shall provide secure mechanisms for backup and recovery of cryptographic keys, ensuring preservation of confidentiality and integrity and enabling restoration of keys without loss of functionality or security.	
2.5.7	The key management module shall support secure import and export of cryptographic keys in standardized formats, ensuring integrity, authenticity, and protection of keys during transfer across systems.	
2.5.8	The key management module shall support interoperability with external systems through standardized protocols such as	

	KMIP, ensuring consistent handling of key formats, attributes, and metadata.	
2.5.9	The key management module shall enforce key attributes and policy controls, including usage restrictions, access control, and lifecycle parameters such as activation and expiration time.	
2.5.10	The key management module shall ensure protection against unauthorized access, key extraction, and leakage, and shall implement safeguards against side-channel and fault-based attacks during all key management operations.	
2.5.11	The key management module shall provide appropriate error handling by detecting and rejecting invalid, corrupted, or unauthorized key operations and ensuring that no sensitive information is disclosed through error responses.	
2.5.12	The key management module shall perform all key management operations within acceptable performance limits and shall support scalability to handle required volumes of key operations under normal and peak load conditions.	
2.5.13	The key management module, where interoperating with external systems, shall support the KMIP for standardized communication between key management systems and client applications, ensuring secure and interoperable key lifecycle operations.	
2.5.14	The key management module shall support KMIP operations including key creation, registration, retrieval, update, revocation, and destruction, and shall ensure correct handling of key attributes, metadata, and lifecycle states in accordance with KMIP specifications.	
2.5.15	The key management module shall ensure secure communication over KMIP interfaces, including authentication, authorization, and protection of data in transit,	

	and shall prevent unauthorized access or manipulation of cryptographic keys through KMIP transactions.	
2.6	Cryptographic Interface and APIs	
2.6.1	The cryptographic interface and API module shall provide standardized and secure interfaces for accessing cryptographic services exposed by the cryptographic system. The module shall support widely adopted cryptographic interface standards and frameworks such as PKCS#11, Cryptography API: Next Generation (CNG), Java Cryptography Extension (JCE), and Web Cryptography APIs, as applicable. Where key delivery through QKD system is implemented, the module shall support standardized interfaces aligned with ETSI specifications for QKD key delivery.	Declaration to be provided by manufacturer regarding the APIs supported. Test case 14
2.6.2	The module shall support abstraction of underlying cryptographic implementations such that applications invoking the APIs are independent of specific algorithms, providers, or hardware implementations.	
2.6.3	The module shall support dynamic selection and configuration of cryptographic algorithms, providers, and parameters through interfaces without requiring changes to application logic.	
2.6.4	The module shall enforce authentication and authorization mechanisms to ensure that only authorized entities can access cryptographic services.	
2.6.5	The module shall support secure session or context management for invoking cryptographic operations, including establishment, maintenance, and termination of sessions.	
2.6.6	The module shall validate all inputs received through interfaces, including parameters, identifiers, and data structures, and shall reject malformed or unauthorized requests.	

2.6.7	The module shall ensure that sensitive information, including cryptographic keys and intermediate data, is not exposed through APIs, logs, error messages, or debug interfaces.	
2.6.8	The module shall support integration with both software-based and hardware-backed cryptographic providers through unified interfaces.	
2.6.9	The module may support remote invocation of cryptographic services, ensuring confidentiality, integrity, and authentication of data in transit.	
2.6.10	The module shall support cryptographic agility at the interface level by enabling addition, removal, or modification of cryptographic algorithms and providers without requiring redesign of the interface.	
2.7	Protocol/Application Layer Requirements	
2.7.1	General Requirements	
2.7.1.1	The system shall implement secure communication protocols at the different TCP layers in compliance with applicable IETF standards and specifications.	Declaration to be provided by manufacturers regarding the protocols supported and documentation regarding hybrid implementation. Test case 15
2.7.1.2	The system shall support cryptographic negotiation mechanisms within protocols to enable selection of approved algorithms, including post-quantum, and hybrid cryptographic schemes.	
2.7.1.3	The system shall ensure interoperability with other compliant implementations of applicable protocols.	
2.7.1.4	The system shall prevent downgrade attacks by ensuring that negotiation mechanisms do not allow fallback to insecure or deprecated algorithms.	
2.7.1.5	The system shall support integration of post-quantum cryptographic mechanisms and hybrid schemes in accordance with applicable standards and evolving specifications.	

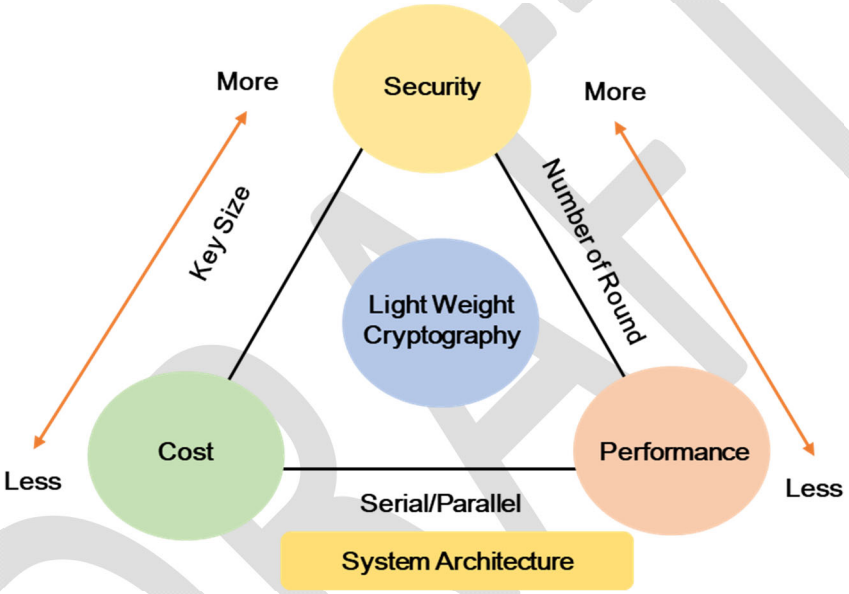
2.7.1.6	Where applicable, the system may support integration of externally generated keying material through Quantum Key Distribution (QKD), into protocol operations.	
2.7.1.7	The system shall support only secure and up-to-date versions of communication protocols and shall disable by default deprecated or insecure protocol versions.	
2.7.1.8	The system shall conform to applicable published RFCs or stable draft specifications for supported protocols and shall ensure correctness of message formats, negotiation procedures, and state transitions.	
2.8	X.509 / PKI Requirements	
2.8.1	The system shall support X.509 certificates for authentication and shall correctly process certificate structures, extensions, and encoding formats.	Test case 16
2.8.2	The system shall support hybrid or PQC certificates and shall correctly associate algorithm identifiers and parameters.	
2.8.3	The system shall support verification of both classical and post-quantum signatures within hybrid certificates.	
2.8.4	The system shall support certificate chain validation including mixed algorithm chains and shall ensure trust path correctness.	
2.8.5	The system shall ensure compatibility and interoperability of certificates across different systems and implementations.	
2.8.6	The system shall ensure that certificate validation mechanisms are resistant to downgrade, truncation, and substitution attacks.	
2.8.7	The system shall correctly process and validate algorithm identifiers (OIDs) and encoding formats for both hybrid and post-quantum cryptographic algorithms.	
2.9	IKEv2 / VPN Requirements	

2.9.1	The system shall implement Internet Key Exchange version 2 (IKEv2) for establishment of secure security associations.	Test case 17
2.9.2	The system shall support secure key exchange mechanisms providing Perfect Forward Secrecy, including post-quantum, and hybrid methods.	
2.9.3	The system shall ensure that negotiated session keys are correctly derived and consistent across communicating entities.	
2.9.4	The system shall support hybrid key exchange mechanisms combining classical and post-quantum algorithms.	
2.9.5	The system shall prevent downgrade to insecure or deprecated key exchange, authentication, or cryptographic algorithms.	
2.9.6	The system shall ensure interoperability with other compliant IKEv2 implementations.	
2.10	TLS Requirements	
2.10.1	The system shall implement Transport Layer Security protocols (with TLS 1.3) supporting secure communication for applications.	Test case 18
2.10.2	The system shall support cipher suite negotiation enabling selection of approved cryptographic algorithms.	
2.10.3	The system shall support hybrid key exchange mechanisms combining classical and post-quantum algorithms.	
2.10.4	The system shall support integration of post-quantum or hybrid certificates and signatures within protocol handshakes.	
2.10.5	The system shall ensure resistance to downgrade attacks and shall prevent fallback to insecure algorithms.	
2.10.6	The system shall ensure interoperability across different compliant TLS implementations.	

2.10.7	The system shall ensure correct implementation of handshake protocols, including key exchange, authentication, and session key derivation.	
2.11	S/MIME Requirements	
2.11.1	The system shall support secure email communication using S/MIME protocols for encryption and digital signatures.	Test case 19
2.11.2	The system shall support integration of post-quantum cryptographic mechanisms for key exchange and digital signatures	
2.11.3	The system shall ensure confidentiality, integrity, authentication, and non-repudiation of email communications.	
2.11.4	The system shall ensure interoperability with other compliant S/MIME implementations.	
2.12	SSH Requirements	
2.12.1	The system shall implement Secure Shell (SSH) protocol for secure remote access and communication.	Test case 20
2.12.3	The system shall support secure key exchange mechanisms providing Perfect Forward Secrecy, including post-quantum or hybrid approaches.	
2.12.3	The system shall support quantum-safe or hybrid authentication mechanisms for host and user authentication.	
2.12.4	The system shall prevent downgrade to insecure or deprecated authentication and key exchange mechanisms.	
2.12.5	The system shall ensure interoperability with other compliant SSH implementations.	
2.13	Crypto Agility Crypto agility is the capability of a system, organisation, or infrastructure to rapidly and securely adapt its cryptographic mechanisms—including algorithms, parameters, keys,	Test case 21

	certificates, and protocols—without requiring major architectural redesign or service disruption. It enables the seamless introduction of new cryptographic algorithms, the coexistence of classical and post-quantum mechanisms (including hybrid constructions), and the timely retirement of deprecated or vulnerable algorithms in response to evolving threats, standards, or regulatory requirements. Crypto agility is a foundational requirement for post-quantum readiness, ensuring that cryptographic transitions can be managed as a controlled, repeatable lifecycle process rather than as disruptive one-time migrations.	
2.13.1	Cryptographic Agility Requirements The below cryptographic agility capabilities can be vendor and product-dependent. For example, a product may support cryptographic switching/configuration but may not support cryptographic negotiation. i. The system shall support cryptographic agility, enabling the addition, replacement, or deprecation of cryptographic algorithms with minimal architectural redesign. ii. The system should support algorithm negotiation mechanisms covering hybrid, and quantum-safe cryptographic algorithms. iii. Cryptographic policies should be externally configurable (e.g., config file, API, or centralized policy server) and shall be protected against unauthorized modification (integrity protection + access control + audit logging). iv. The system shall support hybrid key establishment mechanisms that combine approved classical	

	cryptographic algorithms with post-quantum cryptographic mechanisms. v. Hybrid key derivation mechanisms shall ensure cryptographic independence between classical and post-quantum components, such that compromise of one does not affect the security of the other. Forward secrecy should be preserved during hybrid operation, ensuring protection of past session keys even if long-term keys are compromised. vi. Secure transport mechanisms should use TLS version 1.3 or higher with support for post-quantum cryptography-capable extensions. Legacy and insecure protocols shall be disabled by default. vii. Protocol downgrade attacks shall be prevented through appropriate cryptographic and protocol-level safeguards. viii. The system shall support hybrid certificates that enable the use of both classical and post-quantum public keys and digital signatures. Certificate path validation shall function correctly for certificates employing hybrid and post-quantum cryptographic signatures. ix. The system should support post-quantum cryptographic mechanisms in COSE-based environments, particularly for IoT, embedded systems, and deep packet inspection (DPI) use cases. x. The performance impact and resource utilization implications of post-quantum cryptographic mechanisms should be documented and optimized.	
2.14	Requirements for Constrained devices (Lightweight Cryptography)	Declaration to be provided by manufacturer

	The security of resource-constrained devices is critical for eg. in the IoT field, given that everything is interconnected. The concern is that the limited resources on these devices may cause performance issues when the standard cryptographic algorithms are running on them. Therefore, in recent years, researchers have been working on developing lightweight cryptography and various efficient cryptographic technologies. Its requirements are constrained by security, low cost and high performance.  Figure 7: Block Diagram of Lightweight cryptography design trade-offs. These requirements are balanced accordingly by adjusting the key size, the number of encryption rounds and the system architecture. Thus, the target of lightweight cryptography is to find a better balance between performance and security within cost constraints (refer Figure 7). The chosen algorithms are designed to protect information created and transmitted by the Internet of Things (IoT), including its myriad of tiny sensors and actuators. They are also designed for other	regarding applicability and if yes, then protocol supported.

	miniature technologies, such as implanted medical devices, stress detectors inside roads and bridges, and keyless entry fobs for vehicles. Devices like these need "lightweight cryptography" protection that uses the limited amount of electronic resources they possess. The most important in lightweight cryptography: authenticated encryption with associated data (AEAD) and hashing. AEAD protects the confidentiality of a message, but it also allows extra information, such as the header of a message, or a device's IP address, to be included without being encrypted. The algorithm ensures that all of the protected data is authentic and has not changed in transit. AEAD can be used in vehicle-to-vehicle communications, and it also can help prevent the counterfeiting of messages exchanged with the Radio Frequency Identification (RFID) tags that often help track packages in warehouses. They need to compliant NIST protocols as listed from time to time as per the user requirements.	Test Case 22

Sl. No.	Test Category	Description	Expected outcome

1.	AEAD Functionality	Verify AEAD encryption and decryption with associated data (e.g., header)	Data and associated data are encrypted and authenticated correctly
2.	Confidentiality	Ensure message confidentiality over noisy IoT channel using AEAD	Encrypted data not recoverable by unauthorized parties
3.	Authenticity	Test integrity verification when associated data is tampered with	Verification fails, rejecting tampered messages
4.	Key Size Trade-off	Evaluate security vs performance by adjusting key sizes	Smaller keys increase speed but maintain acceptable security levels
5.	Resistance to Attacks	Simulate side-channel and fault injection attacks on AEAD	System withstands attacks without key leakage or authentication bypass

6.	Performance	Measure encryption/decryption speed and power consumption on IoT hardware	Cryptosystem completes operations within device constraints
7.	Compliance Testing	Confirm compliance with NIST Lightweight Cryptography guidelines	Algorithm passes all mandatory compliance tests

In addition to AEAD and hashing, constrained IoT deployments often require digital signatures for firmware authenticity, secure boot / measured boot attestations, device identity, and non-repudiation in audit logs. FN-DSA or FIPS-206 is a digital signature algorithm that can be used for digital signature requirement for constraint devices.

Test Case 23

Sl. No.	Test Category	Description	Expected outcome
1.	Signature Functionality	Generate signature and verify signature over representative IoT messages (telemetry, control commands)	Valid signatures verify; invalid signatures fail

	2.	Message/Context Binding	Verify signatures bind to correct context (device ID, protocol header, domain separation tag if used)	Replay across contexts fails; correct context verifies	
	3.	Negative Testing	Verify behavior for modified message, modified signature, wrong public key, truncated inputs	Verification fails reliably without crashes	
	4.	Key Generation	Test key-pair generation across supported parameter sets; verify key validity checks	Keys generated correctly; invalid keys rejected	
	5.	Robustness (Malformed Inputs)	Feed malformed/edge-case encodings (oversized, invalid length, non-canonical forms)	Implementation rejects safely; no memory errors	

6.	Side-channel Resistance	Evaluate timing/power leakage under typical signing and verify operations (where applicable)	No exploitable leakage beyond acceptable threshold
7.	Fault Injection Resilience	Simulate fault conditions during signing/verification (glitches, bit flips)	No key leakage; faulty signatures do not verify
8.	Performance on IoT HW	Measure sign/verify time, RAM/Flash footprint, and energy consumption	Operations within device constraints; meets procurement limits
9.	Interoperability	Interop test with a known-good reference implementation across parameter sets	Cross-implementation verification succeeds
10.	Compliance Testing	Confirm compliance with applicable NIST/IETF profiles and test vectors	Passes mandatory vectors and profile requirements

Further, ASCON is a lightweight cryptographic family designed for constrained devices and selected by NIST for lightweight cryptography. For IoT, ASCON AEAD provides confidentiality + integrity with associated data (headers, addresses, counters), while ASCON-Hash supports integrity checks, commitments, and protocol hashing needs.

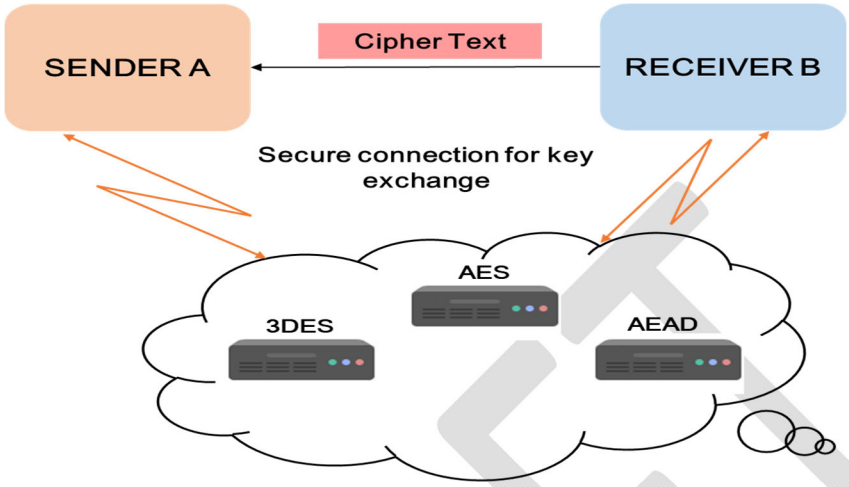
Test Case 24

Sl. No.	Test Category	Description	Expected outcome
1.	Algorithm ID / Variant	Verify the implementation supports the intended standardized functions: Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128, Ascon-CXOF128 (as applicable).	Implemented variants match the claimed functions; unsupported variants are not claimed.
2.	AEAD Correctness	Verify ASCON AEAD encrypt/decrypt with associated data using official test vectors	Ciphertext and tags match vectors; decrypt recovers plaintext
3.	Associated Data Integrity	Modify AD (header/IP/metadata) and verify tag failure	Verification fails; plaintext not released

		4.	Nonce Handling	Verify unique-nonce requirement enforcement (generation/storage/ anti-reuse)	No nonce reuse in normal operation; reuse detected/mitigated per policy	
		5.	Tag Verification Behavior	Ensure constant-time tag verification and " fail closed" behavior	No timing oracle; invalid tags always rejected	
		6.	Misuse / Edge Cases	Test zero-length plaintext/AD, maximum lengths, fragmented inputs	Correct outputs; no crashes; consistent streaming behavior if supported	
		7.	Replay Protection Support	Validate integration with counters/timestamps as AD (where protocol uses it)	Replayed messages detected by system logic; crypto validates binding	
		8.	Hash Correctness	Verify ASCON-Hash outputs for standard vectors and typical IoT payloads	Hash outputs match vectors; stable across platforms	

		9.	XOF Functional Correctness	Validate Ascon-XOF128 supports selectable output length L > 0, correct number of squeezed blocks and correct IV usage.	Output length equals requested L; output matches known-good implementation for various L.	
		10.	CXOF Customization Handling	Validate Ascon-CXOF128: customization string input Z is supported and incorporated as specified; verify the length constraint	--	
		11.	Robustness (Malformed Inputs)	Corrupt lengths/encodings and provide malformed buffers	Implementation rejects safely; no memory corruption	
		12.	Side-channel Resistance	Timing/power analysis on encryption/decryption and hashing (if required)	Leakage within acceptable limits; masking/constant-time validated	
		13.	Performance on IoT HW	Measure throughput, latency, RAM/Flash, and power on target MCU/SoC	Meets device constraints and procurement targets	

	14.	Compliance Testing	Confirm conformance to NIST LWC ASCON specifications and published vectors	Passes all mandatory compliance tests	
2.15	Cloud based PQC products (Cryptography –as – a- service) Cloud services have become ubiquitous due to the rise of high-capacity networks, the decreased cost of computers and data storage devices, and trends toward hardware virtualisation and infrastructure, platform, and software-as-a-service models.				For information purpose only
2.15.1	Cryptography-as-a-Service Deploying cryptographic keys to endpoints such as smartphones, virtual machines in the public cloud and smart grid equipment is risky. Therefore, this proposes a Cryptography as a Service (CaaS) model, which allows cryptographic operations to be performed without exposing cryptographic keys and recommends overcoming the pitfalls associated with this technology. Keyed cryptographic operations, such as encryption and decryption, are performed by a CaaS provider on behalf of a device via web services APIs. Cryptography as a service has been defined as being "Keyed cryptographic operations, such as encryption and decryption that are performed by a CaaS provider on behalf of a device via web service APIs". The way that the "as a service" architecture works is through the implementation of HTTP and systems such as REST and SOAP. The overall architecture is extremely similar to Public Key Authorities (PKA) and Certificate Authorities (CA). The cryptographic keys used to perform these operations are stored within the CaaS provider,				For information purpose only

	so devices do not possess these keys at any time (refer figure 8).  Figure 8: Block Diagram - Cryptography-as-a-Service	
2.15.2	Cryptography Service Provider (CSP) As organisations continue to test and integrate cloud computing into their IT environments, Cryptography-as-a-service and Entropy-as-a-service are into service to safeguard cryptographic keys with the same dynamic and virtualised attributes of cloud computing environments. Additionally, when storing data in multi-clouds, using native encryption from cloud service providers creates silos of data and the risk of not having full control over your keys and data. On-premises Hardware-Secure-Module(HSM) diminish those silos and enable users to know the whereabouts of their keys at all times. BYOE (bring your encryption), or BYOK (bring your keys), is a security model tailored explicitly to cloud computing. It allows cloud service customers to use their encryption tools and manage their encryption keys. A cryptographic Service Provider (CSP) allows Cryptographic applications and services	For information purpose only

	to access secure cryptographic operations and Key management. This provider uses the standard REST API, JCE (Java Cryptographic Extension) programming interface. PKCS#11, Cryptography API: Next Generation (CNG), HTTPS, Web API (W3C), Microsoft CAPI, and OpenSSL.	
2.15.3	Verification of cloud/service based key lifecycle management	Declaration to be provided by manufacturer regarding applicability. Test Case 25
2.15.3.1	Verify integration of Cloud HSM (e.g., AWS Cloud HSM, Azure Key Vault, Google Cloud KMS etc.)	
2.15.3.2	All Cloud HSM deployments shall support complete cryptographic key lifecycle management, including secure key generation, storage, usage, rotation, archival, and destruction.	
2.15.3.3	Cryptographic keys shall be generated and remain within FIPS 140-2 Level 3 (or higher) validated HSM security boundaries, and plaintext export of key material shall not be permitted from the validated HSM security boundaries.	
2.15.3.4	Access to keys shall be governed by role-based access control, enforcing least-privilege, segregation of duties and multi-factor authentication.	
2.15.3.5	Key rotation policies shall be mandatorily enforced, with keys rotation periods as per NIST SP 800-57 Part 1 Rev. 5. The key rotation periods shall be shorter for high-risk systems. Longer crypto period is allowed for Root / Master keys but periodic rotation is recommended.	
2.15.3.6	Automated rotation mechanisms shall be supported without service disruption, and previous key versions shall remain available for decryption or verification only.	
2.15.3.7	All key lifecycle events—including creation, access, rotation, policy changes, and destruction—shall generate immutable audit logs.	

2.15.3.8	Audit logs shall be tamper-evident, exportable to external systems with audit log retention periods as defined in alignment with NIST SP 800-57, NIST SP 800-53, ISO/IEC 11770, ISO/IEC 27001, and applicable national regulatory requirements, including CERT-In cyber security directions. For eg. – The Audit logs may be retained online for a minimum of 400 days, and archived securely for a minimum period of seven years.	
2.15.3.9	Cryptographic destruction of keys shall be irreversible and verifiable through audit evidence.	
2.15.3.10	Check secure key provisioning, distribution, rotation, archival, and destruction mechanisms as per above steps.	
2.15.3.11	Verify that only authorized services or users can access HSM APIs and review IAM policies and role-based access configurations.	
2.15.3.12	Perform dynamic tests on key management APIs for: i. Unauthorized access attempts ii. Replay attacks and injection vulnerabilities iii. Improper error handling revealing sensitive info iv. Data-in-Transit and Data-at-Rest Protection	
2.15.3.13	Verify Cloud HSM complies with FIPS/ISO standards (or equivalent) at least FIPS 140-2 Level 3 (or higher) validated HSMs.	
	CHAPTER-3	
3.1	Operational, Interface and Interoperability Requirements	
3.1.1	Based on network deployment topologies, the cryptographic system should work in point-to-point/ point-to-multipoint / multipoint-to-multipoint mode.	Self-declaration to be provided by the manufacturer.
3.1.2	The cryptographic system shall provide Ethernet payload encryption over a point-to-point network.	

<table>
<tr>
<td>3.1.3</td>
<td>It must be possible for an operator to select a particular encryption scheme for payload encryption system wise.
i. It shall provide confidentiality and protection from firmware upgrades.
ii. It shall support Policy based encryption.
iii. It shall provide data protection against unauthorised access by users and processes in physical, virtual, and cloud environments so that implementation is seamless and transparent to application/presentation of layer of system and its storage. So it can work across an enterprise's entire environment.
iv. Regardless of performance level, the cryptographic system shall be interoperable with the appropriate Application interface.
v. It shall provide confidentiality using standard encryption algorithms in a Quantum-safe cryptosystem and applicable algorithms in asymmetric and hash functions as per the product's specification sheet.</td>
<td>Test Cases already covered in other chapters.</td>
</tr>
<tr>
<td>3.2</td>
<td>Operational requirements of a cryptographic system.
<table>
<tr><th>Sl. No.</th><th>Parameter Type</th><th>Description and range of the Parameters</th><th>Reference Standard(s)</th><th>Remarks</th></tr>
<tr><td>1.</td><td>Traffic type</td><td>Unicast / Multicast / Broadcast over IP networks (IPv4 / IPv6)</td><td>IPv4: RFC 791, IPv6: RFC 8200</td><td>To be confirmed as per applicable RFCs and product scope</td></tr>
</table>
</td>
<td>Test Case 26, 27, 28 and 29 for Clause 8, 9, 10 and 14.

For remaining clauses, self-declaration with values (as applicable) to be submitted by the manufacturer.</td>
</tr>
</table>

	2.	No of Concurrent Connection	User-to-Server mode: support at least 100/500 connections as applicable	TCP: RFC 9293, TLS 1.3: RFC 8446	Exact value to be finalized by procurer based on deployment sizing; verification via load/concurrency testing		
	3.	Direction of data transmission	Full duplex	IEEE 802.3 (Ethernet), as applicable	Low overhead bits		
	4.	Separation of data/control plane	Separation of Control plane and data plane	Product architecture / deployment specification	Physical and logical separation of data and control plane		
	5.	Latency at specified rate	Latency at node (non-aggregation state) ≤ 10 µs for data	-------	Latency requirement to be independent of		

			(server/client)	throughput up to 10 Gb/s		packet/Ethernet frame size		
		6.	Support of Jumbo frames	Support Ethernet frames larger than standard MTU; configurable jumbo frame size	IEEE 802.3	Beyond standard ethernet frame size, exact maximum MTU to be specified by procurer		
		7.	Mode of secure key uploading	Manual/Automatic	ISO/IEC 19790:2025	Applicable according to security level 1/2/3/4		
		8.	Software / Firmware loading	The cryptographic module can load software or firmware from an external source.	ISO/IEC 19790:2025 Clause 7.4.3.4	A validation authority shall validate the integrity/authenticity of loaded software or firmware before loading.		

	9.	Self-test for the integrity of H/W and S/W modules	Cryptographic module pre-operational and conditional self-tests provide the operator assurance that faults have not been introduced that would prevent the module's correct operation.	ISO/IEC 19790:2025 Clause 7.10	Conditional self-tests shall be performed when an applicable security function or process is invoked.		
	10.	Module's version	The cryptographic module shall output the name or module identifier and the versioning information	ISO/IEC 19790:2025 Clause 7.4.3.1	Hardware, software and/or firmware versioning information		

		11.	Status	The cryptographic module shall output the current status	ISO/IEC 19790:2025 Clause 7.4.3.1	Visual indicators in response to a service request/ normal state	
		12.	Self-tests	pre-operational self-tests before loaded code can be executed	ISO/IEC 19790:2025 Clause 7.4.3.1	Status shall reflect completion/ pass/fail	
		13.	Approved Security function test	Approved security functions	ISO/IEC 19790:2025 Clause 7.4.3.1	at least one test in the approved mode of operation	
		14.	Zeroisation	Perform zeroisation (zeroise all unprotected SSPs and key components within the module at all	ISO/IEC 19790:2025 Clause 7.4.3.1	Zeroisation shall be immediate and uninterruptable in Security Level 4	

				security levels)			
		15.	Mode of operation	Normal/degraded	ISO/IEC 19790:2025 Clause 7.2.4.3	Normal mode only if all pre-operational self-tests pass	
		16.	Bypass capability	Indicate whether the Bypass capability is activated or not	ISO/IEC 19790:2025 Clause 7.2.4.3	Bypass allowed only with controls preventing inadvertent plaintext exposure due to a single error	
		17.	Self-Initiated cryptographic output Test	Indicate capability for self-initiated crypto output test without Crypto Officer configuration; show status when	ISO/IEC 19790:2025	this configuration may be preserved over resetting, rebooting, or power cycling of the module	

			enabled				
	18.	Operational environment	i. A non-modifiable operational environment ii. A limited operational environment. A modifiable operational environment	ISO/IEC 19790:2025 Clause 7.6	Functions may be added or modified within the operational environment.		
	19.	Life-cycle assurance	Confirm the best practices by the vendor of a cryptographic module during the design, development, operation and end of life of a cryptographic module.	ISO/IEC 19790:2025 Clause 7.11	Vendor to provide evidence covering defined lifecycle stages		
	20.	Power	AC supply 110-230V	Vendor specific	AC or DC supply or		

				+10% 50/60 Hz AC	ation	both as optional Procurer to specify connector type and redundancy if required	
		21.	DC power	DC supply range: –40 V to –60 V DC (including renewable sources, as applicable)	Vendor specification		
		22.	Size	Dimensions in mm or inches in length, width and height	Vendor specification	Desirable: 1U; multiple 1U options acceptable	
		23.	Cooling	Requirement of Ingress or Egress fans (suck and exhaust kind of setup).	--	Fan may be optional depending on environment; temperature maintenanc	

				e is mandatory	
24.	Min Altitude without any degradation	Normal operation without any degradation at an altitude of upto 3,000 meters.	--	The manufacturer shall guarantee satisfactory performance	
25.	Power Supply Alarm	Visual indicator (e.g., Green/Red) for AC/DC power status		Indicate the status of power AC/DC.	
26.	Encryption/Decryption Alarm	Any visual indicator(G/R or any other colour)		To indicate status	
27.	Fault Indicator Alarm	Any visual indicator(G/R)	Log message and visual indicator	To indicate status	
28.	Capable	Without		Self-certificate	

			of functioning in a saline environment	any degradation system shall be able to function normally		to be submitted if no test environment is available.	
3.3	Interface requirements of a cryptographic system The cryptographic system shall support 10/100/1000/2500/10000 BASE-TX electrical or optical interface or any open standard port for management as per the user requirement. Hardware/Software of Plaintext Interface shall be physically separate from Hardware/Software of Cipher interface.						Test Case 30 for Clause 8. For remaining clauses, self-declaration with details of the interfaces (as applicable) to be submitted by the manufacturer.

Sl. No.	Name of the Sub Parameter	Types of Parameters range	Reference Standard(s)	Remarks
1.	Management Interface	Optical//Ethernet (RJ45) Ethernet data input through the command line interface also. SNMP v3 or above, or XML/JSON shall be supported for	ISO/IEC 19790:2025 Clause 7.3.1, 7.3.2	Applicable interface type(s) depend on module category: i. HMI (Hardware Module Interface): data port + management port;

			EMS/NMS/NOC.		ii. SFM (Software/Firmware Module): logical interfaces; iii. HSMI/HFMI (Hybrid SW/FW Module Interface): plaintext/ciphertext interface separation	
	2.	Data input interface	Interface (plain text, cipher text and SSP)	ISO/IEC 19790:2025 Clause 7.3.3(a)		
	3.	Data	Interface (plain text, cipher text and SSP)	ISO/IEC 19790:2025 Clause 7.3.3(b)		
	4.	Control input interface	All input commands, signals, and control data (e.g., clock input, function	ISO/IEC 19790:2025 Clause 7.3.3(c)	Access control/authentication applies as per module security policy	

				calls, manual controls such as switches/ buttons/ keyboards)			
		5.	Control output interface	All output commands, signals, and control data	ISO/IEC 19790:2025 Clause 7.3.3(d)	Inhibited when the cryptographic module is in an error state unless exceptions are specified	
		6.	Power interface	All external electrical power entering/leaving the cryptographic module (not applicable to pure software-only modules)	ISO/IEC 19790:2025 Clause 7.3.3(g)	Except in the software module, power is provided internally by the source of the battery.	

	7.	Status output	All status output signals, indicators, and status data (including visual/audio/mechanical indicators)	ISO/IEC 19790:2025 Clause 7.3.3(e)	Includes error indicators (return codes), displays/LEDs, buzzer/tone/ring/vibration where implemented	
	8.	Trusted channel (Security Level 3 and above)	Protected link for transmission of unprotected plaintext CSPs/key components and authentication data between module and endpoint	ISO/IEC 19790:2025 Clause 7.3.4	For Security Level 4, multi-factor identity-based authentication shall be employed for all services utilising the trusted channel	

3.4 Interoperable requirements of a cryptographic system

Interoperability is one of the essentials to making seamless internetwork function in a heterogeneous network environment. The application service layer in the cryptographic system communicates with the key management controller. Communication protocol and data format for a quantum key distribution (QKD) network or any Key source network to supply cryptographic keys to an application, i.e., a Cryptographic system.

Sl. No.	Name of the Sub	Types of Parameters range	Reference Standard(s)	Remarks

Test Case 31 for Clause 10.

For Clause 1 to 9, already test cases covered in Chapter 2.

Self-declaration with mention of Link Layer protocol if

		parameter				applicable to be submitted by the manufacturer for Clause 11.
	1.	IP Layer	Internet Protocol (IP) IPV4/IP6, Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP), IPSec	IETF RFCs IPv4: RFC 791; IPv6: RFC 8200; ICMPv4: RFC 792; ICMPv6: RFC 4443; IGMP: RFC 3376; IPsec: RFC 4301 (and related RFCs)	Confirm interworking for both IPv4 and IPv6;	
	2.	Authentication	CA-based authentication and/or AAA-based authentication (as applicable)	X.509/PKI; RADIUS: RFC 2865; TLS: RFC 8446	CA trust model as per (enterprise CA / public CA / local RA); RADIUS server may be used for centralized AAA	
	3.	Encryption	Various encryption methods as listed	NIST-approved algorithms (as applicable)	Device shall support configured algorithms/cipher suites as per security policy	

		4.	Key exchange (KMIP)	During a key exchange with other systems	OASIS standard	------	
		5.	API with QKD interface	REST/HTTPS API for middleware integration; JSON encoding	HTTPS/TLS: RFC 8446; JSON: RFC 8259	-----	
		6.	Inter Secure Application Entity (SAE)	Master SAE ↔ Slave SAE communication for QKD key delivery/control	QKD link as per ETSI GS QKD 004	SAE of cryptographic system connects to the QKD KME	
		7.	SSH	User authentication layer, transport layer, connection layer	RFC 4252, RFC 4253, RFC 4254	SSH may be used in several methodologies like for secure administration/ management access	
		8.	TLS	TLS v1.3 or above	IETF RFC 8446	Use for management/API channels and other interfaces requiring secure transport	

	9.	Entropy source	Proven/validated randomness source for key generation and nonce	NIST SP standards, TEC QRNG standard	Examples: on-chip TRNG, dedicated circuitry, external entropy source;		
	10.	Clock	Internal circuit or External I/O timing source	Product/platform spec	Used for control functions, timeouts, scheduling, counters/ ticks; accuracy requirements to be specified if needed		
	11.	Link layer protocols	L2 options such as Ethernet MAC, PPP, tunneling (as applicable)	IEEE 802.3; PPP: RFC 1661; relevant tunnel RFCs as applicable	Layer 2 protocol communications		
	CHAPTER-4 Security Requirements						
4.1	**Security services requirements of a cryptographic system** The following security services are required for the enhancement of security; i. Authentication mechanisms may be needed within a cryptographic module to authenticate an operator accessing the module and to verify that the operator is authorised to assume the requested role and						Test Case 53

	perform services within that role. The cryptographic system shall support lossless data encryption/ decryption key change. ii. It should implement a key integrity check and authentication mechanism through a suitable hashing algorithm. iii. Encryption keys should be encrypted, stored in a secure device and only accessible to the user, regardless of data and key storage methods.	
4.2	Security/Assurance level classification and requirements The cryptographic techniques (algorithms and protocols) may remain the same across different security/assurance levels; however, the assurance requirements increase with the risk category, usage environment, and deployment criticality. Selection of a PQC-based cryptographic solution shall therefore be based on the assurance level appropriate to the application's risk appetite and the operational environment. The hierarchical structure defines that higher assurance implies compliance with lower assurance requirements, and the quantum-safe cryptographic systems are classified into Level 1, Level 2, Level 3, and Level 4 based on security needs and deployment context. i. Security/Assurance Level 1: Provides a baseline level of security for PQC adoption in non-sensitive, consumer-grade environments. Focus is on basic PQC compliance, correctness, interoperability (including RFC conformance where applicable), and basic performance checks. Basic security requirements are specified for a cryptographic module (e.g. at least one approved security function or approved sensitive security parameter establishment method shall be	Detail of applicable Security/Assurance Level to be submitted by the manufacturer.

	used). Ideally appropriate for security applications where controls, such as physical security, network security, and administrative procedures, are provided outside the module but within the deployable environment. ii. **Security/Assurance Level 2:** Applies to PQC products/solutions handling sensitive data in consumer-grade environments, including cloud-integrated implementations. Level 2 is additionally bifurcated in three sub Levels – Level 2A (Secure Software Assurance for software based PQC products), Level 2B (Secure Hardware Assurance (IoT/IT) for hardware based IT/IoT products) and Level 2C (Hardware Assurance (OT) for hardware based OT products). In addition to Level 1 checks, it emphasizes secure software assurance such as robustness, fuzz/negative testing, vulnerability assessment, and secure coding practices. It enhances the physical security mechanisms of Security Level 1 by adding the requirement for tamper evidence, including tamper-evident coatings or seals or pick-resistant locks on removable covers or doors. Security Level 2 allows a cryptographic software module to be executed in an adaptable environment that implements role-based access controls or, at the minimum, a discretionary access control with the robust mechanism of defining new groups and assigning restrictive permissions through access control lists (e.g. ACLs), and with the capability of setting each user to more than one group, and that protects against unauthorised execution, modification, and reading of cryptographic software.	

	iii. **Security/Assurance Level 3:** Applies to enterprise-grade deployments requiring long-term security for sectors such as finance, telecom, and health. In addition to Level 2 requirements, it introduces stronger enterprise controls such as crypto-agility validation, stronger entropy validation expectations (TRNG/QRNG as applicable), centralized cryptographic management integration, and broader enterprise security assurance practices. It provides additional requirements to mitigate unauthorised access to SSPs held within the cryptographic module. Physical security mechanisms required at Security Level 3 are intended to have a high probability of detecting and responding to attempts at direct physical access, use or modification of the cryptographic module and probing through ventilation holes or slits. The physical security mechanisms may include solid enclosures and tamper detection/response circuitry that zeroise all CSPs when the removable covers/doors of the cryptographic module are opened. Security Level 3 requires identity-based authentication mechanisms, enhancing the security provided by the role-based authentication mechanisms specified for Security Level 2. A cryptographic module authenticates the identity of an operator and verifies that the identified operator is authorised to assume a specific role and perform a corresponding set of services. Security Level 3 requires manually established plaintext CSPs to be encrypted, utilise a trusted channel or use a split knowledge procedure for entry or output.	

	iv. **Security/Assurance Level 4:** Applies to sovereign-grade / critical information infrastructure protection. This level represents the most stringent assurance, including the strongest security assurance expectations (e.g., strategic resilience, rigorous supply chain assurance, nation-state attack simulation–type security validation), and strategic sector-specific requirements. The physical security mechanisms provide a complete envelope of protection around the cryptographic module to detect and respond to all unauthorised attempts at physical access when SSPs are contained in the module, whether external power is applied or not. Penetration of the cryptographic module enclosure from any direction is highly likely to be detected, resulting in the immediate zeroisation of all unprotected SSPs. Security Level 4 introduces the multi-factor authentication requirement for operator authentication. At a minimum, this requires two of the following three attributes. At Security Level 4, a cryptographic module is required to include special environmental protection features designed to detect voltage and temperature boundaries and zeroise all unprotected SSPs to provide a reasonable assurance that the module will not be affected when outside of the normal operating range in a manner that can compromise the security of the module.	
4.2.1	The product shall meet at least Level 2A for software products and Level 2B (ICT products) or 2C (OT products) for commercial usage (i.e. for Medium Risk usage) and therefore, shall cater to below functional requirements:	Detail of applicable Level to be submitted by the manufacturer.

4.2.2	The product shall enforce and validate multi-person (M-of-N) authorization controls for all critical cryptographic operations, including master key generation, activation, and destruction. Validation shall confirm that operations cannot be executed without the required quorum, that minimum M and N values are configurable based on assurance level, and that single-person compromise is technically prevented. All multi-person control events shall be securely logged, auditable, and resistant to bypass or circumvention. The M-of-N quorum shall be mandatorily enforced for any changes to the PQC Transition Policy, including the switching of operating modes from "Hybrid" to "PQC-Only" or "Classical-Only."	Test Case 32
4.2.3	The product shall enforce protocol-level protections against PQC parameter downgrade attacks, ensuring that adversaries cannot force negotiation of weaker security parameter sets when stronger options are available. Validation shall demonstrate strict enforcement of minimum approved parameter sets, rejection of downgrade attempts, immutable policy configuration, and conformance testing across supported protocols to ensure downgrade resistance cannot be bypassed.	Test Case 33
4.2.4	The OEM shall ensure that Vulnerability Assessment and Penetration Testing (VA/PT) of the system is carried out by a designated security lab by NCCS or Information Security Auditing Organization empanelled with CERT-In (MeitY, Government of India). For the cyber security audits that are conducted, ensure the adherence to latest guidelines issued by CERT-In or issued by Sectoral CERTs. The VA/PT report shall be submitted and reviewed for following:	Test Case 34
4.2.4.1	Ensure testing covered all application components — API endpoints, web UI, backend services, and data interfaces.	

4.2.4.2	Verify that cloud integrations and HSM interfaces were included in testing.	
4.2.4.3	Ensure both automated and manual testing were performed.	
4.2.4.4	Review the VA/PT report for: i. Classification of vulnerabilities (Critical, High, Medium, Low) ii. Risk rating and CVSS scoring iii. Recommended mitigations and closure evidence iv. Mitigation Verification v. Check that all Critical and High vulnerabilities have been remediated and re-tested.	
4.2.4.5	Validate that residual risk is documented and approved.	
4.2.4.6	Secure Coding practice shall be verified with steps as under:	
4.2.4.7	Static Analysis using tools.	
4.2.4.8	Manual code inspection to verify: i. Input validation and sanitization ii. Proper authentication and session management iii. Secure cryptographic implementations (e.g., no hardcoded keys) iv. Error/exception handling without information leakage v. Cryptographic Abstraction Layer-Verification shall confirm that all cryptographic functions are called via a standardized Abstraction Layer (API). Direct hard-coding of specific algorithms within application logic shall be prohibited.	
4.2.4.9	The limited lists such as OWASP Top 10, SANS Top 25 and similar, should not be considered as standards or references. Instead, discovery of all known vulnerabilities should be based on the comprehensive standards/frameworks.	
4.2.4.10	Check for vulnerabilities in third-party libraries	

4.2.4.11	Confirm use of version control with restricted access (multi factor authentication)	
4.2.4.12	Ensure code commits and merges require peer review and approval.	
4.2.4.13	Ensure adherence to latest design/secure coding guidelines issued by CERT-In and applicable Sectoral CERTs.	
4.2.4.14	For Hardware Assurance wherever applicable, inspection shall be done to validate Secure element, Trusted Execution Environment (TEE), Physically Unclonable Functions (PUFs), Secure boot attestation and tamper proofness.	
4.3	Additional requirements for Level 3 – Enterprise Grade	
4.3.1	Quantum/True RNG (QRNG/TRNG) Assurance	
4.3.1.1	The product shall integrate QRNG/TRNG compliant with TEC GRs or equivalent globally recognized standards.	Test Case 35
4.3.1.2	The QRNG/TRNG shall undergo validation of its claimed physical entropy source (quantum, optical, or other physical mechanisms).	
4.3.1.3	The entropy source validation shall demonstrate that the source is genuine, continuously operational, and not simulated or emulated.	
4.3.1.4	Validation shall include: i. Auditable scientific documentation of the entropy mechanism ii. Hardware BoM verification of entropy source components iii. Calibrated test evidence demonstrating entropy generation iv. Continuous health monitoring of the entropy source	
4.3.1.5	The QRNG/TRNG shall ensure that the entropy source cannot be spoofed, substituted, or disabled without detection.	

4.3.1.6	Calibration certificates for entropy sources shall be maintained and provided as validation evidence.	
4.3.1.7	The product shall ensure non-repudiation, integrity, and non-repetition of entropy-derived seed material across sessions, restarts, and lifecycle events.	
4.4	Performance and Resource Monitoring	
4.4.1	The product shall support measurement and reporting of memory utilization (heap/stack) and binary footprint for all cryptographic implementations.	Test Case 36
4.4.2	The product shall record CPU/GPU utilization during cryptographic operations including encapsulation/decapsulation and signature generation/verification.	
4.4.3	The product shall measure power consumption under idle, average, and peak workloads and compute energy per cryptographic operation.	
4.4.4	The product shall support scalability validation through concurrent cryptographic sessions under mixed workloads.	
4.4.5	The product shall enable packet-level analysis to assess bandwidth overhead, link utilization, and QoS under encrypted traffic conditions.	
4.4.6	Performance benchmarks shall be configurable based on sector-specific requirements.	
4.5	Crypto-Agility	
4.5.1	The product shall support switching between classical, hybrid, and PQC algorithms through CLI, GUI, or API interfaces.	Test Case 37
4.5.2	Cryptographic switching shall occur without system reboot and with minimal service disruption as per defined SLA requirements.	

4.5.3	The product shall support fallback to classical cryptography in case of PQC module failure, disablement, or overload without data loss.	
4.5.4	The product shall support seamless rollback to previously validated cryptographic configurations.	
4.5.5	The product shall support integration of new cryptographic algorithms through firmware or software updates.	
4.5.6	The product shall include a documented cryptographic transition and upgrade policy.	
4.6	Security Validation and Assessment	
4.6.1	The product shall undergo protocol-level security validation through advanced attack simulations including fault injection and multi-vector attacks.	Test Case 38
4.6.2	The product shall support automated vulnerability discovery and formal security analysis.	
4.6.3	Security audit and assessment reports shall be maintained and made available.	
4.7	Key Derivation and Cryptographic Integration	
4.7.1	The product shall derive final encryption keys using a secure Key Derivation Function (KDF).	Test Case 39
4.7.2	The KDF shall securely combine all applicable entropy sources including PQC outputs, QRNG seeds, and QKD-derived keys (where applicable).	
4.7.3	The product shall support secure integration with centralized cryptographic management systems.	
4.7.4	The integration shall ensure secure communication, authenticated access, and authorized management control.	
4.7.5	The product shall support reporting of cryptographic inventory, algorithm status, versions, and health metrics.	

4.7.6	Supported management protocols (e.g., SNMP, REST APIs) shall be documented and validated for confidentiality and integrity.	
4.8	CI/CD and Regression Validation	
4.8.1	The product shall support integration with CI/CD pipelines for automated regression testing.	Test Case 40
4.8.2	The CI/CD pipeline shall trigger validation upon code, configuration, or cryptographic changes.	
4.8.3	Automated testing shall include unit, integration, security, and interoperability tests.	
4.8.4	Builds failing defined quality or security criteria shall not be promoted.	
4.8.5	Test results and security findings shall be traceable and retained for audit purposes.	
4.8.6	The CI/CD process shall support repeatable builds, version control, and rollback capability.	
4.8.7	The product shall support validation across multiple platforms (e.g., Linux, Windows, ARM-based systems).	
4.9	Supply Chain and Compliance	
4.9.1	The product shall ensure supply chain security across hardware, firmware, software, and critical components.	Declaration to be submitted by manufacturer for applicable compliances. Test Case 41
4.9.2	The vendor shall support compliance with sector-specific cryptographic policies as notified by regulators.	
4.10	Additional requirements for Level 4 – Critical Infrastructure Security	
4.10.1	Indigenous and Customized Cryptography The product shall support validation of customized or indigenous cryptographic implementations – *not covered in this GR.*	For information only

4.10.2	Strategic Cryptographic Resilience	
4.10.2.1	The product shall support rapid cryptographic transition in case of algorithm compromise.	Test Case 42
4.10.2.2	The product shall support transition to alternative PQC algorithms or QKD within defined time limits without loss of security.	
4.10.2.3	The product shall support pre-configured fallback mechanisms and validated transition procedures.	
4.10.2.4	The product shall support algorithm diversification, hybrid models, and multiple trust anchors.	
4.10.3	QKD Integration The product shall support QKD integration readiness as per TEC GRs or equivalent standards.	Test Case 43
4.10.4	Fail-Secure Behavior The product shall ensure fail-secure operation under cryptographic and system failure and shall not fail open under any failure condition.	Test Case 44
4.10.5	Disaster Recovery and Business Continuity The product shall support Disaster Recovery (DR) and Business Continuity (BC). The product shall support multi-site deployment with secure state consistency. The product shall meet defined RTO and RPO requirements. The product shall support validated failover and failback mechanisms. DR testing shall include simulated failures ensuring no data loss or key compromise.	Test Case 45
4.10.6	Zero Trust Architecture The product shall comply with Zero Trust Architecture principles including continuous authentication and authorization.	Test Case 46
4.10.7	Red Teaming and Advanced Threat Validation	Test Case 47

	The product shall undergo independent red team testing simulating real-world attacks. Findings shall be documented, risk-rated, remediated, and revalidated.	
4.10.8	Advanced Supply Chain Assurance The product shall support semi-formal verification of component provenance and integrity. Critical security components shall undergo semi-formal verification for correctness and vulnerability absence. Verification scope and limitations shall be documented.	Test Case 48
4.10.9	Nation-State Threat Evaluation The product shall be evaluated against advanced persistent threat scenarios including supply chain and cryptographic attacks.	Test Case 49
4.10.10	Sector-Specific Extensions Additional requirements may be defined by strategic sectors and shall be treated as extensions to this framework.	Declaration to be submitted by manufacturer for applicable compliances.
	CHAPTER 5 Quality, Safety, EMI/EMC and General Requirements	
5.1	Quality requirements of a cryptographic system	
5.1.1	The manufacturer shall furnish the MTBF values. A minimum value of MTBF shall be 10,000 hours. The calculations shall be based on the guidelines specified in the standard.	The values shall be submitted by the manufacturer along with the calculation methodology.
5.1.2	The product/systems shall be manufactured by the international quality management system ISO 9001:2015, for	Certificate to be submitted for

<table>
<tr><td></td><td>which the manufacturer should be duly accredited. A quality plan describing the manufacturer's quality assurance system must be submitted.</td><td>ISO 9001:2015 compliance.

The Quality plan describing the quality assurance system shall be submitted by the manufacturer along with the quality check report of the finished product.</td></tr>
<tr><td>5.1.3</td><td>The product/systems shall conform to the requirements for the environment specified in document QM 333 {TEC 14016:2010}: " Standard for environmental testing of Telecommunication Equipment" The applicable tests shall be for environmental category B2, including vibration test.
Quality requirements of a cryptographic system
<table>
<tr><th>Sl. No.</th><th>Name of the Sub Parameter</th><th>Description of parameters and its range</th><th>Reference Standards (S)</th><th>Remarks</th></tr>
<tr><td>1.</td><td>Operating Temperature</td><td>0°C to +60°C</td><td>IEC/ISO</td><td>For defence</td></tr>
</table></td><td>Report from TEC designated test lab to be submitted.</td></tr>
</table>

				and space requirements to be met as per user specs.	
	2.	Humidity	10 to 90% RH	IEC/ISO	
	3.	Reliability	Availability / operational uptime expressed as a percentage (e.g., ≥ 99.0%, 99.9%, 99.99%)	--	To be defined by procurer
	4.	Basic environmental Test	Environmental category B2	QM-333	
	5.	MTBF (as per BSNL QM-115)	Metric		At least 100,00 hours
	6.	MTTR	Metric		To be defined by procurer

	7.	Manufactured process compliance	International quality management	ISO 9001:2015		
5.2	EMI/EMC Requirements of a cryptographic system The equipment shall conform to the EMC requirements as per the following standards and limits indicated therein. An accredited test agency shall furnish a test certificate and test report. a) Conducted and radiated emission: Name of EMC Standard: "CISPR 32: 2015 + A1: 2019- Limits and methods of measurement of radio disturbance characteristics of Information Technology Equipment". Limits: - i. To comply with Class B limits of CISPR 32:2015+A1:2019 ii. For Radiated Emission tests, limits below 1 GHz shall be as per relevant limits for measuring the distance of 10m and as per relevant limits for measuring the distance of 3m. b) Immunity to Electrostatic discharge (ESD): Name of EMC Standard: IEC 61000-4-2 {2025) "Testing and measurement techniques of Electrostatic discharge immunity test". Latest version is of 2025: Performance criteria and Applicable Performance Criteria requirements shall be mentioned as per standard practice. Limits: - i. Contact discharge level 2 {± 4 kV} or higher voltage; ii. Air discharge level 3 {± 8 kV} or higher voltage;					Report from TEC designated test lab to be submitted.

	c) Immunity to radiated RF: Name of EMC Standard: IEC 61000-4-3 (2020) "Testing and measurement techniques-Radiated RF Electromagnetic Field Immunity test". Limits: - For Telecom Equipment and Telecom Terminal Equipment with Voice interface (s) i. Under Test level 2 {Test field strength of 3 V/m} for general purposes in the frequency range 80 MHz to 1000 MHz and ii. Under test level 3 (10 V/m) for protection against digital radio telephones and other RF devices in the frequency ranges 800 MHz to 960 MHz and 1.4 GHz to 6.0 GHz. iii. For Telecom Terminal Equipment without Voice Interfaces iv. Under Test Level 2 (Test field strength of 3 V/m) for general purposes over 80 MHz–1000 MHz, and for protection against digital radio telephones and other RF devices over 800 MHz–960 MHz and 1.4 GHz–6.0 GHz. Latest version is of 2020: Performance criteria and Applicable Performance Criteria requirements shall be mentioned as per standard practice. d) Immunity to fast transients (burst): Name of EMC Standard: IEC 61000- 4- 4 {2012) "Testing and measurement techniques of electrical fast transients/burst immunity test". Limits: - Test Level 2, i.e.,	

	i. 1 kV for AC/DC power lines; ii. 0. 5 kV for signal/control/ data/telecom lines; e) Immunity to surges: Name of EMC Standard: IEC 61000-4-5:2014+A1:2017 "Testing & Measurement techniques for Surge immunity test". Limits: i. For mains power input ports: (a)2 kV peak open circuit voltage for line-to-ground coupling (b) 1 kV peak open circuit voltage for a line-to-line coupling ii. For telecom ports: (a) Peak open-circuit voltage of 2 kV, line-to-ground iii. Peak open-circuit voltage of 2 kV for line-to-line coupling Latest version is of 2017: Performance criteria and Applicable Performance Criteria requirements shall be mentioned as per standard practice. f) Immunity to conducted disturbance induced by Radiofrequency fields: Name of EMC Standard: IEC 61000-4-6 (2013) "Testing & measurement techniques-Immunity to conducted disturbances induced by radio- frequency fields". Limits: - i. Under the test level 2 {3 V r.m.s.} in the frequency range 150 kHz-80 MHz for AC / DC lines and Signal /Control/telecom lines.	

	g) Immunity to voltage dips & short interruptions (applicable to AC mains power input ports, if any): Name of EMC Standard: IEC 61000-4-11(2004) "Testing & measurement techniques- voltage dips, short interruptions and voltage variations immunity tests". Limits: i. A voltage dip corresponding to a reduction of the supply voltage of 30% for 500ms (i.e., 70 % supply voltage for 500ms) ii. A voltage dip corresponding to a reduction of the supply voltage of 60% for 200ms; (i.e., 40% supply voltage for 200ms) iii. A voltage interruption corresponds to a reduction of a supply voltage of > 95% for 5s. iv. A voltage interruption corresponds to a reduction of a supply voltage of >95% for 10s. h) Immunity to voltage dips & short interruptions (applicable to DC mains power input ports, if any): Name of EMC Standard: IEC 61000-4-29:2000: Electromagnetic compatibility (EMC) - Part 4-29: Testing and measurement techniques - Voltage dips, short interruptions and voltage variations on d.c. input power port immunity tests. Limits: i. Voltage Interruption with 0% of supply for 10ms. Applicable Performance Criteria shall be B. ii. Voltage Interruption with 0% of supply for 30ms, 100ms, 300ms and 1000ms. Applicable Performance Criteria shall be C.	

	iii. Voltage dip corresponding to 40% & 70% of supply for 10ms, 30 ms. Applicable Performance Criteria shall be B. iv. Voltage dip corresponding to 40% & 70% of supply for 100ms, 300 ms and 1000ms. Applicable Performance Criteria shall be C. v. Voltage variations corresponding to 80% and 120% of supply for 100 ms to 10s as per Table 1c of IEC 61000-4-29. Applicable Performance Criteria shall be B. **Note 1:** Classification of the equipment: Class B: Class B is a category of apparatus that satisfies the class B disturbance Limits. Class B is intended primarily for use in the domestic environment and may include the following: i. Equipment with no fixed place of use; for example, portable equipment powered by built-in batteries; ii. Telecommunication terminal equipment powered by the telecommunication networks iii. Personal computers and auxiliary connected equipment Please note that the domestic environment is an environment where the use of broadcast radio and television receivers may be expected within a distance of 10 m of the apparatus connected. **Class A:** Class A is a category of all other equipment that satisfies the class A limits but not the class B limits. **Note 2:** The testing agency for EMC tests shall be an accredited agency and details of accreditation shall be submitted.	

	Note 3: For checking compliance with the above EMC requirements, the method of measurements shall follow TEC Standard No. TEC 11016:2016 (TEC/SD/DD/EMC-221/05/OCT-16) and the references mentioned therein unless otherwise specified. Alternatively, corresponding relevant Euro Norms of the above IEC/CISPR standards are also acceptable subject to the condition that frequency range and test level are met as per the above mentioned sub clauses (a) to (h) and TEC Standard No. TEC 11016:2016 (earlier no. TEC/SD/DD/EMC-221/05/OCT-16). IEC/CISPR — Euro Norm CISPR 11 — EN 55011 CISPR 32 — EN 55032 IEC 61000-4-2 — EN 61000-4-2 IEC 61000-4-3 — EN 61000-4-3 IEC 61000-4-4 — EN 61000-4-4 IEC 61000-4-5 — EN 61000-4-5 IEC 61000-4-6 — EN 61000-4-6 IEC 61000-4-11 — EN 61000-4-11 IEC 61000-4-29 — EN 61000-4-29	
5.3	**Safety Requirements of a cryptographic system**	
5.3.1	**Electrical safety** The equipment shall conform to IS/IEC 62368-1:2023 "Audio/video, information and communication technology	Report from TEC designated test lab to

	equipment as prescribed under Table no. 1 of the TEC document 'SAFETY REQUIREMENTS OF TELECOMMUNICATION EQUIPMENT": TEC10009: 2024". In case remote line powering is applied, it should comply with [ITU-T K.50], [ITU-T K.51] and [IEC 60950-21]. The safe working practices described in [ITU-T K.64] should be followed when work is carried out outside plant electronic equipment.	be submitted.
5.3.2	Laser safety Laser safety: The equipment shall conform to IEC 60825-2:2021 for products emitting laser radiation as prescribed under Table no. 1 of the TEC document 'SAFETY REQUIREMENTS OF TELECOMMUNICATION EQUIPMENT": TEC10009: 2024". ***Note:*** *This test shall be applicable if laser components are directly mounted in the box.*	Report from TEC designated test lab to be submitted, if applicable.
5.3.3	General Requirements	
5.3.3.1	The system shall support In-field firmware upgrades from time to time for a continuation of functionality with the advancement of technology and interoperable and supporting systems to make it compatible.	Test case 50
5.3.3.2	It shall support remote system Software/Firmware upgrades.	
5.3.3.3	As and when software bugs are found/ determined, the Manufacturer shall provide patches/firmware replacement, if involved, as mutually agreed between the Purchaser of the instrument and supplier. Modified documentation, wherever applicable, shall also be supplied.	Declaration by the manufacturer.
5.3.3.4	The manufacturer/supplier shall furnish the list of recommended spares.	Manufacturer's Declaration along with the List of

		Recommended Spares
5.3.3.5	The supplier shall have a maintenance/repair facility in India. The supplier shall furnish MTBF and MTTR values.	Declaration by the manufacturer.
5.3.3.6	The accessories cables shall have a low attenuation cable link, either optical or Ethernet cable of the latest. The vendor will submit the Specification for the same.	A declaration should be submitted by the manufacturer along with the values of the specifications.
5.3.3.7	It must be possible for an operator to select a particular encryption scheme for payload encryption system wide.	Will be checked as per Crypto Agility Test Case
5.3.3.8	It shall automatically exchange a new session key on a pre-set interval of 1-60 minutes or user configured.	Manual test as per user configured value.
5.3.3.9	The new session key shall be generated automatically by a True Random Number Generator (TRNG) or a Pseudo Random Number Generator (PRNG). QRNGs are preferred over other TRNGs and PRNGs.	RNG testing done in Chapter 2.
5.3.3.10	These devices should support high entropy throughput with very high randomness (entropy)	Test case 51
5.3.3.11	It shall provide confidentiality-protected firmware/software upgrades.	Declaration by the manufacturer.

5.3.3.12	The encryption devices should be future-proof and fully reprogrammable for an upgrade to new algorithms based on the user requirements or availability of technology from time to time.	Declaration by the manufacturer.
5.3.3.13	Cryptographic system can also support Quantum-safe key exchange algorithms under the standardisation process of NIST, along with classical algorithms in a hybrid manner.	Declaration by the manufacturer.
5.3.3.14	Remote management should be possible only through secure Management software with minimum 2-factor authentication with hardware binding.	Declaration by the manufacturer.
5.3.3.15	Cryptographic system shall support SNMPv3 or the latest and shall provide multiple manager support.	Declaration by the manufacturer.
5.3.3.16	Cryptographic system shall support audit and event logging with Syslog support.	Declaration by the manufacturer.
5.3.3.17	Cryptographic system shall be able to work with the NTP server for time synchronisation.	Declaration by the manufacturer.
5.3.3.18	Cryptographic system shall be able to work with RADIUS or TACAS+ server for authentication.	Test case 52
5.3.3.19	Repair procedure; i. List of replaceable parts used to include their sources and the approving authority. ii. Detailed ordering information for all the replaceable parts shall be listed in the manual to facilitate the reordering of spares as and when required. iii. A systematic procedure for troubleshooting and sub-assembly replacement shall be provided. Test fixtures and accessories required for repair shall also be indicated.	Check whether the Repair manual provided by the manufacturer covers the required aspects.

	iv. Systematic troubleshooting procedures shall be given for the probable faults with their remedial actions. ***Note:*** *The Purchaser may mention the repair manual requirement at the time of ordering.*	
5.3.3.20	Technical literature in Hindi or English of the instrument with block schematic diagrams shall be provided. The complete layout and circuit diagrams of various assemblies with test voltages and waveforms at different test points of the units shall be provided, wherever required. All aspects of installation, operation, maintenance and repair shall be covered in the manuals. It is suggested that both the Soft copy / QR code and Hard copy to be provided. Soft copy shall be provided both in English and Hindi. The manual shall include the following two parts: i. Installation, operation and maintenance manual. ii. Safety measures to be observed in handling the equipment. iii. Precautions for setting up, measurements and maintenance iv. Product/equipment required for routine maintenance and calibration, including their procedures. v. Illustration of internal and external mechanical parts. vi. A detailed description of the operation of the software used in the equipment, including its installation, loading and debugging etc.	The details shall be provided by the manufacturer.
5.3.3.21	Identification of Equipment i. Equipment shall be marked with the supplier's or Manufacturer's logo/name. ii. The Model No., serial No., The month and year of manufacture shall be indicated by screen printing on	Physical check

	the body of the equipment or by a tamper-proof sticker pasted on the body of the equipment. iii. Power Supply requirements shall be indicated on the body. iv. The above markings shall be legible, indelible and easily visible.	
	Chapter 6	
6.1	Information for the procurer of the product	
6.1.1	The procurer should require the vendor to submit Bill of Materials (BOM). The BOM submitted by the vendor should be in adherence to the SBOM, CBOM and HBOM guidelines issued by CERT-In or other sectoral CERTs.	For information of the procurer.
6.1.2	The procurer should require the vendor to demonstrate cryptographic agility as an operational capability and not merely as design intent. The vendor shall provide documented procedures for algorithm addition, replacement, and deprecation without system downtime or architectural redesign.	For information of the procurer.
6.1.3	The procurer should clearly specify acceptable cryptographic assurance mechanisms, such as compliance with ISO/IEC 19790, ISO/IEC 24759, FIPS 140-3, or Common Criteria.	For information of the procurer.
6.1.4	The procurer should require that cryptographic policies (algorithm selection, key sizes, protocol versions) be configurable by the procuring entity and protected against unauthorized modification.	For information of the procurer.
6.1.5	During migration phases, the procurer should mandate support for hybrid cryptographic modes (classical + PQC) with explicit controls to prevent downgrade attacks.	For information of the procurer.
6.1.6	The procurer should require the vendor to disclose the performance, latency, power, and resource impact of post-	For information of the procurer.

<table>
<tr><td></td><td>quantum and hybrid cryptographic mechanisms under expected load conditions.</td><td></td></tr>
<tr><td>6.1.7</td><td>The procurer should define minimum support and maintenance periods for cryptographic components, including guaranteed availability of security updates until system end-of-life.</td><td>For information of the procurer.</td></tr>
<tr><td>6.1.8</td><td>The applicability of clauses of GR as per product security Level is also tabulated as under-
<table>
<tr><th rowspan="2">GR Chapter</th><th colspan="4">Security/Assurance Levels</th></tr>
<tr><th>Level 1</th><th>Level 2 (2A, 2B & 2C)</th><th>Level 3</th><th>Level 4</th></tr>
<tr><td>CH-1 Introduction</td><td colspan="4">No test requirements mentioned, Chapter cover introduction of classical and post quantum algorithms with introduction of cryptographic modules</td></tr>
<tr><td>Ch-2 - Functional requirements – covers Core cryptographic modules, Protocol/Application Layer Requirements, Crypto agility, Requirements for constrained devices and</td><td colspan="4">Applicable to all levels</td></tr>
</table></td><td>This is for information purpose.</td></tr>
</table>

	Cryptography –as –a-service.			
	Chapter -3 Operational, Interface and Interoperability Requirements – covers details about interface of product, general description of product like throughput, latency etc. and interoperability requirements like IPv4/IPv6 etc.	Applicable to all levels		

<table>
<tr><td></td><td>
<table>
<tr><td>Chapter 4 – Security Requirements</td><td>--</td><td>Testing as per clause 4.3.</td><td>Additional as per clause 4.4</td><td>Additional as per clause 4.5</td></tr>
<tr><td>Chapter 5- Quality, Safety, EMI/EMC and General Requirements</td><td colspan="4">Applicable to all levels</td></tr>
</table>
</td><td></td></tr>
<tr><td>6.2</td><td>The below clauses from above chapters of GR are to be decided by procurer.
<table>
<tr><th>Sl. No.</th><th>GR clause / location (as written in draft)</th><th>What the procurer must decide</th></tr>
<tr><td>1.</td><td>Ch-3 Operational requirements table – “No of Concurrent connection”</td><td>Concurrency capacity sizing</td></tr>
<tr><td>2.</td><td>Ch-3 Operational requirements table – “Support of Jumbo frames”</td><td>Jumbo frame MTU requirement</td></tr>
<tr><td>3.</td><td>Ch-3 Power table – AC supply connector type & redundancy</td><td>Power feed and redundancy</td></tr>
<tr><td>4.</td><td>Ch-3 Interoperability table – Clock accuracy</td><td>Time/clock requirements</td></tr>
<tr><td>5.</td><td>Ch-5 Quality requirements table – “Reliability (Availability/uptime)”</td><td>Minimum availability target</td></tr>
<tr><td>6.</td><td>Ch-5 Quality requirements table – “MTTR”</td><td>Maximum MTTR target</td></tr>
</table>
</td><td>Values of these to be given by the procurer and to be implemented by the manufacturer.</td></tr>
</table>

	7.	Ch-5 General requirements – patch/firmware replacement ("…as mutually agreed between the Purchaser… and supplier")	Patch/SW update SLA and process		
	8.	Ch-5 General requirements – Supported encryption/hashing/signature algorithms	Required algorithm/cipher-suite		
	9.	Ch-5 General requirements – Session key change interval ("pre-set interval 1–60 minutes or user configured")	Rekey interval policy		
	10.	Ch-5 General requirements – RNG preference ("QRNGs/TRNG")	Required entropy source assurance		
	11.	Ch-5 Repair manual note ("Purchaser may mention the repair manual requirement at the time of ordering")	Whether repair manual is required		

I. TEST SETUP & PROCEDURES:

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 1 - Verification of Symmetric Cryptographic Mechanisms
2.	GR Clause covered	2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.6, 2.1.7, 2.1.8, 2.1.9, 2.1.10, 2.1.15, 2.1.17, 2.1.22, 2.1.23
3.	Objective of the test	To verify that the cryptographic processing module correctly implements approved symmetric cryptographic algorithms, modes of operation, padding schemes, IV/nonce handling, authenticated encryption mechanisms and secure error handling while rejecting insecure algorithms and invalid inputs.
4.	Tools required	• NIST AES Known Answer Test (KAT) vectors and CAVP vectors - https://github.com/usnistgov/ACVP-Server/tree/master/gen-val/json-files. • Hex editor/viewer – input/output comparison
5.	Test procedure	Step 1: Identify all supported symmetric cryptographic algorithms and modes. Step 2: Verify support for approved algorithms such as: • AES-192 / AES-256 • ChaCha20-Poly1305 Step 3: Verify support for approved modes such as: • GCM • CTR • CBC (with integrity protection) • CFB • OFB • XTS for Storage

<table>
<tr><th>Sl. No.</th><th>Test Details</th><th>Details</th></tr>
<tr><td></td><td></td><td>
• Other modes
Step 4: Perform encryption/decryption operations using approved algorithms and compare outputs with NIST KAT vectors.
Note – The NIST KATs are available in prompt.json file of Git repository mentioned in tools required. The expected output is mentioned in expectedResults.json.
Step 5: Verify correct IV/nonce generation and uniqueness and non-deterministic behavior where applicable, such that repeated operations (e.g., encryption or signature generation) using the same input and key produce different outputs when required by the algorithm.

Configure same plaintext, same symmetric key during multiple encryption operations and capture Generated IVs/Nonces and ciphertexts. Verify that IV/nonce changes every operation and the length is correct i.e.
<table>
<tr><th>Mode</th><th>Expected IV/Nonce Length</th></tr>
<tr><td>AES-CBC</td><td>128 bits</td></tr>
<tr><td>AES-GCM</td><td>96 bits recommended</td></tr>
<tr><td>ChaCha20-Poly1305</td><td>96 bits</td></tr>
</table>
Run for 1000 encryptions and export IV/nonces. Check that IV are not reused and also check entropy of IVs using statistical distributions.

Step 6: Verify integrity protection using authenticated encryption modes to ensure AEAD modes detect Ciphertext tampering, Tag modification, AAD modification and Nonce misuse in AES-GCM, AES-CCM and ChaCha20-Poly1305.

Encrypt plaintext and then modify 1 bit in ciphertext, and then attempt decryption. Decryption must fail.

</td></tr>
</table>

Sl. No.	Test Details	Details
		• Modify one bit of GCM tag and check decryption operation is rejected. • If Additional Authenticated Data (AAD) is used, encrypt with AAD and change AAD during decryption. It should result in Authentication failure. • Change nonce during decryption and check for failure. Step 7: Attempt operations using deprecated or insecure algorithms/modes such as: • DES • RC4 • ECB mode Verify these are rejected and disabled by default. Step 8: Attempt operations using malformed ciphertexts, invalid padding, null inputs and oversized inputs. • Generate valid ciphertext. Modify 1 bit or random bytes or truncate ciphertext. Check if module rejects malformed ciphertext. Verify that module shall not hang, crash or leaks memory. • Padding is applicable to AES-CBC, PKCS#7 padding modes. If plaintext length ≠ block size, padding is added. To test, generate valid ciphertext and corrupt padding bytes. Check module should return "Invalid padding" or "Decryption failed". It must NOT output partial plaintext, crash or leak padding details. • Verify secure handling of empty or NULL parameters i.e. NULL plaintext, null key, null nonce or null ciphertext. Verify graceful rejection of null cases without any crash or undefined behaviour.

Sl. No.	Test Details	Details
		• Test for very large plaintexts, cipher texts or oversized AADs (eg. 10 MB/100 MB size file). Verify encryption/decryption succeeds correctly or resource limit error is displayed without crash or memory corruption. • Check rejection for other malformed cases like truncated ciphertext, wrong IV size, wrong key size.
6.	Expected Result	Test passes if: • Approved symmetric algorithms and modes are correctly implemented • Encryption/decryption outputs match reference test vectors • IVs/nonces are generated correctly and uniquely • Integrity verification failures are detected correctly • Deprecated algorithms and insecure modes are rejected • Invalid or malformed inputs are securely rejected • Interoperability with standard implementations is achieved

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 2 - Verification of Correct Implementation of Hashing, Message Authentication mechanism and Digital Signature Mechanisms
2.	GR Clause covered	2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.8, 2.1.9, 2.1.10, 2.1.11, 2.1.12, 2.1.13, 2.1.15, 2.1.17, 2.1.22, 2.1.23
3.	Objective of the test	To verify that the cryptographic processing module correctly implements approved hashing and MAC algorithms and produces

Sl. No.	Test Details	Details
		outputs matching standard test vectors while securely rejecting invalid operations.
4.	Tools required	• NIST SHA and HMAC test vectors - https://github.com/usnistgov/ACVP-Server/tree/master/gen-val/json-files. • Hex editor/viewer
5.	Test procedure	**Step 1:** Identify all supported hashing, MAC and digital signature algorithms. Step 2: Verify support for approved hashing algorithms such as: • SHA-256 / SHA-384 / SHA-512 • SHA3-256 / SHA3-384 / SHA3-512 • SHAKE128 / SHAKE256 • HMAC-SHA-256 / 384 / 512 • HMAC-SHA3 variants • KMAC128 / KMAC256 **Step 3:** Verify support for approved digital signature algorithms and parameter sets such as: • DSA-2048 / 3072 • ECDSA P-256 / P-384 / P-521 • ML-DSA-44 / 65 / 87 • SLH-DSA approved parameter sets **Step 4:** Generate hash/MAC outputs for standard test messages and compare with reference vectors. **Step 5:** Generate digital signatures using KeyGen and generate signatures using SigGen and then verify using SigVer. Verify with outputs of reference vectors. ***Note*** *– The NIST KATs are available in prompt.json file of Git repository mentioned in tools required. The expected output is mentioned in expectedResults.json file.*

Sl. No.	Test Details	Details
		Step 6: Perform incremental hashing by supplying message data in multiple sequential chunks and verify that the final computed hash matches the hash generated from the complete message processed in a single operation. Step 7: Verify rejection of malformed inputs: • Attempt digital signature verification using a modified message while retaining the original signature and verify that the signature verification operation fails securely without module crashing or hanging. • Attempt digital signature verification using a modified or corrupted signature value and verify that the cryptographic module rejects the invalid signature. • Attempt digital signature verification using an invalid, corrupted or unrelated public key and verify that the signature verification operation fails securely. Step 8: Verify rejection of deprecated or insecure algorithms such as: • Attempt hashing operations using deprecated or insecure algorithms such as MD5 and SHA-1 and verify that the cryptographic module rejects, disables or appropriately flags insecure algorithm usage as per security policy. • Attempt digital signature operations using weak or deprecated signature parameter sets and verify that the cryptographic module securely rejects unsupported or insecure cryptographic parameters. Step 9: Verify error handling/logging for invalid operations and unsupported algorithms. • Attempt cryptographic operations using unsupported hashing, MAC or digital signature algorithms and verify that the module

Sl. No.	Test Details	Details
		generates appropriate error handling, logging and controlled failure indications without abnormal termination. • Attempt invalid hashing, MAC or digital signature operations including malformed inputs, incorrect key sizes and invalid parameters and verify that proper error logging and secure failure handling mechanisms are implemented.
6.	Expected Result	Test passes if: • Approved hashing, MAC and signature algorithms operate correctly • Generated outputs match reference vectors • Incremental hashing operates correctly • Modified messages fail MAC, hashing, signature verification • Deprecated algorithms are rejected • Invalid inputs and keys are securely rejected and logging is done.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 3 - Verification of Correct Implementation of Asymmetric Cryptographic Mechanisms
2.	GR Clause covered	2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.8, 2.1.9, 2.1.10, 2.1.15, 2.1.17, 2.1.22, 2.1.23
3.	Objective of the test	To verify that the cryptographic processing module correctly implements approved asymmetric cryptographic algorithms and supports only approved key sizes and parameter sets for encryption/decryption, key exchange, key encapsulation and digital

Sl. No.	Test Details	Details
		signatures. The module shall reject invalid, malformed, weak, deprecated or unsupported keys and algorithms.
4.	Tools required	• NIST Known Answer Test (KAT) vectors and CAVP vectors - https://github.com/usnistgov/ACVP-Server/tree/master/gen-val/json-files. • Hex editor/viewer – Input/output comparison
5.	Test procedure	**Step 1:** Identify all asymmetric cryptographic algorithms supported by the EUT. **Step 2:** Verify support for approved classical algorithms and key sizes such as: • RSA-2048 / RSA-3072 / RSA-4096 • DH-2048 / DH-3072 • ECDH/ECDSA P-256 / P-384 / P-521 **Step 3:** Verify support for approved post-quantum algorithms and parameter sets such as: • ML-KEM-512 / 768 / 1024 **Step 4:** Generate valid key pairs using approved algorithms and key sizes. **Step 5:** Perform encryption/decryption, key exchange/key encapsulation, signing and verification operations using approved keys. **Step 6:** Compare outputs with reference test vectors or known-answer test vectors. ***Note*** *– The NIST KATs are available in prompt.json file of Git repository mentioned in tools required. The expected output is mentioned in expectedResults.json file.* **Step 7:** Attempt operations using invalid, malformed or weak keys such as: • Unsupported key lengths • Corrupted public/private keys

Sl. No.	Test Details	Details
		• Invalid elliptic curve parameters • Invalid PQC parameter sets different from defined in FIPS 203 Step 8: Verify rejection of deprecated or insecure algorithms and parameters such as: • RSA keys below approved size (RSA ≥ 2048-bit) • Weak DH groups • Deprecated curves or parameters (ECC >= 224 bit) Step 9: Verify that appropriate error messages/logs are generated for rejected algorithms or invalid keys.
6.	Expected Result	Test passes if: • Approved asymmetric algorithms and parameter sets are correctly implemented • Encryption/decryption, encapsulation/decapsulation operations succeed correctly • Generated outputs match reference test vectors • Only approved key sizes and parameter sets are accepted • Invalid, malformed, weak or unsupported keys are rejected • Deprecated or insecure algorithms/configurations are rejected • Appropriate error indication/logging is generated for failed operations

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 4 - Verification of Cryptographic Agility and Interoperability Mechanisms
2.	GR Clause covered	2.1.16, 2.1.22

Sl. No.	Test Details	Details
3.	Objective of the test	To verify that the cryptographic processing module supports cryptographic agility through configuration or policy-based control of algorithms, modes and key sizes, and ensures interoperability such that cryptographic outputs generated by one compliant implementation can be successfully processed by another standards-compliant implementation.
4.	Tools required	• OpenSSL CLI/OpenSSL 3.x Provider • Open Quantum Safe (OQS-OpenSSL/OQS Provider) • Python with cryptography / pycryptodome / pqcrypto libraries • Interoperability peer systems implementing standard cryptographic algorithms • Wireshark/packet analyzer – protocol interoperability validation • Configuration management/policy control interface of EUT • NIST Known Answer Test (KAT) vectors and ACVP/CAVP vectors • Hex editor/viewer – output comparison
5.	Test procedure	Step 1: Identify all cryptographic algorithms, modes, key sizes and parameter sets supported by the EUT. Step 2: Verify that cryptographic algorithms, modes and key sizes can be enabled, disabled or restricted through configuration or policy control. Step 3: Configure the module to allow only approved algorithms and verify successful operation using: • AES-256-GCM • RSA-3072 • ECDSA P-384 • ML-KEM / ML-DSA approved parameter sets Step 4: Disable selected algorithms/modes/key sizes through policy configuration and verify that the module rejects their usage.

Sl. No.	Test Details	Details
		Step 5: Attempt operations using deprecated or insecure algorithms/configurations such as: • DES • RC4 • MD5 • SHA-1 • ECB mode • RSA-1024 **Step 6:** Verify that deprecated or insecure algorithms are disabled/restricted without modification. **Step 7:** Generate cryptographic outputs (ciphertexts, hashes, signatures, shared secrets, encapsulated keys) using the EUT. **Step 8:** Process generated outputs using an independent standards-compliant implementation such as OpenSSL/OQS-OpenSSL and verify successful decryption, signature verification, decapsulation or integrity validation. **Step 9:** Generate cryptographic outputs using the external compliant implementation and verify successful processing by the EUT. **Step 10:** Verify interoperability for the following operations where supported: • AES encryption/decryption • RSA encryption/signature verification • ECDSA signing/verification • DH/ECDH shared secret derivation • ML-KEM encapsulation/decapsulation • ML-DSA/SLH-DSA signature verification • Hybrid X25519 + ML-KEM operations **Step 11:** Compare outputs against applicable NIST KAT vectors and verify standards compliance. **Step 12:** Verify appropriate logging/error indication for disabled, restricted or unsupported algorithms and configurations.
6.	Expected Result	Test passes if:

Sl. No.	Test Details	Details
		• Cryptographic algorithms, modes and key sizes can be securely enabled, disabled or restricted through configuration or policy control • Deprecated or insecure algorithms/configurations are successfully disabled or rejected without system redesign • Cryptographic outputs generated by the EUT are successfully processed by compliant external implementations • Cryptographic outputs generated by compliant external implementations are successfully processed by the EUT • Outputs conform to applicable NIST KAT and ACVP/CAVP reference vectors • Only approved algorithms, modes and key sizes are accepted under enforced policy • Appropriate logging/error indication is generated for restricted or unsupported operations • Interoperability with standards-compliant implementations is achieved for all supported cryptographic operations

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 5 - Verification of Protection Against Side-Channel Attacks, performance analysis and secure erasure of cryptographic keys, intermediate values, and sensitive data from memory
2.	GR Clause covered	2.1.18, 2.1.19, 2.1.20
3.	Objective of the test	To verify that the cryptographic processing module implements appropriate protections against side-channel attacks including timing-

Sl. No.	Test Details	Details
		based leakage, memory-based leakage and shall operate within defined performance limits for all supported cryptographic operations under specified operating conditions and shall be suitable for the intended deployment environment and shall securely erase cryptographic keys, intermediate values, and sensitive data from memory after use to prevent residual data exposure.
4.	Tools required	• Timing analysis tools (time/perf/rdtsc based measurement utilities) • Memory monitoring and debugging tools (Valgrind, memory dump analyzers) • Side-channel analysis tools (ChipWhisperer, Cachegrind or equivalent where applicable) • CPU/cache monitoring tools • Python scripting tools for repeated execution and statistical analysis • OpenSSL/Open Quantum Safe implementations • System monitoring/logging utilities
5.	Test procedure	Step 1: Identify all cryptographic operations implemented by the EUT involving sensitive material such as cryptographic keys, intermediate states and shared secrets. Step 2: Execute repeated cryptographic operations using different secret keys and identical plaintext/message inputs while measuring execution time. Record execution times Step 3: Analyze execution timing variations and verify that cryptographic operations do not expose secret-dependent timing differences beyond acceptable implementation variation. Step 4: Monitor and record memory usage during cryptographic operations and verify that cryptographic keys, plaintexts, intermediate values and shared secrets are not exposed through residual memory leakage after operation completion.

Sl. No.	Test Details	Details
		Step 5: Perform repeated encryption, decryption, signature and key agreement operations while monitoring cache access patterns, branch behavior and observable implementation-dependent variations where applicable. **Step 6:** Verify that constant-time implementation techniques or equivalent protections are implemented for sensitive cryptographic operations wherever applicable. **Step 7:** Attempt memory inspection after cryptographic operation completion and verify that sensitive data has been securely cleared or is inaccessible. **Step 8:** Perform stress and repeated-operation testing under varying system load conditions and verify that no additional information leakage occurs. **Step 9:** Review implementation documentation, security architecture and available mitigation mechanisms for protections against timing, cache, memory and related side-channel attacks. **Step 10:** Verify that side-channel protections remain effective for supported classical, PQC and hybrid cryptographic mechanisms where applicable.
6.	Expected Result	Test passes if: • Cryptographic operations do not exhibit observable secret-dependent timing behavior exploitable for side-channel attacks • Sensitive data such as cryptographic keys, intermediate values and shared secrets are not exposed through memory leakage after operation completion • Cache access patterns, branching behavior and implementation characteristics do not reveal sensitive cryptographic information beyond acceptable limits

Sl. No.	Test Details	Details
		• Appropriate constant-time or equivalent side-channel mitigation mechanisms are implemented for sensitive operations • Sensitive information is securely cleared from memory after use • No unintended information disclosure occurs during stress or repeated-operation testing • Side-channel protections operate effectively for supported cryptographic mechanisms in the intended deployment environment

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 6 - Verification of Hybrid Cryptographic Schemes (Classical + PQC)
2.	GR Clause covered	2.2 (i), 2.2 (ii), 2.2 (iii), 2.2 (iv), 2.2 (v)
3.	Objective of the test	To verify that the cryptographic processing module correctly implements hybrid cryptographic schemes combining classical and post-quantum cryptographic mechanisms using approved composition methods, ensures no downgrade to weaker algorithms, validates correctness and security of individual components, supports interoperability across compliant implementations and provides documented validation and verification procedures for hybrid operations.
4.	Tools required	• Open Quantum Safe (OQS-OpenSSL/OQS Provider) • OpenSSL CLI/OpenSSL 3.x Provider • Python with cryptography / pqcrypto libraries

Sl. No.	Test Details	Details
		• NIST PQC Known Answer Test (KAT) vectors • Wireshark/packet analyzer – interoperability validation • Hex editor/viewer – input/output comparison
5.	Test procedure	**Step 1:** Identify all supported hybrid cryptographic schemes and composition methods implemented by the EUT. **Step 2:** Verify support for approved hybrid schemes such as: • X25519 + ML-KEM • ECDHE + ML-KEM • RSA/ECDSA + ML-DSA/SLH-DSA hybrid signature schemes **Step 3:** Review available documentation and verify that documented procedures exist for validation, verification, testing and operational handling of hybrid cryptographic mechanisms. **Step 4:** Verify that hybrid constructions combine classical and PQC components using approved composition methods such as: • Concatenation followed by Key Derivation Function (KDF) • Approved hybrid key derivation mechanisms **Step 5:** Perform hybrid key agreement, encapsulation/decapsulation and signature operations using approved parameter sets. **Step 6:** Independently validate correctness of classical components using applicable NIST KAT and ACVP/CAVP vectors. **Step 7:** Independently validate correctness of PQC components using applicable NIST PQC KAT vectors. **Step 8:** Verify that hybrid outputs such as shared secrets, ciphertexts and signatures are correctly generated and processed. **Step 9:** Attempt downgrade or fallback scenarios by disabling PQC components or forcing weaker classical algorithms and verify that: • Downgrade to weaker algorithms is rejected • Hybrid security policy enforcement remains active

Sl. No.	Test Details	Details
		• Unauthorized fallback operations fail securely **Step 10:** Attempt operations using invalid or malformed hybrid keys, ciphertexts, signatures and parameter combinations. **Step 11:** Capture network traffic during hybrid cryptographic operations using Wireshark or equivalent packet analyzer. **Step 12:** Verify from captured protocol exchanges that: • Both classical and PQC components are negotiated and exchanged • Approved hybrid algorithms and parameter sets are used • No fallback to weaker or deprecated algorithms occurs • Hybrid key exchange/signature negotiation is successfully completed •Cipher suites and negotiated security parameters conform to configured policy **Step 13:** Verify interoperability for both classical and PQC portions of the hybrid scheme by exchanging hybrid cryptographic outputs between the EUT and independent compliant implementations and verifying successful processing of: • Hybrid shared secrets • Hybrid ciphertexts • Hybrid signatures
6.	Expected Result	Test passes if: • Hybrid cryptographic schemes correctly combine classical and PQC components using approved composition methods • Hybrid outputs are correctly generated and processed • Classical and PQC components independently satisfy their respective correctness and security requirements • Downgrade or fallback to weaker algorithms is not permitted • Invalid or malformed hybrid keys, ciphertexts and signatures are securely rejected

Sl. No.	Test Details	Details
		• Interoperability is achieved between compliant implementations for hybrid operations • Hybrid shared secrets, ciphertexts and signatures are successfully exchanged and processed • Appropriate validation and verification procedures for hybrid cryptographic operations are documented and available

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 7 - Verification of Post-Quantum Cryptographic Algorithm Validation
2.	GR Clause covered	2.3.1 (i), 2.3.1 (ii), 2.3.1 (iii), 2.3.1 (iv)
3.	Objective of the test	To verify that the cryptographic processing module correctly implements approved post-quantum cryptographic algorithms using officially published parameter sets, conforms to official Known Answer Tests (KAT), reference implementations and test vectors, rejects reduced or non-standard parameter sets and correctly implements probabilistic components such as randomness generation, noise sampling and rejection sampling in accordance with applicable PQC algorithm specifications.
4.	Tools required	• Open Quantum Safe (OQS-OpenSSL/OQS Provider) • NIST PQC Known Answer Test (KAT) vectors • Official NIST PQC reference implementations

Sl. No.	Test Details	Details
		• ACVP/CAVP validation vectors where applicable • Statistical analysis tools (Dieharder/NIST Statistical Test Suite) • Hex editor/viewer – input/output comparison • Memory and debugging tools
5.	Test procedure	**Step 1:** Identify all supported post-quantum cryptographic algorithms and parameter sets implemented by the EUT. **Step 2:** Verify support only for officially approved or published parameter sets for supported PQC algorithms such as: • ML-KEM approved parameter sets • ML-DSA approved parameter sets • SLH-DSA approved parameter sets **Step 3:** Verify that domain parameters, key sizes, polynomial dimensions, modulus values, security levels and internal constants conform to official algorithm specifications and published standards. **Step 4:** Load official NIST Known Answer Test (KAT) vectors and reference implementation test vectors for supported PQC algorithms. **Step 5:** Perform encapsulation/decapsulation, encryption/decryption, signing and verification operations using the official KAT vector inputs. **Step 6:** Compare generated outputs such as ciphertexts, shared secrets, signatures and recovered plaintexts against expected outputs from official KAT vectors and reference implementations. **Step 7:** Verify interoperability by comparing EUT outputs with outputs generated by official PQC reference implementations and confirming successful cross-verification. **Step 8:** Attempt operations using reduced, modified or non-standard parameter sets such as: • Reduced polynomial dimensions • Modified modulus values

Sl. No.	Test Details	Details
		• Unsupported security levels • Non-approved internal constants **Step 9:** Verify that non-standard, reduced or weakened parameter sets are rejected by the module. **Step 10:** Execute repeated PQC operations such as encapsulation, key generation and signature generation multiple times using identical input conditions and verify that generated random values, ciphertexts, signatures or shared secrets do not exhibit abnormal repetition or deterministic behavior inconsistent with the applicable PQC algorithm specification. **Step 11:** Where supported by the implementation, review randomness generation configuration, entropy source usage and available statistical health-check logs or reports. **Step 12:** Verify appropriate error indication/logging for unsupported, modified or malformed parameter sets and invalid PQC operations.
6.	Expected Result	Test passes if: • Supported PQC algorithms conform to officially published parameter sets and specifications • Domain parameters, key sizes and internal constants match official algorithm definitions • Generated outputs exactly match official NIST KAT vectors and reference implementation outputs • Interoperability with official reference implementations is achieved • Reduced, modified or non-standard parameter sets are securely rejected • Probabilistic components such as randomness generation, noise sampling and rejection sampling conform to required statistical distributions and algorithm specifications

Sl. No.	Test Details	Details
		• Repeated operations produce correct non-deterministic outputs where applicable • Appropriate error indication/logging is generated for invalid or unsupported PQC operations

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 8 - Verification of Code-Based Post-Quantum Cryptographic Algorithms
2.	GR Clause covered	2.3.2 (i), 2.3.2 (ii), 2.3.2 (iii)
3.	Objective of the test	To verify that the cryptographic processing module correctly implements code-based post-quantum cryptographic algorithms including encoding and decoding operations based on the specified error-correcting codes, correctly generates and processes error vectors and validates ciphertexts and public keys against the required algebraic structure and size constraints defined in the applicable algorithm specification.
4.	Tools required	• Open Quantum Safe (OQS-OpenSSL/OQS Provider) • Official reference implementations for code-based PQC schemes (e.g., HQC, Classic McEliece) • NIST PQC Known Answer Test (KAT) vectors • ACVP/CAVP validation vectors where applicable • Hex editor/viewer – input/output comparison
5.	Test procedure	**Step 1:** Identify all supported code-based PQC algorithms and parameter sets implemented by the EUT. **Step 2:** Verify support for approved parameter sets and algorithm specifications for supported code-based schemes such as:

Sl. No.	Test Details	Details
		• Classic McEliece parameter sets • HQC approved parameter sets **Step 3:** Perform key generation, encapsulation and decapsulation operations using approved parameter sets. **Step 4:** Verify that encoding and decoding operations correctly process messages, ciphertexts and shared secrets in accordance with the underlying error-correcting code specification defined by the algorithm. **Step 5:** Compare generated outputs such as ciphertexts, decoded messages and shared secrets against official NIST KAT vectors and reference implementation outputs. **Step 6:** Execute repeated encapsulation and decapsulation operations and verify successful recovery of the original message or shared secret. **Step 7:** Verify that generated ciphertexts, public keys and encapsulated outputs conform to the required size constraints, format requirements and expected structure defined in the applicable algorithm specification. **Step 8:** Attempt operations using malformed or invalid ciphertexts, public keys and encoded messages such as: • Incorrect ciphertext lengths • Modified algebraic structures • Corrupted public keys • Unsupported parameter sets **Step 9:** Verify that malformed or invalid ciphertexts and public keys are securely rejected. **Step 10:** Execute repeated operations and verify that generated error vectors and probabilistic outputs vary appropriately and do not exhibit

Sl. No.	Test Details	Details
		abnormal deterministic repetition inconsistent with the algorithm specification. **Step 11:** Verify interoperability by exchanging ciphertexts, public keys and shared secrets between the EUT and independent compliant reference implementations and confirm successful decapsulation and verification operations.
6.	Expected Result	Test passes if: • Encoding and decoding operations correctly implement the specified error-correcting code mechanisms • Encapsulation and decapsulation operations successfully recover the original message or shared secret • Generated outputs conform to official NIST KAT vectors and reference implementation outputs • Ciphertexts and public keys conform to required format, structure and size constraints • Invalid or malformed ciphertexts, public keys and parameter sets are securely rejected • Generated probabilistic outputs vary appropriately where required by the algorithm specification • Interoperability with compliant reference implementations is achieved • Appropriate error indication/logging is generated for invalid or unsupported operations

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 9 - Verification of Lattice-Based Cryptographic Algorithm Implementation (LWE / R-LWE)
2.	GR Clause covered	2.3.3 (i), 2.3.3 (ii), 2.3.3 (iii), 2.3.3 (iv), 2.3.3 (v)
3.	Objective of the test	To verify that the cryptographic module correctly implements lattice-based cryptographic algorithms in accordance with approved LWE/R-LWE specifications, including polynomial arithmetic, parameter validation, noise sampling, and decryption correctness.
4.	Tools required	• Open Quantum Safe (OQS-OpenSSL/OQS Provider) • Official reference implementations for lattice-based PQC schemes (e.g., ML-KEM, ML-DSA) • NIST PQC Known Answer Test (KAT) vectors • Hex editor/viewer – input/output comparison
5.	Test procedure	• Configure supported LWE/R-LWE algorithms using approved parameter sets • Check polynomial arithmetic operations including modular reduction, NTT, and inverse NTT, and compare outputs with reference implementations • Perform encryption/decryption and verify that ciphertexts and keys satisfy expected LWE/R-LWE mathematical relations. • Validate decryption correctness across repeated executions within defined error bounds. • Verify modulus, lattice dimension, and noise bounds against approved specifications. • Attempt use of reduced or modified parameter sets and verify rejection.

		• Generate noise samples and analyze statistical distribution against required algorithm specifications. • Perform large-scale decryption operations and measure decryption failure probability. • Compare outputs with approved KATs and reference implementations.
6.	Expected Result	• Polynomial arithmetic operations (modular reduction, NTT, inverse NTT) are correctly implemented and consistent. • Ciphertexts and keys satisfy required LWE/R-LWE mathematical relations. • Decryption operations correctly recover plaintext/shared secrets within defined error bounds. • Approved parameter sets meet required security levels and weakened parameters are rejected. • Noise sampling mechanisms generate values conforming to required statistical distributions. • Decryption failure probability remains within algorithm-defined limits.

Sl. No.	Test Details	Details
7.	Name of the test case	Test case 10 - To verify that the cryptographic module correctly implements hash-based cryptographic algorithms
8.	GR Clause covered	2.3.4 (i), 2.3.4 (ii), 2.3.4 (iii),
9.	Objective of the test	To verify that the cryptographic module correctly implements hash-based cryptographic algorithms, including state management, tree-based operations, and domain separation mechanisms in accordance with approved specifications.

10.	Tools required	• Open Quantum Safe (OQS-OpenSSL/OQS Provider) • Official reference implementations for hash-based PQC schemes (ML-DSA, SLH-DSA) • NIST PQC Known Answer Test (KAT) vectors • Hex editor/viewer – input/output comparison
11.	Test procedure	• Configure supported hash-based cryptographic algorithms (e.g., XMSS, SPHINCS+) • Perform signature generation and verification operations using approved test vectors • Validate correct construction and traversal of Merkle trees or equivalent tree-based structures • Verify correct maintenance of internal states such as leaf indices and traversal states across repeated operations • Attempt reuse or duplication of one-time signature states or leaf indices. • Verify implementation of domain separation values and hash input formatting. • Perform operations across multiple protocol instances and validate collision resistance. • Compare outputs with approved KATs and reference implementations.
12.	Expected Result	• Hash-based cryptographic constructions are correctly implemented in accordance with approved specifications. • One-time signatures and tree-based structures operate correctly. • Internal state management is correctly maintained without reuse or duplication. • Reuse of one-time signature states or leaf indices is prevented or rejected. • Domain separation mechanisms correctly prevent cross-protocol or cross-instance collisions.

		• Outputs match approved KATs and reference implementations

Sl. No.	Test Details	Details
13.	Name of the test case	Test case 11 - Verification of Post-Quantum Digital Signature Algorithm Implementation
14.	GR Clause covered	2.3.5 (i), 2.3.5 (ii), 2.3.5 (iii)
15.	Objective of the test	To verify that the cryptographic module correctly implements post-quantum digital signature algorithms in accordance with approved specifications, including secure signature generation, verification, encoding validation, and rejection handling.
16.	Tools required	• Open Quantum Safe (OQS-OpenSSL/OQS Provider) • Official reference implementations for hash-based PQC schemes (e.g., XMSS, LMS, SPHINCS+) • NIST PQC Known Answer Test (KAT) vectors • Hex editor/viewer – input/output comparison
17.	Test procedure	• Generate multiple signatures for identical and different messages using the same signing key. • Verify that randomness or deterministic processes conform to the algorithm specification and that secret-dependent values are not reused. • Perform signature verification using valid signatures and corresponding public keys. • Submit malformed, modified, oversized, truncated, or forged signatures for verification. • Validate bounds checking, mathematical verification conditions, and rejection handling.

		• Verify signature encoding format and signature size against applicable standards. • Compare outputs with approved KATs and reference implementations. • Verify availability and validity of CAVP or equivalent validation certificates, where applicable
18.	Expected Result	• PQC digital signatures are generated and verified correctly in accordance with approved specifications. • Randomness or deterministic processes are correctly implemented without reuse of secret-dependent values. • Mathematical verification conditions, bounds checking, and rejection conditions operate correctly. • Invalid, malformed, modified, or forged signatures are rejected. • Generated signatures conform to approved encoding formats and size constraints. • Outputs match approved KATs and reference implementations. • CAVP or equivalent validation certificates are available and valid, where applicable

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 12 - Verification of Random Number Generator (RNG) Module
2.	GR Clause covered	2.4 Random Number Generator Module

Sl. No.	Test Details	Details
3.	Objective of the test	To verify that the cryptographic processing module correctly implements Random Number Generator (RNG) functionality including entropy generation, random bit generation, seeding and continuous health checking in accordance with ISO/IEC 18031, TEC GR QRNG TEC 91020:2024, Test Guide of GR QRNG TEC 91021:2024 and NIST SP 800-90 series requirements.
4.	Tools required	• NIST Statistical Test Suite (STS) • Dieharder randomness test suite • ENT/randomness analysis tools • NIST SP 800-90A/B/C reference documents
5.	Test procedure	Step 1: Identify the RNG architecture implemented by the EUT including entropy source, TRNG/QRNG component and DRBG/PRNG mechanisms. Step 2: Verify that the RNG module implements approved standards and mechanisms in accordance with NIST SP 800-90 series or applicable TEC QRNG requirements. Step 3: Generate random data samples of sufficient size from the RNG module under normal operating conditions. Step 4: Perform statistical randomness testing on generated random outputs using NIST STS, Dieharder or equivalent statistical test suites and verify acceptable randomness characteristics. Step 5: Verify that entropy generated by the entropy source is correctly used to seed or reseed the DRBG/PRNG mechanism as applicable. Step 6: Verify that repeated RNG operations do not produce abnormal repetition, predictable sequences or deterministic outputs inconsistent with the implemented RNG specification. Step 7: Verify implementation of continuous RNG health tests and fault detection mechanisms where applicable.

Sl. No.	Test Details	Details
		Step 8: Verify that RNG failure conditions do not result in generation of weak or predictable cryptographic keys or parameters.
6.	Expected Result	Test passes if: • RNG module correctly implements approved entropy generation and random bit generation mechanisms • Generated random outputs successfully pass applicable statistical randomness tests • Entropy source correctly seeds or reseeds the DRBG/PRNG mechanism • No abnormal repetition or predictable output patterns are observed • Continuous RNG health monitoring and fault detection mechanisms operate correctly • Weak or predictable cryptographic outputs are not generated during failure conditions

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 13 - Verification of Key Management Module Functionality and Security
2.	GR Clause covered	2.5
3.	Objective of the test	To verify that the key management module securely performs complete key lifecycle operations including generation, verification, storage, exchange, usage control, backup/recovery, import/export, revocation, archival, and destruction of cryptographic keys while ensuring confidentiality, integrity, interoperability, policy enforcement,

		performance, and resistance against unauthorized access and attacks.
4.	Tools required	• OpenSSL or equivalent cryptographic toolkit • OpenKMIP, P6R KMIP Verification Suite, OASIS KMIP Test Cases, luigivezzoso/kmip-simple-test-client, smoonen/kmip-test
5.	Test procedure	• Generate symmetric, asymmetric, and PQC cryptographic keys using approved algorithms and validate key sizes, entropy, and parameter compliance • Store generated keys within the secure cryptographic boundary and verify confidentiality and integrity protections. • Distribute/exchange keys using approved in-band and out-of-band mechanisms and verify secure transmission protections. • Attempt export/distribution of non-exportable keys and verify rejection. • Verify enforcement of key usage policies for encryption, decryption, signing, and verification operations. • Perform complete key lifecycle operations including activation, suspension, revocation, expiration, archival, backup/recovery, and secure destruction. • Attempt usage of revoked, expired, invalid, or destroyed keys and verify rejection. • Import/export cryptographic keys using standardized formats and verify integrity/authenticity during transfer. • Establish interoperability with external systems using KMIP and perform KMIP operations including key creation, registration, retrieval, update, revocation, and destruction.

		• Verify correct handling of key attributes, metadata, activation time, expiration time, and lifecycle states. • Validate authentication, authorization, and secure communication protections for KMIP transactions. • Verify secure error handling and ensure sensitive information is not disclosed in logs or responses. • Perform load and scalability testing for key management operations under normal and peak conditions and measure performance metrics.
6.	Expected Result	• keys should securely generate with approved algorithms, valid entropy, and compliant key sizes. • Keys are securely stored, wrapped/unwrapped, and protected against unauthorized access and leakage. • Secure key exchange/distribution operations function correctly and non-exportable keys cannot be exported. • Key usage restrictions and policy controls are correctly enforced. • Complete key lifecycle management operations execute successfully and invalid/expired keys are rejected. • Key import/export operations maintain integrity, authenticity, and confidentiality. • Unauthorized, malformed, or invalid key operations are securely rejected without disclosure of sensitive information. • Side-channel and fault-based attack protections function correctly. • Key management operations perform within acceptable performance limits under normal and peak loads.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 14 - Verification of Cryptographic Interface and API Module
2.	GR Clause covered	2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.6.8, 2.6.9, 2.6.10
3.	Objective of the test	To verify that the cryptographic interface and API module provides secure, standardized and interoperable interfaces for accessing cryptographic services, supports cryptographic abstraction and agility, enforces authentication and authorization, securely manages sessions and protects sensitive information exposed through cryptographic APIs and interfaces.
4.	Tools required	• PKCS#11 test tools and libraries • OpenSSL/Open Quantum Safe (OQS-OpenSSL/OQS Provider) • Java Cryptography Extension (JCE) test environment • Microsoft CNG/CAPI test utilities • Web Cryptography API test environment • ETSI QKD API/interface reference specifications (where applicable) • API testing tools (Postman/cURL/custom scripts) • Python with cryptography libraries • Network packet analyzer (Wireshark) • Authentication and access-control testing tools • Log monitoring/debugging tools
5.	Test procedure	Step 1: Identify all cryptographic interfaces, APIs and supported standards implemented by the EUT such as PKCS#11, CNG, JCE, Web Cryptography APIs and ETSI QKD interfaces where applicable. Step 2: Verify that cryptographic services such as encryption, decryption, signing, verification, hashing and key management operations are accessible through the supported standardized interfaces.

Sl. No.	Test Details	Details
		Step 3: Verify that applications invoking cryptographic APIs remain independent of underlying cryptographic algorithms, providers and hardware implementations by changing providers or implementations without modifying application logic. **Step 4:** Verify dynamic selection and configuration of cryptographic algorithms, providers and parameters through the interface/API configuration mechanisms. **Step 5:** Attempt access to cryptographic services using authorized and unauthorized entities and verify enforcement of authentication and authorization mechanisms. **Step 6:** Establish, maintain and terminate cryptographic sessions or contexts and verify secure session/context management during cryptographic operations. **Step 7:** Perform cryptographic operations using valid and invalid API inputs including malformed parameters, invalid identifiers and malformed data structures and verify secure input validation and rejection of unauthorized or malformed requests. **Step 8:** Monitor API responses, logs, debug interfaces and error messages during cryptographic operations and verify that sensitive information such as cryptographic keys, intermediate values and confidential data is not exposed. **Step 9:** Verify integration and interoperability with both software-based and hardware-backed cryptographic providers through the same unified interface. **Step 10:** Where remote cryptographic invocation is supported, verify confidentiality, integrity and authentication protection for cryptographic data exchanged over communication channels.

Sl. No.	Test Details	Details
		Step 11: Verify cryptographic agility at the interface level by enabling, disabling or modifying cryptographic algorithms and providers without redesigning or changing the interface structure. **Step 12:** Verify appropriate logging and secure error indication for unauthorized access attempts, malformed API requests and failed cryptographic operations.
6.	Expected Result	Test passes if: • Supported cryptographic interfaces and APIs correctly implement applicable standards and frameworks • Cryptographic services are securely accessible through standardized interfaces • Applications remain independent of underlying cryptographic providers and implementations • Cryptographic algorithms, providers and parameters can be dynamically configured without application redesign • Authentication and authorization mechanisms correctly restrict access to authorized entities only • Session/context establishment, maintenance and termination operate securely • Invalid, malformed or unauthorized API requests are securely rejected • Sensitive information is not exposed through APIs, logs, debug interfaces or error messages • Unified interfaces correctly support both software-based and hardware-backed cryptographic providers • Remote cryptographic invocations are protected for confidentiality, integrity and authentication where applicable • Cryptographic agility is supported at the interface level without redesign of interfaces

Sl. No.	Test Details	Details
		• Appropriate logging/error indication is generated for unauthorized or failed operations

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 15 - Verification of Secure Protocol/Application Layer Requirements
2.	GR Clause covered	2.7.1.1, 2.7.1.2, 2.7.1.3, 2.7.1.4, 2.7.1.5, 2.7.1.6, 2.7.1.7, 2.7.1.8
3.	Objective of the test	To verify that the system correctly implements secure application and transport layer communication protocols such as TLS, IPsec, SSH and X.509-based mechanisms, supports approved cryptographic negotiation including PQC and hybrid schemes, prevents downgrade attacks and ensures interoperability with compliant implementations in accordance with applicable RFCs and standards.
4.	Tools required	Manual verification
5.	Test procedure	**Step 1:** Identify all supported secure protocols, protocol versions and cryptographic negotiation mechanisms implemented by the EUT including TLS, IPsec/IKEv2, SSH and X.509 certificate handling mechanisms. **Step 2:** Verify that only secure protocol versions are enabled by default such as: • TLS 1.2/TLS 1.3 • IKEv2/IPsec • Secure SSH versions/configurations

Sl. No.	Test Details	Details
		Step 3: Verify that deprecated or insecure protocol versions/configurations such as SSLv2, SSLv3, TLS 1.0, TLS 1.1 and weak IPsec/IKE algorithms are disabled or rejected. **Step 4:** Verify interoperability between the EUT and independent compliant implementations such as OpenSSL, StrongSwan or other standards-compliant systems. **Step 5:** Attempt downgrade or fallback attacks by forcing insecure protocol versions, deprecated cipher suites or weak algorithms and verify that: • Downgrade attempts are rejected • Fallback to insecure protocols/algorithms is prevented • Secure session establishment fails securely where applicable **Step 6:** Verify support for PQC and hybrid cryptographic mechanisms in TLS/IPsec/SSH negotiation procedures where supported. **Step 7:** Where QKD integration is supported, verify secure integration of externally generated keying material into protocol operations. **Step 8:** Verify correctness of protocol message formats, handshake procedures and protocol state transitions according to applicable RFCs and standards. **Step 9:** Attempt malformed handshake messages, invalid certificates and unsupported negotiation parameters and verify secure rejection with appropriate error indication/logging.
6.	Expected Result	Test passes if: • Secure application/transport layer protocols are correctly implemented according to applicable RFCs and standards • Only secure protocol versions and approved cipher suites are enabled by default

Sl. No.	Test Details	Details
		• TLS/IPsec/SSH sessions successfully negotiate approved cryptographic algorithms and X.509 certificate validation mechanisms • Interoperability with compliant implementations is successfully achieved • Downgrade and fallback attacks to insecure protocols or algorithms are prevented • PQC and hybrid cryptographic mechanisms are correctly negotiated and processed where supported • QKD-generated keying material is securely integrated where applicable • Protocol message formats, handshake procedures and state transitions are correctly implemented • Invalid certificates, malformed protocol messages and unsupported negotiation parameters are securely rejected with appropriate error indication/logging

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 16 - Verification of X.509 / PKI Requirements
2.	GR Clause covered	2.8.1, 2.8.2, 2.8.3, 2.8.4, 2.8.5, 2.8.6, 2.8.7
3.	Objective of the test	To verify that the system correctly implements X.509 and PKI mechanisms including certificate parsing, validation, hybrid/PQC certificate processing, certificate chain validation, interoperability and secure validation of algorithm identifiers and certificate encodings in accordance with applicable standards and specifications.

Sl. No.	Test Details	Details
4.	Tools required	• OpenSSL/Open Quantum Safe (OQS-OpenSSL/OQS Provider) • X.509 certificate generation and validation tools • Hybrid/PQC certificate test environment • Wireshark/packet analyzer • ASN.1/DER parsing and validation tools • Certificate chain validation tools • Hex editor/viewer
5.	Test procedure	**Step 1:** Identify all supported X.509, PKI, hybrid certificate and PQC certificate mechanisms implemented by the EUT. **Step 2:** Generate and import X.509 certificates containing approved PQC and hybrid cryptographic algorithms and verify correct processing of: • Certificate structure • Extensions • ASN.1/DER encoding formats • Subject/issuer fields • Public key information **Step 3:** Verify correct processing of algorithm identifiers (OIDs), parameters and encoding formats for PQC and hybrid cryptographic algorithms. **Step 4:** Verify support for hybrid and PQC certificates by validating certificates containing: • PQC public keys • Hybrid public keys • Classical + PQC signature combinations **Step 5:** Verify successful validation of both classical and PQC signatures contained within hybrid certificates.

Sl. No.	Test Details	Details
		Step 6: Establish certificate chains including mixed PQC/hybrid chains and verify correct trust path validation, certificate hierarchy processing and chain verification. **Step 7:** Verify interoperability of generated certificates across independent compliant implementations such as OpenSSL/OQS-OpenSSL and confirm successful certificate parsing and validation. **Step 8:** Attempt certificate downgrade, truncation and substitution attacks such as: • Replacing hybrid certificates with classical-only certificates • Removing PQC signature components • Truncating certificate chains • Modifying algorithm identifiers (OIDs) **Step 9:** Verify that invalid, truncated, substituted or downgraded certificates are securely rejected and appropriate error indication/logging is generated. **Step 10:** Attempt operations using malformed certificates, invalid ASN.1/DER encodings, unsupported OIDs and invalid certificate extensions and verify secure rejection.
6.	Expected Result	Test passes if: • X.509 certificates, extensions and encoding formats are correctly processed according to applicable standards • Hybrid and PQC certificates are correctly processed with proper association of algorithm identifiers and parameters • PQC signatures within hybrid certificates are successfully validated • Certificate chain validation including mixed algorithm chains operates correctly • Interoperability with compliant implementations is successfully achieved

Sl. No.	Test Details	Details
		• Downgrade, truncation and substitution attacks against certificates and certificate chains are prevented • Algorithm identifiers (OIDs) and encoding formats for PQC and hybrid algorithms are correctly processed and validated • Malformed certificates, invalid encodings and unsupported OIDs are securely rejected with appropriate error indication/logging

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 17 - Validation of IKEv2 Secure Association Establishment, Key Exchange Mechanisms, Hybrid/Post-Quantum Support, Downgrade Protection, Session Key Consistency, and Interoperability
2.	GR Clause covered	2.9 (2.9.1, 2.9.2, 2.9.3, 2.9.4, 2.9.5, 2.9.6)
3.	Objective of the test	Verify that the system: 1. Implements IKEv2 for establishment of secure Security Associations (SAs). 2. Supports secure key exchange mechanisms providing Perfect Forward Secrecy (PFS), including classical, post-quantum, and hybrid methods. 3. Correctly derives and synchronizes negotiated session keys between communicating entities. 4. Supports hybrid key exchange mechanisms combining classical and post-quantum algorithms.

		5. Prevents downgrade to insecure or deprecated key exchange, authentication, or cryptographic algorithms. 6. Ensures interoperability with other compliant IKEv2 implementations.
7.	Test environment/pre-condition	• VPN Gateway/Router/Endpoint implementing IKEv2 • Standards-compliant IKEv2 implementation (e.g., strongSwan) • Packet capture and logging enabled • NTP synchronized systems
8.	Tools required	• Wireshark, tcpdump, strongSwan, Libreswan • Vendor VPN Client (if applicable) • OpenSSL • Log Collection Tool • Cryptographic Validation Tool • PQC-enabled IKEv2 implementation
9.	Test procedure	• Establish an IKEv2 VPN tunnel and capture traffic using Wireshark/tcpdump. Verify IKE_SA_INIT and IKE_AUTH exchanges. • Configure PFS-enabled key exchange and perform tunnel rekeying. Verify new ephemeral keys are generated for each rekey event. • Configure and establish a VPN tunnel using a supported post-quantum key exchange algorithm. Verify successful negotiation. • Configure and establish a VPN tunnel using a hybrid key exchange mechanism combining classical and post-quantum algorithms. Verify successful negotiation of both components. • Exchange encrypted traffic through the VPN tunnel and verify successful encryption/decryption at both endpoints, confirming session key consistency

		• Attempt tunnel establishment using deprecated or weak algorithms (e.g., weak DH groups, deprecated ciphers, weak authentication methods). Verify the DUT rejects such proposals • Configure a peer to offer both approved and deprecated algorithms. Verify only approved algorithms are selected and no downgrade occurs • Establish VPN tunnels with multiple compliant IKEv2 implementations (e.g., strongSwan, Libreswan, other vendor solutions). Verify tunnel establishment, traffic flow, and rekeying.
10.	Expected Result	• IKE_SA_INIT and IKE_AUTH exchanges complete successfully • PFS should be achieved through ephemeral key generation and successful rekeying • Hybrid key exchange mechanisms (classical + post-quantum) are successfully negotiated and used • Session keys should correctly derive and synchronized between peers • Downgrade attempts to insecure or deprecated algorithms are rejected • cryptographic algorithms should be selected during negotiation • Interoperability is demonstrated with multiple standards-compliant IKEv2 implementations and VPN tunnels remain stable during operation and rekeying.

Sl. No.	Test Details	Details

11.	Name of the test case	Test case 18 - Verification of TLS 1.3 Secure Communication, Cipher Suite Negotiation, PQC/Hybrid Support, Downgrade Protection, Handshake Validation, and Interoperability
12.	GR Clause covered	2.10 (2.10.1, 2.10.2, 2.10.3, 2.10.4, 2.10.5, 2.10.6, 2.10.7)
13.	Objective of the test	Verify that the system: 1. Implements TLS 1.3 for secure application communication. 2. Supports negotiation of approved cipher suites. 3. Supports hybrid key exchange (classical + post-quantum algorithms). 4. Supports post-quantum or hybrid certificates/signatures during TLS handshakes. 5. Prevents downgrade and fallback to insecure algorithms or protocol versions. 6. Interoperates with different compliant TLS implementations. 7. Correctly performs handshake operations including key exchange, authentication, and session key derivation.
8.	Test environment/pre-condition	• Alternate TLS implementations (e.g., OpenSSL, BoringSSL, wolfSSL, GnuTLS) • Isolate test network • Packet capture and logging enabled
9.	Tools required	• Same as given in test case for 2.9
10.	Test procedure	• Establish a TLS session and verify TLS 1.3 is negotiated. • Configure multiple approved cipher suites and verify successful negotiation of an approved suite • Configure classical + PQC key exchange and establish a TLS session • Configure PQC or hybrid certificates/signatures and verify successful certificate exchange and validation during handshake

		• Capture and analyze ClientHello, ServerHello, certificate exchange, key exchange, authentication, and session key establishment • Attempt connections using older TLS versions or weak algorithms and verify rejection or secure negotiation • Establish TLS sessions between the DUT and multiple compliant TLS implementations and verify successful communication
11.	Expected Result	• TLS 1.3 should successfully negotiate and used. • Approved cipher suites should correctly negotiate. • Hybrid key exchange mechanisms operate successfully. • PQC or hybrid certificates/signatures should be accepted and validated. • Handshake completes successfully with correct key exchange, authentication, and session key derivation. • Downgrade attempts and insecure algorithm fallbacks must be rejected. • Secure communication must establish without errors.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 19 - Verification of S/MIME Secure Email Communication, PQC Integration, Security Services, and Interoperability
2.	GR Clause covered	2.11 (2.11.1, 2.11.2, 2.11.3, 2.11.4)
3.	Objective of the test	Verify that the system: 1. Supports S/MIME for secure email encryption and digital signatures. 2. Supports integration of post-quantum cryptographic (PQC) mechanisms for key exchange and digital signatures.

		3. Ensures confidentiality, integrity, authentication, and non-repudiation of email communications. 4. Interoperates with other compliant S/MIME implementations.
5.	Tools required	• S/MIME-enabled Email Clients (e.g., Outlook, Thunderbird) • Same as given in test case for 2.9
6.	Test procedure	• Send an S/MIME encrypted email from Client A to Client B and verify successful decryption. • Send a digitally signed email and verify signature validation at the recipient. • Configure PQC or hybrid certificates/signatures and exchange signed/encrypted emails. • Attempt to view intercepted email content without the recipient's private key. • Modify the signed email content and verify signature validation fails. • Verify sender identity through certificate validation and successful signature verification. • Exchange encrypted and signed emails between the Device under test (DUT) and another compliant S/MIME implementation and verify successful processing.
7.	Expected Result	• S/MIME encrypted emails should successfully decrypt only by authorized recipients. • Digital signatures should successfully verified. • PQC or hybrid cryptographic mechanisms should successfully use for key exchange and/or signatures. Email confidentiality must be maintained • Any modification of signed messages is detected • Sender authentication and non-repudiation are verified through valid signatures and certificates.

		• Signed and encrypted emails should successfully exchanged with other compliant S/MIME implementations

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 20 - Verification of SSH Protocol for Secure Remote Access and Communication
2.	GR Clause covered	2.12.1
3.	Objective of the test	Verify that the system implements the Secure Shell (SSH) protocol and provides secure remote access and communication between client and server systems.
4.	Tools required	• OpenSSH Client/Server • PuTTY (optional) • Wireshark • tcpdump
5.	Test procedure	• Configure and enable the SSH service on the DUT and Configure a valid user account for SSH access. • From the SSH client, initiate an SSH connection to the DUT using valid credentials and Verify successful user authentication and session establishment • Execute basic commands (e.g., directory listing, file access) over the SSH session • Capture network traffic during the session using Wireshark/tcpdump • Terminate the SSH session and review logs for successful connection and disconnection events
6.	Expected Result	• SSH connection should successfully established between client and server

		• User authentication succeeds with valid credentials • Remote commands execute successfully • Data exchanged during the session should be encrypted and not readable in packet captures • SSH session terminates gracefully without errors

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 21 - Verification of Cryptographic Agility, PQC Readiness, Hybrid Cryptography Support, and Secure Cryptographic Lifecycle Management
2.	GR Clause covered	2.13 (2.13.1)
3.	Objective of the test	Verify that the system supports cryptographic agility by enabling secure addition, replacement, negotiation, deprecation, and management of classical, hybrid, and post-quantum cryptographic mechanisms without major architectural changes or service disruption.
4.	Test environment/pre-condition	• Test Client and Server supporting classical, hybrid, and PQC algorithms • TLS 1.3-enabled environment • Policy Management Server/API (if applicable) • PKI environment with classical, hybrid, and PQC certificates • Isolated test network with logging enabled
5.	Tools required	• OpenSSL (TLS 1.3) • Wireshark/tcpdump • PQC-enabled cryptographic libraries (e.g., Open Quantum Safe/OQS) • Certificate Management Tools • Configuration Management Tool/API

		• Log Monitoring Tool • Performance Monitoring Tools (CPU, Memory, Network)
6.	Test procedure	• Add, replace, and deprecate approved cryptographic algorithms through configuration without modifying system architecture or code • Establish secure sessions between endpoints supporting classical, hybrid, and PQC algorithms and verify successful negotiation • Modify cryptographic policies through approved external mechanisms (config file/API/policy server). Verify access controls, integrity protection, and audit logging • Configure hybrid key exchange (classical + PQC) and establish secure sessions. Verify successful operation • Verify session keys are derived from independent classical and PQC components. Confirm forward secrecy through session re-establishment and key rotation • Establish TLS 1.3 sessions with PQC-capable extensions enabled. Verify legacy protocols (SSL, TLS 1.0/1.1/1.2 if prohibited) are disabled by default • Attempt protocol and algorithm downgrade attacks and verify rejection • Deploy hybrid certificates and verify successful certificate validation, chain validation, and signature verification • Measure CPU, memory, bandwidth, latency, handshake time, and key generation time when using PQC and hybrid mechanisms
7.	Expected Result	• Cryptographic algorithms can be added, replaced, or deprecated through configuration without service disruption • Algorithm negotiation successfully selects approved classical, hybrid, or PQC algorithms

		• Cryptographic policies are externally configurable and protected against unauthorized modification • Hybrid key establishment operates successfully using both classical and PQC mechanisms • Session key derivation maintains cryptographic independence between classical and PQC components • Forward secrecy is preserved during hybrid operation • TLS 1.3 is used for secure transport and legacy protocols are disabled by default

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 22 - Verification of AEAD Functionality
2.	GR Clause covered	2.14 - Requirements for constrained devices
3.	Objective of the test	To verify that the cryptographic processing module correctly implements AEAD mechanisms ensuring confidentiality, integrity and authentication of encrypted data and associated data.
4.	Tools required	• NIST AEAD KAT vectors - https://github.com/usnistgov/ACVP-Server/tree/master/gen-val/json-files • Wireshark • Performance measurement tools
5.	Test procedure	**Step 1:** Verify AEAD encryption and decryption operations using associated data (AAD) and compare outputs with NIST AEAD KAT vectors. **Step 2:** Verify confidentiality by transmitting encrypted data over simulated insecure/noisy communication channels and confirm plaintext cannot be recovered without valid keys.

Sl. No.	Test Details	Details
		Step 3: Modify ciphertext, authentication tag and associated data and verify authentication failure and rejection of tampered messages. **Step 4:** Perform AEAD operations using different supported key sizes and verify acceptable security and operational performance. **Step 5:** Simulate side-channel and fault injection scenarios where applicable and verify that authentication bypass or sensitive information leakage does not occur. **Step 6:** Measure and record encryption/decryption throughput, latency and resource consumption under constrained operating conditions. **Step 7:** Verify compliance with applicable NIST AEAD and lightweight cryptography requirements using official test vectors and interoperability validation.
6.	Expected Result	Test passes if: • Data and associated data are correctly encrypted and authenticated • Confidentiality is maintained over insecure communication channels • Modified ciphertext, tags or associated data fail authentication verification • Supported key sizes provide acceptable security and performance trade-off • No authentication bypass or sensitive leakage occurs during attack simulations • Performance remains within acceptable operational limits • AEAD implementation successfully passes applicable NIST compliance validation tests

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 23 - Verification of FIPS 206 Hash-Based Digital Signature Mechanisms
2.	GR Clause covered	2.14 – Lightweight cryptography requirements
3.	Objective of the test	To verify that the cryptographic processing module correctly implements FIPS 206 compliant hash-based digital signature mechanisms including signature generation, verification, context binding, malformed input handling, interoperability and resistance against side-channel and fault injection attacks.
4.	Tools required	• Open Quantum Safe (OQS-OpenSSL/OQS Provider) • Official FIPS 206/FN-DSA reference implementations • NIST Known Answer Test (KAT) vectors • Timing/power analysis tools • Fault injection testing tools • Performance benchmarking tools • Hex editor/viewer
5.	Test procedure	Step 1: Generate FN-DSA signatures and perform signature verification over representative IoT messages such as telemetry data, control commands and protocol messages using supported parameter sets. Step 2: Verify that signatures are correctly bound to associated context information such as device identifiers, protocol headers and domain separation values Step 3: Attempt signature verification using modified messages, modified signatures, incorrect public keys and truncated inputs and verify secure rejection without abnormal behavior or crashes. Step 4: Perform key pair generation using supported parameter sets and verify correctness of generated public/private keys and rejection of invalid or malformed keys.

Sl. No.	Test Details	Details
		Step 5: Attempt operations using malformed or edge-case encodings such as oversized inputs, invalid lengths and verify secure rejection without memory corruption or abnormal execution. **Step 6:** Execute repeated signing and verification operations while monitoring timing behavior, power consumption and observable leakage characteristics where applicable and verify absence of exploitable side-channel leakage beyond acceptable limits. **Step 7:** Simulate fault injection conditions such as bit flips, execution glitches or interrupted signing/verification operations and verify that sensitive key material is not exposed and faulty signatures are not incorrectly accepted. Verify that the implementation uses the exact Gaussian sampler specified by the standard to ensure security and interoperability. **Step 8:** Measure and record signing time, verification time, memory usage, storage footprint and energy consumption on representative IoT or constrained hardware platforms and verify operation within defined device constraints and procurement limits. **Step 9:** Verify interoperability by exchanging signatures, keys and verification data with an independent standards-compliant reference implementation across supported parameter sets and confirm successful cross-verification. **Step 10:** Verify compliance with applicable FIPS 206, NIST and IETF requirements using official test vectors, parameter validation and profile conformance testing.
6.	Expected Result	Test passes if: • Valid SLH-DSA signatures are successfully generated and verified • Signatures are correctly bound to associated context information • Modified messages, signatures, truncated inputs and incorrect public keys are securely rejected

Sl. No.	Test Details	Details
		• Key generation operates correctly and invalid keys are rejected • Malformed and edge-case inputs are safely rejected without crashes or memory errors • No exploitable side-channel leakage is observed beyond acceptable limits • Fault injection attempts do not expose key material and faulty signatures are not accepted • Signing and verification operations operate within defined IoT device constraints • Interoperability with compliant reference implementations is successfully achieved • Implementation successfully passes applicable FIPS 206, NIST and IETF compliance requirements and official test vectors

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 24 - Verification of ASCON Lightweight Cryptographic Algorithms
2.	GR Clause covered	2.14 – Lightweight Cryptography Requirments
3.	Objective of the test	To verify that the cryptographic processing module correctly implements ASCON lightweight cryptographic algorithms including AEAD, Hash, XOF and CXOF functions and ensures confidentiality, integrity, nonce handling, robustness, side-channel resistance and interoperability in constrained IoT environments.
4.	Tools required	• Official NIST ASCON test vectors • Open-source ASCON reference implementations • Timing/power analysis tools • Performance benchmarking tools

Sl. No.	Test Details	Details
		• Hex editor/viewer – ciphertext/hash comparison
5.	Test procedure	**Step 1:** Verify that the implementation supports the claimed ASCON standardized variants such as Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128 and Ascon-CXOF128 and verify that unsupported variants are not accepted or claimed. **Step 2:** Perform ASCON AEAD encryption and decryption operations using associated data and official NIST test vectors and verify generated ciphertexts and authentication tags against expected outputs. **Step 3:** Modify associated data (AAD) such as protocol headers, metadata or addresses and verify that authentication/tag verification fails and plaintext is not released. **Step 4:** Verify nonce generation, storage and anti-reuse handling by executing repeated encryption operations and confirming that nonce reuse does not occur during normal operation or is detected/mitigated. **Step 5:** Perform authentication tag verification using valid and invalid tags and verify constant-time verification behavior and secure fail-closed operation for invalid tags. **Step 6:** Test misuse and edge-case conditions including zero-length plaintext, zero-length associated data, maximum supported lengths and fragmented/streamed inputs where supported and verify stable and correct operation without crashes. **Step 7:** Verify replay protection integration by using counters, timestamps or protocol identifiers as associated data and confirm replayed messages are detected and rejected by the system logic where applicable. **Step 8:** Perform ASCON-Hash operations using official test vectors and representative IoT payloads and verify generated hash outputs against expected values.

Sl. No.	Test Details	Details
		Step 9: Perform Ascon-XOF128 operations using different requested output lengths and verify correct output size, squeezed block generation and output correctness against reference implementations. **Step 10:** Perform Ascon-CXOF128 operations using different customization string values and verify correct incorporation of customization inputs and handling according to the specification. **Step 11:** Attempt operations using malformed inputs, corrupted lengths, invalid encodings and malformed buffers and verify secure rejection without memory corruption or abnormal behavior. **Step 12:** Execute repeated encryption, decryption and hashing operations while monitoring timing and power behavior where applicable and verify absence of exploitable side-channel leakage beyond acceptable limits. **Step 13:** Measure and record throughput, latency, RAM usage, Flash/storage footprint and power consumption. **Step 14:** Verify compliance with applicable NIST Lightweight Cryptography ASCON specifications and official published test vectors using interoperability and conformance testing procedures.
6.	Expected Result	Test passes if: • Supported ASCON variants correctly match claimed standardized functions • ASCON AEAD encryption/decryption outputs match official test vectors • Modified associated data causes authentication/tag verification failure • Nonce reuse does not occur during normal operation or is securely handled • Invalid authentication tags are consistently rejected without timing oracle leakage • Edge-case and malformed inputs are safely handled without crashes or memory corruption

Sl. No.	Test Details	Details
		• Replay protection mechanisms correctly detect replayed messages where applicable • ASCON-Hash, XOF and CXOF outputs match reference vectors and implementations • Side-channel leakage remains within acceptable limits • Performance on representative IoT hardware meets operational and procurement constraints • Implementation successfully passes applicable NIST ASCON compliance and interoperability testing

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 25 - Verification of Cloud/Service-Based Key Lifecycle Management
2.	GR Clause covered	2.15.3.1 to 2.15.3.13
3.	Objective of the test	To verify that the cloud/service-based key lifecycle management system correctly implements secure Cloud HSM integration, cryptographic key lifecycle management, access control, automated key rotation, audit logging, secure API protection and compliance with applicable FIPS/ISO security requirements.
4.	Tools required	• Cloud HSM platforms such as AWS CloudHSM, Azure Key Vault, Google Cloud KMS or equivalent
5.	Test procedure	**Step 1:** Verify integration of the EUT with supported Cloud HSM services such as AWS CloudHSM, Azure Key Vault, Google Cloud KMS or equivalent cloud-based HSM implementations.

Sl. No.	Test Details	Details
		Step 2: Perform cryptographic key lifecycle operations including secure key generation, storage, usage, rotation, archival and destruction using the Cloud HSM environment. **Step 3:** Verify that cryptographic keys are generated and remain within FIPS 140-2 Level 3 (or higher) validated HSM security boundaries and confirm that plaintext export of key material is not permitted outside the validated HSM boundary. **Step 4:** Verify role-based access control (RBAC), least-privilege enforcement, segregation of duties and multi-factor authentication mechanisms for access to cryptographic keys and HSM services. **Step 5:** Configure and verify key rotation policies in accordance with NIST SP 800-57 Part 1 Rev. 5 requirements including shorter crypto periods for high-risk systems and controlled rotation policies for root/master keys where applicable. **Step 6:** Verify automated key rotation mechanisms and confirm that key rotation occurs without service disruption and that previous key versions remain available only for decryption or signature verification operations. **Step 7:** Verify generation of immutable audit logs for all key lifecycle events including key creation, access, usage, rotation, archival, policy changes and destruction. **Step 8:** Verify that audit logs are tamper-evident, exportable to external logging/SIEM systems and retained in accordance with applicable standards and regulatory requirements including NIST SP 800-57, NIST SP 800-53, ISO/IEC 11770, ISO/IEC 27001 and CERT-In directions. **Step 9:** Perform cryptographic key destruction operations and verify that key destruction is irreversible and supported by verifiable audit evidence. **Step 10:** Verify secure key provisioning, distribution, archival, rotation and destruction workflows across integrated cloud services and applications. **Step 11:** Verify that only authorized services, applications and users can

Sl. No.	Test Details	Details
		access HSM APIs by reviewing IAM policies, RBAC configurations and API authorization controls. Step 12: Perform dynamic security testing on key management APIs including: • Unauthorized access attempts • Replay attacks • Injection attacks • Improper error handling revealing sensitive information • Verification of data-in-transit and data-at-rest protection Step 13: Verify that the deployed Cloud HSM implementation complies with applicable FIPS/ISO security requirements and is validated to at least FIPS 140-2 Level 3 or higher equivalent assurance level.
6.	Expected Result	Test passes if: • Cloud HSM integration operates correctly with supported platforms • Complete cryptographic key lifecycle management functions operate securely • Cryptographic keys remain within validated FIPS 140-2 Level 3 (or higher) HSM boundaries and plaintext key export is prevented • RBAC, least privilege, segregation of duties and MFA controls are correctly enforced • Key rotation policies are correctly implemented and automated rotation occurs without service disruption • Previous key versions remain accessible only for authorized decryption/verification purposes • Immutable and tamper-evident audit logs are generated and retained according to applicable requirements • Cryptographic key destruction is irreversible and verifiable through audit evidence

Sl. No.	Test Details	Details
		• Only authorized users/services can access HSM APIs and cryptographic operations • Key management APIs securely resist unauthorized access, replay, injection and information disclosure attacks • Data-in-transit and data-at-rest protections operate correctly • Cloud HSM deployment complies with applicable FIPS/ISO validation requirements

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 26 - Verification of Secure Software/Firmware Loading
2.	GR Clause covered	3.2 – Clause 8 Software/Firmware Loading (ISO/IEC 19790:2025 Clause 7.4.3.4)
3.	Objective of the test	To verify that the cryptographic module validates the integrity and authenticity of software/firmware before installation and rejects unauthorized or modified images.
4.	Tools required	• Vendor firmware image • Modified firmware image • Digital signature verification tools • Firmware loading utility • System logs
5.	Test procedure	**Step 1:** Load a valid vendor-signed firmware image. **Step 2:** Verify successful validation and installation. **Step 3:** Modify the firmware image by changing one or more bytes. **Step 4:** Attempt firmware installation. **Step 5:** Observe module response and audit logs. **Step 6:** Attempt installation of an unsigned image.

Sl. No.	Test Details	Details
		Step 7: Verify module rejects unauthorized images.
6.	Expected Result	Test passes if: • Valid signed firmware is accepted. • Modified firmware is rejected. • Unsigned firmware is rejected. • Integrity verification occurs before loading. • Audit logs record the event.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 27 - Verification of Pre-operational and Conditional Self-Tests
2.	GR Clause covered	3.2 – Clause 9 Self-Test for Integrity of H/W and S/W Modules (ISO/IEC 19790 Clause 7.10)
3.	Objective of the test	To verify that pre-operational and conditional self-tests execute correctly before security functions become available.
4.	Tools required	• System logs • Debug console
5.	Test procedure	**Step 1:** Power ON the module. **Step 2:** Observe execution of power-up self-tests. **Step 3:** Verify cryptographic services remain unavailable until completion. **Step 4:** Invoke a cryptographic function requiring conditional self-test. **Step 5:** Observe execution of the conditional test. **Step 6:** Inject a self-test failure condition (if supported).
6.	Expected Result	• Self-tests execute automatically.

Sl. No.	Test Details	Details
		• Module enters approved mode only after successful completion. • Failure causes error state and service denial.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 28 - Verification of Module Version Information
2.	GR Clause covered	3.2 – Clause 10 Module Version (ISO/IEC 19790 Clause 7.4.3.1)
3.	Objective of the test	To verify that the module outputs module identifier and version information.
4.	Tools required	• CLI/Web GUI • Management interface
5.	Test procedure	Step 1: Access module management interface. Step 2: Request module identification information. Step 3: Record hardware, firmware and software version details. Step 4: Compare with vendor documentation.
6.	Expected Result	Module name, identifier, firmware version and software version are displayed correctly and match documentation.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 29 - Verification of Zeroisation of Sensitive Security Parameters
2.	GR Clause covered	3.2 – Clause 14 Zeroisation (ISO/IEC 19790 Clause 7.4.3.1)

Sl. No.	Test Details	Details
3.	Objective of the test	To verify secure destruction of cryptographic keys.
4.	Tools required	• Memory inspection tools • Debug interface (if applicable)
5.	Test procedure	**Step 1:** Generate cryptographic keys. **Step 2:** Store keys in the module. **Step 3:** Invoke zeroisation service. **Step 4:** Attempt recovery of key material. **Step 5:** Review audit records. **Step 6:** For Level-4 products verify uninterrupted zeroisation.
6.	Expected Result	• All CSPs and SSPs removed. • No recoverable key material remains. • Zeroisation completes successfully. • Audit record generated.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 30 - Verification of Trusted Channel
2.	GR Clause covered	3.3 – Clause 8 Trusted Channel (ISO/IEC 19790 Clause 7.3.4)
3.	Objective of the test	To verify secure transmission of plaintext CSPs and authentication data.
4.	Tools required	• Packet capture tool (Wireshark) • Management workstation • TLS/IPsec analysis tools
5.	Test procedure	**Step 1:** Establish trusted channel. **Step 2:** Transfer CSPs through the channel.

Sl. No.	Test Details	Details
		Step 3: Capture network traffic. **Step 4:** Verify confidentiality protection. **Step 5:** Attempt unauthorized interception. **Step 6:** For Level-4 verify MFA-based authentication.
6.	Expected Result	• CSPs are protected during transmission. • Intercepted traffic does not reveal plaintext secrets. • Authentication requirements are enforced.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 31 - Verification of NTP/PTP Interoperability
2.	GR Clause covered	3.4 – Clause10 – Clock or Time Synchronization Interoperability
3.	Objective of the test	To verify interoperability with NTP/PTP servers and accurate time synchronization.
4.	Tools required	• NTP/PTP Server • Network Analyzer
5.	Test procedure	**Step 1:** Configure NTP/PTP server details on DUT. **Step 2:** Initiate synchronization. **Step 3:** Verify synchronized system time. **Step 4:** Generate cryptographic events and verify timestamps. **Step 5:** Restart synchronization process and verify continuity.
6.	Expected Result	• Time synchronization is successful. • Event timestamps are accurate. • Cryptographic operations are not affected by synchronization.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 32 - Verification of Multi-Person (M-of-N) Authorization Controls
2.	GR Clause covered	4.2.2
3.	Objective of the test	To verify that the product correctly enforces configurable multi-person (M-of-N) authorization controls for critical cryptographic operations and prevents execution of sensitive operations without the required quorum approval.
4.	Tools required	• HSM/Key management system • Administrative consoles • MFA-enabled operator accounts • Audit log monitoring tools • Access control validation tools • SIEM/log analysis tools
5.	Test procedure	**Step 1:** Configure M-of-N authorization policies for critical cryptographic operations such as master key generation, activation, destruction and PQC transition policy modification. **Step 2:** Verify that configurable M and N values can be defined according to applicable procurer requirements. **Step 3:** Attempt execution of critical cryptographic operations using fewer than the required quorum participants and verify secure rejection. **Step 4:** Perform authorized operations using the required quorum and verify successful execution only after valid approvals from the configured number of authorized personnel. **Step 5:** Verify generation of secure audit logs for all quorum approval requests, approvals, denials, failed attempts and policy changes. Verify

Sl. No.	Test Details	Details
		audit logs are tamper-evident, timestamped and attributable to individual authorized operators. Step 6: Attempt bypass or circumvention of quorum mechanisms through direct API access, privilege escalation or session manipulation and verify secure prevention.
6.	Expected Result	Test passes if: • Critical cryptographic operations require valid M-of-N quorum authorization • Configurable M and N values are correctly enforced • Single-person execution of protected operations is technically prevented • Unauthorized PQC transition policy modifications are rejected • All quorum events are securely logged and auditable • Quorum enforcement mechanisms resist bypass and circumvention attempts

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 33 - Verification of Protection Against PQC Parameter Downgrade Attacks
2.	GR Clause covered	4.2.3
3.	Objective of the test	To verify that the product prevents protocol-level downgrade attacks against PQC security parameters and strictly enforces approved minimum cryptographic parameter sets.
4.	Tools required	• TLS/IPsec/SSH interoperability test environment • PQC-enabled protocol implementations • Protocol fuzzing tools

Sl. No.	Test Details	Details
		• Policy management interface
5.	Test procedure	**Step 1:** Configure the product with approved minimum PQC parameter sets and security policies. **Step 2:** Establish protocol sessions using supported PQC and hybrid cryptographic mechanisms and verify successful negotiation of approved strong parameter sets. **Step 3:** Attempt protocol downgrade attacks by forcing negotiation of weaker PQC parameter sets, deprecated modes or unsupported security levels and verify rejection. **Step 4:** Attempt fallback negotiation from PQC or hybrid modes to weaker classical-only modes where prohibited by policy and verify prevention. **Step 5:** Verify that minimum approved parameter policies are immutable or protected against unauthorized modification. **Step 6:** Perform protocol conformance testing across supported protocols such as TLS, IPsec and SSH and verify downgrade resistance behavior.
6.	Expected Result	Test passes if: • Strong approved PQC parameter sets are correctly enforced • Downgrade and fallback attempts are rejected • Minimum parameter set policies cannot be bypassed or modified without authorization • Protocol negotiations remain compliant with configured PQC security policies

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 34 - Verification of VA/PT Compliance and Security Assessment Requirements
2.	GR Clause covered	4.2.4, 4.2.4.1 to 4.2.4.14
3.	Objective of the test	To verify that Vulnerability Assessment and Penetration Testing (VA/PT), secure coding practices, third-party library assessment and hardware assurance activities are performed in accordance with CERT-In/NCCS requirements and applicable security standards.
4.	Tools required	• VA/PT reports from CERT-In empanelled or NCCS-designated lab reports • SAST/DAST tools • Source code review tools • Dependency vulnerability scanners • SIEM/log analysis tools • Version control system audit logs • Hardware inspection tools/debuggers • Secure boot/TEE validation tools
5.	Test procedure	Step 1: Review submitted VA/PT reports issued by NCCS-designated or CERT-In empanelled security assessment organizations and verify compliance with latest CERT-In and Sectoral CERT guidelines. Step 2: Verify that VA/PT scope includes APIs, web UI, backend services, databases, cloud integrations, HSM interfaces and external data interfaces. Step 3: Verify that both automated and manual penetration testing methodologies were performed. Step 4: Review vulnerability classification, CVSS scoring, mitigation recommendations and closure evidence for identified findings. Step 5: Verify that all Critical and High severity vulnerabilities have been remediated and re-tested successfully.

Sl. No.	Test Details	Details
		Step 6: Verify that residual risks are formally documented and approved through risk acceptance procedures. **Step 7:** Perform static code analysis using approved SAST tools and review findings related to input validation, authentication, session management, cryptographic implementation and exception handling. **Step 8:** Perform manual code inspection to verify: • Input validation and sanitization • Proper authentication/session management • Absence of hardcoded cryptographic keys • Secure error handling without information leakage • Use of cryptographic abstraction layers instead of hardcoded algorithms **Step 9:** Verify vulnerability assessment coverage against comprehensive security frameworks rather than limited references such as OWASP Top 10 alone. **Step 10:** Perform vulnerability scanning of third-party libraries and dependencies and verify remediation of known vulnerabilities. **Step 11:** Verify use of version control systems with restricted access controls and MFA protection. **Step 12:** Verify that code commits and merges require peer review and approval workflows. **Step 13:** Verify adherence to latest secure coding and design guidelines issued by CERT-In and Sectoral CERTs. **Step 14:** Where hardware assurance applies, inspect secure elements, TEE, PUFs, secure boot mechanisms and tamper protection features and verify correct implementation and operation.
6.	Expected Result	Test passes if: • VA/PT activities are performed by approved security assessment organizations

Sl. No.	Test Details	Details
		• All relevant application, cloud and HSM components are included in testing scope • Both automated and manual security testing methodologies are used • Critical and High vulnerabilities are remediated and re-tested successfully • Residual risks are documented and formally approved • Secure coding practices are correctly implemented • Cryptographic functions use approved abstraction layers without hardcoded algorithms/keys • Third-party library vulnerabilities are identified and remediated • Version control access protections and peer review workflows are enforced • Hardware assurance mechanisms operate correctly where applicable

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 35 - Verification of QRNG/TRNG Assurance
2.	GR Clause covered	4.3.1.1 to 4.3.1.7
3.	Objective of the test	To verify that the product correctly implements validated QRNG/TRNG mechanisms with authenticated entropy sources, continuous monitoring and protection against spoofing, substitution and entropy manipulation.
4.	Tools required	• QRNG/TRNG validation reports • NIST Statistical Test Suite (STS) • Entropy measurement tools • Hardware inspection tools • Calibration certificates

Sl. No.	Test Details	Details
		• Log monitoring tools
5.	Test procedure	**Step 1:** Verify that the integrated QRNG/TRNG complies with TEC GRs or equivalent internationally recognized standards. **Step 2:** Review scientific documentation and verify the claimed entropy generation mechanism (quantum, optical or physical source). **Step 3:** Verify hardware BoM and physical implementation of entropy source components against design documentation. **Step 4:** Generate entropy samples and validate randomness characteristics using approved statistical analysis tools and entropy tests. **Step 5:** Verify continuous health monitoring mechanisms for entropy source operation and fault detection. **Step 6:** Attempt entropy source spoofing, substitution or disablement scenarios and verify detection and secure failure behavior. **Step 7:** Verify calibration certificates and calibration traceability for entropy source components. **Step 8:** Verify that entropy-derived seeds remain unique, non-repeating and integrity protected across restarts and operational lifecycle events.
6.	Expected Result	Test passes if: • QRNG/TRNG implementation complies with applicable standards • Physical entropy source is genuine and continuously operational • Statistical randomness and entropy requirements are satisfied • Spoofing, substitution or disablement attempts are detected • Calibration evidence is maintained and valid • Entropy-derived seed material remains unique and integrity protected across sessions and restarts

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 36 - Verification of Performance and Resource Monitoring
2.	GR Clause covered	4.4.1 to 4.4.6
3.	Objective of the test	To verify that the product correctly measures, records and reports cryptographic performance, resource utilization, scalability and encrypted traffic overhead metrics.
4.	Tools required	• System monitoring tools • CPU/GPU utilization monitoring tools • Power analyzers • Network traffic analyzers/Wireshark • Performance benchmarking tools • Load/concurrency testing tools
5.	Test procedure	**Step 1:** Execute cryptographic operations and verify measurement/reporting of heap, stack and binary footprint utilization. **Step 2:** Measure CPU/GPU utilization during encapsulation, decapsulation, signature generation and verification operations. **Step 3:** Measure power consumption under idle, average and peak workload conditions and compute energy consumption per cryptographic operation. **Step 4:** Perform scalability testing using concurrent cryptographic sessions and mixed workloads and verify stable operation. **Step 5:** Capture encrypted network traffic and verify packet-level bandwidth overhead, QoS behavior and link utilization metrics. **Step 6:** Verify configurability of performance benchmarks according to sector-specific operational requirements.
6.	Expected Result	Test passes if:

Sl. No.	Test Details	Details
		• Memory and binary footprint metrics are correctly measured and reported • CPU/GPU utilization metrics are accurately recorded • Power consumption measurements are correctly generated • Product remains stable under concurrent workloads • Packet-level overhead and QoS metrics are correctly reported • Performance benchmarks are configurable according to operational requirements

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 37 - Verification of Crypto-Agility Requirements
2.	GR Clause covered	4.5.1 to 4.5.6
3.	Objective of the test	To verify that the product supports dynamic cryptographic transition, rollback and algorithm updates across classical, hybrid and PQC modes with minimal service disruption.
4.	Tools required	• CLI/GUI/API management interfaces • OpenSSL/OQS tools • Configuration management tools • Traffic generators • SLA/performance monitoring tools
5.	Test procedure	**Step 1:** Verify switching between classical, hybrid and PQC algorithms using CLI, GUI and API interfaces. **Step 2:** Verify that cryptographic mode switching occurs without requiring system reboot and within defined SLA disruption limits.

Sl. No.	Test Details	Details
		Step 3: Simulate PQC module failure or overload conditions and verify secure fallback to classical or hybrid cryptography without data loss. **Step 4:** Verify rollback to previously validated cryptographic configurations. **Step 5:** Verify integration and activation of new cryptographic algorithms through firmware/software updates. **Step 6:** Review and verify documented cryptographic transition and upgrade policies.
6.	Expected Result	Test passes if: • Cryptographic mode switching operates correctly through supported interfaces • No reboot is required during transition operations • Fallback and rollback mechanisms operate securely without data loss • New algorithms can be securely integrated through updates • Cryptographic transition policies are documented and enforceable

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 38 - Verification of Security Validation and Assessment
2.	GR Clause covered	4.6.1 to 4.6.3
3.	Objective of the test	To verify that the product supports advanced security validation, automated vulnerability analysis and maintenance of security assessment reports.
4.	Tools required	• Automated vulnerability scanners • Formal verification/security analysis tools

Sl. No.	Test Details	Details
		• Audit and assessment reports • Fault Injection Tool
5.	Test procedure	Step 1: Perform protocol-level attack simulations including multi-vector and fault injection attacks. Step 2: Verify resistance against protocol manipulation, cryptographic misuse and abnormal fault conditions. Step 3: Execute automated vulnerability discovery and security analysis tools against the product. Step 4: Review maintained security audit reports and assessment evidence for completeness and traceability (eg. SOC 2 Type 2 report).
6.	Expected Result	Test passes if: • Product resists protocol-level and multi-vector attacks • Automated vulnerability analysis mechanisms operate correctly • Security audit and assessment reports are maintained and available

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 39 - Verification of Key Derivation and Cryptographic Integration
2.	GR Clause covered	4.7.1 to 4.7.6
3.	Objective of the test	To verify secure key derivation, entropy integration and centralized cryptographic management integration.
4.	Tools required	• KDF validation vectors • Centralized key management systems • REST/SNMP testing tools • Wireshark/network analyzers

Sl. No.	Test Details	Details
5.	Test procedure	**Step 1:** Verify that final encryption keys are derived using approved secure KDF mechanisms. **Step 2:** Verify secure combination of PQC outputs, QRNG entropy and QKD-derived keys where applicable. **Step 3:** Verify secure integration with centralized cryptographic management systems. **Step 4:** Verify confidentiality, integrity and authenticated access during management communications. **Step 5:** Verify reporting of cryptographic inventory, algorithm versions and health metrics. **Step 6:** Verify confidentiality and integrity protection for supported management protocols such as REST APIs and SNMP.
6.	Expected Result	Test passes if: • Secure KDF mechanisms are correctly implemented • Multiple entropy sources are securely integrated • Centralized management integration operates securely • Management communications are authenticated and protected • Cryptographic inventory and health reporting operate correctly

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 40 - Verification of CI/CD and Regression Validation
2.	GR Clause covered	4.8.1 to 4.8.7
3.	Objective of the test	To verify that the product supports automated CI/CD validation, regression testing, traceability and repeatable secure builds.
4.	Tools required	• CI/CD platforms (Jenkins/GitLab/GitHub Actions etc.)

Sl. No.	Test Details	Details
		• Automated test frameworks • Version control systems • Security testing tools • Build verification tools
5.	Test procedure	**Step 1:** Verify integration with CI/CD pipelines for automated regression testing. **Step 2:** Verify automatic validation triggering upon code, configuration or cryptographic changes. **Step 3:** Verify automated execution of unit, integration, interoperability and security tests within the CI/CD pipeline. **Step 4:** Verify that builds failing security or quality criteria are prevented from promotion/release. **Step 5:** Verify traceability and retention of test results and security findings for audit purposes. **Step 6:** Verify support for repeatable builds, version control and rollback mechanisms. **Step 7:** Verify validation support across multiple operating systems and hardware platforms including Linux, Windows and ARM-based systems.
6.	Expected Result	Test passes if: • CI/CD integration and automated regression testing operate correctly • Validation triggers occur automatically upon changes • Security and interoperability tests are automatically executed • Failed builds are blocked from release • Audit traceability and test result retention are maintained • Repeatable builds and rollback capabilities operate correctly • Multi-platform validation support is available

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 41 - Verification of Supply Chain and Compliance Requirements
2.	GR Clause covered	4.9.1, 4.9.2
3.	Objective of the test	To verify that the product implements supply chain security controls and supports compliance with sector-specific cryptographic policies.
4.	Tools required	• SBOM analysis tools • Supply chain verification documentation • Compliance assessment reports
5.	Test procedure	**Step 1:** Verify availability and accuracy of SBOM and component traceability records. **Step 2:** Ask for NSDTS Trusted Source and Trusted Product Certificate if applicable. **Step 3:** Verify support for compliance with applicable sector-specific cryptographic and regulatory policies.
6.	Expected Result	Test passes if: • Supply chain security controls are implemented across all critical components • Component traceability and SBOM records are maintained • Product supports applicable sector-specific compliance requirements

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 42 - Verification of Strategic Cryptographic Resilience

Sl. No.	Test Details	Details
2.	GR Clause covered	4.10.2.1 to 4.10.2.4
3.	Objective of the test	To verify that the product supports rapid and secure cryptographic transition, fallback and diversification mechanisms during algorithm compromise or operational security events.
4.	Tools required	• CLI/GUI/API management tools • Configuration management tools • Traffic generators • SLA monitoring tools
5.	Test procedure	**Step 1:** Simulate compromise or deprecation of an active cryptographic algorithm and verify rapid transition to alternative approved algorithms. **Step 2:** Verify transition to alternative PQC algorithms or QKD-based mechanisms within defined operational time limits without service compromise. **Step 3:** Verify operation of pre-configured fallback and validated transition procedures during algorithm migration events. **Step 4:** Verify support for algorithm diversification and hybrid cryptographic deployment. **Step 5:** Verify continuity of cryptographic services and protection of data during transition events.
6.	Expected Result	Test passes if: • Rapid cryptographic transition mechanisms operate correctly • Alternative PQC/QKD transitions occur securely within defined limits • Fallback and rollback procedures are validated and operational • Hybrid and diversified models are supported • No loss of security or operational integrity occurs during transitions

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 43 - Verification of QKD Integration Readiness
2.	GR Clause covered	4.10.3
3.	Objective of the test	To verify that the product supports readiness for secure integration with Quantum Key Distribution (QKD) systems in accordance with TEC GRs or equivalent standards.
4.	Tools required	• QKD integration test environment • ETSI QKD interface specifications • Key management tools
5.	Test procedure	**Step 1:** Verify supported QKD interfaces, APIs and key delivery mechanisms s per ETSI QKD specification. **Step 2:** Verify secure import and integration of externally generated QKD keys into cryptographic operations. **Step 3:** Verify confidentiality, integrity and authentication protection during QKD key exchange and delivery. **Step 4:** Verify logging and auditability of QKD-related cryptographic operations.
6.	Expected Result	Test passes if: • Product supports QKD integration readiness requirements • QKD-generated keys are securely integrated into operations • QKD communications are protected for confidentiality and integrity • QKD operations are auditable and securely logged

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 44 - Verification of Fail-Secure Behavior

Sl. No.	Test Details	Details
2.	GR Clause covered	4.10.4
3.	Objective of the test	To verify that the product operates in fail-secure mode under cryptographic or system failures and does not enter insecure fail-open conditions.
4.	Tools required	• Fault injection tools • System crash simulation tools • Power interruption tools • Log monitoring tools
5.	Test procedure	**Step 1:** Simulate cryptographic module failures, process crashes and abnormal termination conditions. **Step 2:** Simulate system failures including power interruption, network failure and resource exhaustion scenarios. **Step 3:** Verify that cryptographic protections remain enforced during failure conditions and that unauthorized access is not granted. **Step 4:** Verify that failed operations are securely terminated and sensitive information is not exposed. **Step 5:** Verify logging and audit generation during failure events.
6.	Expected Result	Test passes if: • Product maintains fail-secure operation under all tested failure conditions • No fail-open condition occurs • Sensitive information remains protected during failures • Failure events are securely logged and auditable

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 45 - Verification of Disaster Recovery and Business Continuity
2.	GR Clause covered	4.10.5
3.	Objective of the test	To verify that the product supports secure disaster recovery, business continuity, failover and failback mechanisms while maintaining cryptographic integrity and state consistency.
4.	Tools required	• Multi-site deployment environment • Backup/restore tools • Failover orchestration tools • Replication monitoring tools • Network simulators
5.	Test procedure	Step 1: Deploy the product across multiple sites and verify secure synchronization of cryptographic state and operational data. Step 2: Simulate site failure, communication interruption and disaster recovery conditions. Step 3: Verify failover to backup systems within defined RTO/RPO limits. Step 4: Verify failback operations after restoration of primary systems. Step 5: Verify that no cryptographic keys, sensitive data or operational integrity are lost during DR/BC events. Step 6: Verify auditability and logging of DR/BC operations.
6.	Expected Result	Test passes if: • DR and BC mechanisms operate correctly • Multi-site deployments maintain secure state consistency • RTO and RPO requirements are satisfied • Failover/failback operations succeed securely • No key compromise or data loss occurs during recovery events

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 46 - Verification of Zero Trust Architecture Compliance
2.	GR Clause covered	4.10.6
3.	Objective of the test	To verify that the product supports Zero Trust Architecture principles including continuous authentication, authorization and least-privilege enforcement.
4.	Tools required	• IAM/RBAC management tools • MFA-enabled accounts • Network access control tools • Session monitoring tools
5.	Test procedure	Step 1: Verify continuous authentication and authorization mechanisms during active sessions. Step 2: Verify least-privilege access enforcement across cryptographic operations and management interfaces. Step 3: Attempt unauthorized access escalation and lateral movement scenarios and verify rejection. Step 4: Verify session revalidation and policy enforcement during operational state changes.
6.	Expected Result	Test passes if: • Continuous authentication and authorization mechanisms operate correctly • Least-privilege access controls are enforced • Unauthorized access and lateral movement attempts are prevented • Session revalidation mechanisms operate securely

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 47 - Verification of Red Teaming and Advanced Threat Validation
2.	GR Clause covered	4.10.7
3.	Objective of the test	To verify that the product undergoes independent red team testing and that identified findings are risk-rated, remediated and revalidated.
4.	Tools required	• Red team tools and assessment reports • Attack simulation tools • Vulnerability tracking systems
5.	Test procedure	**Step 1:** Review independent red team assessment reports and testing scope. **Step 2:** Verify simulation of real-world attack scenarios against cryptographic services, infrastructure and operational controls. **Step 3:** Review documented findings, risk ratings and remediation evidence. **Step 4:** Verify revalidation and closure of remediated findings.
6.	Expected Result	Test passes if: • Independent red team testing is performed • Findings are documented and risk-rated • Remediation actions are completed and validated • Revalidation confirms closure of identified risks

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 48 - Verification of Advanced Supply Chain Assurance

Sl. No.	Test Details	Details
2.	GR Clause covered	4.10.8
3.	Objective of the test	To verify semi-formal verification of component provenance, integrity and critical security component correctness.
4.	Tools required	• SBOM verification tools • Integrity validation tools • Provenance documentation • Firmware verification tools
5.	Test procedure	**Step 1:** Verify provenance and integrity records for critical hardware, firmware and software components. **Step 2:** Review semi-formal verification evidence for critical security components. **Step 3:** Verify integrity validation mechanisms for critical components and dependencies. **Step 4:** Review documentation of verification scope, assumptions and limitations.
6.	Expected Result	Test passes if: • Component provenance and integrity are verifiable • Critical components undergo semi-formal verification • Verification scope and limitations are documented • Integrity validation mechanisms operate correctly

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 49 - Verification of Nation-State Threat Evaluation
2.	GR Clause covered	4.10.9

Sl. No.	Test Details	Details
3.	Objective of the test	To verify that the product is evaluated against advanced persistent threat scenarios including cryptographic, infrastructure and supply-chain attacks.
4.	Tools required	• Threat emulation frameworks • APT simulation tools • Supply-chain attack simulation tools • Advanced penetration testing tools
5.	Test procedure	**Step 1:** Execute advanced threat simulations representing nation-state level attack capabilities. **Step 2:** Simulate supply-chain compromise, cryptographic downgrade and persistent access scenarios. **Step 3:** Verify detection, containment and response mechanisms against advanced threats. **Step 4:** Review assessment evidence, mitigation controls and residual risk analysis.
6.	Expected Result	Test passes if: • Product demonstrates resilience against advanced persistent threats • Supply-chain and cryptographic attacks are detected or mitigated • Response and containment mechanisms operate correctly • Residual risks are documented and controlled

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 50 - In-Field Firmware Upgrade Support

2.	GR Clause covered	5.3.3.1
3.	Objective of the test	To verify that the system supports in-field firmware upgrades from time to time to ensure continued functionality, compatibility with advancements in technology, and interoperability with supporting systems.
4.	Tools required	• Valid digitally signed firmware upgrade package • Firmware upgrade utility/software
5.	Test procedure	• Verify current firmware version via system command. • Initiate update using a valid, digitally signed firmware package. Check if authentication mechanisms are implemented. • Verify that the firmware version is updated correctly after successful installation. • Verify that the module operates normally after the firmware update.
6.	Expected Result	• It shall support remote system Software/Firmware upgrades. • The system shall correctly display the firmware version. • The user shall be authenticated before initiating the firmware update. • The firmware is successfully updated.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 51 - High Entropy Throughput and Randomness Verification
2.	GR Clause covered	5.3.3.10
3.	Objective of the test	To verify that the cryptographic module/device supports high entropy throughput and generates high-quality random data with

		sufficient entropy and statistical randomness for cryptographic applications.
4.	Tools required	• Statistical randomness analysis tools
5.	Test procedure	As per NIST SP 800-22 • Selection of a Generator • Binary Sequence Generation • Execute the Statistical Test Suite (Invoke the NIST Statistical Test Suite using the file produced in Stage 2 and the desired sequence length. Select the statistical tests and relevant input parameters (e.g., block length) to be applied.) • Examine the P-values (set of sequence) α=….
6.	Expected Result	• The generator should produce a binary sequence of 0's and 1's of a given length *n* • For a fixed sequence of length *n* and the pre-selected generator, construct a set of *m* binary sequences and save the sequences to a file. • sequence passes a statistical test whenever the P-value ≥ α and fails otherwise.

Sl. No.	Test Details	Details
1.	Name of the test case	Test case 52 - RADIUS/TACACS+ Authentication Support Verification
2.	GR Clause covered	5.3.3.18

3.	Objective of the test	To verify that the cryptographic system supports authentication through RADIUS and/or TACACS+ servers for secure user authentication and access control.
4.	Tools required	• Configured RADIUS and TACACS+ server
5.	Test procedure	• Configure and verify connectivity • Login using valid and invalid credentials and check the result • Check the login for denied user.
6.	Expected Result	• cryptographic system shall authenticate users through RADIUS and TACACS+ servers • Valid users shall be granted access and Invalid users shall be denied access. • The system shall maintain secure access control during authentication server failure conditions.

Sl. No.	Test Details	Details
1.	Name of the test case	Test Case 53 – Verification of Authentication, Key Integrity and Secure Key Storage Mechanisms
2.	GR Clause covered	4.1 (i), (ii), (iii)
3.	Objective of the test	To verify that the cryptographic module implements operator authentication and role-based authorization, supports secure encryption/decryption key change, performs key integrity and authentication verification using an approved hashing mechanism, and stores encryption keys securely with access restricted to authorized users only.

Sl. No.	Test Details	Details
4.	Tools required	• CLI/Web GUI • Management Interface • Administrator/User Credentials • Cryptographic Module Documentation • Key Management Utility
5.	Test procedure	**Step 1:** Access the module management interface using valid operator credentials. **Step 2:** Verify that authentication is required before access to cryptographic services and role-specific functions is granted. **Step 3:** Attempt to access privileged services using credentials associated with a different role and verify that access is denied. **Step 4:** Perform encryption/decryption key change (key update/replacement) operation and verify successful completion without loss or corruption of encrypted data. **Step 5:** Generate, import or store a cryptographic key and verify that the module performs a key integrity/authentication check using the implemented hashing mechanism. **Step 6:** Modify or corrupt the stored key or integrity verification value and verify that the module detects the integrity failure and rejects the key. **Step 7:** Verify that encryption keys are stored only in protected storage and are not accessible in plaintext form through any interface. **Step 8:** Attempt to access stored encryption keys using unauthorized credentials and verify that access is denied. **Step 9:** Review logs, status messages, and vendor documentation to confirm compliance with key protection and authentication requirements.

Sl. No.	Test Details	Details
6.	Expected Result	a) The module authenticates operators before granting access to services. b) Only authorized roles can access corresponding cryptographic services and management functions. c) Encryption/decryption key change operation is completed successfully without data loss. d) Key integrity and authentication verification is performed using the implemented hashing mechanism. e) Any modified or corrupted key is detected and rejected by the module. f) Encryption keys remain encrypted/protected while stored. g) Stored keys are accessible only to authorized users and cannot be retrieved in plaintext by unauthorized entities. h) The observed behavior matches vendor documentation and security policy.

J. SUMMARY OF TEST RESULTS

TEC Standard No. TEC 91010:2026

TEC Test Guide No. TEC 91011:2026

Equipment name & Model No. _____

Clause No.	*Compliance (Complied /Not Complied / Submitted/Not Submitted / Not Applicable)*	*Remarks / Test Report Annexure No.*

Date:

Place: Signature & Name of TEC testing officer/

*Signature of Applicant/ Authorized Signatory