

Government of India

Ministry of Electronics and Information Technology

Office of Controller of Certifying Authorities

Draft Report on IT Act Cryptographic Standards and Post Quantum Cryptography

Expert Committee for Evaluation of IT Act Cryptographic Standards and Recommendations on Post Quantum Cryptography Algorithms

Draft for Public Review and Suggestions

Technology | Implementation Readiness | Techno Policy

Constituted vide Office Memorandum F. No. 19(92)/2026-CCA dated 09 May 2026

October 2026

Comments and suggestions may be sent to ram@cca.gov.in

Last date for receipt of comments and suggestions: [20 November 2026](#)

Purpose and Scope

The Office of the Controller of Certifying Authorities constituted the Expert Committee to evaluate cryptographic standards under the Information Technology Act, 2000, and recommend post-quantum cryptography algorithms. The Committee organised its examination into three tracks covering technology, implementation readiness and techno policy. This consolidated report presents their assessments, recommendations and proposed rule amendments for public review and suggestions. The recommendations and proposed amendments remain subject to review and consideration of the suggestions received.

The technology track examines quantum risks, current cryptographic practice, alternatives and algorithm recommendations. The implementation readiness track assesses the Indian PKI ecosystem, standards, suppliers, applications, performance and migration. The techno policy track examines legal changes, risks to legacy records and arrangements for phased migration, including interim measures. Together, these tracks cover digital signatures, relevant certificate-based authentication and key-establishment uses, and preservation of signed records.

The report does not provide complete implementation specifications for every application, a comprehensive PQC migration plan for all sectors, or detailed treatment of quantum computing infrastructure and quantum key distribution.

The technical considerations for amendments in Part I and the proposed rule amendments in Annexure 1 should be read together. The former addresses technical standards; the latter sets out proposed changes to the operative Rules. The proposed amendments retain their original addition and deletion markings, and their wording and placement may be refined during legal vetting to meet legislative drafting requirements.

Public comments are invited on the report's assessments, recommendations and proposed amendments, including any gaps, ambiguities or relevant PQC aspects not adequately addressed, to inform revision of the report and identify matters requiring further examination.

Contents

Part I Technology Track

1. Description of the problem
2. Current practice
3. Available alternatives
4. Analysis of alternatives
5. Recommendations and technical considerations for amendments

Part II Implementation Readiness Track

1. Current PKI ecosystem in India
2. Readiness of PKI data structures and associated standards
3. Impact on suppliers of PKI systems and components
4. Readiness of supplier and regulator ecosystem
5. Impact on applications
6. Readiness of applications
7. Impact on system performance
8. Migration related aspects
9. Recommendations

Part III Techno Policy Track

1. Assessment of changes needed to the IT Act and Rules
2. Risk to legacy records from quantum computing
3. Migration approaches and interim measures
4. Recommendations

Annexure 1 Proposed Rule Amendments

PART I TECHNOLOGY TRACK

TRACK-1: TECHNOLOGY TRACK

1. Description of the Problem, Including the Impact of Quantum Computing on Existing Cryptographic Algorithms

India's digital-trust framework is founded on the Information Technology Act, 2000, which recognises a digital signature as a method of authenticating an electronic record through an asymmetric cryptosystem and a hash function. The signatory controls a confidential private key, while the corresponding public key enables others to verify the identity of the signer and determine whether the record was altered after signing.

A digital signature primarily provides authentication, integrity and accountability; it does not ordinarily conceal the contents of the signed document. Confidentiality is provided through encryption, which protects information from unauthorised access.

For this report, 'pre-quantum' refers to public key cryptography based on algorithms such as RSA, DSA and ECC.

Although the Information Technology Act is substantially technology-neutral, its subordinate legislation and technical standards currently implement digital signatures mainly through existing public key cryptographic mechanisms. Rule 6 of the Information Technology (Certifying Authorities) Rules, 2000, and the Digital Signature (End Entity) Rules, 2015, refer to RSA, DSA, NIST elliptic curves, SHA-2, PKCS standards and X.509 certificates.

The existing framework also relies on standards such as CMS, RFC 5280, CAdES, PAdES and XML Signature for creating, processing and validating digital signatures. The reliability of India's digital-signature infrastructure therefore depends not only on cryptographic algorithms but also on the standards, devices, services and applications through which those algorithms are implemented.

The Controller of Certifying Authorities, as the statutory authority responsible for licensing and regulating Certifying Authorities, faces a systemic challenge rather than a problem confined to a single algorithm. Quantum-related vulnerabilities could affect the wider national trust framework supporting electronic authentication, e-governance, e-commerce, digital transactions, secure communications and device identities.

RSA, DSA and elliptic-curve cryptography are considered secure because they rely on mathematical problems that ordinary computers cannot solve within a practical period. A sufficiently powerful quantum computer could use methods such as Shor's algorithm to solve the factorisation and discrete-logarithm problems on which these systems depend.

Such a capability could enable an attacker to derive private signing keys from public keys, forge apparently valid signatures, impersonate certificate holders and issue fraudulent certificates. It could also weaken authentication systems, certificate-based trust and the evidentiary reliability of digitally signed electronic records.

Quantum computing also creates risks for information protected through public-key encryption. An attacker may collect encrypted information today and retain it until sufficiently capable quantum computers become available, a threat commonly described as "harvest now, decrypt later." Future quantum attacks may also enable the forgery of signatures, certificates and time stamps, undermining the authenticity of electronic records. This risk is described as "forge later." Premature adoption of inadequately tested PQC systems may introduce new security weaknesses, a risk described as "upgrade now, break later."

Within India's CCA-regulated framework, however, public-key cryptography is used predominantly for creating and verifying digital signatures rather than for encrypting document contents. The principal concern for the Indian

PKI is therefore the possible compromise of signing keys, resulting in forged signatures, impersonation and loss of confidence in the authenticity, integrity and evidentiary value of electronic records.

Use approved symmetric algorithms with suitable key lengths, such as AES-256, for strong quantum-security margins. Larger RSA or ECC keys may provide a limited interim margin by increasing attack-resource requirements, while planned migration to approved PQC algorithms provides the long-term direction.

Quantum computers could weaken hash protection by making it easier to find different data with the same hash value. Quantum computers can make it easier to find data matching a given hash or two different pieces of data with the same hash. Applications should therefore use an approved hash function with sufficient output length for the required protection. This document hash is separate from the hashing built into the PQC algorithm, which must follow its standard.

Increasing an RSA key from 2,048 bits to 3,072 or 4,096 bits does not remove the quantum vulnerability. Larger RSA keys continue to depend on the same factorisation problem and may therefore remain vulnerable to a sufficiently capable quantum computer.

The risk is not limited to transactions conducted after the development of quantum computers. Long-retention records, device certificates, archival signatures, time-stamping certificates and trust-anchor certificates may remain in use or require verification when existing public-key algorithms are no longer considered secure.

The hierarchical structure of the Indian Public Key Infrastructure increases the seriousness of this risk. Compromise of the signing key of the Root Certifying Authority of India, a licensed Certifying Authority or a Time-Stamping Authority could affect large numbers of certificates, digital signatures, transactions and relying applications.

The problem extends across the entire lifecycle of certificates and digital signatures, including their generation, issuance, storage, use, revocation and validation. It therefore affects the Root Certifying Authority of India, licensed Certifying Authorities, CA software, Hardware Security Modules, subscriber smart cards, cryptographic tokens, eSign platforms and device-certificate systems.

The affected infrastructure also includes certificate repositories, Certificate Revocation Lists, Online Certificate Status Protocol services, time-stamping systems, disaster-recovery facilities and applications that create or validate CMS, PDF and XML signatures. A weakness or incompatibility in any of these components could impair the operation of the wider trust framework.

The impact would extend to applications that depend on RSA- or ECC-based mechanisms for authentication, digital signatures and message integrity. In financial systems such as RTGS and NEFT, compromise of these algorithms could enable impersonation, forged transaction instructions or alteration of payment messages. Similar risks could affect e-governance, e-commerce and other critical digital services, resulting in financial loss, operational disruption and loss of trust.

The emerging post-quantum environment also creates operational and interoperability concerns. Existing HSMs and cryptographic tokens may not support new algorithms or may be unable to generate and securely store the required private keys.

Post-quantum certificates, public keys and digital signatures may also be substantially larger than their existing equivalents. Existing applications, communication protocols, document validators and archival systems may be unable to process these larger structures or recognise new algorithm identifiers and certificate profiles.

The problem is further complicated by the continuing dependence of essential public and private services on RSA and elliptic-curve cryptography. Rapid withdrawal of these algorithms could disrupt existing services, while delayed action could prolong exposure to quantum-related vulnerabilities.

The potential adverse impact of quantum computing on the Indian PKI is therefore not limited to the weakening of individual cryptographic algorithms. It extends to the reliability of digital signatures, certificates, trust anchors, cryptographic devices, validation services, relying applications and long-term electronic records.

The central problem is to address the emerging loss of confidence in pre-quantum cryptography without undermining the continuity, interoperability and legal validity of India's existing digital-trust framework.

2. Current Cryptographic Practices and Algorithms Deployed in the PKI Ecosystem

India's Public Key Infrastructure operates through a hierarchical trust model led by the Root Certifying Authority of India and supported by licensed Certifying Authorities, subscribers and relying parties. Digital certificates bind verified identities to public keys and support authentication, digital signatures, message integrity and trusted electronic transactions.

Current practice remains based mainly on existing public key cryptography. The Information Technology (Certifying Authorities) Rules, 2000, and the Digital Signature (End Entity) Rules, 2015, recognise RSA, DSA, NIST elliptic-curve cryptography, SHA-2, X.509 certificates and related PKCS standards.

X.509 version 3 is the principal certificate format, while PKIX principles govern certificate-chain construction and validation. A certificate binds an individual, organisation, device or service to a public key issued under a trusted Certifying Authority.

RSA is the principal public-key algorithm used for certificate signing and end-entity digital signatures. Existing standards provide for 2,048-bit and 4,096-bit RSA keys and refer to the PKCS #1 framework, including PKCS #1 version 1.5 signature mechanisms.

DSA is recognised under the Information Technology (Certifying Authorities) Rules, 2000, although current end-entity standards place greater operational emphasis on RSA and elliptic-curve mechanisms. Elliptic-curve cryptography is supported through the NIST P-256, P-384 and P-521 curves, subject to compatibility across CA software, Hardware Security Modules, subscriber devices and relying applications.

Hash functions are central to India's PKI because a digital signature is generated over a compact digest of the electronic record rather than the entire document. SHA-2 is prescribed for digital-signature operations, and SHA-256 is expressly used in XML signature profiles; any later alteration changes the digest and causes verification to fail.

Section 3 of the Information Technology Act, 2000, defines a digital signature through the combined use of an asymmetric cryptosystem and a hash function. A signature algorithm that does not follow this hash-based construction would therefore require examination of the Act, the Rules and the applicable technical standards.

Hashing also provides much of the efficiency and privacy of online eSign. Only the document hash is sent for signing, allowing the complete document to remain within the Application Service Provider's environment.

Private keys of the Root CA, licensed CAs and Time-Stamping Authorities are protected within controlled cryptographic systems, particularly Hardware Security Modules. Subscriber signing keys may be stored in certified tokens or smart cards, or generated and used within approved remote-signing and eSign environments.

PKCS #10 is used for certificate-signing requests, while PKCS #8 represents private-key information. PKCS #12 supports portable storage or transport of certificates and private-key material, and PKCS #5 provides password-based protection.

CMS and PKCS #7 package digital signatures, certificates and signed attributes in a standard form. They support attached and detached signatures used by applications that create and validate electronic records.

Document-signing applications use CAdES and PAdES for advanced and long-term signatures. PAdES supports PDF documents, while CAdES extends CMS with validation evidence required for continued verification.

PQC migration of PDF signatures requires compatible signature containers, algorithm identifiers and validators, together with preservation of the original signed content and its validation evidence.

XML-based transactions use XML Digital Signature with prescribed rules for canonicalisation, transformation, digest calculation and encoding. Current profiles specify SHA-256 with RSA PKCS #1 version 1.5, X.509 version 3 certificates, Base64 encoding, and enveloped, enveloping or detached signatures.

Certificate validation follows RFC 5280 and the applicable certificate policies. Relying applications examine the chain of trust, validity period, permitted key usage, signature algorithm and status of the issuing authority.

Certificate status is published through Certificate Revocation Lists and Online Certificate Status Protocol services. In eSign deployments, the applicable validation profile uses OCSP evidence and long-term validation information so that a signature can remain verifiable after the short-lived certificate has expired.

RFC 3161 time-stamp tokens establish that a signature or data object existed at a stated time. Together with certificate-chain and revocation evidence, trusted time stamps support long-term verification and evidentiary assessment.

The operational PKI ecosystem extends beyond algorithms and certificates. It includes CA software, Hardware Security Modules, subscriber devices, eSign platforms, certificate repositories, revocation services, time-stamping systems, disaster-recovery facilities and relying applications.

These mechanisms support e-governance, e-commerce, banking and financial services, electronic filing, organisational workflows and device identities. Such applications rely on public key cryptography to authenticate participants and protect the integrity of transaction messages.

Relying applications must recognise the certificate profiles, algorithm identifiers, signature encodings and validation formats used by Certifying Authorities. Differences in implementation may impair interoperability even when the underlying cryptographic algorithm is approved.

At the application layer, many government and commercial systems rely on signing utilities, middleware, browser components and validation tools supplied by specialized service providers. Such tools connect applications with digital certificates, cryptographic tokens and online signing services and are therefore an important operational component of the Indian PKI ecosystem.

The present ecosystem has a large installed base of RSA certificates, cryptographic tokens, Hardware Security Modules, signing applications and document validators. Elliptic-curve cryptography is recognised within the regulatory framework, but the extent of deployment varies across Certifying Authorities, subscriber devices and relying applications.

The traditional digital-signature model is token-based and decentralised. The subscriber obtains a long-term Digital Signature Certificate, retains the private key in a cryptographic token and remains responsible for its secure custody.

Online eSign follows a centrally facilitated, transaction-based model. After e-KYC authentication, the eSign service generates a one-time key pair, the Certifying Authority issues a short-lived certificate, and the signature is applied within the same online transaction.

The eSign private key is deleted after use, which limits the period of key exposure. The model removes the need for a physical token and allows Application Service Providers to integrate signing directly into web and mobile applications.

Mobile-based and remote-key-storage models are planned under phased implementation within the CCA framework. Remote keys remain protected in Hardware Security Modules and are activated after approved multi-factor authentication, while mobile keys may be protected within the device's secure execution environment.

Current cryptographic practice combines existing public key algorithms, SHA-2 hashing, X.509 certificates, PKCS structures, revocation services, trusted time stamps and application-specific signature formats. The extensive deployment of these mechanisms requires phased migration and continued verification of existing signatures.

Summary of Current Cryptographic Practices and Standards

Function or Component	Current Algorithm or Standard	Use within the PKI Ecosystem
PKI architecture	PKIX and hierarchical certificate chains	Establishes trust from the Root Certifying Authority of India through licensed Certifying Authorities to end-entity certificates.
Digital certificates	X.509 version 3	Binds an individual, organisation, device or service to a public key.
RSA signatures	RSA 2,048/4,096 bits; PKCS #1 version 1.5	Supports CA certificate signing and end-entity digital signatures.
Other public-key algorithms	DSA; NIST P-256, P-384 and P-521	Provides recognised alternatives, subject to implementation support across the PKI ecosystem.
Cryptographic hashing	SHA-2; SHA-256 for XML signatures	Creates the document digest used for signing and for detecting any later alteration.
Certificate requests	PKCS #10	Carries the public key and certificate-request information.
Private-key formats	PKCS #8, PKCS #12 and PKCS #5	Supports private-key representation, portable storage and password-based protection.
Signature containers	PKCS #7 and CMS	Packages digital signatures, certificates and signed attributes.
Document and PDF signatures	CAdES and PAdES	Supports embedded signatures, document integrity, portable verification and long-term validation within a single file.
XML signatures	XML Digital Signature and related XML standards	Applies prescribed digest, canonicalisation, transformation and encoding rules.
Certificate validation	RFC 5280	Governs certificate-path construction and validation.
Revocation services	CRLs and OCSP	Provides certificate-status and revocation evidence to relying applications.
Trusted time	RFC 3161 time-stamp tokens	Establishes that signed data existed at a stated time and supports long-term verification.
CA key protection	Hardware Security Modules	Protects the private keys of the Root CA, licensed CAs and Time-Stamping Authorities.

Function or Component	Current Algorithm or Standard	Use within the PKI Ecosystem
Subscriber key custody	Cryptographic tokens and smart cards	Supports decentralised key storage and subscriber-controlled signing.
Online eSign	One-time key pair and short-lived certificate	Provides centrally facilitated, e-KYC-based signing within a single online transaction.
Emerging key models	Remote HSM storage and secure mobile-key storage under phased implementation	Enables authenticated signing without a physical token while preserving protected key custody and transaction-level control.
Application integration layer	Signing utilities, middleware, browser components and validation tools	Connects applications with digital certificates, cryptographic tokens, eSign services and signature-validation systems.
Application use	eSign, e-governance, banking, e-commerce, electronic filing, organisational workflows and device identities	Uses certificates and digital signatures for authentication, message integrity and trusted electronic transactions.

3. Available Post-Quantum Cryptography (PQC) Alternatives

This section identifies and describes the available PQC alternatives, their functions, parameter sets, state models and standards status. It does not rank the alternatives or prescribe their adoption.

3.1 Purpose and Scope

The alternatives fall into two distinct functional classes. ML-DSA, SLH-DSA, LMS/HSS, XMSS/XMSS-MT and future FN-DSA are digital-signature schemes. ML-KEM and future HQC are key-establishment mechanisms. A key-establishment mechanism creates a shared secret for symmetric protection; it does not create a digital signature, establish signer intent or provide contentCommitment. [4-7, 31-32]

This distinction must be retained in certificate key usage, algorithm identifiers, application profiles and eSign processing. The existence of a PQ public key in a certificate does not by itself make the key suitable for every PKI function. [12-19, 24-25]

3.2 Standards Landscape and Functional Classes

The final NIST standards available in the source documents are FIPS 203 for ML-KEM, FIPS 204 for ML-DSA and FIPS 205 for SLH-DSA. NIST SP 800-208 covers selected stateful hash-based signature profiles. FN-DSA/FALCON and HQC remain future standards work in the source material. [4-7, 31-32]

For continuity with the terminology used during the NIST PQC standardisation process, the corresponding standardised terminology and status are shown below:

Terminology used during NIST standardisation	NIST terminology	Standard	Current Status
CRYSTALS-Kyber	ML-KEM	FIPS 203	Final / standardised
CRYSTALS-Dilithium	ML-DSA	FIPS 204	Final / standardised
SPHINCS+	SLH-DSA	FIPS 205	Final / standardised
FALCON	FN-DSA	FIPS 206	Selected for standardisation; FIPS 206 under development

Terminology used during NIST standardisation	NIST terminology	Standard	Current Status
HQC	HQC-KEM	FIPS 207	Selected for standardisation; standard under development

The high-level figures included in the following subsections are explanatory illustrations; the exact processing, parameter sets, algorithm identifiers and application profiles remain governed by the applicable NIST standards and the profiles cited in this report.

Alternative	Standard or status	Cryptographic function	State model
ML-KEM	FIPS 203 - final	Key encapsulation and shared-secret establishment	Stateless per operation; decapsulation private key retained
ML-DSA	FIPS 204 - final	General-purpose digital signature	Stateless
SLH-DSA	FIPS 205 - final	Hash-based digital signature	Stateless
LMS/HSS	NIST SP 800-208; RFC 8554	Stateful hash-based digital signature	Stateful; one-time leaf use
XMSS/XMSS-MT	NIST SP 800-208; RFC 8391	Stateful hash-based digital signature	Stateful; one-time leaf use
FN-DSA/FALCON	FIPS 206 under development in source documents	Future compact lattice-based digital signature	Stateless
HQC	Selected for future NIST standardisation	Future code-based key-establishment mechanism	KEM
Composite or hybrid constructions	Profiles vary; some specifications remain drafts	Combination of pre-quantum and post-quantum components	Depends on components

3.3 Functional Mapping to Existing Public-Key Uses

The table below records the functional mapping only. It does not express preference, ranking or an implementation decision.

Existing public-key function	Available PQC class	Profile implication
Certificate and CRL signatures	ML-DSA, SLH-DSA, specialised LMS/HSS or XMSS/XMSS-MT, and future FN-DSA	New signature identifiers, public-key encodings and validation support are required.
Subscriber or organisational digital signature	ML-DSA, SLH-DSA, future FN-DSA	Crypto device, certificate, CMS/PDF/XML and relying-party profiles must support the selected signature.
Online eSign or remote signature	Stateless PQ signature schemes, principally ML-DSA or SLH-DSA	eSign interfaces and signed-data profiles must identify the exact signature mode and object sizes.

Existing public-key function	Available PQC class	Profile implication
Firmware, software or secure-boot signature	ML-DSA, SLH-DSA, LMS/HSS, XMSS/XMSS-MT and future FN-DSA	The signed-package format and embedded verifier must implement the chosen algorithm and parameters.
RSA key transport or ECC key agreement	ML-KEM and future KEM alternatives	The protocol must support encapsulation, decapsulation and derivation of a symmetric key.
Document or XML reference digest	SHA-2, SHA-3 or SHAKE as permitted by the application profile	The application digest remains distinct from the signature algorithm and its internal hashing.

3.4 ML-KEM: Module-Lattice-Based Key-Encapsulation Mechanism

FIPS 203 specifies ML-KEM-512, ML-KEM-768 and ML-KEM-1024. The recipient generates an ML-KEM key pair and distributes the public key. A sender encapsulates to that public key, producing a ciphertext and a shared secret. The recipient decapsulates the ciphertext with the private key to recover the same shared secret. The shared secret is then processed through an appropriate key-derivation function and used by symmetric encryption or authentication. [4, 8, 18-19]

Parameter set	NIST category	Public key	Ciphertext
ML-KEM-512	1	800 bytes	768 bytes
ML-KEM-768	3	1,184 bytes	1,088 bytes
ML-KEM-1024	5	1,568 bytes	1,568 bytes

ML-KEM is used for secure-channel key establishment, CMS enveloped data, secure email, encrypted document or storage keys, device provisioning and central decapsulation services. RFC 9935 defines ML-KEM public-key identifiers for X.509 PKI and RFC 9936 defines its use in CMS through KEMRecipientInfo. [18-19]

An ML-KEM key is not a digital-signature key. It cannot sign an eSign transaction, a certificate, a CRL, software or a document. Its output is a shared secret used by another symmetric cryptographic mechanism.

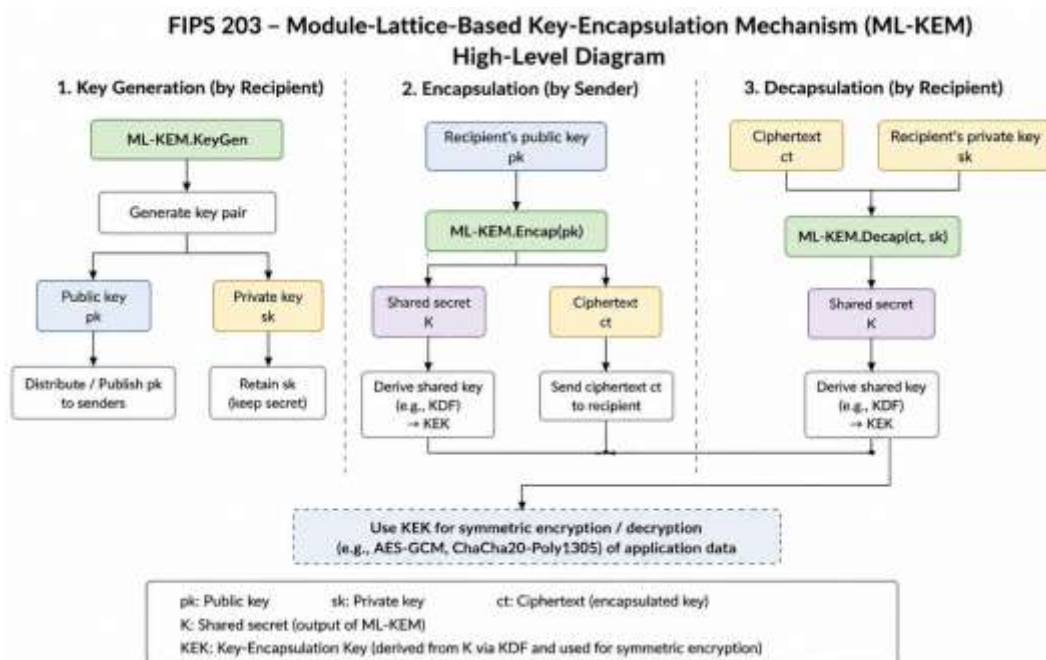


Figure 3-1. High-level operation of ML-KEM: key generation, encapsulation and decapsulation.

Source: Illustrative representation based on NIST FIPS 203.

3.5 ML-DSA: Module-Lattice-Based Digital Signature

FIPS 204 specifies ML-DSA-44, ML-DSA-65 and ML-DSA-87. ML-DSA is a stateless module-lattice digital-signature scheme. Key generation creates a public verification key and a private signing key; signing produces a signature over the supplied message representation; verification checks that signature using the public key. [5]

Parameter set	NIST category	Public key	Signature
ML-DSA-44	2	1,312 bytes	2,420 bytes
ML-DSA-65	3	1,952 bytes	3,309 bytes
ML-DSA-87	5	2,592 bytes	4,627 bytes

RFC 9881 provides algorithm identifiers for ML-DSA public keys and signatures in X.509 certificates and CRLs. RFC 9882 provides ML-DSA identifiers and processing for CMS. These profiles identified in the source documents use pure ML-DSA. [12-13]

FIPS 204 also defines HashML-DSA, in which an explicitly identified approved pre-hash is applied before the ML-DSA processing. Pure ML-DSA and HashML-DSA are different algorithm modes and must use identifiers that bind the exact processing mode and pre-hash function. [5, 12-13]

ML-DSA can be represented in CA, subscriber, remote-signing, eSign, document-signing, code-signing and protocol-authentication profiles when those profiles implement the relevant identifiers, key usages, message processing and object sizes. This statement identifies functional applicability only; relative suitability is analysed in Section 4.

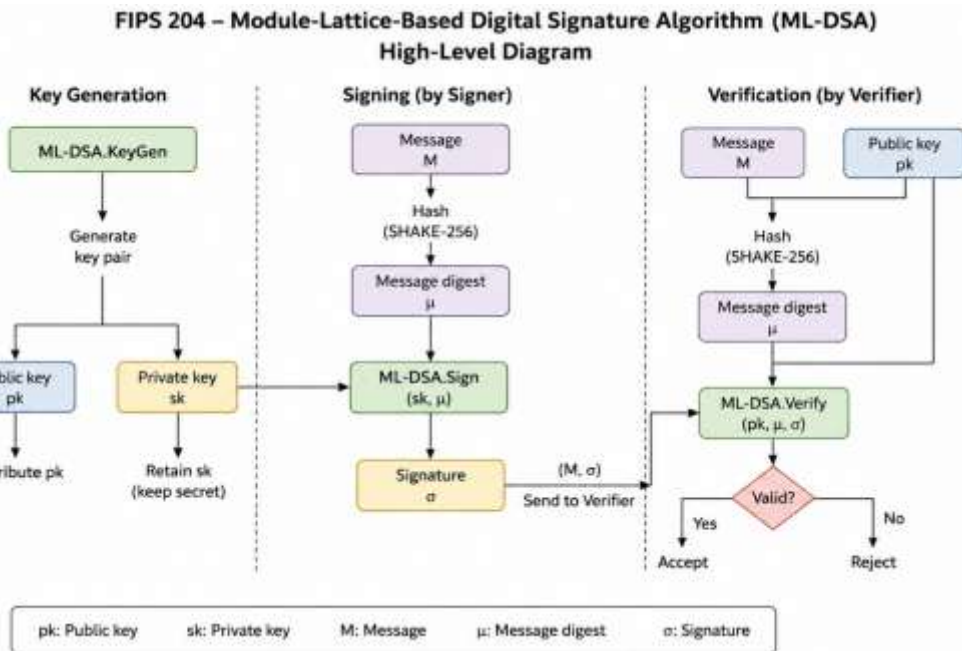


Figure 3-2. High-level operation of ML-DSA: key generation, signing and verification.

Source: Illustrative representation based on NIST FIPS 204.

3.6 SLH-DSA: Stateless Hash-Based Digital Signature

FIPS 205 specifies twelve SLH-DSA parameter sets based on SHA-2 or SHAKE and derived from the SPHINCS+ design. SLH-DSA is stateless. Its security foundation is principally hash-function security, which makes it structurally different from module-lattice signatures. [6]

Family	NIST category	Signature size (s profile)	Signature size (f profile)
SLH-DSA-128	1	7,856 bytes	17,088 bytes
SLH-DSA-192	3	16,224 bytes	35,664 bytes
SLH-DSA-256	5	29,792 bytes	49,856 bytes

The s parameter sets use smaller signatures with slower signing, while the f parameter sets reduce signing time at the cost of larger signatures. RFC 9909 defines pure and pre-hash SLH-DSA identifiers for X.509 certificates and CRLs. RFC 9814 defines pure SLH-DSA for CMS in the source documents. [14-15]

FIPS 205 distinguishes Pure SLH-DSA and HashSLH-DSA. An application-level document digest does not by itself determine the algorithm mode; the signature identifier and application profile must identify the exact processing used. [6, 14-15]

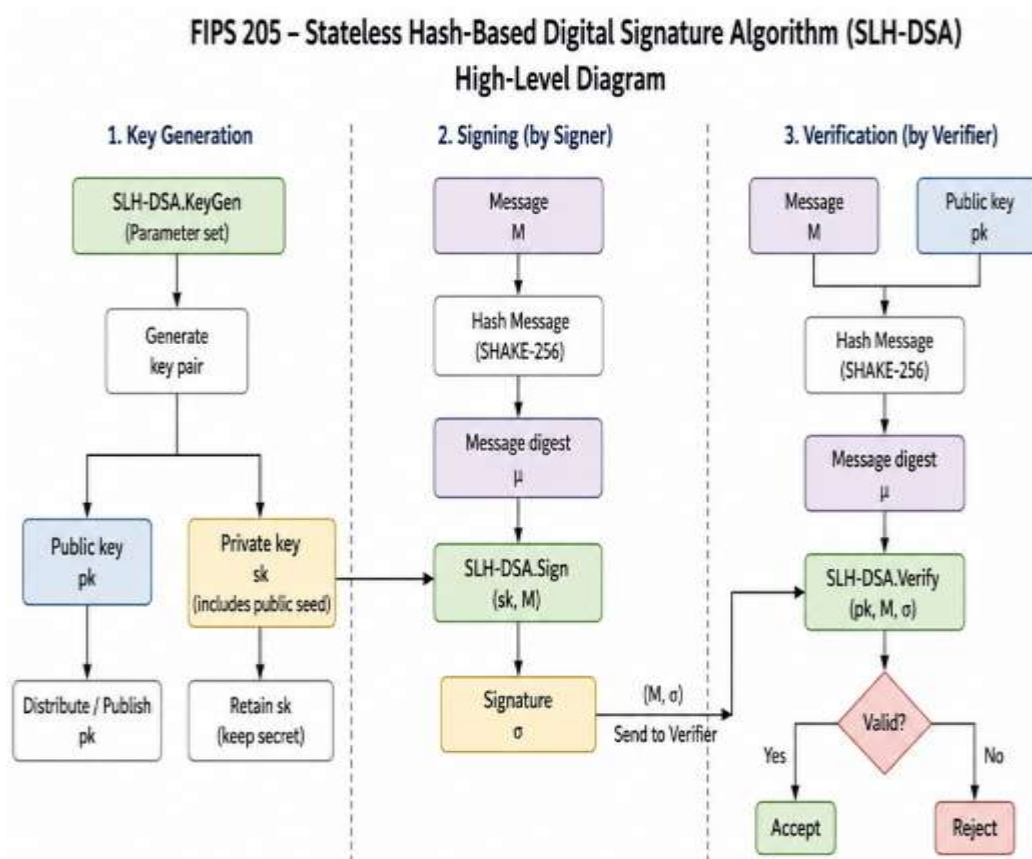


Figure 3-3. High-level operation of SLH-DSA: key generation, signing and verification.

Source: Illustrative representation based on NIST FIPS 205.

3.7 LMS/HSS: Stateful Hash-Based Digital Signature

LMS uses one-time-signature leaves arranged in a Merkle tree. HSS forms a hierarchy of LMS trees. Each signing operation consumes state and a one-time-signature leaf must not be reused. The signer must therefore retain an authoritative counter or equivalent state that survives success, failure, retry, backup and disaster recovery. [7]

NIST SP 800-208 identifies approved parameter sets and requires stringent protection of the private key and state. RFC 9708 defines HSS/LMS use in CMS, while RFC 9802 covers HSS in X.509 PKI in the source documents. [7, 16-17]

The total signature capacity is finite and determined by the selected tree structure. Key rollover, exhaustion monitoring and recovery procedures are part of the cryptographic lifecycle, not merely operational administration.

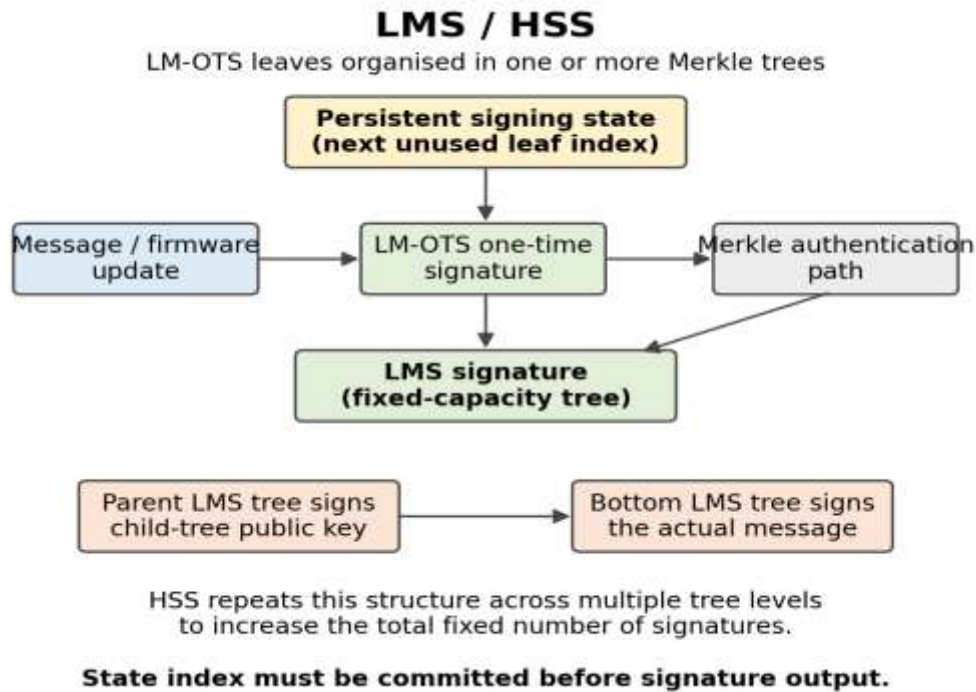


Figure 3-4. High-level stateful signing operation of LMS/HSS.

Source: Illustrative representation based on NIST SP 800-208.

3.8 XMSS and XMSS-MT: Stateful Hash-Based Digital Signature

XMSS is an eXtended Merkle Signature Scheme and XMSS-MT uses multiple tree layers. Like LMS/HSS, each signature consumes state that must never be reused. Backup restoration, cloning, concurrent signers and rollback can cause reuse unless the state lifecycle is designed as an atomic and durable cryptographic control. [7]

NIST SP 800-208 identifies selected XMSS/XMSS-MT profiles. RFC 9802 provides the X.509 profile identified in the source documents. Application support must cover key capacity, state exhaustion, key replacement and compliant disaster recovery. [7, 17]

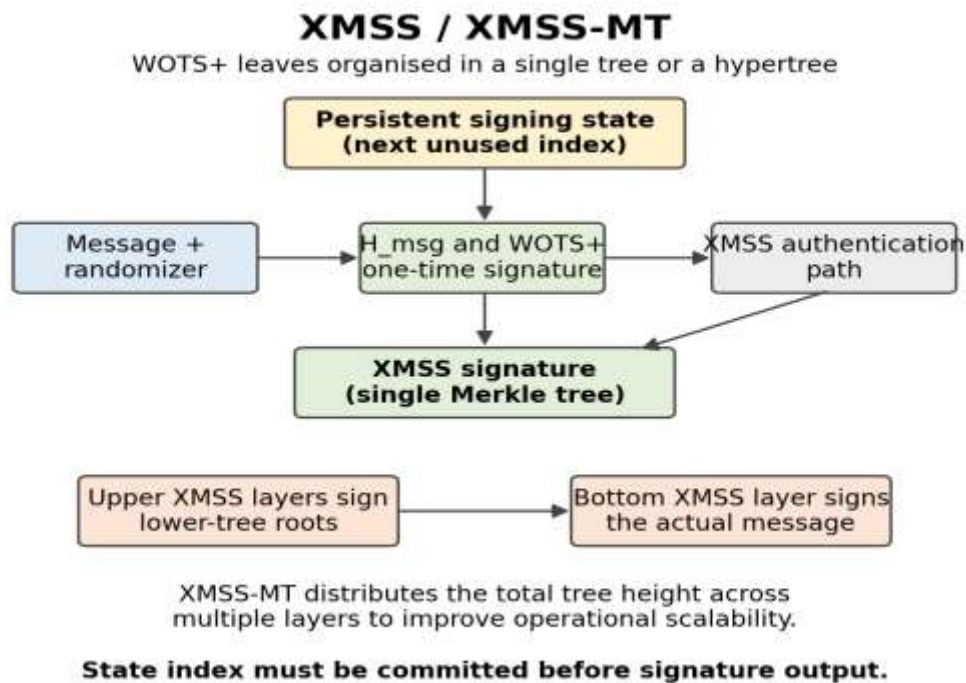


Figure 3-5. High-level stateful signing operation of XMSS/XMSS-MT.

Source: Illustrative representation based on NIST SP 800-208.

3.9 FN-DSA/FALCON: Future Compact Lattice-Based Signature

NIST selected FALCON as the basis for FN-DSA. The design is intended to provide compact lattice-based public keys and signatures and therefore has potential relevance to constrained certificates, tokens, network protocols, firmware and document signatures. The implementation involves complex sampling and numerical requirements that must be captured by the final standard and validated implementations. [11, 32]

In the source documents, FIPS 206 remains under development as of 1 August 2026. Final parameter sets, encodings and supporting X.509, CMS, XML and protocol profiles are therefore not treated as completed production standards in this section. [32]

FIPS 206 (Draft) – FALCON Digital Signature Algorithm (FN-DSA)

High-Level Diagram (Under Development)

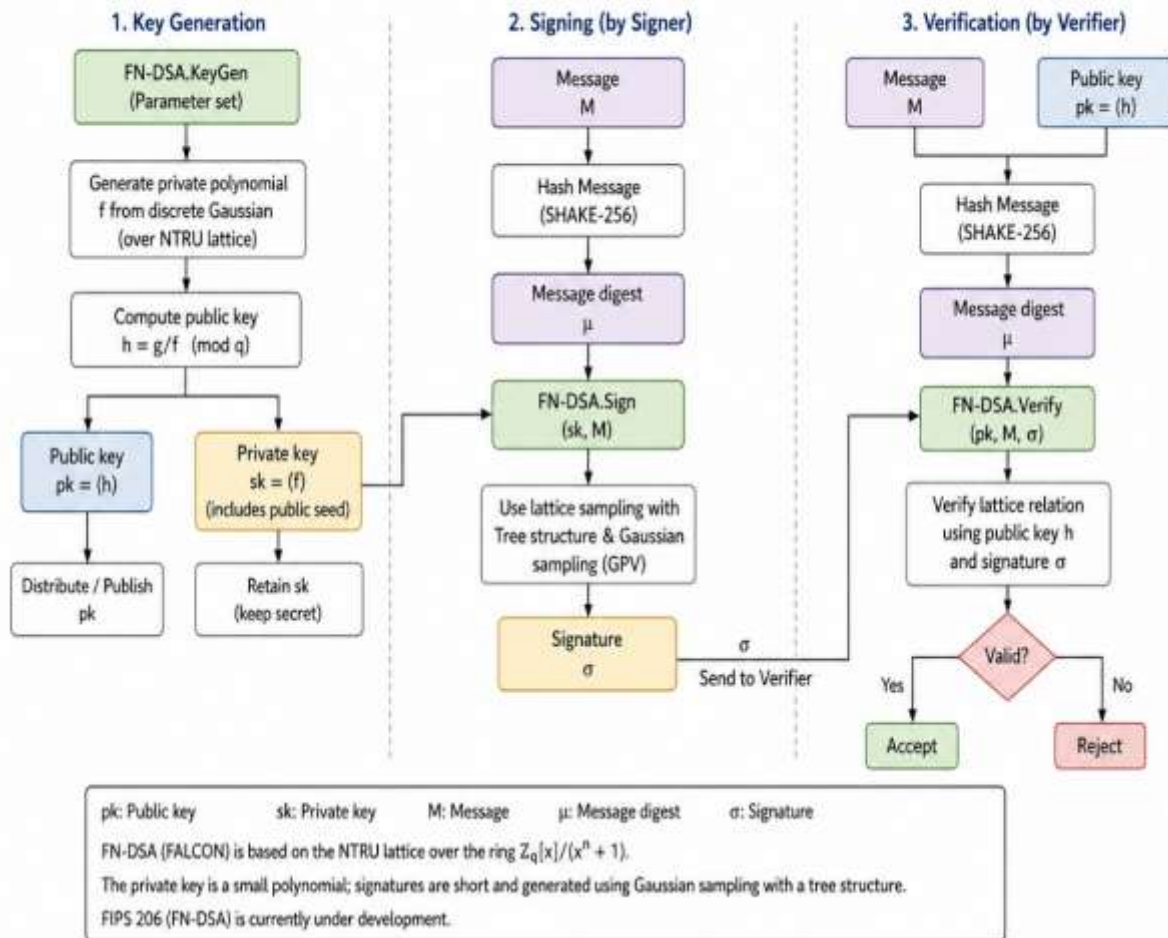


Figure 3-6. High-level FN-DSA/FALCON signing flow (FIPS 206 under development).

Source: Illustrative representation based on FIPS 206 draft/status material; standard under development.

3.10 HQC and Other Future KEM Alternatives

HQC is a code-based KEM selected by NIST for future standardisation. Its intended role is key establishment and algorithmic diversity from module-lattice KEMs. In the source documents, the final standard and application profiles are not yet available. [11, 31]

Other proprietary, experimental or pre-standard KEMs may be relevant to research and interoperability work, but they are not equivalent to a final standardised alternative for statutory or production profiles.

3.11 Composite and Hybrid Constructions

A composite signature places pre-quantum and post-quantum signature components within a defined combined structure. A hybrid-signature deployment may use separate pre-quantum and post-quantum signatures or certificates under an explicit verifier policy. The acceptance rule must state whether both components are required or whether either component is sufficient. [20-22]

Hybrid key establishment combines a pre-quantum key-establishment component with a post-quantum KEM through a standards-defined combiner. The objective is to derive a session secret that remains secure when the specified security condition for the components is met. The construction belongs to the relevant protocol profile and should not be inferred from the mere presence of two certificates. [8, 19-20]

The source documents identify the composite ML-DSA X.509 work as an Internet-Draft. Composite and hybrid constructions must therefore be distinguished from the final pure ML-DSA, SLH-DSA and ML-KEM profiles. [12-15, 18-22]

3.12 Pure Modes, Pre-Hash Modes and Hash Functions

A document format may calculate a content digest and still use a pure PQ signature mode. In CMS, for example, the application may hash the document, place the digest in signed attributes and ask the signature algorithm to sign the encoded signed-attributes value. Pure or pre-hash is an algorithm-level distinction concerning the input and processing defined by the signature standard. [5-6, 13, 15, 25]

Algorithm	Mode structure	Standards position recorded in source drafts
ML-DSA	ML-DSA and HashML-DSA	FIPS 204 defines both; RFC 9881 and RFC 9882 identify pure ML-DSA profiles.
SLH-DSA	Pure SLH-DSA and HashSLH-DSA	FIPS 205 defines both; RFC 9909 covers X.509/CRL identifiers and RFC 9814 covers pure CMS.
LMS/HSS	No separate pure/pre-hash variant	Message hashing is part of the scheme; an external pre-hash would require a separate profile.
XMSS/XMSS-MT	No separate pure/pre-hash variant	Message-digest processing is part of the scheme; a private mode must not be inferred.
FN-DSA	Subject to final FIPS 206	Final mode and profile treatment remains dependent on the completed standard.

SHA-2 is not invalidated by Shor-type quantum attacks on RSA and ECC. Quantum search changes generic security margins for hash functions, which are addressed through suitable output lengths. FIPS 204 uses SHAKE-based processing internally; FIPS 205 defines SHA-2 and SHAKE parameter families; ML-KEM uses the internal functions prescribed by FIPS 203. Application-level digests and internal algorithm hashing must remain separately identified. [4-6, 38-40]

3.13 Standards and Application-Profile Integration

Layer	Available profile or structure	Unresolved or profile-dependent element
X.509 certificates and CRLs	RFC 9881 for ML-DSA; RFC 9909 for SLH-DSA; RFC 9802 for HSS/XMSS; RFC 9935 for ML-KEM public keys	Final FN-DSA profiles and any statutory parameter restrictions remain dependent on later work.
CMS	RFC 9882 for ML-DSA; RFC 9814 for SLH-DSA; RFC 9708 for HSS/LMS; RFC 9936 for ML-KEM KEMRecipientInfo	Composite-signature profiles identified in the source documents remain works in progress.
PDF/CAAdES/PAdES	Uses CMS-based signature structures and long-term validation evidence	Libraries, size limits, timestamping and archival validation must implement the selected PQ profiles.
XML Signature	W3C XML Signature 1.1 provides an algorithm-extensible structure	Recognised PQ SignatureMethod identifiers and an interoperability profile are required; private statutory identifiers should not be assumed.

Layer	Available profile or structure	Unresolved or profile-dependent element
Cryptographic modules and devices	FIPS 140-3 and CCA device/key-lifecycle requirements provide the security boundary framework	Each algorithm requires validated key generation, key storage, signing or decapsulation and zeroisation mechanisms.
eSign and remote signing	Existing interfaces provide transaction and certificate workflows	The exact PQ signature mode, signed input, object sizes and relying-party validation must be profiled.

3.14 Summary

The following matrix closes Section 3 by recording what each alternative is. It intentionally does not assign a recommended status or India-PKI ranking.

Alternative	Function	Standard maturity in source drafts	Defining characteristic
ML-DSA	Digital signature	Final FIPS and final X.509/CRL/CMS profiles	Stateless module-lattice signature with signatures of approximately 2.4-4.6 KB.
SLH-DSA	Digital signature	Final FIPS and X.509/CRL/CMS profiles	Stateless hash-based signature with signatures of approximately 7.8-49.9 KB.
LMS/HSS	Digital signature	NIST-approved stateful profiles; CMS/X.509 profiles identified	Finite one-time-signature state managed through Merkle-tree hierarchies.
XMSS/XMSS-MT	Digital signature	NIST-approved stateful profiles; X.509 profile identified	Stateful Merkle signatures with strict rollback and capacity requirements.
FN-DSA/FALCON	Digital signature	FIPS 206 under development	Potentially compact lattice-based signature with complex implementation requirements.
ML-KEM	Key establishment	Final FIPS and X.509/CMS profiles	Encapsulation and decapsulation produce a shared secret for symmetric protection.
HQC	Key establishment	Selected for future standardisation	Future code-based diversity KEM.
Composite/hybrid	Transition construction	Profile-dependent; some specifications remain drafts	Combines pre-quantum and post-quantum components under an explicit security and validation rule.

References and Source Basis

- [4] NIST FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), final, 13 August 2024.
[5] NIST FIPS 204, Module-Lattice-Based Digital Signature Standard (ML-DSA), final, 13 August 2024.

- [6] NIST FIPS 205, Stateless Hash-Based Digital Signature Standard (SLH-DSA), final, 13 August 2024.
- [7] NIST SP 800-208, Recommendation for Stateful Hash-Based Signature Schemes (LMS/HSS and XMSS/XMSS-MT), final.
- [8] NIST SP 800-227, Recommendations for Key-Encapsulation Mechanisms, final, 18 September 2025.
- [9] NIST CSWP 39upd1, Considerations for Achieving Crypto Agility: Strategies and Practices, final, 29 June 2026.
- [10] NIST IR 8547, Transition to Post-Quantum Cryptography Standards, Initial Public Draft, 12 November 2024.
- [11] NIST Post-Quantum Cryptography Project - standards status, migration information and current work programme.
- [12] RFC 9881, Algorithm Identifiers for ML-DSA in X.509 Certificates and CRLs, October 2025.
- [13] RFC 9882, Use of ML-DSA in the Cryptographic Message Syntax (CMS), October 2025.
- [14] RFC 9909, Algorithm Identifiers for SLH-DSA in X.509 Certificates and CRLs, April 2026.
- [15] RFC 9814, Use of SLH-DSA in the Cryptographic Message Syntax (CMS), June 2025.
- [16] RFC 9708, Use of HSS/LMS in the Cryptographic Message Syntax (CMS), January 2025.
- [17] RFC 9802, Use of HSS and XMSS Hash-Based Signature Algorithms in X.509 PKI, July 2025.
- [18] RFC 9935, Algorithm Identifiers for ML-KEM in X.509 PKI, March 2026.
- [19] RFC 9936, Use of ML-KEM in the Cryptographic Message Syntax (CMS), March 2026.
- [20] RFC 9958, Post-Quantum Cryptography for Engineers, June 2026.
- [21] RFC 9955, Hybrid Signature Spectrums, June 2026.
- [22] IETF LAMPS Internet-Draft: Composite ML-DSA for X.509 PKI, draft-ietf-lamps-pq-composite-sigs-19, April 2026 (work in progress).
- [23] W3C Recommendation: XML Signature Syntax and Processing Version 1.1.
- [24] RFC 5280, Internet X.509 Public Key Infrastructure Certificate and CRL Profile.
- [25] RFC 5652, Cryptographic Message Syntax (CMS).
- [26] RFC 3161, Internet X.509 Public Key Infrastructure Time-Stamp Protocol.
- [27] Controller of Certifying Authorities, Acts, Rules, Regulations and Notifications.
- [28] Controller of Certifying Authorities, Guidelines issued by CCA, including Interoperability, CMS and XML Signature Profiles.
- [29] Controller of Certifying Authorities, eSign API Specifications.
- [30] Controller of Certifying Authorities, Long-Term Archival guidance for XAdES, CAdES and PAdES.
- [31] NIST IR 8545, Status Report on the Fourth Round of the NIST PQC Standardization Process (selection of HQC for future standardisation), March 2025.
- [32] NIST FIPS 206 status information for FN-DSA/FALCON; FIPS 206 remains under development as of 1 August 2026.

4. Comparative Analysis of the Identified Alternatives

This section compares the post-quantum alternatives described in Section 3 against the technical, operational and interoperability requirements of the CCA-regulated Public Key Infrastructure.

4.1 Scope and Method

The analysis covers RCAI, licensed Certifying Authorities, subscriber Digital Signature Certificates, eSign, remote signing, time-stamping, CRL and OCSP services, CMS/PDF/XML signatures, software and firmware signing, encrypted CMS and other key-establishment uses. Signature schemes are compared only with signature schemes, and key-establishment mechanisms are compared only with key-establishment mechanisms. ML-KEM is therefore not treated as an eSign or digital-signature algorithm. [4-6, 18-19, 27-30]. The operative comparison is centred on the final NIST post-quantum FIPS standards:

4.2 Comparative Baseline

Family	Function and principal strengths	Principal limitations	Comparative application implication
ML-DSA	Final FIPS 204 stateless digital signature; moderate post-quantum key and signature	Larger objects than RSA/ECDSA; new lattice arithmetic, rejection-sampling, side-channel, fault and	Strongest general-purpose fit for CA, subscriber, remote-signing, eSign, document-signing and

Family	Function and principal strengths	Principal limitations	Comparative application implication
	sizes; final X.509, CRL and CMS profiles.	implementation-assurance requirements.	protocol-authentication roles.
SLH-DSA	Final FIPS 205 stateless hash-based digital signature; independent non-lattice security foundation; final X.509, CRL and CMS profiles.	Signatures are materially larger and signing can be slower; high-volume transport, storage, parser and HSM effects are substantial.	Selective fit for cryptographic diversity, offline or low-frequency signing, long-lived signatures, firmware and archival roles; weaker fit for mass eSign, tokens, TSA and OCSP.
ML-KEM	Final FIPS 203 key-encapsulation mechanism; establishes a shared secret for an approved KDF and symmetric encryption or authentication.	Does not provide authentication, signer intent or a digital signature; requires a complete protocol profile including KDF, symmetric algorithm, authentication and downgrade protection.	General post-quantum key-establishment mechanism for encrypted CMS, secure channels, storage-key wrapping, device provisioning and central decapsulation services.
RSA, NIST ECC and SHA-2	Mature installed base and broad relying-party support. SHA-2 remains suitable for application-level digests when an appropriate output length is selected.	RSA and ECC remain quantum-vulnerable; increasing key size does not remove that failure mode.	Retained for migration, coexistence and verification of existing certificates, signatures and archived records; not the target post-quantum replacement.

4.2.1 Parameter and Object-Size Comparison

Mechanism	NIST category	Public key	Signature or ciphertext	Comparative use
ML-DSA-44	2	1,312 bytes	2,420-byte signature	Lowest ML-DSA size and processing burden; strongest fit for high-volume subscriber, eSign and remote-signing profiles.
ML-DSA-65	3	1,952 bytes	3,309-byte signature	Higher-assurance CA and longer-lived signature profiles.
ML-DSA-87	5	2,592 bytes	4,627-byte signature	Highest-assurance signatures where the greater object and processing cost is acceptable.
SLH-DSA-128	1	Profile-dependent	7,856 bytes (s); 17,088 bytes (f)	Cryptographic diversity with significant size cost.
SLH-DSA-192	3	Profile-dependent	16,224 bytes (s); 35,664 bytes (f)	Higher category with larger signatures and greater operational cost.

Mechanism	NIST category	Public key	Signature or ciphertext	Comparative use
SLH-DSA-256	5	Profile-dependent	29,792 bytes (s); 49,856 bytes (f)	Highest category; suited only where very large signatures are operationally acceptable.
ML-KEM-512	1	800 bytes	768-byte ciphertext	Lower-category key establishment with the smallest ML-KEM payload.
ML-KEM-768	3	1,184 bytes	1,088-byte ciphertext	Balanced general key-establishment profile.
ML-KEM-1024	5	1,568 bytes	1,568-byte ciphertext	Highest-category key establishment with the greatest payload and computation.

Note: Profile-dependent means that the applicable algorithm, parameter set, processing mode, identifier and security controls are determined by the requirements of the specific certificate, signature, protocol or application profile.

Certificate-chain growth is cumulative because both subject public keys and CA signatures increase in size. The same certificate chain may be embedded again in CMS, PDF and long-term validation evidence. Capacity assessment must therefore use complete chains, CRLs, responses and signed containers rather than isolated primitive sizes. [12-15, 24-26, 30]

4.3 Hash Functions and Signature Processing Modes

Application-level hashing and the internal processing of a post-quantum algorithm are separate layers. SHA-2, as specified in FIPS 180-4, is preferred for document digests, XML reference digests, CMS messageDigest values, timestamp message imprints and pre-hash processing. SHA-256, SHA-384 and SHA-512 are selected according to the security category, signature profile, assurance objective and retention period. A hash function or extendable-output function under FIPS 202 is used only where SHA-2 cannot be used, and the exact function and output length must be identified. This rule does not permit substitution of the internal functions fixed by FIPS 203, FIPS 204 or FIPS 205. [4-6, 38-40]

Issue	ML-DSA treatment	SLH-DSA treatment
Preferred mode	Pure ML-DSA is preferred.	Pure SLH-DSA is preferred.
Pre-hash mode	HashML-DSA is used only where pre-hashing is required.	HashSLH-DSA is used only where pre-hashing is required.
Identification	The identifier distinguishes pure ML-DSA from HashML-DSA and, for HashML-DSA, identifies the pre-hash function or XOF and output length.	The identifier distinguishes pure SLH-DSA from HashSLH-DSA and, for HashSLH-DSA, identifies the pre-hash function or XOF and output length.
Internal processing	The SHAKE processing prescribed by FIPS 204 is mandatory and is not replaced by the application digest.	The internal SHA-2 or SHAKE family is fixed by the selected FIPS 205 parameter set.
Application digest	A separately identified SHA-2 digest may be used in CMS, PDF or XML without changing pure ML-DSA into HashML-DSA.	A separately identified SHA-2 digest may be used in CMS, PDF or XML without changing pure SLH-DSA into HashSLH-DSA.

Note: XOF means Extendable-Output Function, a cryptographic function that can generate an output of the required length, for example, SHAKE256 with a 512-bit output.

4.3.1 ML-DSA Content-Digest Strength

ML-DSA parameter set	Preferred SHA-2 content digest for signed-attribute profiles	Comparative effect
ML-DSA-44	SHA-256 or stronger	Suitable for the high-volume profile while retaining the intended comparative strength.
ML-DSA-65	SHA-384 or SHA-512	SHA-256 would constrain the overall signed-content strength below the parameter-set objective.
ML-DSA-87	SHA-512	A shorter digest would materially constrain the overall signed-content strength.

Implementations shall support SHA-512 as required by RFC 9882, section 3.3. The existing parameter-specific SHA-2 mapping for CMS signed attributes is retained.

4.3.2 SLH-DSA Parameter-Family Preference

Parameter family	Treatment	Operational effect
SHA-2 sets	Preferred: SLH-DSA-SHA2-128s, -128f, -192s, -192f, -256s and -256f.	Uses the preferred SHA-2 family while retaining the selected category and s/f trade-off.
SHAKE sets	Used only where the corresponding SHA-2 set cannot be used: SLH-DSA-SHAKE-128s, -128f, -192s, -192f, -256s and -256f.	Requires explicit identification of the selected SHAKE parameter set.
s sets	Smaller-signature configuration.	Smaller signatures with slower signature generation.
f sets	Fast-signing configuration.	Faster signature generation with larger signatures.

4.4 eSign and Remote-Signing Processing

eSign and remote signing are high-volume and legally significant uses of the CCA-regulated PKI. The present eSign model is hash-oriented: the Application Service Provider retains the document and transmits its SHA-256 digest for signing. This separation of document custody from signature-key custody should be preserved, but a post-quantum profile must define the exact bytes signed, the digest algorithm, the signature mode, the container, the transaction binding and the verification procedure. A document digest, DER-encoded CMS SignedAttrs, XML SignedInfo, a PDF byte-range representation and an ML-DSA message representative are different cryptographic inputs and shall not be treated as interchangeable. [5, 13, 25, 29]

4.4.1 Recommended CMS Processing Model

For CMS-based eSign and remote signing, the preferred model is a detached CMS signature with mandatory signed attributes. The application computes a SHA-2 content digest and the digest is included in the messageDigest signed attribute. Pure ML-DSA, or pure SLH-DSA where that profile is used, signs the complete DER encoding of

SignedAttrs. The complete document need not be transmitted to the ESP or remote-signing service. Pure ML-DSA follows RFC 9882 and pure SLH-DSA follows RFC 9814. [13, 15, 25, 29]

Stage	Required processing
1. Document digest	The ASP or application computes the selected SHA-2 digest over the exact document or detached-content byte range.
2. Signed attributes	The responsible component constructs deterministic DER-encoded SignedAttrs containing at least contentType and messageDigest, together with required policy and transaction attributes.
3. Authorisation	The subscriber authenticates and authorises the same document, digest and transaction represented in the signed input.
4. Signature	Pure ML-DSA, or pure SLH-DSA where applicable, signs the complete DER encoding of SignedAttrs.
5. Response	The service returns the signature, subscriber certificate and CMS data required by the response profile.
6. Verification	The relying party recalculates the content digest, compares it with messageDigest, validates the certificate path and status, and verifies the pure signature over the same DER-encoded SignedAttrs.

The profile shall fix responsibility for constructing SignedAttrs. Either the service constructs deterministic SignedAttrs from the submitted digest and profile metadata, or the application submits the complete DER-encoded SignedAttrs for signing. Silent variation between these models is not acceptable. The returned SignedAttrs shall be checked against the submitted digest, the document displayed to the subscriber and the authorised transaction. [13, 25, 29]

4.4.2 Comparative Treatment of Pure ML-DSA, HashML-DSA and Hash-Only Input

Processing option	Signed input and identifier effect	Comparative conclusion
Pure ML-DSA over CMS SignedAttrs	Pure ML-DSA signs the complete DER-encoded SignedAttrs containing the document digest. The pure parameter-set OID and ordinary pure verifier are used.	Preferred initial eSign and remote-signing model. It preserves document locality, aligns with the final X.509/CRL/CMS profiles and avoids unnecessary pre-hash combinations.
HashML-DSA over the original content	The original content is processed using the identified pre-hash function or XOF, output length, context and FIPS-defined domain separation. A distinct HashML-DSA identifier and verification procedure are required.	Used only where pre-hashing is required, such as justified large-message, streaming or separated-module cases, under a separately defined profile.
Pure ML-DSA over the existing InputHash	The supplied SHA-256 digest is treated merely as the message byte string. The result remains a pure ML-DSA signature over that byte string.	Mathematically valid but not the initial document-signature profile because the bare digest does not independently bind the digest algorithm, document type, transaction, signer intent, policy or reconstruction rules.

Processing option	Signed input and identifier effect	Comparative conclusion
HashML-DSA over the existing InputHash	HashML-DSA would pre-hash the already calculated document digest.	Not equivalent to HashML-DSA over the original document and shall not arise through accidental reuse of the existing InputHash field.
External-mu pure ML-DSA	A 64-byte message representative is derived from the exact public key, context and message, normally the DER-encoded SignedAttrs. The ordinary pure ML-DSA OID and verifier remain unchanged.	May be used as a controlled signer-side or HSM optimisation after integrity, transaction-binding and conformance requirements are defined. It is not HashML-DSA and is not a replacement for the document digest.

Pure ML-DSA is preferred because FIPS 204 generally prefers the pure mode and the final X.509, CRL and CMS profiles use pure ML-DSA. HashML-DSA remains a valid FIPS 204 mode but introduces a separate identifier, pre-hash selection, output-length requirement, context and verifier path. A key or certificate profile shall be bound to the permitted mode; the recipient shall process explicit identifiers and shall not infer the mode from the signature length, document content or presence of a digest. [5, 12-13]

The eSign or remote-signing API shall explicitly identify the ML-DSA parameter set, signature mode, content-digest algorithm, signed-input type and CMS profile. It shall reject unknown, ambiguous or inconsistent combinations and shall not silently fall back to another algorithm, digest or mode. SHA-256 is proportionate for the ML-DSA-44 CMS profile; ML-DSA-65 and ML-DSA-87 require the stronger content-digest treatment stated in paragraph 4.3.1. [5, 13, 29]

4.5 Certificate, Container and Protocol Integration

Application layer	Required post-quantum treatment	Comparative consequence
X.509 certificates and CRLs	ML-DSA identifiers and processing under RFC 9881; SLH-DSA identifiers and processing under RFC 9909. Exact public-key, signature, parameter-set and mode identifiers are mandatory.	Path builders, trust stores, CA software and revocation clients must process larger objects and reject unknown or ambiguous identifiers.
CMS / PKCS #7	Pure ML-DSA under RFC 9882 and pure SLH-DSA under RFC 9814. SignedAttrs carry the separately identified content digest.	Provides the standard basis for eSign, remote signing, CAdES and detached signatures without transferring the complete document.
PDF, CAdES and PAdES	Use the corresponding CMS processing together with certificates, timestamps and revocation evidence.	The signature, chain and long-term validation evidence enlarge the final document; independent validator testing(interoperability) is required.
XML Signature	SignatureMethod and KeyInfo identify the exact ML-DSA or SLH-DSA algorithm, parameter set and mode. SHA-2 is preferred for Reference digests. Private or provisional identifiers are not used.	Canonicalisation, transforms, reference processing and algorithm identifiers must be tested together.

Application layer	Required post-quantum treatment	Comparative consequence
TSA and OCSP	The certificate and response signature use the exact post-quantum identifiers; SHA-2 is preferred for timestamp message imprints.	High-frequency signing, response size, caching and broad verifier dependency make throughput and client testing critical.
ML-KEM in X.509 and CMS	RFC 9935 provides X.509 PKI identifiers and RFC 9936 provides CMS KEMRecipientInfo processing. The shared secret is passed through an approved KDF to symmetric protection.	Applies only to key establishment; it does not sign certificates, documents, eSign transactions, CRLs or software.
Long-term validation	Preserve exact identifiers, certificates, revocation evidence, timestamps, validation policy and software capable of processing historical algorithms.	Pre-quantum and post-quantum records coexist for long periods and require continuing validation support and evidence renewal.

The relying party must process explicit OIDs, parameters, signed attributes, digest identifiers, key usage and certificate policy. It must not infer pure or pre-hash mode from the signature length, the existence of a document digest or the content being signed. Unknown, mismatched, private or unprofiled identifiers fail closed, and silent fallback to RSA, ECDSA, a weaker digest or another processing mode is not permitted. [5-6, 12-15, 24-25]

4.6 Key Generation, Hardware Custody and Performance

Algorithm standardisation alone does not establish production readiness. The key pair, seed, signing or decapsulation operation, entropy source, DRBG, self-tests, role separation, backup, restore and zeroisation must be assessed within the exact cryptographic-module and service configuration. A product statement that an HSM or token supports an algorithm is not sufficient evidence. [4-6, 33-37, 43]

Control area	Minimum comparative requirement
Native secure-boundary operations	Native key generation, non-exportable private-key or protected-seed storage, signing or decapsulation, self-tests, role separation, zeroisation and the exact FIPS parameter set inside the protected boundary.
Entropy and randomness	Approved entropy source and DRBG, health tests, unique key generation under peak load, safe failure handling and hedged/randomised signing where required by the implementation profile.
Implementation assurance	Constant-time or otherwise protected arithmetic, rejection-sampling controls, side-channel and fault-testing, secure firmware and supply-chain evidence.
Backup and disaster recovery	Protected backup and restore only where permitted by the applicable key-custody policy and cryptographic-module security policy, with key consistency, auditability and recovery testing. Stateful rollback controls are not required for ML-DSA or SLH-DSA because both are stateless, but ordinary key-custody controls remain mandatory.
Token and middleware	Storage, RAM, APDU or command-size limits, output transfer, certificate storage, driver support and independent application interoperability. Host-side emulation does not provide token-protected signing.
HSM and service capacity	KeyGen, first and sustained signing or decapsulation, p50/p95/p99 latency, operations per second, session limits, concurrency, memory, entropy performance and failover.

Control area	Minimum comparative requirement
Object and parser safety	Revised size limits for certificates, signatures, CRLs, OCSP responses, APIs, databases and archives; bounded parsing, negative tests, quotas and denial-of-service controls.

ML-DSA is stateless and has the lowest final post-quantum signature sizes among the compared signature families, making it the strongest relative fit for high-volume services. SLH-DSA is also stateless but transfers substantial cost to signing time, signature size, transport, storage and verification. ML-KEM operations are stateless per transaction, but the decapsulation private key remains a high-value protected key. [4-6, 20, 33]

4.7 India-PKI Application Comparison

Use case	Relative algorithm fit	Essential evidence before use
RCAI offline root and highest-assurance signing	ML-DSA-65 or ML-DSA-87 provides the principal lattice-based fit; SLH-DSA provides selective non-lattice diversity where its size and signing cost are acceptable.	Test root ceremony, native HSM KeyGen, permitted protected backup/restore, CRL generation, full path validation and long-term verification.
Licensed issuing CA and organisational signing	ML-DSA-65 has the strongest balance of assurance, clustering, issuance volume and object size. ML-DSA-87 is reserved for the highest-assurance profiles.	CA issuance, CRL/OCSP, repository, HSM cluster, disaster recovery and independent relying-party benchmarks.
Subscriber DSC token	ML-DSA-44 has the strongest relative fit because it minimises post-quantum key, signature and transfer burden. SLH-DSA is constrained by signature size and hash workload.	Next-generation token profile, native KeyGen, non-exportability, PIN and key policy, APDU limits, middleware and side-channel/fault testing.
Online eSign and remote signing	ML-DSA-44 with pure CMS SignedAttrs has the strongest high-volume fit. SLH-DSA is not the default because of larger signatures and processing cost.	Versioned API, SHA-2 content digest, deterministic SignedAttrs, transaction binding, HSM isolation, throughput and complete ASP/ESP/relying-party interoperability.
TSA and OCSP	ML-DSA is comparatively more practical after throughput testing. SLH-DSA has a narrower low-frequency role.	Response size, HSM throughput, network and cache impact, client verification and archival evidence.
CMS/PDF/CAdES/PAdES	Pure ML-DSA is the primary fit; pure SLH-DSA provides selective diversity where object growth is acceptable.	Independent generation and validation, signed-attribute processing, timestamps, revocation evidence and long-term validation.
XML Signature	ML-DSA or SLH-DSA after exact recognised SignatureMethod and KeyInfo profiles are implemented.	Canonicalisation, transforms, reference digests, certificate processing and verifier interoperability.
Software, firmware and secure boot	ML-DSA provides balanced general use; SLH-DSA provides non-lattice diversity	Embedded verifier footprint, update-package size, device lifetime, offline verification and key-lifecycle tests.

Use case	Relative algorithm fit	Essential evidence before use
	where package size and signing time are acceptable.	
Encrypted CMS and data exchange	ML-KEM-768 is the balanced general key-establishment profile; ML-KEM-512 and ML-KEM-1024 correspond to lower and higher category profiles.	Complete KEMRecipientInfo or protocol profile, authentication, KDF, symmetric algorithm, key confirmation, failure handling and downgrade protection.

4.8 Transition and Coexistence

The clearest initial migration architecture is a separate CCA-controlled test post-quantum hierarchy containing a root, issuing CA, subscriber certificate, CRL/OCSP, TSA, eSign-style short-lived certificate, repository and representative relying applications. This isolates testing from the production RCAI hierarchy and permits clean rollback. [9-10, 24, 27-30]

Pure post-quantum certificates and signatures provide the baseline for testing. Separate pre-quantum and post-quantum certificates support controlled coexistence where applications select the correct certificate through explicit policy. Draft composite signatures, private OIDs and undocumented hybrid constructions are not part of the operative profile. A post-quantum end-entity certificate inside an otherwise pre-quantum validation path does not by itself provide end-to-end post-quantum assurance. [9-10, 20-22, 24]

RSA and NIST curves P-256, P-384 and P-521 remain supported during migration and for verification of existing certificates, signatures and archived records. Historical trust anchors, policies, CRLs, OCSP evidence, timestamps and validation software must remain available for the applicable retention period. SHA-2 remains the preferred application-level hash family throughout migration. [9-10, 27-30, 38, 40]

The transition may use separate pre-quantum and post-quantum certificates, dual signatures or approved hybrid key-establishment mechanisms, depending on the application. Where assurance requiring successful validation of both pre-quantum and post-quantum signatures is claimed, both signatures and their certificate paths must validate. Composite certificates and signatures require standardisation, CCA-approved profiles and interoperability testing before production use. The duration of coexistence and retirement of pre-quantum-only signing shall depend on the phased programme and each application's demonstrated readiness, rather than a universal cut-off date. Permitted transition mechanisms and acceptance rules are addressed in Part II, section 8.3, and Part III, sections 3.1 and 4.4. Their duration shall follow Part III, section 3.7, and use-case-specific readiness choices.

4.9 Minimum End-to-End Evidence

Test domain	Minimum evidence
Algorithm correctness	Known-answer and negative tests, strict public-key/signature/ciphertext length checks, parameter and OID validation, and cross-mode rejection.
Key generation and hardware boundary	Entropy and DRBG evidence, native KeyGen, private-key non-exportability, signing or decapsulation, self-tests, zeroisation, backup and restore.
Performance and capacity	KeyGen, first and sustained operation, verification, p50/p95/p99 latency, concurrency, memory, HSM sessions, response size and failure behaviour.
Certificates and status services	CSR, certificate issuance, complete chains, path building, CRL, OCSP, TSA, repository growth and independent client validation.

Test domain	Minimum evidence
eSign and remote signing	Document display, authentication, consent, transaction binding, SHA-2 digest, SignedAttrs construction, pure/pre-hash separation, API payload, HSM isolation and relying-party verification.
CMS/PDF/XML	Independent generation and validation, DER and canonicalisation checks, signed attributes, exact identifiers, large-object handling, timestamps and negative tests.
Tokens and middleware	Storage, RAM, command limits, transfer, driver support, certificate storage, application integration and secure failure handling.
Security and resilience	Side-channel and fault testing, parser fuzzing, resource-exhaustion limits, firmware integrity, supply-chain evidence, failover and disaster recovery.
Archival validation	Preservation of pre-quantum and post-quantum algorithms, certificates, status evidence, timestamps, policies, validation software and evidence-renewal procedures.

References and Source Basis

The reference numbering follows the source materials used for this comparative analysis.

- [4] NIST FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), final, 13 August 2024.
- [5] NIST FIPS 204, Module-Lattice-Based Digital Signature Standard (ML-DSA), final, 13 August 2024.
- [6] NIST FIPS 205, Stateless Hash-Based Digital Signature Standard (SLH-DSA), final, 13 August 2024.
- [7] NIST SP 800-208, Recommendation for Stateful Hash-Based Signature Schemes.
- [8] NIST SP 800-227, Recommendations for Key-Encapsulation Mechanisms, final, 18 September 2025.
- [9] NIST CSWP 39upd1, Considerations for Achieving Crypto Agility: Strategies and Practices, final, 29 June 2026.
- [10] NIST IR 8547, Transition to Post-Quantum Cryptography Standards, Initial Public Draft, 12 November 2024.
- [12] RFC 9881, Algorithm Identifiers for ML-DSA in X.509 Certificates and CRLs, October 2025.
- [13] RFC 9882, Use of ML-DSA in the Cryptographic Message Syntax (CMS), October 2025.
- [14] RFC 9909, Algorithm Identifiers for SLH-DSA in X.509 Certificates and CRLs, April 2026.
- [15] RFC 9814, Use of SLH-DSA in the Cryptographic Message Syntax (CMS), June 2025.
- [18] RFC 9935, Algorithm Identifiers for ML-KEM in X.509 PKI, March 2026.
- [19] RFC 9936, Use of ML-KEM in the Cryptographic Message Syntax (CMS), March 2026.
- [20] RFC 9958, Post-Quantum Cryptography for Engineers, June 2026.
- [21] RFC 9955, Hybrid Signature Spectrums, June 2026.
- [22] IETF LAMPS Internet-Draft: Composite ML-DSA for X.509 PKI, draft-ietf-lamps-pq-composite-sigs-19, April 2026 (work in progress).
- [24] RFC 5280, Internet X.509 Public Key Infrastructure Certificate and CRL Profile.
- [25] RFC 5652, Cryptographic Message Syntax (CMS).
- [26] RFC 3161, Internet X.509 Public Key Infrastructure Time-Stamp Protocol.
- [27] Controller of Certifying Authorities, Acts, Rules, Regulations and Notifications.
- [28] Controller of Certifying Authorities, Guidelines, including Interoperability, CMS and XML Signature Profiles.
- [29] Controller of Certifying Authorities, eSign API Specifications.
- [30] Controller of Certifying Authorities, Long-Term Archival Guidance for XAdES, CAdES and PAdES.
- [31] NIST IR 8545, Status Report on the Fourth Round of the NIST PQC Standardization Process, March 2025.
- [32] NIST FIPS 206 status information for FN-DSA/FALCON; FIPS 206 remained under development in the source material.
- [33] NIST FIPS 140-3, Security Requirements for Cryptographic Modules.
- [34] NIST SP 800-90A Rev. 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators.
- [35] NIST SP 800-90B, Recommendation for the Entropy Sources Used for Random Bit Generation.
- [36] NIST SP 800-90C, Recommendation for Random Bit Generator Constructions.
- [37] NIST SP 800-133 Rev. 2, Recommendation for Cryptographic Key Generation.
- [38] NIST FIPS 180-4, Secure Hash Standard.
- [39] NIST FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions.
- [40] NIST SP 800-57 Part 1 Rev. 5, Recommendation for Key Management - General.
- [41] NIST IR 8413, Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process.
- [42] RFC 6960, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP.

5. Recommendations

These recommendations convert the findings of Sections 1 to 4 into a limited technical position on algorithm families, their permitted role, and the conditions for migration and production adoption.

The transition may use separate pre-quantum and post-quantum certificates, dual signatures or approved hybrid key-establishment mechanisms according to application needs. Both signatures and their certificate paths must validate where dual assurance is claimed, while composite constructions require standardisation, CCA-approved profiles and interoperability testing. Coexistence shall continue according to the phased programme and demonstrated readiness. References: Part II, section 8.3; Part III, sections 3.1 and 4.4; and Part III, section 3.7 for duration and readiness-based transition.

Transition mechanisms should be selected according to their purpose. Hybrid key establishment combines pre-quantum and post-quantum methods to protect confidentiality, while dual or composite signatures address authenticity and integrity. Separate pre-quantum and post-quantum certificates support coexistence and independent credential management during migration. References: Part II, section 8.3, and Part III, section 4.4.

5.1 Recommended Algorithm Portfolio

The technical classification below distinguishes (i) operative post-quantum algorithms, (ii) specialised stateful hash-based signature schemes, (iii) pre-quantum algorithms retained for transition, coexistence and legacy validation, and (iv) algorithms for future consideration. The operative post-quantum portfolio consists of ML-DSA and SLH-DSA for digital signatures and ML-KEM for key establishment. XMSS/XMSS-MT and LMS/HSS may be considered only for tightly controlled applications where signing state can be securely maintained; they are not recommended for general-purpose CA/eSign use. FN-DSA and HQC-KEM remain future considerations and shall not form part of the operative CCA algorithm profile until the relevant final NIST standards and the supporting PKI implementation ecosystem are available. SHA-2 remains the preferred family for external document digests, reference digests and pre-hash processing. A hash function or extendable-output function under FIPS 202 shall be used only where SHA-2 cannot be used and the exact function and output length are specified. This preference does not alter the internal SHAKE processing mandated by FIPS 204 or the hash family fixed by the selected FIPS 205 parameter set.

Alternative	Recommended status	Minimum technical position
ML-DSA	Primary general-purpose post-quantum digital-signature family.	1. The Committee recommends ML-DSA-44, as specified in FIPS 204, for high-volume subscriber, remote-signing and eSign profiles. 2. The Committee recommends ML-DSA-65, as specified in FIPS 204, for CA, organisational, longer-lived and higher-assurance profiles. 3. The Committee recommends ML-DSA-87, as specified in FIPS 204, for the highest-assurance profiles. 4. Pure ML-DSA is preferred. 5. HashML-DSA shall be used only where pre-hashing is required. 6. SHA-2 is preferred for the pre-hash. 7. A FIPS 202 hash function or extendable-output function shall be used only where SHA-2 cannot be used. 8. The internal SHAKE processing prescribed by FIPS 204 is mandatory and shall not be replaced.

Alternative	Recommended status	Minimum technical position
SLH-DSA	Secondary stateless hash-based digital-signature family for cryptographic diversity.	9. The FIPS 205 SHA-2 parameter sets are preferred. 10. The preferred SHA-2 parameter sets are SLH-DSA-SHA2-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-256s and SLH-DSA-SHA2-256f. 11. The corresponding SHAKE parameter sets shall be used only where SHA-2 cannot be used. 12. The corresponding SHAKE parameter sets are SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-256s and SLH-DSA-SHAKE-256f. 13. Pure SLH-DSA is preferred. 14. HashSLH-DSA shall be used only where pre-hashing is required. 15. The s parameter sets provide smaller signatures with slower signing. 16. The f parameter sets provide faster signing with larger signatures.
ML-KEM	Post-quantum key-establishment family; not a digital-signature or eSign algorithm.	17. The Committee recommends ML-KEM-768, as specified in FIPS 203, for the general key-establishment profile. 18. ML-KEM-512 and ML-KEM-1024 shall be used for the corresponding NIST security categories and application profiles. 19. ML-KEM establishes a shared secret. 20. The shared secret shall be processed through an approved key-derivation function. 21. The derived key material shall be used with approved symmetric encryption or authentication mechanisms.
XMSS / XMSS-MT and LMS/HSS	Specialised stateful hash-based signature schemes for tightly controlled applications; not recommended for general-purpose CA/eSign use.	22. These schemes may be considered for specialised controlled applications such as firmware/code signing or other low-frequency signing environments. 23. Their use shall conform to NIST SP 800-208 and the applicable standardized profiles. 24. Secure maintenance of signing state is mandatory. 25. Controls shall prevent reuse of state/one-time signing keys. 26. Backup, recovery and disaster-recovery procedures shall preserve correct signing state. 27. These algorithms shall not be used for general-purpose subscriber/eSign signing where reliable state management cannot be assured. 28. Use for Root CA or other critical signing functions shall be subject to specific evaluation of state management, HSM support, recovery and operational controls before adoption.
RSA/ECC and SHA-2	Transition, coexistence and legacy validation.	29. RSA and NIST curves P-256, P-384 and P-521 shall be retained during the migration period. 30. These algorithms shall continue to be supported for verification of existing certificates, signatures and archived records. 31. SHA-2 shall be retained during migration and for verification of existing certificates, signatures and archived records. 32. SHA-2 shall remain the preferred application-level hash family.

FUTURE CONSIDERATIONS

Alternative	Recommended status	Minimum technical position
FN-DSA (Falcon) - FN-DSA-512 / FN-DSA-1024	Selected by NIST for standardization; FIPS 206 under development. Future consideration.	1. FN-DSA is being standardized by NIST as FIPS 206. 2. FN-DSA-512 and FN-DSA-1024 provide comparatively compact public keys/signatures and may be relevant to applications where bandwidth, storage or transmission overhead is important. 3. They shall not presently form part of the operative CCA algorithm profile because FIPS 206 is not final. 4. The exact parameter sets and processing requirements shall be taken from the final FIPS 206 rather than preliminary specifications. 5. Inclusion shall be reconsidered after publication of final FIPS 206 and availability of applicable X.509/CMS identifiers, validated implementations, HSM/token support and end-to-end interoperability. 6. IETF documents relating to constrained-device implementation may be considered as technical guidance but shall not substitute for the final normative standard.
HQC-KEM	Selected by NIST for standardization as an additional post-quantum KEM; future consideration.	7. HQC has been selected by NIST for standardization as an additional KEM to complement ML-KEM and provide cryptographic diversity. 8. HQC is based on a different cryptographic family from ML-KEM and may therefore provide diversity in future key-establishment profiles. 9. The corresponding NIST FIPS standard is not yet final. 10. HQC-KEM shall therefore not presently form part of the operative CCA algorithm profile. 11. It shall remain under review for future inclusion after publication of the final NIST standard. 12. Inclusion shall also depend on availability of approved protocol/PKI identifiers, validated cryptographic implementations, HSM support, performance evaluation and interoperability testing.

5.2 Conditions before Production Adoption

1. For general-purpose operative post-quantum profiles, use only the final NIST standards FIPS 203, FIPS 204 and FIPS 205. Each signature profile shall explicitly distinguish pure ML-DSA from HashML-DSA and pure SLH-DSA from HashSLH-DSA and shall bind the exact parameter set, context, key usage, encoding, pre-hash function and verification rule. Unknown, private, provisional or unprofiled identifiers shall fail closed.
2. Require native key generation, private-key custody and signing or decapsulation within the approved cryptographic boundary where hardware protection is prescribed. Product claims alone shall not be accepted as evidence of secure implementation.
3. Require recognised algorithm and module testing, together with independent evidence for entropy and DRBG operation, side-channel and fault resistance, parser safety, secure firmware, backup, restore, zeroisation and supply-chain controls.
4. Complete end-to-end interoperability and performance testing across certificate issuance and validation, CRL, OCSP, TSA, CMS, PDF, XML, eSign, remote signing, tokens, HSM clusters, repositories, archives and relying applications.
5. Require crypto-agility, authenticated algorithm selection, downgrade resistance, explicit coexistence rules and preservation of the exact signed input. Undefined bare-hash migration, mode inference and silent

fallback shall be prohibited. SHA-2 shall be used for application-level document and reference digests and for pre-hashing. A FIPS 202 hash function or XOF shall be used only where SHA-2 cannot be used. Internal functions mandated by FIPS 203, FIPS 204 and FIPS 205 shall not be substituted.

6. XMSS/XMSS-MT and LMS/HSS shall be restricted to specialised controlled applications specified in the applicable profiles. Key generation and signature generation shall conform to NIST SP 800-208, section 8.1, and shall be performed within validated hardware cryptographic modules providing Level 3 or higher physical security, with a non-modifiable or limited operational environment. The modules shall operate only in approved mode, shall not implement a bypass mode, and shall not permit export of private keying material. The entropy source shall be located within the module's physical boundary. On accepting a signing request, the module shall increment the signing-key leaf index and store the incremented value in non-volatile storage before releasing the signature or accepting another signing request. One-time signing keys shall not be reused. Recovery and disaster-recovery arrangements shall comply with these requirements and shall not permit private-key export, cloning or rollback of signing state. Root CA and other critical uses shall require prior evaluation of the complete implementation and recovery arrangements. These schemes shall not be used for general-purpose subscriber/eSign signing where reliable state management cannot be assured.
7. FN-DSA and HQC-KEM shall remain outside production profiles until the corresponding final NIST standards are published. Any future adoption shall additionally require approved X.509, CMS or protocol identifiers as applicable, validated implementations, HSM/token support, performance evaluation and end-to-end interoperability testing. Private or provisional identifiers shall not be used in production.

5.3 Migration Approach

1. Establish a separate CCA-controlled laboratory post-quantum hierarchy before altering the production RCAI hierarchy. The test shall include a root, issuing CA, subscriber certificate, CRL/OCSP, TSA, eSign-style short-lived certificate, repository and representative relying applications.
2. Use pure post-quantum certificates and signatures based on ML-DSA or SLH-DSA for the general-purpose operative profile. Pure ML-DSA and pure SLH-DSA are the preferred modes. HashML-DSA or HashSLH-DSA shall be used only where pre-hash processing is required. For eSign and remote signing through CMS signed attributes, the document digest shall be generated using SHA-2 and the pure signature algorithm shall sign the complete DER-encoded SignedAttrs. Separate pre-quantum and post-quantum certificates shall support controlled coexistence during migration.
3. Evaluate XMSS/XMSS-MT and LMS/HSS only in a separate controlled test track for specialised low-frequency signing use cases. Testing shall cover state exhaustion, concurrent use, prevention of state reuse, HSM support and failover and recovery arrangements compliant with NIST SP 800-208, without private-key export, cloning or signing-state rollback, before any production decision. These schemes shall not be introduced into general-purpose subscriber/eSign profiles.
4. Keep FN-DSA and HQC-KEM in non-production technology-watch or prototype status until their final NIST standards are available. They shall not be added to the operative CCA algorithm schedule or production profiles before final standards, approved identifiers, validated implementations and interoperability support are available.

5. Prepare coordinated revisions to the CCA algorithm schedule, X.509/CRL, CMS/PDF/XML, OCSP/TSA, eSign and remote-signing, crypto-device, audit and functional-testing profiles. Any specialised stateful-signature use shall be separately profiled with mandatory state-management controls.
6. Move to controlled production coexistence only after functional, security, interoperability, performance, hardware, disaster-recovery and legal-readiness gates are satisfied.
7. Preserve verification of existing RSA/ECC certificates and signed records. Historical trust anchors, certificates, revocation evidence, time stamps, policies and validation software shall be retained, with evidence renewal for long-retention records where required.

5.4 Algorithm Recommendation

1. The operative profile shall use:
 - FIPS 204 ML-DSA as the primary digital-signature family;
 - FIPS 205 SLH-DSA for stateless hash-based cryptographic diversity; and
 - FIPS 203 ML-KEM only for key establishment.
2. XMSS/XMSS-MT and LMS/HSS under NIST SP 800-208 shall be restricted to specialised, tightly controlled applications and shall not be used as general-purpose CA/eSign algorithms.
3. Use of XMSS/XMSS-MT or LMS/HSS for Root CA or other critical signing functions shall require specific evaluation of state management, HSM support, backup, recovery and operational controls.
4. FN-DSA and HQC-KEM shall remain future considerations and shall not form part of the operative CCA profile until the corresponding final NIST standards, required identifiers, validated implementations, hardware support and end-to-end interoperability are available.
5. Pure signature modes and SHA-2-based external hashing shall be preferred.
6. Pre-hash modes and FIPS 202 hashing shall be used only where the preferred processing cannot be used and shall be explicitly identified.
7. Implementation shall proceed through defined profiles, validated hardware and software, a separate laboratory hierarchy, end-to-end testing, controlled coexistence and continued validation of legacy records.
8. The 5.5, accordingly propose considerations in amendments for the operative standards and, under Rule 6, provide a restricted provision for specialised stateful hash-based signatures.
9. FN-DSA and HQC-KEM shall not be proposed for inclusion in the standards at this stage.
10. The modifications required in the standards prescribed under the IT Act are shown in the Annexure: red for addition, strikeout for deletion and black for no change.

5.6 Considerations for - Information Technology (Certifying Authorities) Rules, 2000 - Rule 6 “Standards”

The suggested modifications, and their appropriate placement in the Rules under the IT Act, the operative provisions of the Interoperability Guidelines, or other applicable CCA instruments, are consolidated in the table below. The requirements may accordingly be distributed across the appropriate regulatory and technical layers for effective implementation and enforcement by CCA.

6. Standards.-The Information Technology (IT) Rules for Certifying Authorities may support open standards and accepted de facto standards; the most important standards that may be considered for different activities associated with the Certifying Authority’s functions are as under-

THE PRODUCT	THE STANDARD
Public Key Infrastructure	PKIX
Digital Signature Certificates and Digital Signature revocation list	X.509 version 3 certificates and certificate revocation lists conforming to ITU-T X.509, RFC 5280 and the applicable CCA interoperability profiles.

THE PRODUCT	THE STANDARD
	<i>ML-DSA certificate and CRL identifiers: RFC 9881. SLH-DSA certificate and CRL identifiers: RFC 9909.</i>
<i>Directory (DAP and LDAP)</i>	<i>X500 for publication of certificates and Certification Revocation Lists (CRLs)</i>
<i>Database Management Operations</i>	<i>Use of generic SQL</i>
<i>Public Key algorithm</i>	<i>DSA, RSA and Curves NIST P-256, P-384 or P-521 ML-DSA-44, ML-DSA-65 and ML-DSA-87 under FIPS 204. SLH-DSA under FIPS 205.</i>
<i>Stateful hash-based signature algorithms</i>	<i>1. XMSS/XMSS-MT and LMS/HSS may be considered only for specialised controlled signing applications in accordance with NIST SP 800-208 and the applicable standardized profiles.</i> <i>2. They shall not be used for general-purpose subscriber/eSign signing where reliable signing-state management cannot be assured.</i> <i>3. Secure maintenance of signing state is mandatory, and controls shall prevent reuse of state or one-time signing keys.</i> <i>4. Backup, recovery and disaster-recovery procedures shall preserve the correct signing state.</i> <i>5. Use for Root CA or other critical signing functions shall require specific evaluation of state management, HSM support, recovery and operational controls before adoption.</i>
<i>ML-DSA processing modes</i>	<i>6. Pure ML-DSA is preferred.</i> <i>7. HashML-DSA shall be used only where pre-hashing is required.</i> <i>8. The signature identifier shall distinguish between pure ML-DSA and HashML-DSA.</i> <i>9. For HashML-DSA, the signature identifier shall identify the pre-hash function or extendable-output function.</i> <i>10. For HashML-DSA, the signature identifier shall also identify the output length of the pre-hash function or extendable-output function.</i> <i>11. SHA-2 is preferred for pre-hashing.</i> <i>12. A FIPS 202 hash function or extendable-output function shall be used only where SHA-2 cannot be used.</i> <i>13. The internal SHAKE processing prescribed by FIPS 204 is mandatory and shall not be replaced.</i>
<i>SLH-DSA SHA2 parameter sets</i>	<i>14. The SLH-DSA SHA-2 parameter sets under FIPS 205 shall be preferred.</i> <i>15. The preferred SHA-2 parameter sets are SLH-DSA-SHA2-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-256s and SLH-DSA-SHA2-256f.</i>
<i>SLH-DSA SHAKE parameter sets</i>	<i>16. The following SLH-DSA SHAKE parameter sets shall be used only where the corresponding SHA-2 parameter sets cannot be used: SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-256s and SLH-DSA-SHAKE-256f.</i>

THE PRODUCT	THE STANDARD
SLH-DSA processing modes	17. Pure SLH-DSA is preferred. 18. HashSLH-DSA shall be used only where pre-hashing is required. 19. The signature identifier shall distinguish between pure SLH-DSA and HashSLH-DSA. 20. For HashSLH-DSA, the signature identifier shall identify the pre-hash function or extendable-output function. 21. For HashSLH-DSA, the signature identifier shall also identify the output length of the pre-hash function or extendable-output function. 22. SHA-2 is preferred for pre-hashing. 23. A FIPS 202 hash function or extendable-output function shall be used only where SHA-2 cannot be used.
Post-quantum key-establishment algorithm	24. FIPS 203 specifies ML-KEM-512, ML-KEM-768 and ML-KEM-1024. 25. RFC 9935 specifies the X.509 PKI algorithm identifiers for ML-KEM. 26. RFC 9936 specifies the use and processing of ML-KEM in CMS. 27. ML-KEM shall be used only for key establishment. 28. ML-KEM shall not be used for digital signatures or eSign.
Digital Hash Function	29. SHA-2, as specified in FIPS 180-4, is preferred. 30. SHA-256, SHA-384 and SHA-512 shall be selected according to the applicable security category. 31. The selection shall also consider the signature profile, assurance objective and retention period. 32. A hash function or extendable-output function under FIPS 202 shall be used only where SHA-2 cannot be used. 33. Where a FIPS 202 hash function or extendable-output function is used, the exact function and output length shall be explicitly identified. 34. This provision shall not permit substitution or modification of the internal functions prescribed by FIPS 203, FIPS 204 or FIPS 205.
RSA Public Key Technology	PKCS#1 RSA Encryption Standard (2048, 4096 bit). Retained for transition, coexistence and legacy validation.
	PKCS#5 Password Based Encryption Standard
	35. PKCS#7 Cryptographic Message Syntax standard. 36. Pure ML-DSA CMS processing: RFC 9882. 37. Pure SLH-DSA CMS processing: RFC 9814. 38. For eSign and remote signing, SHA-2 shall be used for the content digest in messageDigest and the pure signature algorithm shall sign the complete DER-encoded SignedAttrs.
	PKCS#8 Private Key Information Syntax standard
	PKCS#9 Selected Attribute Types
	PKCS#10 Certification Request. Certification requests shall carry the ML-DSA or SLH-DSA public key and proof of possession using the same signature family and parameter set.

THE PRODUCT	THE STANDARD
	<i>PKCS#12 Portable format for storing/transporting a user's private certificates</i>
<i>Distinguished name</i>	<i>X.520</i>
<i>Digital Encryption and Digital Signature</i>	39. <i>PKCS#7 and CMS RFC 5652.</i> 40. <i>Pure ML-DSA signatures use RFC 9882; pure SLH-DSA signatures use RFC 9814; ML-KEM key establishment uses RFC 9936.</i>
<i>Digital Signature Request Format</i>	41. <i>PKCS#10.</i> 42. <i>The request shall identify ML-DSA or SLH-DSA, the exact parameter set and the proof-of-possession signature.</i>

B. Digital Signature (End Entity) Rules, 2015 - Rule 7 “Digital signature standards”

7. Digital signature standards. - The most important standards that shall be applicable for different activities associated with digital signature functions are as under-

Products	Standards
<i>Cryptographic hash function</i>	1. <i>SHA-2, as specified in FIPS 180-4, is preferred.</i> 2. <i>SHA-256, SHA-384 and SHA-512 shall be selected according to the applicable security category.</i> 3. <i>The selection shall also consider the signature profile, assurance objective and retention period.</i> 4. <i>A FIPS 202 hash function or extendable-output function shall be used only where SHA-2 cannot be used.</i> 5. <i>Where a FIPS 202 hash function or extendable-output function is used, the algorithm and its output length shall be explicitly identified.</i>
<i>RSA Public Key Technology</i>	<i>PKCS#1 RSA Encryption Standard (2048, 4096 bit); Version 1.5. Retained for transition, coexistence and validation of legacy certificates and signed records.</i>
<i>ML-DSA digital signature</i>	6. <i>The Committee recommends ML-DSA-44, as specified in FIPS 204, for high-volume subscriber, remote-signing and eSign profiles.</i> 7. <i>The Committee recommends ML-DSA-65, as specified in FIPS 204, for CA, organisational, longer-lived and higher-assurance signature profiles.</i> 8. <i>The Committee recommends ML-DSA-87, as specified in FIPS 204, for the highest-assurance signature profiles.</i> 9. <i>Pure ML-DSA is preferred.</i> 10. <i>HashML-DSA shall be used only where pre-hashing is required.</i> 11. <i>SHA-2 is preferred for the pre-hash.</i> 12. <i>A FIPS 202 hash function or extendable-output function shall be used only where SHA-2 cannot be used.</i>

Products	Standards
	13. The internal SHAKE processing prescribed by FIPS 204 is mandatory and shall not be changed or replaced.
SLH-DSA digital signature	14. The FIPS 205 SHA-2 parameter sets are preferred. 15. The preferred SHA-2 parameter sets are SLH-DSA-SHA2-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-256s and SLH-DSA-SHA2-256f. 16. The corresponding SHAKE parameter sets shall be used only where the corresponding SHA-2 parameter sets cannot be used. 17. The applicable SHAKE parameter sets are SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-256s and SLH-DSA-SHAKE-256f. 18. Pure SLH-DSA is preferred. 19. HashSLH-DSA shall be used only where pre-hashing is required. 20. The s parameter sets provide smaller signatures with slower signature generation. 21. The f parameter sets provide faster signature generation with larger signatures.
Encryption and digital signature	22. PKCS #7 and CMS shall be used for cryptographic message syntax and digital-signature containers. 23. Pure ML-DSA signatures in CMS shall be implemented in accordance with RFC 9882. 24. Pure SLH-DSA signatures in CMS shall be implemented in accordance with RFC 9814. 25. For eSign and remote-signing transactions, SHA-2 shall be used to compute the content digest. 26. The content digest shall be included in the CMS messageDigest signed attribute. 27. The pure ML-DSA or pure SLH-DSA signature shall cover the complete DER-encoded SignedAttrs structure.
Validation of Digital Signature Certificate	28. RFC 5280. 29. ML-DSA certificate and CRL identifiers use RFC 9881; SLH-DSA certificate and CRL identifiers use RFC 9909.
ECC curve	NIST P-256, P-384, or P-521. Retained for transition, coexistence and legacy validation.
Long term signature formats	30. CAdES RFC 5126. 31. CAdES signatures support pure ML-DSA under RFC 9882 and pure SLH-DSA under RFC 9814, together with certificates, revocation evidence and timestamps.
	32. PAdES with CAdES.

Products	Standards
	33. PAdES signatures support pure ML-DSA and pure SLH-DSA through the corresponding CMS processing and long-term validation evidence.
Time stamp token	34. As specified in RFC 3161. 35. The TSA certificate and timestamp signature shall use ML-DSA or SLH-DSA with the corresponding certificate and signature identifiers. 36. SHA-2 is preferred for the message imprint.

C. Digital Signature (End Entity) Rules, 2015 - Rule 11 “The XML digital signature standards”

11. The XML digital signature standards. - The most important standards that shall be applicable for different activities associated with XML digital signature functions are as under-

The Product	Standard
XML Digital Signature Standard	RFC 3275 with the following constraints: • Manifest is not permitted inside Object. • KeyInfo containing the X509Certificate element is mandatory. • Reference Processing shall use Exclusive Canonicalization (without comments) in addition to other transforms. • For an XML resource, XSLT shall be the last transform for rendering the document on screen. • Each referenced XML resource shall be implemented using XSLT. • Each non-XML resource shall use the MimeType attribute in the Object.
XML Namespace	W3C Namespaces in XML 1.0; namespace URI syntax as specified in RFC 3986, where applicable.
Signature encoding	UTF-8 RFC 3629
Signature Value Encoding	Base64 RFC 4648
Reference element Digest	SHA-256, SHA-384 or SHA-512 under FIPS 180-4. SHA-2 is preferred and shall be selected according to the security category, assurance objective and retention period. A FIPS 202 hash function or XOF shall be used only where SHA-2 cannot be used and the exact identifier and output length are specified.
Signature Algorithm	SHA-256 with RSA PKCS#1 Version 1.5. Retained for transition, coexistence and legacy validation.
Post-quantum Signature Algorithm	1. ML-DSA shall be used in accordance with FIPS 204 using ML-DSA-44, ML-DSA-65 or ML-DSA-87. 2. SLH-DSA shall be used in accordance with FIPS 205. 3. For SLH-DSA, the SHA-2 parameter sets are preferred.

<i>The Product</i>	<i>Standard</i>
	4. The corresponding SHAKE parameter sets shall be used only where the SHA-2 parameter sets cannot be used. 5. Pure ML-DSA and pure SLH-DSA are preferred. 6. HashML-DSA or HashSLH-DSA shall be used only where pre-hashing is required. 7. SignatureMethod and KeyInfo shall identify the exact algorithm, parameter set and processing mode. 8. Private or provisional identifiers shall not be used.
<i>Signature block Canonicalization</i>	<i>Exclusive Canonicalization without comments, XML-EXC-C14N, RFC 3741; Canonical XML 1.0 without comments; Canonical XML 1.1 without comments.</i>
<i>Transform Algorithms</i>	<i>Exclusive Canonicalization without comments, XML-EXC-C14N, RFC 3741; Canonical XML 1.0; Canonical XML 1.1; XSLT Version 1.0; XPath transform—W3C XPath 1.0 as specified in XML Signature; XPath Filter 2.0—RFC 3653, where permitted by the applicable profile.</i>
<i>Signature Type</i>	<i>Enveloped, enveloping or detached</i>
<i>Digital Signature Certificate</i>	<i>DER-encoded X.509 version 3 certificate issued as per interoperability guidelines. ML-DSA certificates use RFC 9881; SLH-DSA certificates use RFC 9909.</i>
<i>Public Key Algorithms</i>	<i>RSA PKCS#1 Version 1.5 and ECC NIST P-256, P-384 or P-521 are retained for transition, coexistence and legacy validation. ML-DSA under FIPS 204 and SLH-DSA under FIPS 205.</i>
<i>Long Term Signature formats</i>	<i>1. XMLERS RFC 6283 and XAdES. XAdES shall preserve the ML-DSA or SLH-DSA algorithm, parameter set, mode, certificates, status evidence and timestamps.</i>
	<i>2. XMLERS RFC 6283 and PAdES with XAdES. The long-term validation record shall preserve the ML-DSA or SLH-DSA signature and supporting evidence.</i>
<i>Time Stamp Token</i>	<i>As specified in RFC 3161 in XML notation. The TSA certificate and timestamp signature shall use ML-DSA or SLH-DSA. SHA-2 is preferred for the message imprint.</i>

PART II IMPLEMENTATION READINESS TRACK

TRACK-2 IMPLEMENTATION READINESS TRACK

The purpose of this track is to assess whether the present Indian Public Key Infrastructure can migrate to post-quantum certificates and digital signatures without loss of security, interoperability, service continuity or the ability to verify existing records. The assessment covers the complete implementation chain: the Root Certifying Authority of India, licensed Certifying Authorities, Hardware Security Modules, CA software, certificate and revocation structures, subscriber devices, middleware, signature applications, validation services and long-term archival systems. It also addresses the migration of document signing, authentication, device certificates, secure communication, tender encryption and long-term validation of already signed records.

The principal conclusion is that final cryptographic standards are available for ML-DSA, SLH-DSA and ML-KEM, but the Indian PKI ecosystem is not yet ready for an immediate production-wide migration. Readiness must be demonstrated for each exact combination of algorithm, profile, product model, hardware, firmware, software, cryptographic provider, interface and operating configuration. A phased programme of inventory, profiling, laboratory testing, cross-vendor pilots and controlled production introduction is therefore required.

1. Current PKI ecosystem in India – {Rationale, Criticality, Extent}

1.1 Rationale

India's PKI is a statutory and operational trust framework established under the Information Technology Act, 2000, the Information Technology (Certifying Authorities) Rules, 2000, the Digital Signature (End Entity) Rules, 2015, and the technical guidelines issued by the Controller of Certifying Authorities. It supports legally recognised digital signatures by linking verified identities to public keys through X.509 certificates issued under the RCAI hierarchy.

Migration cannot be treated as a change confined to RSA or elliptic-curve algorithms. The selected post-quantum mechanism must pass through the entire lifecycle of key generation, certificate request, issuance, publication, signing, revocation, validation, time-stamping, archival preservation and later verification. Failure at any one layer can make a technically correct certificate or signature unusable.

1.2 Criticality

The ecosystem supports e-governance, e-commerce, banking, taxation, procurement, corporate filing, organisational approvals, eSign, remote signing and device identities. Its extensive installed base makes service continuity and verification of previously signed records essential during PQC migration.

The hierarchy also creates concentration risk. Failure of a Root CA, licensed CA, Time-Stamping Authority, revocation service or widely used relying application can affect a large number of certificates and records. PQC implementation must therefore preserve key protection, auditability, high availability, disaster recovery and secure failure behaviour.

1.3 Extent of the ecosystem

Layer	Principal components	Readiness concern
National trust infrastructure	RCAI, CA hierarchy, licensed CAs, certificate policies and trust stores	Trust-anchor rollover, coexistence of pre-quantum and post-quantum certificates, policy control and national interoperability.
CA cryptographic infrastructure	Validated Network and PCIe HSMs, key ceremonies, backup, replication, high availability and disaster recovery	Validated PQ algorithms, protected keys, backup and restore, performance and safe rollover.

Layer	Principal components	Readiness concern
CA and trust-service software	Certificate issuance, CSR processing, CRL, OCSP, TSA, repositories, audit and databases	New OIDs, larger objects, mixed chains, status services, logging and migration.
Subscriber environment	USB tokens, smart cards, eSign, remote signing, mobile keys and middleware	Secure PQ key generation and use, device capacity, API support, authentication and lifecycle.
Applications and relying parties	Government portals, financial systems, PDF/CMS/XML tools, desktop, web and mobile validators	Parsing, signing, validation, display, business rules, failure handling and backward compatibility.
Archival environment	Long-term validation, evidence records, timestamps, record repositories and preservation services	Continued verification of historical signatures and renewal of evidence before algorithms become weak.

The first implementation deliverable should be a National PKI Cryptographic Dependency Inventory, maintained as a machine-readable Cryptography Bill of Materials (CBOM) using a CCA-approved schema, such as CycloneDX 1.6 or a later approved version. The inventory should cover cryptographic components, algorithms, certificates, trust anchors, protocols and keystore or HSM references. Each entry should identify the responsible owner, vendor, deployed version, cryptographic dependencies, PQC capability, validation status, interoperability requirements, upgrade path and planned replacement date. Key-related entries should contain metadata and controlled references only, excluding private-key material, passwords, PINs and recovery secrets.

2. Readiness of the PKI data structure and associated standards

The principal data structures do not need to be discarded. X.509, CMS and related standards are algorithm-extensible, and final IETF profiles now define important uses of ML-DSA, SLH-DSA and ML-KEM. The immediate issue is whether deployed implementations correctly process these profiles and their larger keys, signatures and certificate chains.

Data structure or standard	Present baseline	Required PQC readiness	Current assessment
X.509 v3 certificates and CRLs	PKIX, RFC 5280, pre-quantum RSA, DSA and NIST elliptic curves	Recognise final PQ OIDs and encodings; issue and validate PQ CA, subscriber and CRL signatures; support larger objects and mixed migration paths.	Standards are available for ML-DSA, SLH-DSA, HSS/XMSS and ML-KEM public keys. Certified products and Implementation readiness must be tested.
PKCS #10 and enrolment	CSR carrying a public key and proof of possession	Generate PQ keys, carry the public key and request signature, process new OIDs and retain approval and audit workflows.	Product-specific; evidence required from CA, RA modules, HSM and device suppliers.
CMS and PKCS #7	SignedData, detached signatures, signed attributes and certificate packaging	Create and validate PQ SignedData, multiple signatures, timestamps,	Final profiles exist for ML-DSA, SLH-DSA and HSS/LMS; cross-vendor testing remains necessary.

Data structure or standard	Present baseline	Required PQC readiness	Current assessment
		embedded chains, negative vectors and large documents.	
CAAdES and PAdES	Advanced CMS and PDF signatures with validation evidence	Carry selected PQ signatures, certificates, revocation data and timestamps without breaking long-term validation.	Application support is incomplete; major relying products have not published complete PQC readiness dates.
XML Signature and XAdES	Algorithm-extensible XML signature with canonicalisation and transformations	Standard SignatureMethod identifiers, PQ-capable libraries, size handling, canonicalisation and interoperable validation profile.	A common Indian profile and reference vectors are required.
CRL and OCSP	Status publication and online validation	PQ-signed CRLs and responses, larger objects, caching, throughput, responder rollover, failover and mixed-chain validation.	Requires capacity and interoperability testing.
RFC 3161 and long-term evidence	Trusted time stamps and evidence used after certificate expiry	PQC-capable TSA profile, evidence renewal and preservation across pre-quantum and post-quantum periods.	Transition design is required before production migration.
PKCS #11, CSP and JCA/JCE (Java Crypt Arch/ Ext)	Interfaces between applications, devices and providers	Final PQ mechanisms, stable parameter handling, provider discovery, error reporting and non-exportable keys.	Support differs by vendor, release and firmware; no ecosystem-wide baseline exists.

2.1 Mandatory standards tests

- Issue root, issuing-CA, subscriber, CRL, OCSP and time-stamp objects using the selected PQ profiles.
- Validate each object independently in at least two implementations and across different suppliers.
- Test pure and pre-hash modes only under their correct identifiers and processing rules.
- Test malformed, oversized, unknown and withdrawn algorithms to confirm secure rejection.
- Confirm continued validation of existing RSA and ECC certificates and historical signatures.

2.2 Functional separation of RSA, ML DSA and ML KEM

Algorithm	Primary standard	Permitted cryptographic function	PKI consequence
RSA	FIPS 186-5 for signatures; SP 800-56B for RSA-based key establishment	Signature and RSA-based key establishment/encryption functions exist in the RSA family.	A single RSA implementation may support both operations, but certificate policy and keyUsage should restrict each certificate to its approved purpose.
ML-DSA	FIPS 204	Key generation, signature generation and signature verification.	Use for subscriber signatures, CA certificate signatures, CRL signatures and CSR proof of

Algorithm	Primary standard	Permitted cryptographic function	PKI consequence
			possession. It cannot encrypt or establish keys.
ML-KEM	FIPS 203 and SP 800-227	Key generation, encapsulation and decapsulation to establish a shared secret.	Use the derived secret with an approved symmetric authenticated-encryption scheme. It cannot create a digital signature or self-sign a PKCS #10 request.

2.3 Subscriber key generation, CSR submission and proof of possession

Certificate purpose	Key generation	Request and proof of possession	Overall finding	Implementation requirement
ML-DSA signing certificate	Subscriber generates an ML-DSA key pair in an approved device or module. The private key remains protected.	PKCS #10 can retain its existing structure: subject, SubjectPublicKeyInfo and requested attributes. The request is signed with the corresponding ML-DSA private key.	Same business sequence; technical modification required.	Add ML-DSA mechanisms and OIDs; support larger keys and signatures; enforce RFC 9881 encoding and proof-of-possession verification.
ML-KEM encryption or key-establishment certificate	Subscriber generates an ML-KEM encapsulation/decapsulation key pair. The decapsulation key remains protected.	The ML-KEM public key can be represented in SubjectPublicKeyInfo under RFC 9935. A standard PKCS #10 request signed by that same key is not possible because ML-KEM has no signing operation.	Material process modification required.	Use CRMF/CMP or another expressly profiled enrollment flow with KEM-capable proof of possession; alternatively authenticate the request with a separate signing credential and perform a CA challenge/confirmation that proves control of the decapsulation key. Where expressly permitted by the applicable Certificate Policy, RFC 9883 may instead provide a signed statement of possession for a key-establishment certificate. This is an alternative to technical proof, not decapsulation proof. Apply its identity, signature-certificate validation, security-strength and linked-revocation provisions;

Certificate purpose	Key generation	Request and proof of possession	Overall finding	Implementation requirement
Separate signing and encryption certificates	Subscriber generates two distinct key pairs and requests two certificates.	ML-DSA request proves possession by signature. ML-KEM request proves possession by decapsulation challenge or other approved method. A policy-authorised RFC 9883 statement may be accepted instead of technical proof only under the conditions in section 2.3.	Recommended target model.	Enrollment portal, token/HSM object labels, certificate profiles, lifecycle records, renewal and revocation must distinguish the two key purposes. Enrollment and lifecycle profiles shall manage separate pre-quantum and post-quantum credentials explicitly. RFC 9883 may assist eligible key-establishment enrollment; composite certificates and hybrid validation require their own adopted profiles.
Existing RSA signing and encryption certificates during transition	Existing RSA request and issuance procedures continue under current policy.	RSA may coexist with PQ certificates during the approved migration period.	No immediate structural change to legacy requests.	The CA and subscriber application must avoid replacing a functioning RSA path before all relying applications can process the new PQ certificate and signature formats.

2.3.1 Proof of possession for RSA and ML DSA signing requests

Process step	RSA signing certificate	ML-DSA signing certificate	Finding
Generate subscriber key pair	Subscriber device generates the RSA signing public/private key pair.	Subscriber device generates the selected ML-DSA public/private key pair.	Sequence unchanged. The mechanism, parameter set, key representation, device firmware and API change.
Create CertificationRequestInfo	CSR software inserts the subject name, RSA SubjectPublicKeyInfo and requested attributes or extensions.	CSR software inserts the same classes of information, but the SubjectPublicKeyInfo contains the ML-DSA public key and the OID for ML-DSA-44, -65 or -87. Algorithm parameters are absent.	Logical structure unchanged; parser, OID, encoding and size support change.
Create proof of possession	The subscriber signs the DER-encoded CertificationRequestInfo with the RSA private key corresponding to the public key in the request.	The subscriber signs the same DER-encoded CertificationRequestInfo with the corresponding ML-DSA private key.	Same proof principle and same position in the sequence. The signature algorithm and output size change.

Process step	RSA signing certificate	ML-DSA signing certificate	Finding
Submit request	Signed PKCS #10 CSR is sent through the CA/RA enrollment channel.	Signed ML-DSA PKCS #10 CSR is sent through the same enrollment channel, provided that its size and OID are accepted.	No necessary business-sequence change. Transport limits and allowlists require modification.
Authenticate applicant	CA/RA establishes the real-world identity and authority of the applicant under the applicable identity-verification and certificate policy.	The same identity and authorization process applies unless policy introduces a new assurance requirement.	No change caused solely by ML-DSA. Identity authentication is separate from cryptographic proof of possession.
Verify integrity and proof of possession	CA reconstructs CertificationRequestInfo, verifies the CSR signature with the RSA public key contained in the CSR, and confirms that the request has not been altered.	CA reconstructs CertificationRequestInfo, verifies the CSR signature with the ML-DSA public key contained in the CSR, and confirms that the request has not been altered.	Verification logic and algorithm provider change; security objective is unchanged.
Authorize and issue	CA applies policy, approves requested fields, creates the certificate and signs TBSCertificate with the CA signing key.	CA performs the same decision process. The subject public key is ML-DSA. The CA may sign the certificate with an approved RSA or ML-DSA CA key according to the transition profile.	Issuance sequence unchanged. Subject-key algorithm and issuer-signature algorithm must be processed as separate fields.

2.3.2 Scope of proof of possession

A valid CSR signature establishes that the entity creating the request controlled the private key corresponding to the public key carried in that request at the time of CSR generation. It also protects the signed CertificationRequestInfo against undetected alteration. The CA performs this check by verifying the CSR signature with the public key contained in the CSR. The method is the same for an RSA signing key and an ML-DSA signing key.

The CSR signature does not, by itself, prove the subscriber's civil, organizational or legal identity. The CA or Registration Authority must separately authenticate the applicant, validate the organization and authorization where applicable, and bind the verified identity to the public key only after both controls succeed. A CA must reject the request if either identity authentication or CSR proof-of-possession verification fails.

Root keys shall be established through approved ceremonies and authenticated trust-anchor distribution. A root self-signature does not independently establish trust. Verify subordinate-CA key possession and test issuer signatures, root rollover, path validation, revocation and time-stamping dependencies.

2.3.3 Sequence for a signing-only subscriber

Area	RSA	ML-DSA	Determination
Subscriber action	Generate key pair; create CSR; sign CSR; submit CSR; receive certificate.	Generate key pair; create CSR; sign CSR; submit CSR; receive certificate.	No additional subscriber business step.

Area	RSA	ML-DSA	Determination
Cryptographic implementation	RSA key generation and RSA CSR signature.	ML-DSA key generation and ML-DSA CSR signature.	Mandatory software, firmware, provider and mechanism change.
CA identity check	Identity and authorization under the applicable policy.	Same identity and authorization controls.	No additional identity procedure solely because ML-DSA is used.
CA proof-of-possession check	Verify RSA CSR signature with CSR public key.	Verify ML-DSA CSR signature with CSR public key.	Same control; different verification algorithm.
Encryption key or certificate	Not required when service scope is signature only.	ML-KEM is not required when service scope is signature only.	No additional encryption enrollment process.

2.4 Applicability of PKCS 10 to ML DSA and ML KEM

PKCS #10, defined by RFC 2986, contains CertificationRequestInfo, a signature-algorithm identifier and a signature. The request is signed using the private key corresponding to the public key being certified. This simultaneously protects the request and proves possession of the private signing key. ML-DSA satisfies this model because it supplies KeyGen, Sign and Verify operations. RFC 9883 separately permits a key-establishment request to be signed using an existing signing credential where the applicable Certificate Policy permits its statement-of-possession procedure; this does not make the ML-KEM key a signing key.

ML-KEM supplies KeyGen, Encaps and Decaps operations. It has no Sign operation. Placing an ML-KEM public key in the SubjectPublicKeyInfo field does not create a valid method for signing the CertificationRequestInfo with the corresponding ML-KEM private key. Unless the applicable Certificate Policy expressly permits the RFC 9883 statement described in section 2.3, the CA shall verify possession of the corresponding ML-KEM decapsulation key through a cryptographic challenge–response procedure. When the challenge–response procedure is used, the CA shall encapsulate a fresh secret using the submitted ML-KEM public key, and the subscriber shall demonstrate successful decapsulation by returning a cryptographic confirmation value derived from that secret. The proof shall be bound to the certificate request, subscriber identity, public key and enrolment transaction, and shall incorporate freshness and replay protection

The functional difference is:

- Signing key (RSA or ML-DSA): The applicant signs the certificate request; the CA verifies the signature using the submitted public key.
- Encryption key (RSA): The CA encrypts a challenge using the applicant’s public key; the applicant proves possession by decrypting it.
- KEM key (ML-KEM): The CA uses the applicant’s public key to create a ciphertext and shared secret; the applicant decapsulates the ciphertext and proves that it obtained the same secret.

RSA encryption proves private-key possession by decrypting a CA-encrypted challenge, whereas ML-KEM proves it by decapsulating a CA-generated ciphertext and confirming the resulting shared secret. The policy-authorised RFC 9883 statement is a separately identified alternative and shall not be described as technical proof of decapsulation-key possession.

2.5 End-to-end certificate request and issuance process

Step	Current RSA model	PQC model and impact
1. Select purpose	Signing or encryption purpose selected under policy.	Select ML-DSA for signing and ML-KEM for key establishment. One certificate cannot substitute for the other.

Step	Current RSA model	PQC model and impact
2. Generate key	Generate RSA pair with permitted modulus and mechanism.	Generate the relevant ML-DSA or ML-KEM pair using the FIPS algorithm and an approved random-bit generator inside the validated module boundary.
3. Build request	Place RSA public key and requested attributes in PKCS #10.	ML-DSA: place its public key and OID in PKCS #10. ML-KEM: carry its SubjectPublicKeyInfo in an approved request protocol that supports suitable proof of possession. A policy-authorised RFC 9883 request may use an existing signing credential as described in section 2.3.
4. Prove possession	Sign PKCS #10 with RSA private key; RSA encryption keys may also rely on an approved enrollment-specific POP method.	ML-DSA: sign request with ML-DSA private key. ML-KEM: complete decapsulation challenge, encrypted-certificate confirmation, RA verification or another nationally profiled method. Alternatively, process an expressly policy-authorised RFC 9883 statement under section 2.3.
5. Submit and authenticate	Subscriber submits request through existing authenticated channel.	The authenticated submission channel may continue, but must accept new sizes, OIDs and enrollment messages. Channel authentication does not replace ML-KEM private-key proof. An expressly policy-authorised RFC 9883 statement remains a separate enrollment control under section 2.3; channel authentication alone is insufficient.
6. CA validate	Validate identity, authorization, request signature, policy and RSA parameters.	Validate identity and authorization; parse new OIDs; validate ML-DSA request signature or ML-KEM POP; enforce purpose-specific keyUsage and profile. Where the applicable Certificate Policy permits RFC 9883, apply its statement-validation controls instead of technical ML-KEM proof.
7. Issue certificate	CA signs certificate using its approved CA signing key.	CA signs using its approved CA signing algorithm. The subject key may be ML-DSA or ML-KEM; the CA certificate-signature algorithm is a separate field and decision.
8. Use certificate	RSA signing key signs; RSA encryption key performs the profiled RSA operation.	ML-DSA signs. For encryption, sender encapsulates with recipient ML-KEM public key, derives a symmetric key and encrypts the data; recipient decapsulates and decrypts.

Certificate enrollment and issuance: signing and key-establishment branches

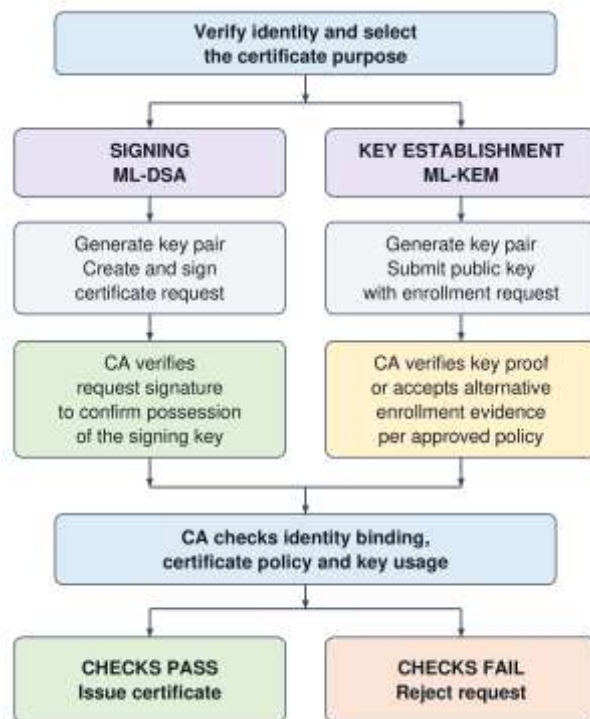


Figure 3-7. High-level certificate enrollment and issuance flow for signing and key-establishment certificates.

3. Impact assessment on suppliers of PKI systems and components

Suppliers will have to change products, interfaces, operating procedures and assurance evidence. A general statement that a supplier supports PQC is insufficient because support may be limited to a software library, an experimental build or a firmware release outside the validated boundary.

Supplier or component	Principal impact	Mandatory evidence before production
Network and PCIe HSM suppliers	New key types, larger key objects, signing or decapsulation mechanisms, PKCS #11/JCA support, backup, cloning, HA and firmware changes.	Exact model and firmware; CMVP certificate or published validation status; algorithm-validation scope; performance; backup and disaster-recovery demonstration.
CA software suppliers	New OIDs, profiles, CSR processing, certificate and CRL issuance, OCSP, audit, rollover and mixed hierarchies.	Exact release; supported profiles; HSM matrix; root-to-subscriber issuance; CRL/OCSP; rollback and migration test results.
Crypto-device and token suppliers	Secure-element capacity, applet and middleware changes, PQ key generation, signatures, authentication and device replacement.	Exact chip, firmware, applet and middleware; on-device ML-DSA capability; non-exportability; performance; validation mapping and lifecycle roadmap.

Supplier or component	Principal impact	Mandatory evidence before production
Cryptographic-library suppliers	Final algorithms, OIDs, ASN.1, CMS, X.509, provider interfaces and security maintenance.	Exact build and provider mode; standards conformance; FIPS boundary where required; negative tests and software bill of materials.
Signature-application suppliers	PDF, CMS, XML, bulk and server signing; certificate display; trust and status processing.	Create and verify reference signatures; cross-product validation; backward verification; unsupported-algorithm behaviour and roadmap.
TSA, OCSP and repository suppliers	Larger signed responses, signing-key migration, storage, bandwidth, caching and availability.	Load tests, failover, rollover, freshness, publication and relying-party compatibility.
Archival and long-term validation suppliers	Evidence renewal, preservation of certificates and revocation data, algorithm transition and historical verification.	End-to-end preservation test spanning pre-quantum and post-quantum signatures with documented renewal triggers.

3.1 Current product evidence

Category	Observed position	Treatment
HSMs	PQC-capable releases are emerging for network and PCIe HSM classes, but FIPS status is exact-version and firmware-specific.	Do not name an HSM production-ready until the deployed PQ firmware and algorithms are within the applicable validated boundary.
CA platforms	Nexus, eMudhra, Odyssey and Entrust require exact-release assessment. Public evidence is uneven; crypto-agility claims do not prove complete issuance and validation readiness.	Require one common CCA test suite and vendor evidence for root, issuing CA, subscriber, CRL, OCSP and rollover functions.
Libraries	OpenSSL 3.5, Bouncy Castle and other libraries implement final PQ algorithms, while FIPS status varies by module and release. liboqs remains primarily useful for research and prototyping.	Separate functional support from FIPS validation and from application-level standards support.
Subscriber devices	Current CCA records establish that the identified devices comply with applicable requirements for pre-quantum cryptographic functionality but do not establish native ML-DSA capability.	Assess each exact secure element, applet, firmware and middleware combination; plan replacement where upgrade is not possible.

3.2 Findings from vendor readiness submissions

The submissions show usable PQC building blocks, but not uniform end-to-end readiness. The statements below are vendor-reported positions and must be confirmed against the exact product release, cryptographic boundary, interface, certificate profile and customer configuration before regulatory or production reliance.

Vendor or product	Brief finding	Applicable treatment in this report
Entrust nShield HSM and CSP PKI	Reports production support for ML-KEM, ML-DSA and SLH-DSA, in-HSM key generation, hybrid and pure PQ certificates, and CAVP record 39895. Certification/validation is marked partly complete; the offered PQ Lab is expressly non-production.	Treat this as a strong candidate for laboratory testing. Confirm the exact 5C model, firmware, software release, CMVP boundary, mechanisms, backup/restore, CA/TSA functions and cross-product validation before approval.
Thales Luna Network HSM 7 and CipherTrust Manager	Reports native ML-KEM and ML-DSA in Luna firmware 7.9.x with CAVP record 39968, while the stated PQ-enabled FIPS 140-3 boundary for firmware 7.9.2 remains under review. CipherTrust Manager has partial PQ protection for management interfaces; native application-key management remains on the roadmap.	Separate Luna functional and algorithm evidence from CMVP status and from CipherTrust orchestration capability. Pilot the exact firmware/client combination and verify HA, clone, backup, restore, PKCS 11, CSR, attestation and disaster recovery.
JISA Softtech CryptoBind HSM	Reports ML-KEM, ML-DSA and SLH-DSA algorithm capability and in-device key generation at pilot stage. Certificate/signature interfaces and certification remain planned; complete readiness is targeted for Q1 2027, with testing stated to be under way with C-DAC.	Do not treat algorithm availability as PKI readiness. Require interface specifications, exact hardware/firmware evidence, validation scope, certificate and CSR tests, performance results and independent interoperability results.
Nexus Smart ID Certificate Manager 8.15.x	Reports native ML-KEM, ML-DSA and SLH-DSA support, including CA/end-entity signatures, key establishment and hybrid/composite certificates. Customer testing is continuing; EAL4+ is cited, while PQ cryptographic-module validation depends on the integrated HSM/provider.	Use in CA software pilots only with an approved HSM and exact integration matrix. Test root and issuing-CA creation, PKCS 10/POP, CRL, OCSP, TSA, rollover, mixed chains, rollback and standards-conformant hybrid handling.
Odyssey Certrix	Reports ML-KEM and ML-DSA integration at testing/pilot stage. SLH-DSA is planned for Q1 2027-28 and overall readiness for Q1-Q2 2027-28. Dependencies include hybrid-certificate policy, PQ-capable hardware, application test beds, stable libraries and IETF protocol profiles.	Include Certrix in cross-vendor pilots, not production approval. Require exact release and HSM matrix, larger-object capacity tests, CRL/OCSP/TSA performance, rollback procedures and interoperability evidence.
Precision InnaIT IAM and DSC-token ecosystem	Reports engineering integration and architecture validation, not certified product readiness. It proposes a cryptographic abstraction layer with pre-quantum cryptography, hybrid, PQC-preferred and PQC-only operating profiles; HSM/token	Adopt the abstraction and policy-profile approach as a useful crypto-agility pattern. Production use must wait for an exact certified provider/device combination, on-device key protection, middleware support, performance testing and end-to-end CA/application qualification.

Vendor or product	Brief finding	Applicable treatment in this report
	and third-party interoperability is expected within 9-18 months.	
Adobe application ecosystem	Reports a phased approach, private DigiCert certificate testing in managed-device environments and a platform target of 2029 to support migration before 2030. Readiness depends on foundational libraries and supported client operating-system/browser versions.	Maintain coexistence of pre-quantum and post-quantum mechanisms and backward validation through the application transition. Obtain product-specific evidence for Acrobat/Reader, Acrobat Sign and enterprise products, including PDF/CMS profiles, trust processing, supported clients and failure behaviour.

The submissions support immediate creation of an India PKI interoperability test programme, but they do not establish ecosystem-wide production readiness. The most mature claims relate to ML-KEM and ML-DSA functions in selected HSM and CA-platform releases. The principal remaining gaps are exact-version module validation, certificate and protocol profiling, subscriber device capability, cross-vendor application validation, performance, lifecycle operations and documented rollback.

4. Readiness of supplier & regulator ecosystem

The ecosystem is at different stages. Final standards exist, software implementations and some supplier products are available, and several FIPS submissions are in progress. Production readiness across the regulated Indian PKI has not yet been demonstrated. Regulation, product assurance and procurement must be aligned so that one component does not advance ahead of the dependencies needed to use it safely.

Readiness level	Definition	Evidence
No capability	No supported PQ implementation or committed roadmap.	Inventory record and confirmed product limitation.
Roadmap available	Supplier has identified changes and target releases.	Dated product roadmap, affected versions and support policy.
Laboratory capable	PQC is implemented but assurance or interoperability remains incomplete.	Laboratory build, conformance results and known limitations.
Pilot ready	Standards-conformant implementation passes the CCA reference tests in a controlled environment.	Cross-vendor results, security assessment, operations plan and rollback.
Production ready	Security validation, interoperability, performance, HA/DR and lifecycle support are demonstrated.	Approved exact-version configuration and completed production-readiness review.

4.1 Regulatory responsibilities

Entity	Implementation-readiness responsibility
CCA	Define national certificate and signature profiles, readiness criteria, common test vectors, audit requirements, phased approval and reporting.

Entity	Implementation-readiness responsibility
Licensed Certifying Authorities	Create dependency inventories, obtain supplier evidence, conduct laboratory and pilot testing, revise CP/CPS and demonstrate continuity and recovery.
MeitY and national cyber institutions	Coordinate digital-trust, cybersecurity, incident-response and government adoption requirements within their legal mandates.
STQC and notified laboratories	Test cryptographic modules, products, certificate structures, application conformance, security and interoperability under notified criteria.
Application-owning ministries and organisations	Identify relying applications, remove hard-coded assumptions, fund upgrades and accept migration risk only after business and continuity testing.
Suppliers and OEMs	Publish exact-version roadmaps, software bills of materials, validated configurations, support periods, vulnerabilities and migration constraints.

The regulator should approve a configuration, not a brand. Approval should identify the exact hardware, firmware, software, provider, algorithm profile and operating mode. Any change affecting that combination should trigger defined reassessment.

4.2 Controls arising from the vendor analysis

- PQC hardware acceptance shall follow the validation, configuration and operational-testing requirements in Part III, sections 3.4–3.6. Supplier readiness shall be assessed against those requirements for the exact product version and intended service.
- Establish a common India PKI test bed using at least two HSM suppliers, two CA-platform suppliers and representative subscriber devices and applications. Test root-to-subscriber issuance, PKCS 10 and ML-KEM proof of possession, CRL, OCSP, TSA, CMS, PDF, XML, HA, backup, restore, disaster recovery and rollback.
- Permit vendor participation by readiness stage: architecture and library work where a supplier roadmap is available, integrated testing where laboratory capability is demonstrated, cross-vendor pilots where pilot-entry conditions are met, and production only with exact approved configurations. Self-declared production readiness shall not bypass these gates.
- Require crypto-agile operating profiles for pre-quantum cryptography, coexistence or hybrid, PQC-preferred and eventual PQC-only states. Each transition shall have explicit acceptance rules, downgrade protection, telemetry, failure handling and a tested return path.
- Prioritise ML-DSA signing and ML-KEM key establishment in the first test cycle. Track SLH-DSA and FN-DSA separately so that roadmap claims do not delay deployment testing of the finalised priority profiles.
- Plan application coexistence through the stated 2029-2030 vendor transition horizon unless earlier product-specific evidence supports migration. Minimum-client requirements, unsupported-platform behaviour and long-term validation of existing records shall be documented before production change.

5. Impact assessment of proposed changes to applications

Application change is likely to be the largest source of hidden cost and delay. Many applications contain assumptions that are not visible at the cryptographic interface, including RSA-only allow lists, fixed key-length checks, limited database fields, certificate-size limits, token-specific middleware and user-interface logic that labels unknown algorithms as invalid.

Application boundary	Required change	Failure if omitted
Certificate acceptance	Recognise PQ OIDs, public-key encodings, certificate signatures, larger chains, key usages and policy identifiers.	Valid certificates may be rejected, mislabelled or fail during import.
Signature creation	Use a PQ-capable provider or device; build the correct CMS, PDF or XML object; integrate time stamps and handle larger output.	Signing fails, produces a non-standard object or falls back to an unapproved mechanism.
Signature validation	Verify PQ signatures and paths, status, timestamps, permissions and historical evidence while retaining RSA/ECC support.	False rejection, false acceptance or inability to verify older records.
Business rules	Replace fixed algorithm, key-size, buffer-length and certificate assumptions with configurable application and database settings.	Operational errors remain even when the cryptographic library is correct.
User interface and audit	Display algorithm, trust, status and failure reason accurately; record evidence sufficient for audit.	Users cannot distinguish unsupported, invalid, revoked or untrusted signatures.
Deployment and rollback	Distribute providers, middleware, trust data and application updates in a controlled order.	Partial deployment creates inconsistent validation results and service interruption.

5.1 Application-product dependencies

Product class	Known dependency	PQC position
Adobe Acrobat Pro and Reader	Adobe application processing; embedded OpenSSL FIPS module on Windows in FIPS mode; Windows certificate interfaces; macOS Keychain; external PKCS #11 devices. Bouncy Castle is not publicly identified as the Acrobat/Reader core.	Adobe's September 2026 response states that it plans phased infrastructure upgrades and aims to have platform PQC features ready by 2029, allowing customer migration before a 2030 target. The response identifies dependencies on foundational libraries such as OpenSSL and Java and on supported client environments, including Windows 11 or later and updated Chrome versions. This is a planning position, not evidence that general production validation of ML-DSA or SLH-DSA PDF signatures is presently available.
Adobe Acrobat Sign	Adobe-operated cloud service; detailed internal cryptographic dependencies are not publicly specified.	Adobe's response describes initial testing of private DigiCert PQC certificates in managed-device environments, with expansion contemplated as vendor-issued certificates become available. Exact Acrobat Sign algorithm, certificate-

Product class	Known dependency	PQC position
		profile, validation and production dates still require product-specific confirmation.
Other Java, .NET and native applications	May depend on JCA/JCE providers, Windows CNG/KSP, OpenSSL, PKCS #11 middleware or vendor libraries.	Readiness is product- and version-specific; underlying library support does not establish application readiness.

5.2 Impact on signature creation

Stage	Present RSA process	ML-DSA process	Required change
Selection of certificate	RSA signing certificate selected by keyUsage and application policy.	ML-DSA certificate selected. ML-KEM certificate must never be selected for signing.	Mandatory algorithm- and keyUsage-aware selection.
Data preparation	Application forms the content to be signed and applies the specified RSA signature encoding and hash treatment.	Application forms the same business content, but uses the adopted ML-DSA profile. RFC 9881 profiles pure ML-DSA for PKIX and requires the ML-DSA OID parameters to be absent.	Signing API and profile configuration change; business document need not change solely because the key algorithm changes.
Private-key operation	RSA private-key signature operation in token, HSM or software module.	ML-DSA.Sign in a supporting cryptographic module.	Device, middleware, provider and API must expose ML-DSA and keep the private key inside the approved boundary.
Signature encoding	RSA signature length follows modulus size; algorithm parameters follow the applicable RSA profile.	Signature is a fixed-size ML-DSA value: 2,420, 3,309 or 4,627 bytes for ML-DSA-44, -65 or -87.	Buffers, database columns, CMS/PDF/XML containers, network limits and user interfaces require size and parser review.
Validation	Relying party recognises RSA OIDs, validates the certificate path and verifies the RSA signature.	Relying party must recognise ML-DSA OIDs, enforce absent parameters, validate an ML-DSA-capable path and verify the ML-DSA signature.	All relying applications require exact-version interoperability testing.
Coexistence	Existing RSA records remain verifiable.	Migration may use separate pre-quantum and post-quantum certificates or an expressly profiled hybrid/composite approach.	Do not use proprietary dual-signature structures, as they may lack interoperability, standard algorithm identifiers, independent security analysis, validator support and long-term verifiability. Use only an adopted standard or CCA-approved hybrid/composite profile, and preserve all evidence required to validate historical RSA signatures..

5.3 Impact on PKI services and application flows

Function	RSA flow	ML-DSA flow	Practical impact
Subscriber document signature	Application creates the prescribed signed object and invokes the RSA signing private key.	Application invokes ML-DSA. For CMS, RFC 9882 specifies pure ML-DSA and ML-DSA algorithm identifiers; signed attributes remain applicable.	User-facing sequence need not change. Signing provider, OIDs, encoding, buffer sizes and validation logic must change. ML-DSA signatures are much larger.
Document encryption	An RSA encryption certificate may be used for the profiled key-transport or encryption arrangement. It is separate from the signing purpose under sound policy.	ML-DSA does not encrypt. Encryption remains a separate service using ML-KEM or another approved key-establishment mechanism plus symmetric encryption.	No impact where the subscriber only signs. An ML-DSA signing certificate must not be presented as an encryption certificate.
Timestamp request	Client hashes the signed document or signature value and sends a TimeStampReq. The TSA returns a CMS SignedData time-stamp token signed by its TSA key.	The requester-side sequence is unchanged. If the TSA remains RSA, the token remains RSA-signed. If the TSA migrates to ML-DSA, its CMS token must follow the applicable ML-DSA CMS profile and all validators must support it.	No additional subscriber enrollment step. TSA, CMS, token-size and relying-application changes arise when the TSA migrates to a different signing algorithm.
Timestamp validation	Validate TSA certificate path, RSA token signature, message imprint, policy, nonce and time.	Validate the same semantic fields. If ML-DSA is used by the TSA, validate the ML-DSA certificate path and CMS signature.	Validation sequence unchanged; algorithm support and object size change.
OCSP request	Ordinary OCSP requests are normally unsigned. Client identifies the certificate and sends the request.	The same request can be used for an ML-DSA subscriber certificate, subject to parser and identifier support.	No additional subscriber proof of possession. OCSP checks certificate status, not possession of the subscriber private key.
OCSP response	Responder signs BasicOCSPResponse using its authorized RSA responder key and certificate.	Responder may remain RSA during coexistence. If migrated, it signs with an authorized ML-DSA responder key and clients must recognise and verify the ML-DSA signature and chain.	Response semantics are unchanged. Responder HSM/provider, OIDs, response size and client verification support change only if the responder algorithm changes.
Certificate and CRL issuance	CA signs certificates and CRLs with its RSA CA signing key.	RFC 9881 permits ML-DSA signatures on certificates and CRLs. The subject may use ML-DSA even while the issuer remains RSA, if the	Subscriber-key migration and CA-signing-key migration can be phased separately.

Function	RSA flow	ML-DSA flow	Practical impact
		national transition profile permits that mixed path.	

5.4 Signature-only process comparison

If the subscriber requires only a digital-signature certificate, migration from RSA to ML-DSA does not require the subscriber to obtain an ML-KEM key pair, an encryption certificate or an additional proof-of-possession transaction. The subscriber continues to generate one signing key pair and one PKCS #10 request and proves possession by signing that request with the corresponding private key. The CA continues to authenticate the applicant separately, verify the CSR signature, approve the certificate profile and issue the certificate.

The modifications occur inside the technical implementation: selection of the ML-DSA parameter set; key generation in a supporting validated module; ML-DSA SubjectPublicKeyInfo and signature OIDs; absent algorithm parameters; larger CSR, certificate and signature objects; ML-DSA-capable CA validation; and relying-application support. These are mandatory changes, but they do not add a new subscriber-facing process step.

5.5 Representative application migration scenarios

The migration requirement depends on the function performed by the certificate. A system certificate used to authenticate a server or payment gateway is not the same as a certificate used to sign a payment instruction. Similarly, eSign separates identity authentication, generation of a short-lived signing key and certificate, document signing and long-term validation. The following scenarios apply the general migration principles to two important India PKI uses.

5.5.1 Bank-to-bank fund transfer using system certificates

Bank-to-bank payment systems commonly use system certificates to authenticate participating institutions, establish protected communication channels and, where the payment protocol requires it, sign transaction messages or files. These functions shall remain separately identified. A transport-layer certificate proves the identity of the communicating system; it does not by itself prove that an individual payment instruction was authorized. Message-level signatures and the application audit trail provide that evidence when required by the scheme.

Process stage	Present RSA or ECC implementation	PQC migration requirement	Effect on sequence
System key and certificate	The bank or payment-system operator generates a protected key pair, submits a signed certificate request and receives a system certificate.	Introduce an approved ML-DSA certificate profile for system authentication or message signing. Keep keys non-exportable in an approved HSM and retain proof of possession in the certificate request.	The business sequence remains unchanged. Algorithms, object identifiers, key sizes, HSM mechanisms and certificate-validation software change.
Secure channel	Mutual TLS or an equivalent protocol authenticates the endpoints and protects data in transit.	For TLS-based migration, use TLS 1.3 with an adopted, tested hybrid key-establishment profile where supported. Specify endpoint authentication separately, prevent downgrade and qualify the exact configuration. Other protocols shall follow their approved profiles. ML-KEM	Connection establishment remains recognizable, but protocol negotiation, cipher suites, certificates, libraries and monitoring require modification.

Process stage	Present RSA or ECC implementation	PQC migration requirement	Effect on sequence
		protects key establishment; ML-DSA or an approved pre-quantum signature authenticates the endpoint.	
Payment instruction or file	The application may sign the payment message, batch file or transaction digest using a system signing key.	Where message-level signing is required, define an ML-DSA-capable message profile or a controlled dual-signature profile. Do not treat ML-KEM as a signing algorithm.	The authorization flow remains unchanged. Signature fields, encodings, size limits, canonicalization rules and verification logic require modification.
Receipt and validation	The receiving system validates the peer certificate, message signature, authorization data, freshness and duplicate status.	Validate the selected pre-quantum and post-quantum validation paths according to a stated transition policy. Preserve algorithm, certificate path, revocation status, time evidence and transaction identifiers.	The logical checks remain the same; validators, trust stores, status services, logs and evidence formats must become crypto-agile.
Continuity and recovery	HSM redundancy, certificate rollover, disaster recovery and archived transaction evidence support service continuity and dispute resolution.	Test larger certificates and signatures, HSM capacity, rollover, fallback controls and validation of historical records. Existing signed records shall not be re-signed merely because a new algorithm is introduced.	Operational procedures require rehearsed PQC rollover and recovery, but the control objectives remain unchanged.

Recommended position: payment-system operators should first inventory whether each system certificate is used for endpoint authentication, message signing or both. Hybrid protection should be introduced first for communication channels exposed to harvest-now-decrypt-later risk. ML-DSA system certificates and message signatures should enter production only after the payment message format, HSM, CA, trust stores, counterparties, monitoring and disaster-recovery sites have passed bilateral and end-to-end interoperability tests. Acceptance rules for pre-quantum and post-quantum mechanisms must be explicit so that fallback cannot silently weaken the service.

5.5.2 eSign and eOffice document-signing scenario

In an eSign-based workflow, the user is authenticated under the approved eSign process, a signing key is generated and used within the controlled signing environment, a short-lived signing certificate is issued, and the resulting signature is returned to the application. eOffice or another government application then embeds or associates the signature with the document and relies on certificate-path, revocation and time evidence for later validation. Migration to ML-DSA does not require an additional action from the signer, but it requires coordinated changes across the eSign provider, CA, application and long-term validator.

Process stage	Present eSign or eOffice flow	PQC migration requirement	Effect on user and system
User authentication and consent	The application obtains the prescribed consent and sends an authenticated eSign request through the approved interface.	Retain the approved identity and consent process. Protect the application-to-eSign channel using an approved pre-quantum or hybrid secure-communication profile.	No additional user step is required solely because ML-DSA is used for the document signature.
Signing key and certificate	The controlled eSign environment generates or activates the signing key and obtains the corresponding short-lived certificate.	Generate an ML-DSA key in the approved cryptographic module; bind the ML-DSA public key to the certificate request; demonstrate proof of possession by signing the request or equivalent CA-controlled evidence.	The service sequence remains the same. HSM firmware, CA software, certificate profiles, OIDs, audit evidence and capacity planning require modification.
Signature creation	The service signs the application-provided document hash or prescribed signed object and returns the signature response.	Adopt an approved ML-DSA signature profile for CMS, PDF or the applicable eSign response format. Define whether the transition requires a pre-quantum signature, a post-quantum signature or both.	The user experience can remain unchanged. APIs, schemas, size limits, signature containers and error handling require modification.
eOffice integration and validation	eOffice embeds or records the returned signature and validates the certificate chain, signature, revocation status and signing-time evidence.	Add ML-DSA certificate and signature parsing, trust-path processing, revocation handling and clear display of the validation result. Continue validating existing RSA and ECC signatures.	Application screens and workflow need not change materially, but all signing and validation components must be upgraded together.
Long-term preservation	Signed records and validation evidence are retained according to the applicable record-retention period.	Preserve the original signed document. Before the existing algorithm or evidence becomes unreliable, collect the certificate path, revocation evidence and trusted time evidence and renew the archival evidence under an approved preservation profile.	A later PQC time-stamp strengthens the preservation chain only when the earlier signature and evidence were validated before renewal; it does not repair an invalid signature.

Recommended position: eSign and eOffice should retain the present user-facing sequence. The migration should be implemented through crypto-agile APIs and signature containers, controlled ML-DSA pilots and parallel validation of pre-quantum and post-quantum signatures. Production migration should occur only when the eSign provider, issuing CA, HSM, application, PDF or CMS validator, time-stamping service and archival system support the same approved profile. Existing RSA- or ECC-signed documents should remain validatable through long-term validation and evidence renewal rather than bulk re-signing

5.5.3 MCA21 electronic filing and digital-signature scenario

MCA21 uses digital signatures extensively to bind statutory electronic filings to the person acting in an authorized capacity, such as a director, designated partner, company secretary, chartered accountant, cost accountant or other permitted professional. Depending on the service and form, the workflow may include portal registration and association of the Digital Signature Certificate with the user profile, preparation of a web form or electronic form, attachment of supporting records, signature through local signing middleware or an approved remote-signing service, submission to MCA21, acknowledgement and later verification of the filed record. The exact roles and signature requirements remain those prescribed for the relevant form and shall not be inferred merely from possession of a certificate.

For PQC migration, MCA21 should preserve the legal and business meaning of the existing filing process. The principal change is not an additional step for the filer. It is coordinated enablement of ML-DSA across subscriber devices or remote-signing services, Certifying Authority systems, certificate profiles, portal interfaces, form and attachment formats, validation services, payment and acknowledgement workflows, inspection systems and the long-term corporate record. Identity, organizational role and statutory authority must continue to be verified separately from cryptographic proof of possession.

Process stage	Present MCA21 use of digital signatures	PQC migration requirement	Effect on process
User registration and DSC association	The registered user associates a valid DSC with the MCA21 profile. The portal links the certificate identity to the declared user type and permitted role.	Accept approved ML-DSA certificate profiles and public-key identifiers. Reassociation or certificate update shall follow the same identity and role controls; algorithm change alone shall not expand authority.	The user sequence remains substantially unchanged. Portal, middleware, certificate parsing and role-mapping logic require modification.
Key generation and proof of possession	For a token-based DSC, the subscriber key is generated or held in the approved cryptographic device. For remote signing, the key is generated and used in the controlled signing environment. The CA verifies proof of possession during certificate issuance.	Generate the ML-DSA key in an approved module. Bind its public key to the certificate request and prove possession through a valid ML-DSA signature over the request or through an equivalent CA-controlled protocol for remote signing.	No additional business step is required. Token or HSM firmware, PKCS 11 or signing APIs, CA software, request profiles and audit evidence must change.
Form preparation and attachments	MCA21 constructs the prescribed web form or electronic form and binds the filing data and attachments to the submission workflow.	Ensure that form schemas, document containers, canonicalization rules, digest handling, attachment limits and client-server interfaces can carry larger PQC certificates and signatures without altering the signed meaning.	Form fields and statutory declarations need not change solely due to PQC. Technical size limits, serialization and validation rules require testing.
Signature creation	The authorized signatory or professional selects the associated DSC and signs through the supported client utility,	Add ML-DSA signing and define a controlled transition rule: pre-quantum only, dual or composite where formally profiled, or PQC only after complete ecosystem readiness.	The visible signing action can remain unchanged. Signature APIs, OIDs, buffers, form containers, error messages and

Process stage	Present MCA21 use of digital signatures	PQC migration requirement	Effect on process
	middleware or remote-signing route.	ML-KEM shall not be used for signing.	verification components require modification.
Submission and portal validation	MCA21 receives the filing and validates the signature, certificate chain, validity period, revocation status, signer identity and role-related filing rules before acceptance or further processing.	Validate ML-DSA signatures and certificate paths against the approved India PKI profile. During transition, apply an explicit policy for pre-quantum and post-quantum validation results and reject downgrade, unknown-algorithm and partial-validation conditions.	The filing sequence remains unchanged. Trust stores, CRL or OCSP processing, validation services, application rules, logs and operational monitoring require coordinated upgrade.
Professional certification and multiple signatures	Some filings require signatures or certifications by more than one person or professional, each acting in a defined capacity.	Permit independently verifiable signatures using pre-quantum algorithms and ML-DSA without invalidating an earlier signature when another signature is added. Record the algorithm, certificate, role and validation result for each signer separately.	The order and statutory purpose of signatures remain unchanged. Containers and workflow engines must support mixed-algorithm signatures during migration.
Acknowledgement and evidence	The portal issues acknowledgement and retains the filed form, attachments, receipt data, certificate information and processing history.	Bind the accepted signature result, certificate path, revocation evidence, trusted time evidence, filing identifier and acknowledgement to an auditable preservation record.	No new filer action is required. Evidence collection, storage capacity, audit format and retrieval tools require modification.
Public inspection and long-term validation	Filed corporate records may remain legally and operationally relevant long after submission and may be inspected or relied upon later.	Preserve existing RSA- and ECC-signed filings unchanged. Validate and collect supporting evidence before algorithms or certificates become unreliable, then renew archival evidence under an approved preservation profile, including PQC protection when ready.	Historical records shall remain readable and verifiable. Bulk re-signing of old filings is neither necessary nor sufficient because it can obscure the original act and time of filing.

Recommended position: MCA21 should use a staged, backward-compatible migration. The first phase should inventory every signature point, client utility, middleware component, form container, certificate and revocation dependency, role-mapping rule and archival record. The second phase should add ML-DSA certificate and signature validation while continuing to accept valid existing RSA and ECC filings. Controlled pilots should then test token-based and remote-signing routes, multiple-signature forms, professional certification, rejection handling, acknowledgements and record inspection. New ML-DSA signing should enter production only after CCA, MCA, Certifying Authorities, signing-service providers, token and HSM suppliers and MCA21 application operators adopt the same profile and pass end-to-end interoperability and recovery tests.

6. Readiness of applications

An application should be classified as ready only after it passes a complete reference corpus. Successful validation of one PQ certificate or one signature does not establish readiness for certificate paths, revocation, timestamps, malformed inputs, mixed migration states or long-term evidence.

Test class	Minimum acceptance requirement
Existing cryptography baseline	Existing valid RSA and ECC certificates and signatures continue to operate without change in their historical validity.
PQ certificate parsing	Root, CA and subscriber certificates are imported, displayed and processed with correct algorithm and policy information.
PQ path validation	Complete paths build to the intended trust anchor and enforce basic constraints, key usage, policy and validity.
Signature creation	Signature is created through the approved secure-key mechanism using the adopted CMS, PDF or XML profile.
Independent verification	At least one independent implementation verifies the certificate path and signature.
Revocation and status	CRL and OCSP processing, caching, freshness, responder rollover and failure modes operate correctly.
Time stamp and long-term validation	Time stamps and embedded validation evidence are created, renewed and validated across algorithm transitions.
Cross-vendor interoperability	Test interoperability between independent suppliers according to each product's supported functions; test signing and verification in both directions where both products support those functions.
Mixed ecosystem	Pre-quantum, hybrid where approved, and pure post-quantum objects coexist under an explicit validation policy.
Negative and malformed input	Unknown OIDs, altered signatures, oversized fields, withdrawn roots and malformed encoding fail securely.
Upgrade and rollback	Deployment and rollback do not corrupt keys, trust stores, audit records or existing signed documents.

6.1 Readiness conclusion

Current applications generally range from having no demonstrated PQC capability to supporting laboratory testing. Cryptographic libraries and experimental PDF implementations provide a basis for laboratory work, but broader pilot readiness requires an Indian reference profile and cross-vendor tests. Production readiness should be confirmed only after the exact application release, provider, operating system and trust configuration have passed security, interoperability, performance and lifecycle testing.

6.2 FIPS validation and assurance implications

PQC hardware acceptance shall follow the validation, configuration and operational-testing requirements in Part III, sections 3.4–3.6. Supplier readiness shall be assessed against those requirements for the exact product version and intended service.

Assurance area	Requirement
Algorithm implementation	ML-DSA shall conform to FIPS 204; ML-KEM shall conform to FIPS 203. Parameter sets and approved modes must be fixed by national profile.
Module boundary	Key generation and private-key operations shall occur in the approved operational mode of the exact validated hardware, firmware or software module configuration.
Randomness	Key generation and randomized operations shall use an approved DRBG and entropy source as required by the module security policy and applicable NIST guidance.
Algorithm validation	Evidence should include the applicable CAVP/ACVP algorithm-validation records where required, together with the CMVP certificate covering the operational module.
Interfaces	PKCS #11, JCA/JCE, CNG/KSP or proprietary interfaces must expose the correct mechanisms without exporting private key material or silently falling back to a non-approved implementation.
Key consistency	Implement required public/private key consistency checks and the ML-KEM expanded-key hash check. Imported seed and expanded representations must be handled exactly as the applicable standard and profile require.
Certificate profile	Enforce RFC 9881 for ML-DSA and RFC 9935 for ML-KEM, including exact OIDs, absent parameters and restricted keyUsage. For ML-KEM, RFC 9935 specifies keyEncipherment as the only keyUsage bit.
Operational evidence	Test CSR generation, POP, issuance, renewal, rekey, revocation, recovery, rollover, malformed input, cross-vendor interoperability and performance using the exact production versions.

6.3 Suggested India PKI implementation position

1. Retain the existing subscriber-to-CA business sequence for ML-DSA signing certificates, but mandate new algorithm, encoding, size, device, provider and interoperability support.
2. Do not use an ML-KEM key to sign a PKCS #10 request. Define a separate enrollment and proof-of-possession profile for ML-KEM certificates. An expressly policy-authorised RFC 9883 statement may be used as specified in section 2.3.
3. Where both functions are required, issue separate ML-DSA signing certificates and ML-KEM key-establishment certificates. Signing-only subscribers require only the signing certificate. Apply purpose-specific keyUsage, certificate policy OIDs, lifecycle controls and audit records.
4. Keep subscriber authentication and cryptographic proof of possession as separate controls. Aadhaar, organizational approval or an authenticated portal session establishes applicant identity but does not by itself prove control of an ML-KEM private key. Any policy-authorised statement accepted in place of technical proof shall meet the separate RFC 9883 controls in section 2.3.
5. Require CA software, RA software, HSMs, subscriber devices, middleware and relying applications to demonstrate the complete workflow. Support at library level alone is insufficient.
6. Permit legacy RSA services to coexist during a controlled transition. Preserve verification of historical RSA signatures and avoid forcing PQ certificates into applications that have not passed interoperability and performance tests.
7. Publish an India PKI PQC profile covering parameter sets, OIDs, CSR/CRMF/CMP enrollment, POP, certificate and CRL signatures, CMS/PDF/XML formats, keyUsage, renewal, revocation, archival validation and failure behavior.

PQC root migration shall preserve applicable RCAI key-custody, ceremony, backup and disaster-recovery requirements. Any additional domestic-location or remote-access restriction shall be adopted expressly through the relevant policy and CP/CPS.

6.4 Readiness determination

For a subscriber requiring only a signature certificate, the sequence remains: generate one signing key pair, construct a PKCS #10 CSR, sign the CSR with the corresponding private key, submit it to the CA, complete the existing identity and authorization process, allow the CA to verify the CSR signature as proof of possession, and receive the certificate. ML-DSA introduces no additional subscriber-facing step and does not require ML-KEM.

The statement that the existing process will continue unchanged is accurate only at the level of this administrative sequence. It is technically inaccurate if it implies that existing RSA key-generation commands, CSR software, token middleware, CA parsers, signature validators, field sizes and certificate profiles require no modification. They require modification and validation for ML-DSA. Applicant authentication and proof of possession remain separate mandatory controls: identity verification binds the applicant to the request, while verification of the RSA or ML-DSA CSR signature proves control of the corresponding private signing key.

7. Impact assessment of proposed changes on performance of systems

PQC migration must be measured as a systems-engineering change. ML-DSA, SLH-DSA and ML-KEM have different key, signature and processing characteristics. The impact will occur not only during the cryptographic operation but also in certificate chains, CRLs, OCSP responses, signed documents, databases, network transfer, backups and disaster recovery.

Performance area	Measurements required	Operational decision supported
Cryptographic operations	Key generation, signing, verification, encapsulation, decapsulation and HSM transactions per second.	Algorithm parameters, HSM capacity, number of partitions and service sizing.
Object size	Root, CA and subscriber certificates; chain; CRL; OCSP; CMS; PDF; XML; timestamp and evidence records.	Database fields, repository storage, bandwidth, client limits and archival growth.
CA and trust services	Issuance rate, mass renewal, CRL generation, OCSP/TSA throughput, queue depth and peak response time.	Capacity expansion, rollout schedule and service-level objectives.
Applications	Interactive signing latency, bulk-signing rate, validation time, memory use, mobile and constrained-device behaviour.	User experience, batch-window design and minimum client requirements.
Operations	Key ceremony duration, backup and restore, replication, DR recovery, rollover, incident containment and software distribution.	Operational procedures, staffing, maintenance windows and recovery targets.

7.1 Performance-test conditions

- Measure the existing baseline and each PQ parameter set under the same workload and hardware conditions.
- Test single certificates and realistic chains, including revocation and timestamp evidence.
- Include average, peak and failure conditions; do not rely only on laboratory micro-benchmarks.
- Test network loss, stale status data, HSM failover, repository delay and disaster-recovery operation.
- Record acceptable thresholds before the pilot so that results lead to an objective deployment decision.

8. Migration related aspects

A single migration method should not be imposed on every use of the India PKI. The migration should be separated by cryptographic function and by the period for which evidence must remain verifiable. Document signatures require continuity of legal and evidentiary validation; secure communication requires rapid protection against harvest-now-decrypt-later risk; authentication and device certificates require interoperability; and tender encryption requires recoverable confidentiality without weakening private-key protection.

The recommended approach is a phased, use-case-based migration. Existing RSA- or ECC-signed documents should remain unchanged. Their validation material should be collected and protected through long-term validation and preservation evidence before the relevant algorithms or certificates become unreliable. New secure communication should move first to standardized hybrid key establishment where supported. New document-signing services should retain the presently interoperable RSA or ECC path during the transition and introduce ML-DSA through controlled pilots, dual signatures or separate certificates only after certificate profiles, cryptographic modules, tokens, HSMs, PDF validators and government applications are interoperable.

A post-quantum time-stamp over an existing RSA- or ECC-signed document is relevant only as part of a correctly created preservation chain. Before the new time-stamp is applied, the original signature, certificate path, revocation status and earlier time evidence must be validated and bound into the preserved object. A later PQC time-stamp cannot repair a signature that was already invalid, nor can it establish when an untrusted signature was originally created.

8.1 Scope and system context

The current ecosystem contains several distinct uses of public-key cryptography. They must not be treated as one technical flow merely because RSA can historically support both signature and encryption operations.

Use case	Present function	Primary migration concern	Recommended direction
PDF and other document signatures	Subscriber signs a document using an IT Act recognised certificate and an RSA or ECC private key.	Legal continuity, verifier compatibility and validation after 5 to 10 years or longer.	Preserve existing signatures; establish long-term validation evidence; pilot ML-DSA for new signatures without changing old documents.
eOffice and government applications	The application invokes token, remote-signing or built-in signature validation mechanisms.	Application libraries, certificate profiles, UI, workflow and record-management systems may assume RSA/ECC sizes and identifiers.	Make the application crypto-agile; support parallel pre-quantum and post-quantum validation before changing the production signing profile.
Device and infrastructure certificates	RSA or ECC certificates authenticate devices and secure machine-to-machine communication, including deployments such as IDRBT systems.	Protocol negotiation, constrained devices, firmware, HSM support and certificate size.	Use protocol-specific hybrid or PQC profiles; retain pre-quantum fallback only under explicit downgrade-resistant policy.
Authentication and transaction signing	Certificates support identity verification; application rules	The relying party must distinguish authentication from document-signing permissions and proof.	Use separate key usage and policy controls; migrate authentication according to the protocol

Use case	Present function	Primary migration concern	Recommended direction
	determine transaction authority.		and signing according to the signature format.
Tender encryption	A public key protects a symmetric content-encryption key; the private key is later used for decryption. Some implementations generate an exportable key on a desktop for backup and then import it into a token.	Confidentiality, key recovery, custody, token capability and replacement of RSA key transport by KEM-based key establishment.	Redesign around ML-KEM plus approved symmetric encryption. Generate operational keys in approved hardware; use separately controlled recovery or escrow where legally required.
Application-to-CA and user-authentication channels	TLS or another authenticated secure channel protects enrolment, CSR submission, status queries and administrative access.	Harvest-now-decrypt-later exposure and protection of credentials and enrolment data.	Treat secure communication as a mandatory supporting control; adopt standardized hybrid key establishment as products and protocol profiles become available.
Archived signed records	Stored documents must remain provable after certificate expiry, revocation-data loss or algorithm weakening.	A signature that is valid today may not be independently provable years later without historical validation evidence.	Use PAdES-LT/LTA or an equivalent preservation service and periodically renew evidence before algorithms or certificates weaken.

8.2 Governing technical principles

- Separate signature, key establishment and content encryption. ML-DSA signs and verifies. ML-KEM establishes a shared secret. Symmetric algorithms such as AES-GCM encrypt the document or communication payload.
- Do not force a universal migration mechanism. Direct PQC replacement is suitable only where every participant and validator is ready. Hybrid or dual operation is appropriate where continuity with existing systems is required.
- Do not alter an already signed document merely to make it post-quantum. Preserve the original signed bytes and add external or embedded preservation evidence in a standards-conformant manner.
- Do not treat certificate expiry as destruction of historical validity. Validation must be performed against the time of signing or a proven time, using preserved certificate, policy and revocation evidence.
- Apply cryptographic renewal before an algorithm becomes unreliable. Renewal performed after practical forgery becomes possible may not distinguish an authentic historic signature from a newly forged one.
- Use FIPS-approved algorithms only through implementations operating within the applicable validated cryptographic-module boundary when FIPS compliance is claimed. Algorithm standardization alone does not establish product validation.

8.3 Migration models

Model	Meaning	Benefits	Risks and conditions	Recommended use
Direct PQC	Replace the pre-quantum algorithm with ML-DSA or ML-KEM.	Clean final state and no continuing dependence on the replaced primitive.	Requires end-to-end readiness of certificates, HSMs, tokens, APIs, applications and validators.	New closed systems or coordinated ecosystems after interoperability and compliance testing.
Hybrid key establishment	Combine a shared secret established using a pre-quantum algorithm with an ML-KEM shared secret under a defined protocol combiner.	Protects confidentiality if at least one component remains secure, subject to the construction.	Must follow a standardized protocol; requires downgrade protection and larger handshakes.	TLS, VPN and high-value communication during transition.
Dual signatures	Attach separate signatures using pre-quantum algorithms and ML-DSA to the same content or transaction.	Pre-quantum validators continue to operate while upgraded validators obtain post-quantum protection.	Policy must state whether both signatures or either signature is required; size and validation complexity increase.	Controlled pilots, software signing, selected records and applications supporting multiple signatures.
Separate pre-quantum and post-quantum certificates	Maintain distinct key pairs and certificates for each algorithm.	Avoids a proprietary composite certificate and allows independent lifecycle management.	Two enrolment, renewal, revocation and discovery paths must be managed.	Existing X.509 ecosystems during staged deployment.
Preservation renewal	Validate and bind the document, signature, certificates, revocation evidence and time evidence under a newer time-stamp or evidence record.	Protects historical proof without requiring the original signer to sign again.	Must be completed while the existing evidence is still trustworthy and repeated when necessary.	All records requiring validation beyond the safe life of their original algorithms or evidence sources.

The transition mechanism should suit each use. Hybrid key establishment protects confidentiality. Dual signatures provide assurance through both pre-quantum and post-quantum signatures. Separate certificates allow pre-quantum and post-quantum systems to coexist. Composite constructions should be adopted only after meeting the prescribed approval and testing conditions (Part III, section 4.4).

Migration should follow a phased programme (Part III, section 3.7). Each use case should have clear conditions and dates for ending pre-quantum-only signing. The duration of hybrid key establishment and dual signatures

should be reviewed separately. July 2030 is not a universal deadline for moving entirely to PQC. The transition to PQC should follow use-case-specific readiness, approved timelines and clearly defined exit criteria.

8.3.1 Comparative decision on direct PQC migration and coexistence

Section 8.3 already describes direct PQC, hybrid key establishment, dual signatures, separate pre-quantum and post-quantum certificates, and preservation renewal. The following comparison is therefore limited to the implementation decision between immediate replacement and controlled coexistence; it does not repeat the technical description of those models.

Decision factor	Direct PQC migration	Coexisting pre-quantum and post-quantum mode	Assessment
Meaning	The existing algorithm is replaced by the selected PQC algorithm for new keys, certificates, signatures or key establishment.	Pre-quantum and post-quantum mechanisms operate in parallel under an explicit transition policy. This may use separate certificates, dual signatures or standardized hybrid key establishment, depending on the function.	Coexistence is a transition architecture, not a new certificate algorithm.
Security benefit	Removes continuing dependence on the pre-quantum algorithm for the migrated function and avoids future exposure from newly created pre-quantum objects.	Adds PQC protection while retaining a proven pre-quantum path. A correctly designed hybrid key-establishment scheme can remain secure if at least one component remains secure.	Direct migration gives the cleanest final state; coexistence reduces transition risk.
Interoperability	Requires every issuing, signing, transmitting, receiving, validating and archival component to support the same PQC profile from the migration date.	Allows upgraded and legacy components to operate during a controlled period, subject to clear rules for what must validate.	Coexistence is preferable where several independent organizations or products must interoperate.
Implementation effort	Requires coordinated replacement or upgrade of HSMs, tokens, CA software, libraries, APIs, schemas, applications, validators, trust stores, monitoring and recovery systems.	Permits phased upgrades, but temporarily requires management of two algorithms, key pairs, certificate paths, validation results and operational procedures.	Direct migration concentrates effort at cutover; coexistence distributes effort but creates temporary complexity.
Failure and downgrade risk	A missing or defective PQC dependency can stop the entire service because no pre-quantum path remains.	A poorly defined fallback can allow an attacker or faulty implementation to force the weaker path. Acceptance rules and downgrade protection are mandatory.	Both models require end-to-end negative testing; coexistence additionally requires strict fallback control.
Certificate and key management	One target algorithm and lifecycle remain after migration. Legacy validation	Separate keys and certificates are preferable to an unprofiled composite certificate. Issuance, renewal, revocation, discovery and	Coexistence shall not permit one key or certificate to be used for an unintended function.

Decision factor	Direct PQC migration	Coexisting pre-quantum and post-quantum mode	Assessment
	capability must still be retained for historical records.	audit must identify the algorithm and purpose of each credential.	
Application and data formats	All formats must accept PQC object identifiers, public keys, certificates and larger signatures before cutover.	Formats may carry a pre-quantum object, a post-quantum object or both. Canonicalization, size limits and validation semantics must be defined for each permitted state.	Coexistence is practical only where formats can represent the transition without changing the signed meaning.
Performance and capacity	Avoids duplicate operations after cutover but may increase bandwidth, storage and verification cost because PQC objects are larger.	May temporarily increase signing, verification, handshake, storage and logging loads because both mechanisms are processed.	Capacity testing must cover peak load, HSM throughput, response size, archives and disaster-recovery sites.
Historical validation	Does not remove the obligation to validate RSA- and ECC-signed records created before migration.	Naturally retains pre-quantum validation while post-quantum validation is introduced. Long-term evidence must still be renewed before older protection becomes unreliable.	Neither model justifies bulk re-signing or alteration of historical signed records.
Governance	Requires a formal cutover date, approved PQC profile, complete readiness evidence and a rollback or continuity plan.	Requires approved combinations, acceptance logic, transition dates, downgrade controls, incident rules and objective exit criteria.	Coexistence shall be time-bound and governed; indefinite dual operation is not the target state.
Best fit	New closed systems, single-operator environments, coordinated bilateral deployments and functions for which every dependency is already validated and interoperable.	Large established ecosystems, long-lived signed records, multi-party workflows, public-facing services and applications with uneven supplier readiness.	Selection shall be made by use case and cryptographic function, not by one national cutover rule.

Controlled coexistence is the preferred default for the present India PKI migration because the ecosystem contains many Certifying Authorities, device suppliers, remote-signing services, government applications, financial systems, document formats and relying parties that will not become PQC-ready at the same time. It preserves service continuity and validation of existing RSA and ECC records while PQC capability is introduced and tested. Coexistence shall not mean unrestricted fallback. Each approved profile must state whether both mechanisms must validate, whether either mechanism is sufficient, how an unsupported algorithm is handled, how downgrade is prevented, and when the pre-quantum production path will be retired. Separate key pairs and certificates for pre-quantum and post-quantum algorithms should be used unless a composite or hybrid certificate format has been formally standardized, profiled and independently tested.

Direct PQC migration should be adopted where the complete operational chain is under coordinated control and has passed conformance, interoperability, performance, security, rollover, revocation and disaster-recovery testing. It is also the appropriate target state for new systems that do not require compatibility with legacy participants. The long-term objective is direct PQC operation; coexistence is the controlled route to that objective. Crypto-agility

shall continue after PQC migration and support controlled replacement of algorithms, parameters, credentials and implementations, with interoperability testing and downgrade protection.

8.3.2 Applications suitable for controlled coexistence

Application or service	Why coexistence is suitable	Recommended coexistence method	Condition for direct PQC
PDF and other long-term document signatures	Documents signed today may require verification for many years, and relying applications will be upgraded at different times.	Continue valid RSA or ECC signing during transition; pilot a separate ML-DSA signature or certificate; preserve validation evidence and renew archival protection.	All signers, PDF or CMS producers, validators, trust stores, time-stamping services and archives support the approved ML-DSA profile.
eSign and eOffice	The service spans user authentication, remote key generation, short-lived certificate issuance, application APIs, document containers and long-term validation.	Retain the present user sequence; add ML-DSA through separate certificates and parallel validation; use dual signatures only where the signed format and policy define their meaning.	The eSign provider, CA, HSM, application, validator, TSA and archival system pass the same end-to-end profile.
MCA21 statutory filings	Filings involve many user roles, token and remote-signing paths, multiple signatures, professional certification, inspection and long retention.	Accept existing RSA or ECC filings while adding ML-DSA parsing and validation; pilot mixed-algorithm multi-signature forms with explicit role and acceptance rules.	Every filing route, form container, CA, client utility, verifier, evidence store and public-inspection function supports the approved profile.
Bank-to-bank payment and settlement systems	Participating institutions, HSMs, gateways, message formats and disaster-recovery sites cannot safely change on different assumptions.	Use standardized hybrid key establishment for secure channels; introduce separate ML-DSA system or message-signing certificates with bilateral validation during transition.	All members and central infrastructure complete coordinated cutover testing, including peak performance, key rollover, revocation and recovery.
Secure communication to CA and authentication services	Confidentiality may require early protection against harvest-now-decrypt-later attacks, while certificate authentication remains dependent on deployed clients and trust stores.	Use a standardized hybrid key-establishment profile that combines a pre-quantum algorithm with ML-KEM and provides downgrade protection; retain established certificate authentication until PQ signatures are interoperable.	Client and server implementations, certificate profiles and trust stores support stable PQ authentication and an approved protocol profile.
Tender encryption	Legacy RSA-based tender encryption and recovery arrangements may remain	Use an approved RSA-OAEP path where still required and pilot an ML-KEM or	Publishers, bidders, tokens or HSMs, envelope format,

Application or service	Why coexistence is suitable	Recommended coexistence method	Condition for direct PQC
	operational while new confidentiality protection is introduced.	standardized hybrid envelope with separately governed recovery keys.	recovery process and archival decryption procedure support the same ML-KEM profile.
Device and infrastructure certificates	Devices have long service lives, constrained resources, unequal firmware support and staged field replacement.	Use separate pre-quantum and post-quantum credentials, protocol-specific hybrid operation or a gateway-assisted transition where formally designed.	The entire device population or a defined deployment domain supports PQC credentials, update, revocation and recovery without unsafe fallback.
Software and firmware signing	Older bootloaders and update clients may validate only pre-quantum signatures, while new releases require protection against future forgery.	Attach separate signatures using pre-quantum algorithms and ML-DSA where the package format defines both and policy states whether both must validate.	All supported validators accept ML-DSA and the pre-quantum verification path can be retired without stranding deployed systems.
OCSF, CRL and time-stamping services	PKI status and time evidence must remain consumable by existing relying applications while PQC validators are introduced.	Retain interoperable production signatures and issue ML-DSA test or parallel objects under a controlled profile; archive evidence for both.	CAs, responders, TSAs and relying applications consistently generate and validate the approved PQC objects at required service levels.

An application should enter coexistence only after its transition states are defined and tested. The approved design shall identify the authoritative signature or key-establishment result, required validation outcomes, treatment of unsupported algorithms, downgrade and rollback rules, certificate and revocation sources, audit evidence, performance limits and the condition for ending pre-quantum production. Historical validation capability shall be retained for as long as the corresponding records must remain verifiable.

8.4 Use-case-specific migration recommendations

8.4.1 PDF and other legally recognised document signatures

For documents signed today, continuity and broad validation support are more important than an immediate replacement that current PDF readers, tokens or government applications cannot process. Existing RSA and ECC signature generation should continue only for the approved transition period and with currently permitted parameters. At the same time, the signing ecosystem should introduce crypto-agility and controlled ML-DSA pilots.

Period	Signature production	Validation requirement	Preservation action
Immediate	Continue approved RSA/ECC for production where ML-DSA is not interoperable. Do not weaken current controls to obtain a quick PQC label.	Validate the signature, certificate path, key usage, policy, signing time evidence and revocation status.	For records requiring long retention, upgrade to PAdES-LT/LTA or equivalent evidence packaging without changing signed content.

Period	Signature production	Validation requirement	Preservation action
Transition	Issue separate ML-DSA test certificates and support dual-signature or parallel verification in selected applications.	Clearly display the result of each signature and apply an explicit acceptance rule.	Collect OSCP responses or CRLs, certificates, policies and trusted time evidence at or near signing.
PQC production	Use ML-DSA after national profiles, X.509/CMS/PDF bindings, validated modules and application support are established.	Validators must support ML-DSA algorithm identifiers, parameters, sizes and policy processing.	Continue archival renewal; PQC signatures also require preservation because certificates expire and implementations or algorithms may later change.

During early migration, separate pre-quantum and post-quantum signatures are preferable to a composite signature scheme not defined by an adopted standard or CCA-approved profile, because each signature can be represented and validated independently. The signed content must be identical, and policy must specify whether both signatures are mandatory or whether the pre-quantum signature is temporarily sufficient for legacy verifiers.

8.4.2 eOffice and government application signing

The central change is application readiness rather than a change in the user action. The user may continue to select a document, authenticate to the signing device or service, review the content and approve the signature. Internal components must, however, stop assuming fixed RSA or ECC lengths, identifiers and certificate structures.

Component	Required modification
Document workflow	Permit multiple signature algorithms and multiple signatures without changing business approval semantics.
Crypto interface	Support PQC-capable PKCS #11, remote-signing or other approved interfaces when standardized and implemented; avoid hard-coded RSA operations.
CSR and certificate handling	Accept ML-DSA subject public keys and proof-of-possession signatures once the adopted X.509/PKCS #10 profile supports them.
Validator	Process signatures using pre-quantum algorithms and ML-DSA independently and retain evidence for the applicable validation time.
Database and message fields	Increase limits for public keys, CSRs, certificates, signatures, CRLs, OSCP responses and audit records.
User interface	Identify algorithm and validation status accurately without presenting a PQC pilot signature as universally valid.
Archive integration	Send final signed objects and validation evidence to a preservation service governed by retention class.

8.4.3 Device certificates and secure machine communication

Device authentication and secure communication should be migrated according to the protocol used by the device, not by replacing an algorithm name inside an existing certificate without testing the protocol. For systems presently using RSA or ECC device certificates, inventory firmware, TLS libraries, gateways, HSMs, certificate-management software and remote-update capability.

Hybrid key establishment should receive priority for communication carrying information that must remain confidential for several years. The secrets established using the pre-quantum algorithm and ML-KEM must be combined by the approved protocol construction. Pre-quantum-only fallback must be logged, time-limited and

protected against downgrade. Device authentication may remain pre-quantum temporarily if the deployed TLS or device profile does not yet support PQ signatures, while the key-establishment layer is upgraded separately.

8.4.4 User authentication and transaction signing

Authentication and signing should use separate certificate policy, key usage and relying-party rules even if one device hosts both credentials. ML-DSA can replace RSA or ECC for proof-producing signatures. ML-KEM is not used to authenticate a subscriber and is not placed into a signing CSR as a substitute for ML-DSA. Where a user performs only signing, no ML-KEM key pair or encryption certificate is required.

8.4.5 Tender encryption and key recovery

Tender confidentiality should use hybrid encryption: a random symmetric content key encrypts the tender, while a public-key mechanism protects or establishes that content key. Under PQC, ML-KEM encapsulates a shared secret from which a key-encryption or content-encryption key is derived. The recipient decapsulates with the ML-KEM private key. This is not an RSA private-key decryption operation and therefore requires changes in the application envelope, algorithm identifiers, APIs and token or HSM services.

Step	Current RSA-oriented flow	PQC flow	Required change
Key generation	Generate an RSA encryption key pair, sometimes on a desktop for backup before import to a token.	Generate an ML-KEM key pair in an approved token or HSM, preferably as non-exportable.	The key-generation and custody design must change; desktop generation of an exportable operational private key should not be the default high-assurance model.
Recovery	Restore the backed-up private key when authorized.	Use a separately governed recovery key, escrow service, threshold custody or an application-level rewrapping design supported by policy.	Define encryption-key recovery independently from the live operational key. Subscriber document-signing private keys shall not be backed up; infrastructure signing-key recovery shall follow the applicable CCA and cryptographic-module security policies.
Encryption	Encrypt or wrap the symmetric key using RSA-OAEP.	Encapsulate with the ML-KEM public key, derive a key using the specified KDF, and protect the tender using an approved authenticated symmetric scheme.	Introduce a versioned envelope containing KEM ciphertext, KDF and content-encryption identifiers.
Decryption	Use the RSA private key to recover the symmetric key.	Decapsulate the KEM ciphertext and derive the same symmetric key.	Token middleware and application must expose approved ML-KEM decapsulation and handle failure safely.
Transition	RSA recipient certificate is understood by all bidders.	Publish information on recipients' support for pre-quantum and post-quantum algorithms or use a standardized hybrid envelope.	Avoid proprietary formats; maintain test vectors and reject downgrade or algorithm substitution.

8.4.6 Application-to-CA and user-authentication channels

Although TLS certificates used for communication are not the same as subscriber document-signing certificates, secure communication is an essential dependency of the PKI service. CA enrolment, CSR submission, identity-verification interfaces, administrative portals, OCSP, repository access and remote-signing authorization must be protected against interception, modification and impersonation.

The near-term recommendation is to deploy protocol-standard hybrid key establishment where the server, client and validated cryptographic implementation support it. Server authentication can remain under the established X.509 path during a controlled transition. Direct use of ML-DSA server certificates should follow only after browser, operating-system, TLS-profile and certificate-policy support is stable.

8.5 Treatment of already signed documents

Already signed documents should not be re-signed merely because PQC standards have been issued. Re-signing by an archive operator changes the evidence model and does not recreate the original subscriber act. The preservation system should protect the original document and its existing signature as evidence.

1. Determine the retention class and identify documents requiring proof beyond five years, ten years or permanently.
2. Validate the existing signature while the relevant certificate path, signature algorithm and status services remain trustworthy.
3. Capture the complete certificate chain, applicable certificate policy, OCSP response or CRL, trusted time evidence and validation result.
4. Bind the document, signature and validation material into PAdES-LT/PAdES-LTA or an equivalent evidence-record or preservation container.
5. Apply an archive time-stamp or preservation evidence using the then-approved algorithm and a trusted time source.
6. Monitor algorithms, certificates and evidence sources. Renew the time-stamp or evidence record before the current protection becomes weak or expires.
7. When changing the digest algorithm, perform a hash-tree or evidence renewal that covers the original object and the complete prior evidence chain.

8.6 Relevance of a PQC time-stamp over an RSA- or ECC-signed document

Question	Finding
Does a PQC time-stamp have relevance?	Yes. If created while the RSA/ECC signature and its validation evidence are still trustworthy, it can provide new evidence that the complete preserved object existed before the PQC time-stamp and has not subsequently changed.
Does it convert the original signature into a post-quantum signature?	No. The signer's original signature remains RSA or ECC. The PQC time-stamp adds preservation evidence; it does not change the signer's algorithm or certificate.
Can it repair an invalid or unverifiable signature?	No. A new time-stamp cannot prove that an already invalid signature was once valid. The original signature and status evidence must be verified before renewal.
Is a PQC time-stamp alone sufficient?	No. It must cover the document, original signature, certificate path, revocation information and relevant earlier time evidence, directly or through a defined evidence-record construction.
Can it be applied immediately?	Only when the time-stamp format, algorithm identifiers, TSA certificate profile, HSM implementation, validation software and policy are interoperable and approved. Until then, use current long-term time-stamping and renew to PQC before the current protection becomes unsafe.

Question	Finding
Must every document receive a PQC time-stamp today?	No. Apply preservation according to retention period and risk. Short-lived records may expire before quantum risk is relevant; long-lived and high-value records require early evidence capture and scheduled renewal.

The decisive issue is the time of renewal. If a practical quantum attack could already forge the original signature before the preservation evidence is created, the archive cannot reliably determine whether it is preserving the authentic historic signature or a recent forgery. Long-term records therefore require preventive renewal, not post-compromise repair.

8.7 Recommended migration profile

Use case	Immediate position	Transitional mechanism	Target state	Priority
Long-term PDF and official records	Continue compliant RSA/ECC signing; create LT/LTA evidence for retention classes requiring it.	Dual signatures only in approved pilots; scheduled archival renewal.	ML-DSA signing plus continuing evidence renewal under approved profiles.	Highest for preservation inventory; measured for new signing.
eOffice and government applications	Implement crypto inventory, size testing and algorithm agility.	Parallel validation and separate pre-quantum and post-quantum certificates.	Native ML-DSA generation and validation.	High.
Tender encryption	Retain approved RSA-OAEP where necessary; prohibit weak RSA v1.5 key transport.	Pilot ML-KEM or a standardized hybrid envelope; establish recovery governance.	ML-KEM plus approved KDF and authenticated symmetric encryption.	High for long-confidentiality tenders.
Secure communication to CA and authentication services	Harden current TLS and inventory protocol dependencies.	Standardized hybrid key establishment with downgrade protection.	PQC-capable protocol profile; PQ authentication when mature.	High because of harvest-now-decrypt-later risk.
Device certificates	Assess firmware, field-upgrade and constrained-resource limits.	Gateway-assisted or protocol-specific hybrid deployment where necessary.	PQC-capable device identity and communication profiles.	Risk-based.
OCSP, CRL and TSA services	Retain interoperable signatures; ensure evidence is captured.	Sign test responses and time-stamps with ML-DSA in a controlled environment.	ML-DSA service signatures under approved profiles and validated modules.	High for infrastructure readiness.

8.8 Minimum-effort implementation plan

8.8.1 Phase 1 Preservation and crypto agility

- Do not modify existing signed documents.
- Classify records by required validation period and immediately collect validation evidence for long-lived records.
- Make PDF, eOffice, CA, OCSP, TSA, tender and device applications accept configurable algorithms and larger objects.
- Create separate inventories for signing, authentication, key establishment, encryption and archival evidence.

8.8.2 Phase 2 Controlled parallel operation

- Issue non-production ML-DSA certificates and test CSR, issuance, revocation, OCSP, time-stamping, PDF and application validation flows.
- Pilot standardized hybrid ML-KEM key establishment for sensitive communication and tender-encryption envelopes.
- Use separate pre-quantum and post-quantum certificates or signatures during transition unless a formally adopted composite profile exists.
- Test archival renewal from RSA/ECC evidence to PQC-protected evidence before production deployment.

8.8.3 Phase 3 Production migration and retirement

- Enable ML-DSA for production signatures only after national profiles, relying-party acceptance, module validation and application interoperability are established.
- Enable ML-KEM for key establishment and tender encryption through versioned, standardized envelopes and approved key-recovery controls.
- Set an end date for pre-quantum-only issuance by use case; do not use the same date mechanically for all applications.
- Continue preservation renewal after PQC deployment because no algorithm or certificate remains trustworthy indefinitely.

8.9 Mandatory policy decisions

Decision	Required rule
Acceptance during transition	State whether both pre-quantum and post-quantum signatures must validate, or whether either is sufficient for each application.
Retention classes	Assign preservation controls and renewal frequency according to legal and operational retention requirements.
Proof time	Define the trusted time at which certificate and revocation status is evaluated.
Algorithm profiles	Publish permitted parameter sets, identifiers, certificate extensions, key usages, KDFs, content-encryption algorithms and failure handling.
Private-key custody	Prohibit plaintext export and escrow of signing private keys. Permit protected infrastructure signing-key backup only where expressly allowed by the applicable CCA and cryptographic-module security policies. Stateful-signature private keys shall not be exported or cloned. Define separately controlled recovery for encryption keys where continuity requires it.
Downgrade control	Prevent silent negotiation from PQC or hybrid operation to pre-quantum-only operation.
Module assurance	Require the relevant algorithms and services to operate within appropriately validated cryptographic modules when validation is claimed.
Archival renewal	Require renewal before an algorithm, TSA certificate, trust anchor or validation source ceases to provide reliable evidence.
Legacy validation	Maintain trusted software and historical policy data capable of validating older signatures without allowing obsolete algorithms for new signatures.

8.10 Migration conclusion

The most suitable migration for India is not immediate replacement of every RSA or ECC use. It is a controlled separation of functions. Secure communication and long-confidentiality encryption should receive early ML-KEM or standardized hybrid protection. New document-signing systems should prepare for ML-DSA through crypto-agility, parallel validation and carefully scoped pilots. Existing signed documents should remain intact and be protected through timely long-term validation and evidence renewal.

For records that may require validation after five to ten years, the immediate priority is not to add a post-quantum signature indiscriminately. It is to preserve trustworthy evidence now: the original bytes, signature, full certificate path, applicable policy, revocation status and trusted time. A future PQC time-stamp is useful when it renews this already validated evidence before the pre-quantum protection becomes unreliable. This approach minimizes application disruption while preserving the ability to establish authenticity and integrity over the required retention period.

9. Recommendations

9.1 Establish a national implementation programme

CCA should establish a time-bound programme covering inventory, profiles, test infrastructure, supplier evidence, pilots, regulatory amendments and controlled production introduction. The programme should retain one configuration register for every approved product and version.

9.2 Complete the cryptographic dependency inventory

RCAI, licensed CAs and major relying organisations should identify every dependency on RSA, ECC, key size, OID, certificate format, API, device, middleware, database field and validation rule. Unknown dependencies should be treated as migration risks.

9.3 Adopt a narrow initial technical baseline

ML-DSA should be the primary signature candidate for implementation pilots. ML-KEM should be tested only for key-establishment use cases. SLH-DSA should be evaluated for algorithm diversity and limited high-assurance uses. FN-DSA should not enter the production baseline until the standard and validated implementations are complete.

9.4 Publish India PKI profiles and test vectors

CCA should define exact algorithms, parameter sets, OIDs, pure or pre-hash mode, certificate profiles, CMS/PDF/XML use, CRL/OCSP/TSA treatment, transition policy and failure behaviour. The same reference corpus should be used by all suppliers and laboratories. Maintain a versioned test corpus of test keys, CSRs, certificates and chains, signed containers, status and time-stamp objects and ML-KEM envelopes, with expected results and negative cases. Link to authoritative algorithm vectors, including NIST's ML-DSA and ML-KEM vectors at <https://github.com/usnistgov/ACVP-Server/tree/master/gen-val/json-files>. Mark all disclosed private keys TEST ONLY and prohibit production use. An appendix or maintained repository shall identify the corpus version and source references.

9.5 Create a CCA reference test environment

The environment should include a PQ-capable CA, network or PCIe HSM, software provider, sample subscriber mechanisms, CRL, OCSP, TSA, repositories and relying applications. It should support positive, negative, malformed, cross-vendor, performance, rollover and disaster-recovery tests.

9.6 Require exact-version supplier evidence

A supplier declaration should not be accepted as proof. Evidence should identify model, firmware, software, cryptographic provider, interface, FIPS or other validation scope, supported profile, performance, lifecycle and security-maintenance period.

9.7 Separate functional support from assurance

Implementation of an algorithm, IETF conformance, FIPS validation, CCA acceptance, application interoperability and production approval are different states. Reports and procurement documents should record each state separately.

9.8 Conduct phased pilots

The sequence should be laboratory certificates and signatures, closed cross-vendor pilot, non-critical production pilot, selected trust-service migration and later broad deployment. Each phase should have entry criteria, exit criteria and a tested rollback plan.

9.9 Protect continuity and historical verification

Pre-quantum certificates and signatures must remain verifiable for their legitimate historical use. Long-term records should receive renewed timestamps or evidence protection before the relevant algorithms lose acceptable security.

9.10 Include applications in the approval boundary

A PQ-capable CA and HSM do not create a usable service unless the relying application accepts the certificate and signature. Major PDF, CMS, XML, browser, mobile, banking, government and enterprise applications should be tested before certificate issuance is expanded.

9.11 Establish performance thresholds

CCA and participating organisations should agree measurable thresholds for issuance, HSM throughput, signing, validation, CRL/OCSP/TSA response, file size, storage growth, backup and DR. Capacity upgrades should precede production use.

9.12 Maintain crypto agility

Policies, software and procurement should permit controlled addition, rollover and retirement of algorithms without redesigning the complete system. Proprietary encodings and vendor-specific transition structures should not be adopted as the national baseline.

9.13 Broad implementation sequence

Implementation shall follow the consolidated phased programme in Part III, section 3.7, subject to the technical readiness and acceptance requirements specified in this Part.

References and source basis

[1] Information Technology Act, 2000; Information Technology (Certifying Authorities) Rules, 2000; Digital Signature (End Entity) Rules, 2015.

[2] Controller of Certifying Authorities, Interoperability Guidelines, Identity Verification Guidelines, eSign specifications and applicable cryptographic-device requirements.

[3] NIST FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard; FIPS 204, Module-Lattice-Based Digital Signature Standard; and FIPS 205, Stateless Hash-Based Digital Signature Standard, 13 August 2024.

[4] NIST SP 800-208, Recommendation for Stateful Hash-Based Signature Schemes; NIST SP 800-227, Recommendations for Key-Encapsulation Mechanisms; NIST IR 8547, Transition to Post-Quantum Cryptography Standards; and NIST crypto-agility guidance.

[5] RFC 5280, RFC 5652, RFC 3161, RFC 9708, RFC 9802, RFC 9814, RFC 9881, RFC 9882, RFC 9909, RFC 9935 and RFC 9936.

Additional enrollment reference: RFC 9883, An Attribute for Statement of Possession of a Private Key, sections 1–6: <https://www.rfc-editor.org/rfc/rfc9883.html> . CBOM reference: <https://cyclonedx.org/capabilities/cbom/> . TLS 1.3: RFC 8446, <https://www.rfc-editor.org/rfc/rfc8446.html> .

[6] NIST Cryptographic Module Validation Program validated-modules, Modules in Process and Implementation Under Test records.

[7] OpenSSL 3.5 PQC and FIPS-validation material; Bouncy Castle, Open Quantum Safe, AWS-LC and supplier technical material referenced in the readiness component assessment.

[8] Adobe Acrobat digital-signature standards documentation; Adobe product documentation; iText PQC support material; PDF Association and BFO reports on post-quantum PDF-signature interoperability work.

PART III TECHNO POLICY TRACK

TRACK-3 TECHNO POLICY TRACK

Changes to the standards mentioned in the Act, changes to rules and manner of interoperability when hybrid certificates are in use, as well as protection/migration of already signed documents

Terms of Reference

iv. To identify the required changes in existing rules, standards, and guidelines for enabling PQC-based certificates and signatures, while ensuring interoperability and continuity of services.

v. To recommend safeguard measures for already digitally signed records and suggest phased migration timelines at the user, Certifying Authority, and application levels, keeping in view ecosystem readiness and global standardization developments.

This report examines the legal and technical changes needed to enable post-quantum certificates and signatures, the protection of already signed records, and the arrangements required during migration. The transition should align approved algorithms and procedures with tested implementations, preserve validation evidence for legacy records and maintain continuity of services.

1. Assessment of changes needed to IT Act and Rules

1.1 Accommodating ML-DSA and SLH-DSA in the IT Act

A digital signature links an electronic record to the subscriber's private key and allows another person to check the signature with the corresponding public key. ML-DSA and SLH-DSA perform these functions.

Section 3 requires an asymmetric cryptographic system and a hash function to authenticate an electronic record. ML-DSA and SLH-DSA use a signing key, a verification key and cryptographic hashing. Once the necessary PQC-related rule changes have been made, section 3 can accommodate both algorithms without a further amendment to the Act. The signature must still be verifiable against the electronic record that the subscriber intended to sign.

Section 3A permits a reliable electronic-signature technique specified in the Second Schedule. It concerns the authorised technique and procedure for authentication; it does not prescribe the internal steps like hashing. Section 3A itself therefore needs no amendment to accommodate these algorithms, provided their use follows a technique covered by the Second Schedule and satisfies section 3A's reliability conditions.

The Second Schedule describes authentication, a subscriber's key pair, a Digital Signature Certificate and creation of a digital signature. These procedures refer to the applicable e-authentication guidelines and standards without naming specific algorithms. Their references to "hash" and an "asymmetric crypto system" are also capable of covering the algorithms ML-DSA and SLH-DSA. A separate Schedule entry or modification is not necessarily required where the existing procedure is retained and the referenced requirements permit the chosen PQC algorithm and mode. The present wording can accommodate PQC-based algorithms in eSign, subject to alignment of the referenced requirements.

Provision	Required change for ML-DSA and SLH-DSA
IT Act, Section 3	No Act amendment; amend relevant PQC rules.
IT Act, Section 3A	No amendment; comply with authorised techniques and reliability conditions.
IT Act, Second Schedule	No separate entry or modification where existing procedures cover the chosen algorithm and mode.

1.2 Changes to creation verification and standards provisions

A digital signature allows a person to sign an electronic record with a private key. Another person uses the corresponding public key to check who signed it and whether the record has changed. The question is whether the rules describing these steps also accommodate the signature algorithms in FIPS 204 and FIPS 205.

The Information Technology (Certifying Authorities) Rules, 2000, prescribe a separate hashing and signing sequence. Rules 4 and 5 should be amended to expressly cover creation and verification of signatures on Digital Signature Certificates in accordance with the approved algorithm and its specified mode. Rule 6 should include the approved PQC standards, and the applicable Interoperability Guidelines under Rule 7 should specify the corresponding certificate and CRL profiles.

For subscriber signatures, rules 4 and 5 of the Digital Signature (End Entity) Rules, 2015, should also be amended. Rule 4(1)–(3) prescribes a separate hashing and signing sequence, while rule 5 specifies the corresponding verification process. These provisions should permit creation and verification in accordance with the approved algorithm and its specified mode, while retaining authentication of the intended electronic record and detection of any alteration.

FIPS 204 specifies ML-DSA, and FIPS 205 specifies SLH-DSA. Both provide a pure mode and a pre-hash mode. In pure mode, the message is given to the signature procedure, which performs the processing required by the standard. In pre-hash mode, the message is hashed first, but the standard also specifies how that result must be used. Hashing is therefore present in PQC signatures; the problem is the rules' requirement that it occur as one fixed, separate step before signing.

To accommodate RSA, ECC, ML-DSA and SLH-DSA, rules 4 and 5 of the Digital Signature (End Entity) Rules, 2015, should specify that digital signatures are “to be created and verified in accordance with the approved algorithm and its specified mode.” This wording allows each algorithm to process the record as prescribed by its standard, while retaining verification of the signer's public key and detection of any alteration to the signed record.

A further amendment is needed in rule 7 of the 2015 End Entity Rules. Its standards table names SHA-2 and identifies RSA and ECC technologies, but does not identify ML-DSA or SLH-DSA. The table should state which PQC signature standards, parameter sets and modes are permitted, and specify a separate hash function only where the chosen mode requires one. Existing uses of SHA-2 can remain.

FIPS 203 requires separate treatment. It specifies ML-KEM for establishing a shared secret key. ML-KEM does not sign a record, so it should not be inserted into a rule as a digital signature algorithm.

CCA policies, interoperability guidelines, cryptographic-device requirements and audit criteria, together with CA Certification Practice Statements, should be updated consistently with the revised End Entity Rules. Certificate issuance, subscriber signatures, certificate-chain validation, CRLs, OCSP responses and time-stamping profiles must use compatible algorithm identifiers and verification procedures.

The proposed modifications to the IT (Certifying Authorities) Rules, 2000, are set out in **Annexure 1** and cover digital signature creation and verification on certificates, together with updated cryptographic standards. The Annexure also includes modifications to the Digital Signature (End Entity) Rules, 2015, covering creation and verification of digital signatures, including XML signatures, certificate validation and standards for approved algorithms, their parameters, processing modes and hashing requirements.

1.3 Changes to the e-authentication guidelines and eSign API

In the existing eSign process, the Application Service Provider (ASP) sends a document hash rather than the document to the eSign Service Provider (ESP). This separation should be retained. Sending a document hash does not, by itself, require HashML-DSA or HashSLH-DSA.

For CMS-based eSign, the preferred approach is a detached CMS signature with mandatory signed attributes. The ASP calculates the approved SHA-2 document digest, which is included in the CMS messageDigest attribute. Pure ML-DSA signs the complete DER-encoded SignedAttrs in accordance with RFC 9882. Pure SLH-DSA may be used under an approved secondary profile following RFC 9814. The original document need not be transmitted to the ESP. SignedAttrs shall include content-type and message-digest attributes. The signature input shall use the DER SET OF encoding prescribed by RFC 5652, rather than the context-specific [0] encoding used in the transmitted SignerInfo. The signature context shall be empty.

The e-authentication guidelines and eSign API shall specify whether the ESP constructs deterministic SignedAttrs from the submitted digest and prescribed metadata, or the ASP submits the complete DER-encoded SignedAttrs. Subscriber authentication and consent shall be bound to the exact document digest, signed attributes and authorised transaction. Returned signed attributes shall be checked against the submitted digest and transaction.

The revised API shall identify the algorithm, parameter set, signing mode, content-digest algorithm, signed-input type, context, identifiers, encoding, response format and verification procedure. Digest selection shall match the approved parameter-set profile; the existing SHA-256 input shall not automatically be retained for every configuration.

HashML-DSA and HashSLH-DSA shall be used only where pre-hashing is separately justified and covered by an approved interoperable profile. Such profiles shall specify the exact signed input, permitted pre-hash function and output length, context, domain separation, identifiers and verification procedure. Signing CMS SignedAttrs containing a document digest in pure mode shall not be described as a FIPS pre-hash signature. For CMS signed-attribute profiles, use SHA-256 or stronger with ML-DSA-44, SHA-384 or SHA-512 with ML-DSA-65, and SHA-512 with ML-DSA-87.

1.4 Treatment of FN-DSA and key-establishment certificates

FN-DSA is not expected to require amendments to Sections 3 and 3A. Its coverage under the Second Schedule should be confirmed against the final standard. The e-authentication guidelines and eSign API are likely to require updates specifying how the document hash is used for FN-DSA signing and verification.

Where ML-KEM certificates are required, their purpose, key usage and proof-of-possession procedure should be specified separately. ML-KEM cannot generate a signature for a conventional signing-based certificate request. The procedure should therefore demonstrate possession of the decapsulation key through an approved challenge-response or other suitable mechanism. Where expressly permitted by the applicable Certificate Policy, the RFC 9883 signed statement described in Part II, section 2.3, may be accepted instead of technical proof of possession.

2. Analyse risk to legacy records from quantum computing

2.1 Nature of the risk

Legacy records are electronic documents signed using existing algorithms such as RSA or elliptic-curve cryptography. Many of these records must remain verifiable for years after the signer's certificate expires. A future verifier must still be able to establish that the document was signed by the stated person, remained unchanged and was signed while the certificate was valid.

A sufficiently capable quantum computer could undermine the public-key algorithms used in digital signatures and in the certificates and time stamps supporting them. It could then become possible to forge signatures or challenge the reliability of older validation evidence. This is a future risk; existing signatures are not invalid today. However, replacing the signing algorithm for new documents alone will not protect records that have already been signed.

The transition to post-quantum cryptography must address the following risks to confidentiality, authenticity and secure implementation:

2.2 Harvest Now, Decrypt Later

Attackers may collect encrypted data today and decrypt it later using a capable quantum computer. Sensitive information that must remain confidential for many years should therefore be identified and protected early with quantum-resistant or hybrid encryption.

This is a confidentiality risk affecting encrypted communications and records, including signed records that are also encrypted. A digital signature establishes authenticity and integrity; it does not prevent future decryption. Re-signing a document or adding an archival time stamp therefore does not address Harvest Now, Decrypt Later. The exposure lies in quantum-vulnerable public-key encryption, key establishment or wrapping of the key used to encrypt the data.

Application and record owners should identify the information that must remain confidential for many years, the channels through which it is transmitted, and the public-key mechanisms protecting its encryption keys. Such information should receive priority for migration to approved quantum-resistant or hybrid key-establishment arrangements. For retained encrypted records, migration may require re-encryption or replacement of vulnerable key-wrapping arrangements. A change made now cannot recover confidentiality for ciphertext and vulnerable key material already harvested by an attacker.

ML-KEM under FIPS 203 establishes a shared secret; it is not a digital-signature algorithm. Its use for confidentiality protection requires a separate approved encryption or key-establishment profile. This assessment belongs to the application or data custodian and the relevant sector authority, alongside CCA's certificate and signature migration work.

2.3 Forge Later

A capable quantum computer may break RSA or ECC keys and create false signatures, certificates or time stamps. Long-term records should therefore be protected by preserving validation evidence, applying trusted time stamps and renewing the protection with quantum-resistant algorithms.

2.4 Upgrade Now, Break Later

PQC systems introduced too quickly may contain implementation or security weaknesses. Therefore, migration should use tested standards, validated cryptographic modules and systems that allow algorithms to be replaced when required.

The risk is particularly relevant to long-lived Indian banking records and documents signed through government or commercial applications. It is greatest where the application has not preserved the evidence required to establish the signing time, the complete India PKI certificate chain and the revocation status of the certificates at the relevant time.

A certificate's subsequent expiry does not, by itself, invalidate an earlier signature. However, a time stamp obtained only after the original signature can no longer be trusted cannot establish that the signature was valid when it was created.

2.5 Preservation and renewal of validation evidence

Records should be prioritised according to their required retention period and the consequences if their authenticity is disputed. While the existing cryptography remains reliable, each important record should be validated and the following information preserved:

- Exact signed document and digital signature;
- Complete certificate chain;
- Applicable validation policy;
- Relevant CRL or OCSP evidence;
- Signature-validation result;

- Trusted evidence of signing time; and
- Relevant application audit trail.

These elements should be maintained as one linked package in an appropriate long-term validation or evidence-record format. The package should be protected with a trusted archival time stamp or standards-based evidence record.

At every renewal, the complete package should be verified. Its time stamps and cryptographic protection should be renewed using stronger algorithms before the existing protection ceases to provide adequate assurance. Post-quantum protection may be used for such renewal when an appropriate trusted service becomes available, while retaining the ability to verify the original signature.

This process preserves evidence established while the original signature was trustworthy. It does not convert the original signature into a post-quantum signature or make an invalid signature valid. Confidential records encrypted using quantum-vulnerable public-key algorithms require a separate confidentiality-risk assessment.

Renewal must also address weakening of the hash function used to bind the record and its evidence. Where necessary, the underlying record and evidence should be protected with a suitable stronger hash before the earlier hash ceases to provide adequate assurance. Missing historical evidence should be recorded as a limitation and should not be represented as established proof.

2.6 eSign records and CA evidence

In eSign, the signed document and the record of the signing transaction may be stored in different places. If quantum computing makes the original signature unreliable, it may become difficult to prove who signed the document and when.

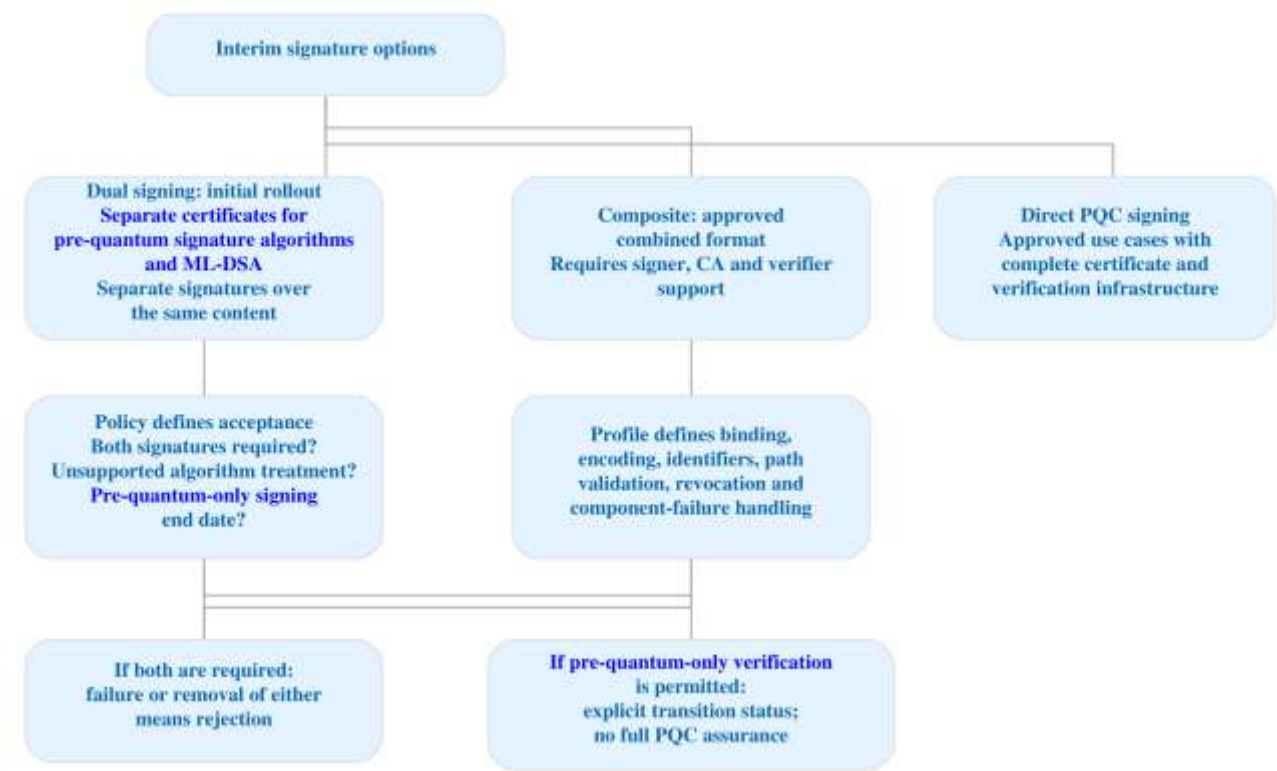
While the signature remains trustworthy, the signed document, certificate details, signing-transaction record and proof of signing time should be linked and preserved together. This evidence should be protected with an archival time stamp and renewed with stronger algorithms before the existing protection becomes weak.

Quantum computing could eventually make RSA and elliptic-curve CA keys vulnerable, allowing forged certificates or revocation information and casting doubt on older certificate chains.

The risk should be managed by protecting CA keys, preserving certificate and revocation evidence for existing records, and migrating certificate issuance and validation to approved post-quantum methods within an appropriate timeframe.

3. Possible approaches to migration including interim measures

3.1 Interim use of pre-quantum and post-quantum signatures



During the transition, existing and post-quantum signatures should be used together where required. For the initial rollout, separate certificates for pre-quantum signature algorithms and ML-DSA and separate signatures over the same intended content provide a practical approach. Each can be issued and tested independently. This is dual signing, rather than a composite certificate or composite signature.

The applicable policy must specify whether both signatures are required to verify, how an unsupported algorithm is treated and when pre-quantum-only signing ends. Where both signatures are required, failure or removal of either component should prevent acceptance under that policy. Pre-quantum-only verification during a defined transition should be identified explicitly and should not be presented as full PQC assurance.

A composite signature or certificate may be considered as a future migration option only after the combined format is formally standardised, adopted through a CCA profile and independently tested for interoperability. The profile shall define component binding, identifiers, encoding, path validation, revocation and treatment of component failure. Separate pre-quantum and post-quantum certificates and signatures shall remain the preferred initial approach. Direct PQC signing may begin for approved use cases once the complete certificate and verification infrastructure is ready.

The following table gives indicative migration recommendations, subject to approved profiles, application compatibility and successful testing.

Sector / application	Recommended migration method
Government documents, eSign and contracts	Dual signing during transition
Banking and interbank messages	Dual signing during transition; composite only after formal standardisation, CCA profiling and independent interoperability testing
eProcurement	Dual signing for bids and approvals; separate PQC key-establishment migration for bid confidentiality

Sector / application	Recommended migration method
Software and firmware updates	Dual signing where existing devices support it; direct PQC for new PQC-ready devices
Connected devices and industrial systems	Separate pre-quantum and post-quantum credentials during transition; direct PQC for fully ready deployments
CA, CRL, OCSP and time-stamp services	Parallel pre-quantum and post-quantum trust services, followed by direct PQC
Websites, APIs and VPNs	Approved hybrid key establishment combining a pre-quantum algorithm with ML-KEM during transition; migrate certificate authentication separately
User, device and service authentication	Parallel pre-quantum and post-quantum credentials during transition; direct PQC when all participants are ready
New internal applications	Direct PQC where all endpoints and dependencies are ready

Composite certificates and signatures are future options, subject to formal standardization, CCA profiling and independent interoperability testing; draft constructions shall not form part of the initial production profile.

Dual and composite signing protect authenticity; hybrid key establishment protects confidentiality. Website and VPN migration must address both functions

3.2 Algorithm agility and ecosystem interoperability

The situation, with regard to what algorithms will eventually be used, is not completely settled. Churn is expected, and agility is needed in public key and signature algorithms (ML-DSA/SLH-DSA), certificate structure and content, requirements for certificate transparency and logging, and revocation. Cryptographic settings need to be configurable. Maintain an algorithm-failure response procedure covering affected services and records, restrictions on issuance and signing, approved alternative algorithms, rekeying, certificate and trust-anchor replacement, and preservation of reliable evidence. Test recovery and prohibit fallback that misrepresents the achieved assurance.

An internal testbed should be established to generate and test ML-DSA Certificate Signing Requests, including PQC test roots, with SLH-DSA as a secondary algorithm where appropriate.

For Web PKI, developments in Merkle Tree Certificates and ML-DSA X.509 certificates should be monitored and evaluated separately against standards, browser and library support, TLS and ACME compatibility, and applicable root-program requirements. Adoption of these approaches shall not be a prerequisite for migration of Indian document and transaction signatures.

Outside the Web PKI ecosystem, where digital signatures are used for transactions between banks, MCA21, Income Tax, GST and similar applications, the initial priority should be ML-DSA X.509 certificates. This is particularly relevant to CCA's document and transaction signature framework.

For those, the end applications, for example digital records of submitted e-Filings, need to prepare for the sizes of these signatures to go up significantly.

In this case, testing and piloting should focus on document signing and verification, whether that happens individually or in bulk.

The profile must distinguish the subscriber public-key algorithm from the CA certificate-signature algorithm. Migration should cover the trust chain, CRLs, OCSP responses and time stamps, as well as subscriber signatures. A PQC subscriber key supported only by a quantum-vulnerable certificate chain does not provide complete quantum-resistant trust-path assurance.

3.3 Actions during the interim period

Approved pre-quantum services should continue while regulatory changes and production readiness are completed. In parallel, CAs and application owners should inventory cryptographic dependencies, identify long-lived records,

preserve validation evidence and prepare systems to support approved PQC profiles. Existing long-term validation, time-stamping and archival arrangements should be reviewed for evidence renewal and algorithm replacement.

Experimental PQC equipment may be used in isolated laboratory environments with test keys and test data. Such use should be distinguished from production acceptance. Pilots should test issuance, signing, verification, revocation, time-stamping, larger object sizes, concurrency, failure handling and recovery. The user interface should clearly state the validation result and the assurance available during transition.

3.4 Risk in acceptance of FIPS-claimed PQC HSMs and tokens

HSM vendors make various claims about the post-quantum cryptography (PQC) capabilities of their products. These claims can make it difficult for an organisation, to determine whether an HSM meets its requirements. The Criteria for Acceptance of Post Quantum Cryptography Based Hardware Security Modules sets out an evidence-based assessment of an HSM's validation status, configuration and PQC operations. The audit can be carried out by an auditor with the competence to assess the applicable cryptographic module and workflow.

Products are frequently described as “PQC-ready,” “PQC-compliant,” “NIST-certified” or “FIPS-certified” when only the cryptographic algorithm has been tested, the module is still undergoing validation, or PQC functions are provided through software outside the validated cryptographic boundary. The technically correct description is a FIPS 140-3 validated cryptographic module. An algorithm certificate, entropy certificate, benchmark, demonstration or modules-in-process entry does not establish that the complete HSM or token is FIPS 140-3 validated.

Such claims do not prove that the supplied product securely generates, stores, uses, protects and zeroises post-quantum private keys. Acceptance shall therefore require three cumulative forms of evidence:

Official validation: An active NIST CMVP certificate covering the exact cryptographic module and deployed hardware, firmware, operating environment and configuration.

Documentary evidence: The applicable Security Policy, ACVP/CAVP algorithm certificates, entropy-source evidence and configuration guidance.

Operational evidence: Witnessed testing of approved-mode operation, key generation, signing and verification for signature services, encapsulation and decapsulation for key-establishment services, access control, applicable non-exportability requirements, zeroisation and the complete intended workflow.

The first two requirements establish the FIPS validation position. Witnessed testing is an additional acceptance control required to confirm that the validated module has been configured and integrated correctly for DSC, CA, eSign, time-stamping or OCSP use.

3.5 Checks for the supplied configuration and intended service

The following checks shall be conducted even when the product is represented as FIPS validated:

Validated configuration: Verify that the CMVP certificate and Security Policy cover the exact cryptographic module, hardware revision, firmware version, operating environment and deployed configuration. Where the validated module is embedded in a larger commercial product, establish that the supplied product uses that module without changes outside the permitted configuration.

Algorithm and service coverage: Confirm that every PQC algorithm, parameter set, mode and operation required for the intended deployment is covered by the exact module's validation, Security Policy and supporting ACVP/CAVP records. This includes ML-DSA and, where used, SLH-DSA key generation and signing, and ML-KEM key generation, encapsulation and decapsulation, as applicable to the module's role. Verification operations shall also be checked where provided by the module. An algorithm certificate alone is insufficient

Approved-service indication: Verify approved-mode or approved-service operation using the visible, programmatic or status-output mechanism specified in the Security Policy. Key generation and signing shall not silently fall back to non-approved software or an unvalidated implementation.

ML-DSA parameters and modes: Specify the permitted ML-DSA parameter set and whether Pure ML-DSA, HashML-DSA or both are allowed. The permitted configuration shall correspond to the validated services. Malformed parameters, unauthorised modes and downgrade attempts shall be rejected.

Key generation boundary: For DSC and CA applications requiring onboard key generation, confirm that the secret seed, private-key construction and associated secret values are generated and processed within the validated cryptographic boundary. Any external entropy arrangement or key-import facility shall be permitted only when expressly covered by the Security Policy and applicable CMVP requirements.

Entropy and DRBG: Verify that the entropy source, conditioning functions and deterministic random bit generator used for ML-DSA key generation are those covered by the validated design and applicable NIST SP 800-90 requirements. For DSC key generation, caller-controlled, repeated or unapproved externally supplied seeds shall not be permitted.

Private-key protection: Require DSC, CA, eSign, time-stamping and OCSP private keys to be created as sensitive objects and protected against unauthorised extraction. Where the applicable operational policy requires non-exportability, test attempted export through PKCS #11, middleware, backup, administrative, diagnostic and maintenance interfaces. FIPS validation alone shall not be assumed to prohibit every form of protected key export.

Deletion and zeroisation: For eSign and other one-time keys, verify zeroisation of the secret seed, expanded private key, cached copies, session objects and temporary secret values after successful and failed transactions. Deleting only the object handle, metadata or database entry does not demonstrate zeroisation. The test shall use the zeroisation method documented in the Security Policy.

Firmware integrity: Verify the running firmware version and confirm that it forms part of the validated configuration. Firmware integrity checks and authenticated update controls shall operate as documented. Secure boot, anti-rollback protection and firmware-hash verification shall be required as additional controls where appropriate to the product architecture and deployment risk.

Cryptographic interface: Test the supported PKCS #11 or other cryptographic interface using the OEM-approved middleware. Testing shall cover mechanism identifiers, parameter mapping, key attributes, authentication, signing modes, session isolation, access control, concurrency and error handling. Interface interoperability and application correctness are additional acceptance controls and are not established by CMVP validation alone.

Side-channel and fault resistance: Review the Security Policy, CMVP certificate and available independent assessment reports to determine the actual coverage of timing, power, electromagnetic, fault-injection and other non-invasive attacks. FIPS Level 3 or an algorithm certificate shall not be treated as proof of resistance to every such attack.

Debug and manufacturing interfaces: Verify that JTAG, scan, trace, test and manufacturing interfaces are disabled, removed or cryptographically controlled in the production lifecycle state. Where such interfaces fall within the validated boundary, their treatment shall correspond to the Security Policy.

Zeroisation events: Verify the module's documented zeroisation service and every automatic zeroisation event claimed in the Security Policy. Separately test the behaviour following key deletion, session closure, reset, authentication failure, error conditions, power interruption and tamper detection. Automatic zeroisation on every listed event is not implied by FIPS validation. Any additional event-based zeroisation requirement shall be specified for the deployment and tested separately.

Middleware integrity: Record the approved middleware, driver or DLL version and its SHA-256 hash. Establish whether it is inside or outside the validated cryptographic boundary. Any change affecting a component within the validated boundary may invalidate the validated configuration; changes outside the boundary shall undergo controlled security and interoperability assessment.

Supply-chain integrity: Verify OEM provenance, device model and serial number, hardware and firmware identity, tamper evidence, authorised distribution and secure delivery. These are procurement and deployment controls and do not replace CMVP validation.

End-to-end testing: Conduct witnessed testing of the complete intended workflow, including key generation, certificate-request generation and proof of possession where applicable, certificate loading, access control, failed transactions, concurrent operation and key deletion or zeroisation. Test signing and verification where the product provides signature services, and encapsulation or decapsulation where it provides ML-KEM key-establishment services. Performance claims establish capacity only; they do not establish secure or FIPS-approved operation.

3.6 Production acceptance laboratory use and rejection

A PQC HSM or token shall be accepted for production only when the exact deployed module and configuration are covered by an active FIPS 140-3 validation; every required PQC algorithm and operation is an approved service; approved-service operation is independently verifiable; key generation, entropy use, storage and zeroisation conform to the validated design; applicable private-key protection requirements are met; firmware and integration integrity are established; and the complete intended workflow passes witnessed testing. Coverage shall include ML-DSA, SLH-DSA and ML-KEM operations wherever required by the intended deployment.

A product having only an algorithm or entropy certificate, a pending CMVP submission, a modules-in-process entry, or experimental PQC support outside the validated configuration may be used only for controlled laboratory and interoperability testing. It shall not generate, store or protect production subscriber, CA, eSign, time-stamping or OCSP private keys.

A product shall be rejected where the CMVP certificate covers a different module, hardware revision, firmware version or operating environment; the certificate is inactive or otherwise inapplicable; any required PQC algorithm, parameter set, mode or operation is absent from the module's approved-service coverage; approved operation cannot be demonstrated; required private-key protection or zeroisation cannot be established; or the vendor does not provide the Security Policy, configuration details and supporting test evidence.

Brochures, performance figures, demonstrations, customer references, algorithm certificates and vendor declarations are not proof of FIPS 140-3 module validation. Acceptance shall be based on the official NIST CMVP validated-modules records, the applicable FIPS 140-3 Security Policy, supporting algorithm and entropy evidence, and witnessed testing of the exact supplied configuration.

3.6.1 Indian certification and acceptance of cryptographic modules

The Committee recommends establishment of an Indian cryptographic-module testing and certification mechanism under MeitY, in coordination with BIS, STQC and the relevant accreditation bodies. The mechanism should be based on applicable Indian Standards aligned with ISO/IEC 19790 for cryptographic-module security requirements and ISO/IEC 24759 for testing requirements, supplemented by CCA-approved PQC algorithms, security levels and service-specific requirements.

The mechanism should provide for testing by laboratories accredited for the relevant scope, independent certification by a designated Indian authority, and publication of a verifiable register of certified modules. Certification shall identify the exact hardware, firmware, cryptographic boundary, operating configuration, security level, approved algorithms and operations. It should include procedures for vulnerability reporting, reassessment following changes, renewal, suspension and withdrawal.

Once the mechanism is operational and recognised through the applicable rules and CCA guidelines, certification issued under it shall constitute an accepted route for HSM and token deployment within the CCA framework, alongside the recognised FIPS 140-3 validation route. Indian-certified modules meeting the prescribed requirements shall be eligible for production acceptance without requiring a separate FIPS certificate solely for domestic acceptance. Production approval shall additionally require the documentary, configuration, security and witnessed workflow checks specified in Sections 3.4–3.6.

CCA should publish the recognition criteria, required assurance levels, transition arrangements and acceptance procedures. This would establish a domestic certification pathway for Indian products while maintaining the security assurance required for CA, DSC, eSign, time-stamping, OCSP and applicable key-establishment services.

3.7 Phased implementation and responsibilities

Establishment of a CCA-led National PQC PKI Interoperability Testbed is proposed to commence in January 2027. The periods below are indicative planning windows. Progression shall depend on satisfaction of the applicable exit criteria, rather than the passage of time. These dates do not predict when a quantum computer will become capable of breaking existing cryptography. Preservation of legacy validation evidence and assessment of Harvest Now, Decrypt Later exposure shall begin immediately.

Phase 1—Testbed establishment and profiling: January–June 2027. CCA shall establish the shared testbed, publish HSM acceptance criteria, coordinate initial ML-DSA and ML-KEM profiles, and publish the common test corpus and evidence template. CAs, ESPs, vendors and application owners shall complete cryptographic and dependency inventories, identify dependency owners, and connect experimental HSMs, CA test PKIs, token simulators, reference libraries, validation tools and application test instances. Record custodians shall identify long-lived signed and confidential records and preserve available validation evidence. This phase shall conclude when the testbed is operational, baseline profiles are approved, participating interfaces are connected, and common test and evidence requirements are published.

Phase 2—Integrated laboratory testing: July 2027–June 2028. CAs, ESPs, vendors and application owners shall test certificate issuance, proof-of-possession, signing, key establishment, path validation, CRLs, OCSP, time stamps, malformed inputs, failure handling, performance, backup, restoration and disaster recovery using exact recorded configurations. Each participating configuration's achieved readiness level shall be recorded, together with material defects, validation gaps, performance limits and unresolved dependencies; only configurations that pass the required laboratory tests shall proceed to cross-vendor pilots.

Phase 3—Cross-vendor pilots: July 2028–June 2029. Participants shall conduct bidirectional interoperability tests across HSMs, CA platforms, libraries, token simulators, physical devices and selected applications. Testing shall cover CMS, PDF, XML, rollover, high availability and disaster recovery. This phase shall conclude when at least two independent implementations complete the common test suite and selected configurations demonstrate pilot readiness, with documented limitations and tested rollback arrangements.

Phase 4—Controlled production: July 2029–June 2030. CAs, ESPs and application owners shall deploy approved exact-version configurations in bounded, non-critical services only after those configurations meet the production acceptance conditions and receive production-readiness approval. Entry requires effective legal and technical amendments, approved profiles, applicable cryptographic-module validation, and successful security, interoperability, performance, continuity and complete workflow testing. Monitoring, incident procedures, rollback and controlled coexistence of pre-quantum and post-quantum mechanisms shall be operational. Evidence collected during controlled production shall confirm continued readiness and support wider deployment.

Phase 5—Progressive migration: July 2030 onward. CCA, CAs, ESPs and application owners shall extend approved ML-DSA and ML-KEM profiles to additional trust services and applications according to risk and demonstrated readiness. Each production cohort shall publish its date for ending pre-quantum-only signing and maintain arrangements for verification of earlier records. Defined migration coverage shall be achieved while preserving long-term validation. Retirement of legacy mechanisms shall depend on ecosystem-wide readiness evidence.

CCA shall coordinate legal and guideline changes, profiles, testing and readiness reviews. CAs and ESPs shall upgrade issuance and signing services, protect private keys, preserve certificate and transaction evidence, and maintain access to historical revocation information. Vendors shall provide configuration details, interface support, validation evidence and defect resolution. Application owners shall implement compatible signing, key establishment and verification, define interim acceptance policies, and preserve validation evidence.

Record owners and custodians shall classify holdings by retention period, confidentiality requirements and consequence of disputed authenticity, preserve linked evidence packages, and renew their protection as required. Users shall migrate to approved certificates and devices as their applications become ready. Retirement of pre-quantum signing shall be distinguished from continued verification of records already signed using RSA or ECC.

4. Recommendations

The following recommendations identify the changes and deliverables proposed for adoption by the Committee. The implementation timetable follows the indicative dates in section 3.7.

4.1 Retain Sections 3 and 3A and the existing eSign Schedule procedure

The Committee recommends no amendment to Sections 3 and 3A of the Information Technology Act for accommodating ML-DSA and SLH-DSA. Both algorithms use a signing key, a verification key and cryptographic hashing. Their introduction should be implemented through the applicable Rules, standards and guidelines. No separate Second Schedule entry is recommended merely to name these algorithms where the existing notified authentication procedure is retained and its referenced requirements permit the approved PQC algorithm and mode.

4.2 Amend the operative signature creation and verification Rules

The Committee recommends amendment of rules 4 and 5 of the Digital Signature (End Entity) Rules, 2015, to replace the fixed separate hashing and signing sequence with the requirement that digital signatures are “to be created and verified in accordance with the approved algorithm and its specified mode.” The revised provisions must require authentication of the intended electronic record, signature verification using the corresponding public key, and detection of changes to the record.

The Committee recommends amendment of the standards table in rule 7 to include ML-DSA under FIPS 204 and SLH-DSA under FIPS 205, together with the permitted parameter sets and signing modes. A separate hash function should be specified where the approved mode requires one. Existing approved RSA, ECC and SHA-2 use should continue during the defined transition. ML-KEM under FIPS 203 should be covered separately as a key-establishment mechanism and should not be listed as a digital-signature algorithm.

4.3 Publish a PQC eSign API and matching verification profile

The Committee recommends revision of the CCA e-authentication guidelines and eSign API to adopt detached CMS signatures with mandatory signed attributes as the preferred initial PQC eSign model. The ASP shall calculate the approved SHA-2 document digest, and pure ML-DSA shall sign the complete DER-encoded SignedAttrs containing that digest. Pure SLH-DSA may be supported through an approved secondary profile. The document shall remain with the ASP.

The API shall define responsibility for constructing SignedAttrs and specify the algorithm, parameter set, mode, digest, signed-input type, context, identifiers, encoding and verification procedure. Authentication and consent shall remain bound to the exact digest, signed attributes and transaction. Unknown, ambiguous or inconsistent combinations and silent fallback shall be rejected.

HashML-DSA and HashSLH-DSA shall be permitted only through separately justified and approved profiles. Reference test vectors shall demonstrate matching ASP, ESP and verifier results, including rejection of altered digests, signed attributes, identifiers, parameters and modes. FN-DSA shall be considered separately after its final standard and required interfaces have been assessed.

4.4 Adopt separate certificates and dual signing for the initial transition

The Committee recommends separate approved certificates for pre-quantum signature algorithms and ML-DSA, with separate signatures over the same intended content, as the initial approach for applications requiring both during migration. For a transaction claiming assurance requiring successful validation of both pre-quantum and post-quantum signatures, both signatures and their certificate paths must validate; removal, failure or unsupported processing of either component must prevent that assurance claim. A verifier processing only the pre-quantum signature may report pre-quantum assurance during the permitted transition, but must not report PQC protection.

Composite certificates and signatures shall enter production only after formal standardization, adoption through a CCA profile and independent interoperability testing, with defined component binding, identifiers, encoding, path validation, revocation and failure handling.

4.5 Preserve legacy records as linked evidence packages

The Committee recommends preservation of each priority legacy record with the exact signed document and signature, complete certificate chain, applicable validation policy, relevant CRL or OCSP evidence, signature-validation result, trusted evidence of signing time and application audit trail. The package should receive a trusted archival time stamp or standards-based evidence record while the original signature and evidence remain trustworthy. For eSign, the ESP's transaction evidence must be linked to the signed document held by the application or record owner.

Record custodians should complete classification of holdings, using required retention period, consequences of disputed authenticity and availability of validation evidence. Permanent records and records with long retention periods or high consequences should receive first priority. Each protected package should have an assigned custodian and a recorded renewal trigger. Renewal must protect the complete package with stronger algorithms before its existing signature, time-stamp or hash protection becomes inadequate. Missing historical evidence must be recorded; a later time stamp must not be represented as proof that an already untrustworthy signature was valid when created.

4.6 Address Harvest Now, Decrypt Later separately from signature preservation

The Committee recommends that application and data owners identify, information requiring long-term confidentiality and the quantum-vulnerable public-key mechanisms protecting its transmission, storage or encryption keys. These systems should receive first priority for approved quantum-resistant or hybrid key establishment and for replacement of vulnerable key-wrapping arrangements. This recommendation applies also to digitally signed records that are encrypted.

Confidentiality migration should be undertaken by the application or data custodian under the applicable sector requirements. Re-signing, evidence preservation and archival time-stamping must not be accepted as substitutes for confidentiality protection. The assessment must record any residual exposure from ciphertext already harvested before migration.

4.7 Publish and apply the HSM acceptance criteria

The Committee recommends that CCA publish the Criteria for Acceptance of Post Quantum Cryptography Based Hardware Security Modules on its website to assist organisations in assessing PQC-based HSMs during migration. Production acceptance should require all three forms of evidence: official module validation, documentary evidence and witnessed operational testing of the exact supplied configuration. An assessment by a competent auditor should record the module certificate, hardware and firmware identity, Security Policy, approved services, configuration and workflow test results.

A product having only an algorithm or entropy certificate, a pending CMVP submission, a modules-in-process entry or experimental PQC support outside the validated configuration should be confined to controlled laboratory and interoperability testing. It must not generate, store or protect production subscriber, CA, eSign, time-stamping

or OCSP private keys. A supplied configuration that does not meet the acceptance conditions in sections 3.4 to 3.6 should be rejected for production.

4.8 Establish the India PKI testbed and publish interoperable profiles

The Committee recommends establishment of an internal CCA testbed for ML-DSA certificate requests, PQC test roots and certificate chains, with SLH-DSA as a secondary algorithm. Initial pilots should cover document and transaction signing in applications, including individual and bulk signing. Test completion should demonstrate cross-vendor issuance, signing and verification, CRLs, OCSP, time stamps, evidence renewal, larger object sizes, concurrency, failed transactions and recovery.

CCA should publish matching certificate, signature, revocation and time-stamping profiles before production approval. The profiles must distinguish the subscriber public-key algorithm from the CA certificate-signature algorithm and cover the complete trust path. A PQC subscriber certificate supported only by a quantum-vulnerable chain must not be reported as providing complete quantum-resistant trust-path assurance. Web PKI deployments should follow the relevant browser, root-program and protocol requirements as those standards develop.

4.9 Adopt a phased programme with explicit production entry conditions

The Committee recommends adoption of the phased programme in section 3.7. Its planning windows are indicative; progression shall depend on satisfaction of the applicable exit criteria.

CAs shall demonstrate compliant issuance, private-key protection and certificate-status services. ESPs shall demonstrate signing through approved CMS signed attributes and linked evidence. Application owners shall demonstrate correct signing, key establishment where applicable, verification and archival handling before users migrate to production PQC certificates and devices.

Controlled production shall use approved exact-version configurations that have met the production acceptance conditions and received production-readiness approval before live deployment. Pilot readiness permits controlled pilots and does not authorise production use. Entry requires effective legal changes, approved CCA profiles, compliance with cryptographic-module acceptance and validation requirements, and successful complete workflow, security, interoperability, performance and continuity tests. Monitoring, incident procedures and tested rollback shall be operational. Subsequent operational evidence shall confirm continued readiness and support wider deployment

Where a required condition remains unmet, the affected cohort shall remain in testing and retain approved pre-quantum services. CCA shall review progress at six-month intervals and publish revised planning milestones for affected cohorts. Preservation of legacy validation evidence and assessment of Harvest Now, Decrypt Later exposure shall begin immediately and continue throughout the programme. Each production cohort shall publish its date for ending pre-quantum-only signing and maintain continued verification of earlier records.

4.10 Retain historical verification when pre-quantum signing is retired

The Committee recommends that CCA publish a use-case-specific retirement date for pre-quantum-only issuance and signing after the relevant production cohort is ready. CAs should retain access to historical certificate and revocation evidence, and application owners should retain the capability to verify previously signed records and their renewed evidence packages. Retirement of pre-quantum signing must not remove historical verification services or change the original signed record. Users should migrate as their applications are approved, without being required to re-sign all earlier records.

References

Information Technology Act, 2000, Sections 3 and 3A and Second Schedule; Digital Signature (End Entity) Rules, 2015, G.S.R. 660(E); and Information Technology (Certifying Authorities) Amendment Rules, 2015, G.S.R. 662(E), dated 25 August 2015. Official notifications: https://cca.gov.in/eSign_gazette_notification.html and <https://cca.gov.in/sites/files/pdf/ACT/GSR660.pdf>

CCA e-authentication guidelines and eSign API specifications: <https://cca.gov.in/sites/files/pdf/esign/CCA-EAUTH.pdf> and <https://cca.gov.in/eSignAPI.html>

NIST FIPS 203, 204 and 205, and signature algorithm identifiers: <https://csrc.nist.gov/pubs/fips/203/final> ; <https://csrc.nist.gov/pubs/fips/204/final> ; <https://csrc.nist.gov/pubs/fips/205/final> ; <https://csrc.nist.gov/projects/computer-security-objects-register/algorithm-registration>

NIST Cryptographic Module Validation Program: <https://csrc.nist.gov/projects/cryptographic-module-validation-program>

RFC 4998, Evidence Record Syntax: <https://www.rfc-editor.org/rfc/rfc4998> . NIST, What Is Post-Quantum Cryptography: <https://www.nist.gov/cybersecurity-and-privacy/what-post-quantum-cryptography>

MODIFICATION TO RULES

Deletions are struck out; additions are italicised. The proposed CA Rules provisions address certificates. The End Entity Rules address subscriber signatures and certificate validation. Algorithm-neutral processing applies to RSA, ECC, ML-DSA and SLH-DSA; the standards specify the permitted algorithms and modes.

Information Technology (Certifying Authorities) Rules, 2000

4. Creation of Digital Signature on a Digital Signature Certificate.—~~To sign an electronic record or any other item of information~~ *To sign a Digital Signature Certificate, the signer shall first apply the hash function in the signer's software; the hash function shall compute a hash result of standard length which is unique (for all practical purposes) to the electronic record; the signer's software transforming the hash result into a Digital Signature using signer's private key the signer shall create the Digital Signature using the signer's private key in accordance with the signature algorithm specified in these rules, using the mode and hashing requirements prescribed by the applicable standard and the applicable profiles; the resulting Digital Signature shall be unique to both electronic record Digital Signature Certificate and private key used to create it; the Digital Signature and the digital signature Certificate attached to its electronic record shall be stored or transmitted along with its electronic record the Digital Signature shall be stored or transmitted with the signed Digital Signature Certificate.*

5. Verification of Digital Signature on a Digital Signature Certificate.—The verification of a Digital Signature shall be accomplished ~~by computing a new hash result of the original electronic record by means of the hash function used to create a Digital Signature and by using the public key and the new hash result using the public key in accordance with the signature algorithm specified in these rules, using the mode and hashing requirements prescribed by the applicable standard and the applicable profiles, the verifier shall check—~~
 (i) if the Digital Signature was created using the corresponding private key; and
 (ii) ~~ii the newly computed hash result matches the original result which was transformed into Digital Signature during the signing process the Digital Signature Certificate has not been altered.~~ The verification software will confirm the Digital Signature as verified if—
 (a) the signer's private key was used to digitally sign the ~~electronic record Digital Signature Certificate~~, which is known to be the case if the signer's public key was used to verify the signature because the signer's public key will verify only a Digital Signature created with the signer's private key; and
 (b) the ~~electronic record Digital Signature Certificate~~ was unaltered, which is known to be the case if ~~the hash result computed by the verifier is identical to the hash result extracted from the Digital Signature during the verification process~~ verification succeeds for the Digital Signature Certificate under the signature algorithm specified in these rules and the mode prescribed by the applicable standard and the applicable profiles.

6. Standards.—The Information Technology (IT) architecture for Certifying Authorities may support open standards and accepted de facto standards; the most important standards that may be considered for different activities associated with the Certifying Authority's functions are as under—

Product / activity	Standard and modification
Public Key Infrastructure	PKIX
Digital Signature Certificates and Digital Signature revocation list	X.509 version 3 certificates and certificate revocation lists conforming to ITU-T X.509, RFC 5280 and the applicable CCA interoperability profiles. <i>RFC 5280 and the applicable Interoperability Guidelines apply; algorithm-specific certificate/CRL encodings are specified below.</i>

Product / activity	Standard and modification
Directory (DAP and LDAP)	X500 for publication of certificates and Certification Revocation Lists (CRLs)
Database Management Operations	Use of generic SQL
Public Key algorithm	DSA, RSA and Curves NIST P-256, P-384 or P-521. <i>ML-DSA and SLH-DSA; XMSS/XMSS-MT and LMS/HSS only under the restricted profiles below. ML-KEM is separately specified for key establishment.</i>
ML-DSA	<i>FIPS 204: ML-DSA-44, ML-DSA-65 and ML-DSA-87. The parameter set for each application shall be specified in the applicable profiles.</i> <i>Certificates and CRLs: pure ML-DSA under RFC 9881; the OID identifies the parameter set, AlgorithmIdentifier parameters shall be absent, and the context shall be empty.</i> <i>CMS signatures: pure ML-DSA under RFC 9882 with the selected parameter-set OID and absent signatureAlgorithm parameters. HashML-DSA requires a interoperable profile issued by the Controller identifying mode, permitted pre-hash and output length. Internal SHAKE processing shall remain as specified in FIPS 204.</i>
SLH-DSA	<i>FIPS 205: SLH-DSA-SHA2-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-256s and SLH-DSA-SHA2-256f shall be preferred. SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-256s and SLH-DSA-SHAKE-256f shall be used only where the corresponding SHA-2 set cannot be used.</i> <i>Certificates and CRLs: RFC 9909; identify the exact parameter set and pure or pre-hash mode by the standard OID; parameters shall be absent and context empty. Pure mode shall be preferred.</i> <i>CMS signatures: pure SLH-DSA under RFC 9814. HashSLH-DSA requires a signature-container profile issued by the Controller. Pre-hash functions and output lengths shall conform to the selected standard OID. The selected parameter set's internal hashing shall remain unchanged.</i>
XMSS/XMSS-MT	<i>XMSS and XMSS-MT: parameter sets specified in NIST SP 800-208 only, using RFC 8391 encodings. Identify the exact registered parameter-set code, hash family, output length, tree height and, for XMSS-MT, layer count.</i> <i>Certificate public keys and certificate/CRL signatures: RFC 9802. Document signatures require a container profile issued by</i>

Product / activity	Standard and modification
	<i>the Controller; an X.509 profile alone does not authorise CMS or XML use.</i>
<i>LMS/HSS</i>	<i>LMS/HSS: parameter sets specified in NIST SP 800-208 only, using RFC 8554 encodings. Identify the LMS and LM-OTS typecodes, hash/output length, tree height, Winternitz parameter and HSS level count.</i> <i>Certificate public keys and certificate/CRL signatures: RFC 9802. CMS signatures: RFC 9708. The exact parameter choices shall be carried in the standard key/signature structures.</i>
<i>Stateful-signature controls</i>	<i>XMSS/XMSS-MT and LMS/HSS shall be restricted to specialised controlled applications specified in the applicable profiles. Key generation and signature generation shall conform to NIST SP 800-208, section 8.1, and shall be performed within validated hardware cryptographic modules providing Level 3 or higher physical security, with a non-modifiable or limited operational environment. The modules shall operate only in approved mode, shall not implement a bypass mode, and shall not permit export of private keying material. The entropy source shall be located within the module's physical boundary. On accepting a signing request, the module shall increment the signing-key leaf index and store the incremented value in non-volatile storage before releasing the signature or accepting another signing request. One-time signing keys shall not be reused. Recovery and disaster-recovery arrangements shall comply with these requirements and shall not permit private-key export, cloning or rollback of signing state. Root CA and other critical uses shall require prior evaluation of the complete implementation and recovery arrangements. These schemes shall not be used for general-purpose subscriber/eSign signing where reliable state management cannot be assured.</i>
<i>Post-quantum key establishment</i>	<i>FIPS 203: ML-KEM-512, ML-KEM-768 and ML-KEM-1024. Certificate subject public keys: RFC 9935; the OID identifies the parameter set, parameters shall be absent and key usage shall conform to that RFC. CMS key establishment: RFC 9936 using KEMRecipientInfo under RFC 9629. ML-KEM shall be used for key establishment and shall not be used for digital signature generation or verification. Its application-specific use shall follow the The applicable profiles. An ML-KEM certificate shall be signed by a signature algorithm specified in these rules.</i>
<i>Digital Hash Function</i>	<i>SHA-2 under FIPS 180-4 shall be preferred for external digests. FIPS 202 functions shall be used for external digests only where SHA-2 cannot be used, with the exact function and output length specified in the The applicable profiles. Only pre-hash functions</i>

Product / activity	Standard and modification
	<i>permitted by the selected algorithm and mode shall be used. Internal hashing prescribed by the applicable algorithm standard shall remain unchanged.</i>
RSA Public Key Technology	PKCS#1 RSA Encryption Standard (2048, 4096 bit); PKCS#5 Password Based Encryption Standard; PKCS#7 Cryptographic Message Syntax standard; PKCS#8 Private Key Information Syntax standard; PKCS#9 Selected Attribute Types; PKCS#10 RSA Certification Request; PKCS#12 Portable format for storing/transporting a user's private certificates. <i>Existing RSA use is retained for transition, coexistence and legacy validation.</i>
Distinguished name	X.520
Digital Encryption and Digital Signature	PKCS#7. CMS RFC 5652; <i>ML-DSA signatures: RFC 9882;</i> <i>SLH-DSA signatures: RFC 9814;</i> <i>LMS/HSS signatures: RFC 9708;</i> <i>ML-KEM key establishment: RFC 9936. XMSS/XMSS-MT CMS use requires a applicable standard profile issued by the Controller.</i>
Digital Signature Request Format	PKCS#10. <i>ML-DSA/SLH-DSA requests shall identify the subject key's parameter set and prove possession using that key and the signature mode prescribed by the applicable standard and the applicable profiles. ML-KEM possession requires an key-establishment proof specified in the applicable profiles, not an ML-KEM signature.</i>
Identifier and parameter binding	<i>The certificate subjectPublicKeyInfo shall identify the subject's key algorithm and parameter set; the certificate/CRL signature fields shall identify the issuer's signing algorithm and mode. These need not be the same algorithm or parameter set. Each signature shall carry its required identifier, and the verifier shall use the corresponding signing public key. Parameter choices shall be expressed through the OID or standard key/signature encoding; an ASN.1 parameters field shall remain absent wherever the applicable RFC requires. Unknown, private, provisional or mismatched identifiers shall be rejected.</i>

The algorithms and standards specified in these rules shall be implemented in accordance with the applicable interoperability guidelines, signature profiles and other technical requirements issued by the Controller, including requirements for parameter selection, processing modes, identifiers, encodings and application-specific procedures.

Explanation.—The Digital signature certificate granted before the commencement of the information technology (Certifying Authorities Amendment) Rules, 2011 using SHA-1, digital hash function standard shall continue to be valid till the date of expiry of such certificate.

Digital Signature (End Entity) Rules, 2015

4. Creation of digital signature.

- (1) ~~The signatory shall, while signing an electronic record or any other item of information, first apply an hash function in the signatory's hardware or software~~ *create the digital signature using the signatory's private key in accordance with the signature algorithm specified in these rules, using the mode and hashing requirements prescribed by the applicable standard and the applicable profiles.*
- (2) ~~The hash function shall produce a hash result.~~
- (3) ~~The signatory's hardware or software shall then transform the hash result into a digital signature using signatory's private key and signature algorithm.~~
- (4) The contextual information like date and time, shall be then made part of the digital signature.
- (5) The counter signatures or parallel signatures or both may also be applied to electronic record.
- (6) The following information may also be a part of signature,-
- (a) the signatory's public key signature certificate(s);
 - (b) the public key certificate(s) of the licensed Certifying Authorities which used to verify the authenticity of the digital signature certificate issued to the signatory;
 - (c) the self signed certificate generated by the Controller used to verify the authenticity of the public key certificate of the licensed Certifying Authorities;
 - (d) the certificate revocation list(s) maintained by the licensed Certifying Authorities, and the controller which is used to check whether the digital signature certificate has been revoked under section 38 the Act;
 - (e) online certificate status protocol responder certificates and online certificate status protocol responses that may be used in lieu of certificate revocation list.
- (7) To create long term valid digital signature,-
- (a) a time stamp shall be applied initially to the signed data including the certificates and revocation information;
 - (b) ensure that initial time stamp shall cover all the data and signature(s);
 - (c) a nested time stamp option shall be used to ensure signature validity past the time stamping service provider's (tssp) key or algorithm expiry where the nesting of time stamps implies that a subsequent time stamp shall be applied to the prior time stamp;
 - (d) signature(s) and time stamps may be embedded in the data itself or stored separately as standalone

5. Verification of digital signature.

- (1) ~~The verification of a digital signature shall be accomplished by computing a new hash result of the original electronic record by means of the hash function used to create a digital signature and by using the public key and the new hash result~~ *using the signatory's public key in accordance with the signature algorithm specified in these rules, using the mode and hashing requirements prescribed by the applicable standard and the applicable profiles, the verifier shall check-*
- (a) if the digital signature was created using the corresponding private key and shall be applicable for parallel and counter signatures applied on the electronic record, if present;
 - (b) the time when the digital signature was created;
 - (c) *whether the signed electronic record has been altered.*
- (2) To verify counter signature, the signature on electronic record and thereafter signature on previous signature serially shall be verified.
- (3) To verify parallel signature, signature on electronic record shall be verified independently .

(4)To verify long term signature, the initial time stamp and all subsequent time stamp applied on the each prior time stamp shall be verified.

6. Verification of Digital Signature Certificate.

(1)The self signed certificate generated by the Controller, which begins the trust chain for the public key infrastructure, shall be used to verify the authenticity of the public key certificate of the licensed Certifying Authorities.

(2)The public key certificate of the licensed Certifying Authorities shall be used to verify the authenticity of the digital signature certificate issued to the subscribers.

(3)The certificate revocation list maintained by the licensed Certifying Authorities shall be checked to confirm whether the certificate of the licensed Certifying Authorities is valid or whether it has been revoked under section 38 the Act.

(4)While verifying the validity of a digital signature the corresponding digital signature certificate shall chain up through the public key certificate of the issuing Certifying Authority to the self signed certificate of the Controller and if any of the certificates in the trust chain is not trusted the signature shall not be verified.

(5)The Digital Signature Certificate shall be verified with respect to time of signature created.

(6)The chain of certificates shall be verified in accordance with the standards specified in rule 7.

(7)If the certificate validity is less than one hour, the checking of revocation list shall not be required.

7. Digital signature standards.—The most important standards that shall be applicable for different activities associated with digital signature functions are as under—

Product / activity	Standard and modification
Cryptographic hash function	<i>SHA-2 under FIPS 180-4 shall be preferred for external digests. FIPS 202 functions shall be used for external digests only where SHA-2 cannot be used, with the exact function and output length specified in the applicable profiles. Only pre-hash functions permitted by the selected algorithm and mode shall be used. Internal hashing prescribed by the applicable algorithm standard shall remain unchanged</i>
RSA Public Key Technology	PKCS#1 RSA Encryption Standard (2048, 4096 bit); Version 1.5. <i>Retained for transition, coexistence and legacy validation.</i>
<i>ML-DSA digital signature</i>	<i>FIPS 204: ML-DSA-44, ML-DSA-65 and ML-DSA-87. The parameter set for each application shall be specified in the applicable profiles.</i> <i>Certificates and CRLs: pure ML-DSA under RFC 9881; the OID identifies the parameter set, AlgorithmIdentifier parameters shall be absent, and the context shall be empty.</i> <i>CMS signatures: pure ML-DSA under RFC 9882 with the selected parameter-set OID and absent signatureAlgorithm parameters. HashML-DSA requires a interoperable profile issued by the Controller identifying mode, permitted pre-hash and output length. Internal SHAKE processing shall remain as specified in FIPS 204.</i>
<i>SLH-DSA digital signature</i>	<i>FIPS 205: SLH-DSA-SHA2-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-256s</i>

Product / activity	Standard and modification
	<i>and SLH-DSA-SHA2-256f shall be preferred. SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-256s and SLH-DSA-SHAKE-256f shall be used only where the corresponding SHA-2 set cannot be used.</i> <i>Certificates and CRLs: RFC 9909; identify the exact parameter set and pure or pre-hash mode by the standard OID; parameters shall be absent and context empty. Pure mode shall be preferred.</i> <i>CMS signatures: pure SLH-DSA under RFC 9814. HashSLH-DSA requires a signature-container profile issued by the Controller. Pre-hash functions and output lengths shall conform to the selected standard OID. The selected parameter set's internal hashing shall remain unchanged.</i>
XMSS/XMSS-MT	<i>XMSS and XMSS-MT: parameter sets specified in NIST SP 800-208 only, using RFC 8391 encodings. Identify the exact registered parameter-set code, hash family, output length, tree height and, for XMSS-MT, layer count.</i> <i>Certificate public keys and certificate/CRL signatures: RFC 9802. Document signatures require a container profile issued by the Controller; an X.509 profile alone does not authorise CMS or XML use.</i>
LMS/HSS	<i>LMS/HSS: parameter sets specified in NIST SP 800-208 only, using RFC 8554 encodings. Identify the LMS and LM-OTS typecodes, hash/output length, tree height, Winternitz parameter and HSS level count.</i> <i>Certificate public keys and certificate/CRL signatures: RFC 9802. CMS signatures: RFC 9708. The exact parameter choices shall be carried in the standard key/signature structures.</i>
Stateful-signature controls	<i>XMSS/XMSS-MT and LMS/HSS shall be restricted to specialised controlled applications specified in the applicable profiles. Key generation and signature generation shall conform to NIST SP 800-208, section 8.1, and shall be performed within validated hardware cryptographic modules providing Level 3 or higher physical security, with a non-modifiable or limited operational environment. The modules shall operate only in approved mode, shall not implement a bypass mode, and shall not permit export of private keying material. The entropy source shall be located within the module's physical boundary. On accepting a signing request, the module shall increment the signing-key leaf index and store the incremented value in non-volatile storage before releasing the signature or accepting another signing request. One-time signing keys shall not be reused. Recovery and disaster-</i>

Product / activity	Standard and modification
	<i>recovery arrangements shall comply with these requirements and shall not permit private-key export, cloning or rollback of signing state. Root CA and other critical uses shall require prior evaluation of the complete implementation and recovery arrangements. These schemes shall not be used for general-purpose subscriber/eSign signing where reliable state management cannot be assured.</i>
Key-establishment certificates	<i>FIPS 203: ML-KEM-512, ML-KEM-768 and ML-KEM-1024. Certificate subject public keys: RFC 9935; the OID identifies the parameter set, parameters shall be absent and key usage shall conform to that RFC. CMS key establishment: RFC 9936 using KEMRecipientInfo under RFC 9629. ML-KEM shall be used for key establishment and shall not be used for digital signature generation or verification. Its application-specific use shall follow the applicable profiles. An ML-KEM certificate shall be signed by a signature algorithm specified in these rules.</i>
Encryption and digital signature	<i>PKCS#7, CMS. CMS RFC 5652; ML-DSA signatures: RFC 9882; SLH-DSA signatures: RFC 9814; LMS/HSS signatures: RFC 9708. ML-DSA and SLH-DSA CMS signatures shall use pure mode with an empty context and signed-attribute processing and digest selection prescribed by the applicable CMS signature profile issued by the Controller. ML-KEM CMS key establishment shall follow RFC 9936; it is not a digital-signature operation.</i>
Validation of Digital Signature Certificate	<i>RFC 5280. ML-DSA: RFC 9881; SLH-DSA: RFC 9909; XMSS/XMSS-MT and HSS: RFC 9802; ML-KEM subject keys: RFC 9935. Certificate-path signatures shall be verified under the issuer's signature profile, independently of the subject-key algorithm.</i>
ECC curve	<i>NIST P-256, P-384, or P-521. Retained for transition, coexistence and legacy validation.</i>
Long term signature formats	1. CAdES RFC 5126; 2. PAdES with CAdES. PQC signatures shall use the corresponding CMS signature profile issued by the Controller and preserve identifiers, certificates, revocation evidence and timestamps.
Time stamp token	<i>As specified RFC 3161. PQC TSA certificates and token signatures shall use the corresponding ML-DSA or SLH-DSA certificate and CMS profiles. SHA-2 shall be preferred for the message imprint.</i>

Product / activity	Standard and modification
Identifier and parameter binding	<i>The certificate subjectPublicKeyInfo shall identify the subject's key algorithm and parameter set; the certificate/CRL signature fields shall identify the issuer's signing algorithm and mode. These need not be the same algorithm or parameter set. Each signature shall carry its required identifier, and the verifier shall use the corresponding signing public key. Parameter choices shall be expressed through the OID or standard key/signature encoding; an ASN.1 parameters field shall remain absent wherever the applicable RFC requires. Unknown, private, provisional or mismatched identifiers shall be rejected.</i>

The algorithms and standards specified in these rules shall be implemented in accordance with the applicable interoperability guidelines, signature profiles and other technical requirements issued by the Controller, including requirements for parameter selection, processing modes, identifiers, encodings and application-specific procedures.

9. Creation of xml digital signature.

(1) To sign an electronic record or any other item of information, the signatory shall first construct reference elements, xml digital signature element, signed info, key info and signature value.

(2) For the purpose of reference element generation, the signing software shall-

(a) create reference(s) element(s) with reference to the item of information, xml transform element(s) (optional), digest algorithms and digest value;

(b) optionally apply xml transform(s) to each referenced object in a sequential order;

(c) apply the hash function in the signatory's hardware or software to each reference elements, store the hash result in the reference element;

(d) ensure that if the object element is created, it shall not have a manifest element;

(e) ensure that exclusive canonicalization "without comments" has been mandatorily specified in addition to any other transforms.

(3) For the purpose xml digital signature generation, the signing software shall-

(a) create signed info element with signature method, canonicalisation method and reference(s);

(b) apply canonicalisation to signed info and ~~calculate the hash value of canonicalised signed info using the hashing algorithms implied by the signature method~~ process canonicalised signed info in accordance with the signature algorithm specified in these rules, using the mode and hashing requirements prescribed by the applicable standard and the applicable profiles;

(c) ensure that,-

(i) the signatory has seen all the contents of the document before signing;

(ii) the contents display requirements, in the case of automated signing process, is not required;

(iii) to sign multiple resources together,-

(a) each resource shall be rendered on the screen;

(b) each referenced xml resource shall be rendered using xslt and the xslt shall be the last transform done to render the resource on the screen;

(c) each non xml resource shall be rendered using mime-type attribute mentioned in the object;

(d) ~~generate the signature using the signature algorithm and the hash, the signatory's private key, and the public key parameters (if applicable)~~ generate the signature over canonicalised signed info using the signatory's private key and the signature algorithm specified in these rules, using the mode, hashing requirements and public key parameters (if applicable) prescribed by the applicable standard and the applicable profiles and perform base64 encoding of the signature result and use it to form signature value;

(e) construct the signature element that includes signed info, items of information, key info with x 509 certificate

element and signature value and the x509 certificate element shall carry the signatory's x 509 public key certificate.

(4)The contextual information like date and time, shall be then made part of the xml digital signature.

(5)The counter signatures or parallel signatures or both may also be applied to electronic record.

(6)The following information may also be a part of signature-

(a) the public key certificate(s) of the licensed Certifying Authorities which used to verify the authenticity of the digital signature certificate issued to the signatory;

(b) the self signed certificate generated by the Controller used to verify the authenticity of the public key certificate of the licensed Certifying Authorities;

(c) the certificate revocation list(s) maintained by the licensed Certifying Authorities and controller which is used check whether the digital signature certificate has been revoked under section 38 the Act;

(d) online certificate status protocol responder certificates and online certificate status protocol responses may be used in lieu of certificate revocation list.

(7)To create long term valid xml digital signature-

(a) a time stamp shall be applied initially to the signed document, where the Initial time stamp shall cover all the electronic record and signature(s);

(b) a nested time stamp option shall be used to ensure signature validity past the time-stamping service provider (tssp)'s key or algorithm expiry where as nesting of time stamps implies that a subsequent time stamp shall be applied to the prior time stamp;

(c) signature(s) and time stamps may be embedded in the data itself or stored separately as standalone.

10. Verification of xml digital signature.

(1)The verification of the xml digital signature shall be accomplished by signature validation, reference validation and certificate validation and an xml digital signature shall be treated valid only if signature validation, reference validation and certificate validation as specified below are complied with.

(2)To accomplish signature validation—

(a) canonical form of signed info as produced during reference validation shall be used;

~~(b) canonical form of signature method using the canonicalization method shall be obtained~~ *the signature algorithm specified in these rules, its mode and applicable parameters prescribed by the applicable standard and the applicable profiles shall be identified from the signature method in accordance with the XML signature profile issued by the Controller;*

~~(c) the public key contained in the signatory's x509 certificate that is included in the xml digital signature, the canonicalised form of signed info signature value, and canonicalised form of signature method shall be used to verify the signature~~ *the canonicalised signed info and the signature value shall be used to verify the signature in accordance with the signature algorithm specified in these rules, using the mode, hashing requirements and parameters prescribed by the applicable standard and the XML signature profile issued by the Controller.*

(3)To accomplish reference validation—

(a) canonicalise the signed info element based on the canonicalisation method in the signed info shall be used;

(b) for each reference in the signed info—

(i) if transformations were applied by the signatory, then the verifier software may de-reference the uniform resource identifier (uri) and execute transforms provided by the signatory in the reference element;

(ii) compute digest of referenced item using the digest method specified in its reference element;

~~(iii) compare the computed digest value against digest value in the signed info reference; if there is any mismatch or validation fails~~ *compare the computed digest value with the digest value in the signed info reference; any mismatch shall cause reference validation to fail;*

(4)The Digital Signature Certificate included in the xml digital signature shall be verified in accordance with the provisions specified in rule 6.

(5)To verify counter signature, the signature on electronic record first and signature on previous signature shall be verified serially.

(6)To verify parallel signature, signature on electronic record shall be verified independently.

(7)To verify long term signature, the initial time stamp and all subsequent time stamp applied on the each prior time stamp shall be verified.

11. The xml digital signature standards.—The most important standards that shall be applicable for different activities associated with xml digital signature functions are as under—

Product / activity	Standard and modification
XML Digital Signature Standard	RFC 3275 with the following constraint: Manifest is not permitted inside Object; Key info containing X509 Certificate element is mandatory; Reference Processing shall use Exclusive Canonicalization (without comments) in addition to other transforms; for XML resource, XSLT shall be the last transform done to enable rendering; each referenced XML resource shall be implemented using XSLT; each non XML resource shall be implemented using Mime Type attribute mentioned in the object.
XML Namespace	W3C Namespaces in XML 1.0; namespace URI syntax as specified in RFC 3986, where applicable.
Signature encoding	UTF-8 RFC 3629
Signature Value Encoding	Base64 RFC 4648
Reference element Digest	~~SHA-256~~ SHA-256, SHA-384 or SHA-512 under FIPS 180-4 shall be used as specified in the applicable XML signature profile. FIPS 202 functions shall be used for reference digests only where SHA-2 cannot be used, with the exact function and output length specified in that profile
Signature Algorithm	SHA256 with RSA PKCS-1 Version 1.5. ECC, ML-DSA and SLH-DSA shall be used only through XML signature profiles issued by the Controller specifying algorithm identifiers, parameter sets, modes and key encodings. ML-DSA uses FIPS 204 and SLH-DSA uses FIPS 205 with the parameter sets and mode restrictions specified in rule 7. XMSS/XMSS-MT and LMS/HSS require a XML signature profile issued by the Controller and rule 7 state controls. ML-KEM shall not be used as an XML signature algorithm.
Signature block Canonicalization	Exclusive (without comments), XML-EXC-C14N, RFC 3741; Canonical XML 1.0 (omits comments); Canonical XML 1.1 (omits comments).

Product / activity	Standard and modification
Transform Algorithms	Exclusive (without comments), XML-EXC-C14N, RFC 3741; Canonical XML 1.0 and 1.1 (omit comments); XSLT—XSL Transforms Version 1.0, W3C; XPath transform—W3C XPath 1.0 as specified in XML Signature; XPath Filter 2.0—RFC 3653, where permitted by the applicable profile.
Signature Type	enveloped or enveloping or detached
Digital Signature Certificate	(DER) X.509 V3 issued as per interoperability guidelines. <i>ML-DSA: RFC 9881; SLH-DSA: RFC 9909; XMSS/XMSS-MT and HSS: RFC 9802. KeyInfo and SignatureMethod shall identify the corresponding subject key and signature profile; issuer and subject algorithms shall be validated separately.</i>

Composition of the Expert Committee

Expert Committee for Evaluation of IT Act Cryptographic Standards and Recommendations on Post-Quantum Cryptography (PQC) Algorithms

Constituted by the Office of the Controller of Certifying Authorities vide Office Memorandum F. No. 19(92)/2026-CCA dated 09 May 2026.

S. No.	Name	Designation / Organisation	Role
1	Dr. Sundeep Oberoi	Associate Professor, IIT Bombay	Chairman
2	Prof. Goutam Paul	ISI Kolkata	Member
3	Mr. Sitaram Chamarty	Principal Consultant, TCS Innovation Labs	Member
4	Smt. Reena Garg	Scientist 'G' & Head, LITD, BIS	Member
5	Mr. C. V. Sridhar	Mission Director, AP State Quantum Mission	Member
6	Dr. Rajan M. A.	Principal Scientist, TCS Research	Member
7	Mr. Rahul Aggarwal	Partner, Cybersecurity, PwC India	Member
8	Ms. Neelu Tripathy	Cloud Security Architect, Adobe	Member
9	Mr. Rajesh Mittal	Founder & CEO, Capricorn CA	Member
10	Mr. P. Ramachandran	Scientist 'G', Office of CCA, MeitY	Member
11	Dr. K. K. Soundra Pandian	Scientist 'E', Office of CCA, MeitY	Member Convener