# A Post-Quantum Evidence Layer for Incremental Migration of Legacy Digital Signature Systems

**Gabriel Dilly**[1], **Jean Everson Martina**[1]

[1] Departamento de Informática e de Estatística
Universidade Federal de Santa Catarina (UFSC)
Florianópolis, SC – Brazil

gabriel.dilly@posgrad.ufsc.br, jean.martina@ufsc.br

***Abstract.*** *Deployed digital-signature infrastructures cannot migrate atomically because cryptographic algorithms are coupled to certificates, policies, validation services, formats, and document workflows. This paper proposes an incremental migration model that introduces post-quantum protection first at the evidence layer. Before practical classical compromise, the target construction commits a document digest, its classical signature, validation material, recorded result, and policy to a Merkle root protected by a post-quantum timestamp token. The model defines a security and deployment boundary identifying which evidence must be captured and which guarantees survive without replacing the original signing infrastructure. Under correct intake validation, collision-resistant hashing, and a trusted timestamping service, it provides tamper evidence and existence-before-time for the captured validation state; it neither upgrades the original signature nor repairs evidence captured too late. A PreserveEU-based prototype evaluates a first implementation step by committing CMS bytes and generating local ML-DSA and SLH-DSA root tokens; enriched package encoding and independently operated post-quantum timestamping remain outside the implementation. Direct CMS measurements contextualize immediate signature-migration costs rather than providing an end-to-end deployment comparison. On the evaluated software-only host, ML-DSA provided the lowest post-quantum token latency and smaller records than the evaluated SLH-DSA variants, whereas SLH-DSA offered diversity from lattice-based assumptions at higher measured cost.*

## 1. Introduction

Digital signatures are widely used to guarantee authenticity, integrity, and non-repudiation of electronic documents. In archival, governmental, legal, and compliance scenarios, however, these guarantees may need to remain verifiable for decades, beyond the expected secure lifetime of the algorithms originally used to sign the documents. Long-term trust therefore cannot rely only on the original signature; it also requires preservation mechanisms capable of maintaining verifiable evidence over time [Gondrom et al. 2007, Blazic et al. 2011, Joseph et al. 2022, Moody et al. 2024].

The post-quantum transition makes this preservation problem more urgent. Quantum-capable adversaries threaten public-key algorithms such as RSA and elliptic-curve cryptography [Shor 1994, Joseph et al. 2022, Moody et al. 2024], while practical migration must be staged across cryptographic assets, services, policies, and operational

dependencies [Moody et al. 2024, NIST NCCoE 2026]. This is especially difficult for document-signing infrastructures, where the signing algorithm is embedded in certificate policies, trust anchors, validation services, revocation distribution, timestamping services, signature formats, client software, and audit rules. The Brazilian ICP-Brasil framework illustrates this coupling by defining profiles and policies for digital signatures and their validation [Instituto Nacional de Tecnologia da Informação 2020]. In this paper, such deployments are called *legacy-bound digital signature systems*: their classical cryptography is expected to become insufficient against future quantum-capable adversaries, but their surrounding infrastructure cannot be replaced atomically.

Hybrid and composite signatures combine classical and post-quantum authentication in the signing layer [Driscoll et al. 2025, Ounsworth et al. 2026, Fall et al. 2025]. They are preferable for newly created signatures when signers, PKI, formats, and relying parties can be upgraded together. This paper studies a complementary boundary: an external evidence layer for existing signatures and workflows that cannot yet change. Both approaches can coexist—new signatures may be hybrid while legacy collections are sealed and later renewed. The central contribution is an incremental migration model that protects the evidence of what was validated, when it was validated, and under which policy, without requiring immediate replacement of every document-signing workflow.

The security obtained by this strategy is intentionally limited. A post-quantum evidence layer does not transform an existing RSA or ECDSA document signature into a post-quantum signature. Nor can it repair a signature that was forged, incorrectly validated, or preserved only after the classical algorithm had already become practically breakable. Its purpose is narrower: if a document, its classical signature, its certificate and revocation material, and its validation state are captured before practical quantum compromise, then a post-quantum evidence layer can help preserve the integrity, existence, and validation state of that captured evidence against later tampering.

This framing leads to two research questions. First, what guarantees can a post-quantum evidence layer provide for legacy-bound digital signature systems, and what guarantees does it explicitly not provide? Second, what are the time and storage costs of validating legacy signatures at intake and protecting Merkle-root evidence records with NIST-standardized post-quantum signature algorithms, contextualized by direct document-signature migration baselines?

To address these questions, we specify a target evidence-layer migration model for systems that continue to rely on classical document signatures while adding post-quantum protection to captured evidence. We partially integrate this model in PreserveEU and evaluate its first implemented step using ML-DSA and SLH-DSA, the NIST-standardized algorithms in FIPS 204 and FIPS 205 [National Institute of Standards and Technology 2024a, National Institute of Standards and Technology 2024b]. Legacy RSA intake validation characterizes the first-stage budget; CMS-byte commitment, local root-token issuance, and XMLERS size characterize the evaluated post-quantum integration; and direct CMS signing and verification contextualize immediate signature migration.

At the implementation level, the prototype's enabling change is the adaptation of PreserveEU's root-token path to issue and encode ML-DSA- and SLH-DSA-protected

RFC 3161 tokens. This implementation step alone, however, does not determine which validation state must be committed, when it must be sealed relative to classical compromise, or which later claims follow from trusted-time and renewal assumptions; defining those boundaries is a central contribution of the proposed migration model.

The contributions are threefold: (i) a migration model that identifies the evidence that must be committed before classical compromise and positions this boundary relative to embedded long-term signatures and hybrid migration; (ii) an assumption-based security analysis of the guarantees preserved across that boundary; and (iii) a partial PreserveEU integration and single-host cost characterization of PQC-signed root tokens, with direct CMS signing and verification as a baseline. The resulting contribution combines established evidence-record mechanisms with a security model and an evaluated first step for introducing PQC before the surrounding document-signing infrastructure can be replaced.

## 2. Background and Positioning

This section positions the migration model among signature-profile and evidence-record mechanisms and introduces the evaluated post-quantum candidates.

### 2.1. Legacy-Bound Signatures and Incremental PQC Migration

Classical digital signature systems rely on public-key algorithms (e.g., RSA, ECDSA), certificate chains, revocation information, validation policies, timestamping services, and signature containers. In ordinary verification, these elements are checked with respect to a validation time and a policy. In long-term scenarios, the evidence supporting that validation must remain meaningful after certificates expire, services change, or algorithms are deprecated. Within the broader preservation context defined by OAIS, this paper concentrates on the cryptographic evidence layer [International Organization for Standardization 2025].

For the incremental strategy studied here, the relevant object is therefore not only the document signature, but the validation state that binds a document digest, a signature, certification material, revocation status, validation policy, and timestamp context. Cryptoagility matters because this evidence must be renewable or complemented as algorithms and policies age [NIST 2026, Alnahawi et al. 2023].

ETSI baseline profiles retain validation material and archive timestamps within CAdES, XAdES, and PAdES, while ETSI EN 319 102-1 specifies validation procedures [ETSI 2021, ETSI 2024b, ETSI 2024c, ETSI 2024a]. ERS/XMLERS protect detached objects and inclusion proofs, including integration with CAdES [ETSI 2017]. Profile elevation therefore suits supported containers; detached evidence suits heterogeneous repositories. Geihs et al. prove an ERS variant secure in a long-lived computational-security model provided that the archive, acting as notary, validates preceding timestamps at renewal [Geihs et al. 2015]. Their result formalizes validation and trust assumptions inherited here.

Clupek et al. propose a lattice-based archive that applies quantum-resistant signatures to documents, certificates, and timestamps [Clupek et al. 2015]; Thiel and Thiel evaluate PQC candidates against PKI and eIDAS-oriented long-term evidential requirements [Thiel and Thiel 2021]. Neither defines our boundary for existing signatures: validate at intake, leave the signer, format, and PKI unchanged, and protect the captured state

with standardized PQC root tokens before practical compromise. The distinction is where PQC first enters and which validation state it binds.

Kusber et al. apply RFC 4998 to serialized DLT blocks and linked off-chain records, timestamping a Merkle root and renewing the tree before a hash weakens [Kusber et al. 2021]. This is the closest construction, but it preserves crypto-agility for DLT data; our layer instead commits a legacy-PKI validation state, protects the root with standardized PQC tokens, and requires capture before practical compromise.

Brînzea et al. retain ordinary RFC 3161 validation but asynchronously anchor each classical token's serial-number–hash mapping in Ethereum or Hyperledger Fabric [Brînzea et al. 2025]. Their supplementary evidence depends on ledger consensus, finality, and availability, plus public-chain cost and confirmation latency; checking requires a ledger query. Our PQC token instead protects a Merkle commitment to document-validation state, avoiding ledger dependence while retaining trust in the PQ timestamping service and timely renewal.

Table 1 summarizes the migration boundaries. Brînzea et al. remain outside it because their classical TSA gains non-PQC ledger evidence.

**Table 1. Mechanism boundaries relevant to incremental PQC migration.**

| Mechanism | Primary protected object | Changes signer/PKI? | Best fit |
|---|---|---|---|
| AdES long-term profiles | Embedded validation material and archive timestamps | No, for elevation | Supported signature containers that can be elevated to T/LT/LTA. |
| Classical ERS/XMLERS | External object or DLT-record commitments and renewable existence evidence | No | Repositories requiring detached renewal, including DLT records. |
| Hybrid/composite signatures | New signatures combining classical and PQC authentication | Yes | New workflows able to migrate issuance, formats, and relying parties together. |
| Direct/archive-wide PQC | New signatures, certificates, and timestamp credentials | Yes | Infrastructures able to replace classical signing and trust-service paths. |
| External PQ evidence layer | Captured validation state of existing signatures | No | Legacy signatures sealed before classical compromise and renewed over time. |

## 2.2. Evidence Records and Merkle-Root Timestamping

Evidence records were standardized to support long-term non-repudiation of the existence and integrity of data, including renewal before cryptographic primitives become insecure [Gondrom et al. 2007]. XMLERS provides an XML representation of the same evidence-record model for XML-oriented environments [Blazic et al. 2011]. In this paper, an evidence record is relevant because it can bind more than a document digest: it can preserve commitments to the document, the original classical signature, certificate and revocation material, validation results, policy identifiers, and timestamp metadata.

Merkle trees are the scalability mechanism behind the construction used in this work. A preservation system can hash individual evidence packages, aggregate them into

a Merkle tree, and timestamp only the root. The timestamped root then acts as a compact commitment to the covered set, while each preserved package remains verifiable through its inclusion path. This avoids timestamping every package independently and creates a natural point at which post-quantum signatures or timestamping credentials can protect many legacy-signed artifacts at once.

The security of this layer is bounded by its assumptions. A timestamp token provides evidence that a digest or Merkle root existed no later than the claimed time, provided that the timestamping authority and its signing credential are trustworthy [Adams et al. 2001]. Evidence records also rely on the collision resistance of the hash function and on timely renewal when algorithms age [Gondrom et al. 2007, Blazic et al. 2011]. These assumptions motivate the explicit threat model introduced in Section 3.

### 2.3. Post-Quantum Signature Algorithms for Evidence Chains

NIST standardized ML-DSA and SLH-DSA as post-quantum digital signature algorithms in FIPS 204 and FIPS 205 [National Institute of Standards and Technology 2024a, National Institute of Standards and Technology 2024b]. ML-DSA is the standardized module-lattice-based signature algorithm derived from Dilithium, whereas SLH-DSA is the standardized stateless hash-based signature algorithm derived from SPHINCS+ [Ducas et al. 2018, Bernstein et al. 2019]. NIST has registered the corresponding object identifiers in the CSOR, and the IETF has specified conventions for using these algorithms in CMS [National Institute of Standards and Technology 2026, Salter et al. 2025, Housley et al. 2025].

Their roles in an evidence layer are different. ML-DSA is evaluated as a candidate for a primary evidence chain when throughput, storage, and integration cost dominate. SLH-DSA had a higher cost in our software measurements, but its hash-based design offers diversity from lattice-based assumptions; hardware acceleration and compatibility with existing SHA-2 infrastructure may change this trade-off. Performance alone should therefore not determine deployment policy: high-assurance systems may prefer parallel or renewable evidence chains even when an additional chain increases processing time and storage.

## 3. Threat Model and Security Goals

The threat model follows the usual discipline of making explicit what is being protected, what can go wrong, and which assumptions are required for the stated guarantees [Shostack 2014]. We use this structure to avoid overstating the effect of a post-quantum evidence layer: the protected object is not the original classical signature in isolation, but the validation state captured and committed before a relevant cryptographic transition point.

### 3.1. System Model

Let $D_i$ be the $i$-th document signed at time $t_s$ with a classical signature $\sigma_{C,i}$, such as RSA or ECDSA. Let $\mathsf{Cert}_{C,i}$ be the corresponding certificate chain, $\mathsf{Rev}_{C,i}$ the revocation information available to the validator, and $V_{t_p,i}$ the validation state observed at preservation

time $t_p$ under a policy identifier $\pi_{t_p}$. The preserved evidence package for document $D_i$ is modeled as

$$P_i = \{\mathsf{id}_i, h(D_i), \sigma_{C,i}, \mathsf{Cert}_{C,i}, \mathsf{Rev}_{C,i}, V_{t_p,i}, \pi_{t_p}\}.$$

Here, $\mathsf{id}_i$ is a stable document or preservation identifier used to associate the evidence package with the archived object and to distinguish it from other preserved packages. The document itself may remain in external archival storage; the evidence layer commits to its digest and to the validation material needed to interpret the signature state at $t_p$. We assume a canonical, unambiguous encoding $\mathsf{encode}(P_i)$ with explicit domain separation, field identifiers, and algorithm identifiers before hashing. The evidence layer computes a digest $h(\mathsf{encode}(P_i))$ for each package, aggregates package digests into a Merkle tree, and obtains a root value $\mathsf{root}_{t_p}$. A post-quantum timestamp token $T_{PQ}$ then binds $\mathsf{root}_{t_p}$ to the responder's time and TSA policy; the document-validation policy is bound through its inclusion in $P_i$. The resulting evidence record is denoted by

$$ER_{t_p} = \{\mathsf{root}_{t_p}, T_{PQ}, \mathsf{path}_i, \mathsf{meta}_{t_p}\},$$

where $\mathsf{path}_i$ is the Merkle inclusion path for $P_i$ and $\mathsf{meta}_{t_p}$ contains evidence metadata such as algorithms, policy identifiers, and timestamping context.

### 3.2. Adversarial Timeline

We distinguish three times. The document is originally signed at $t_s$, preserved at $t_p$, and later faces an adversary with practical quantum capability at $t_q$. The intended migration window is $t_s < t_p < t_q$: the document was signed with classical cryptography, validated and preserved while the classical signature was still acceptable under policy, and only later exposed to a quantum-capable adversary.

Before $t_q$, the adversary is assumed to be classical. After $t_q$, the adversary may be able to forge or undermine classical public-key signatures whose security depends on RSA or elliptic-curve hardness. At any time, the adversary may try to replace $D_i$, $\sigma_{C,i}$, certificate material, revocation data, validation results, Merkle paths, or evidence records. The adversary may also try to fabricate evidence retroactively for a package that was not preserved at $t_p$.

The model assumes collision resistance of the hash function used for package and Merkle commitments, correctness of Merkle inclusion verification, unforgeability of the post-quantum signing credential used by $T_{PQ}$ during its validity interval, a trustworthy time assertion by the timestamping service, and an honest validation result at $t_p$. It also assumes that the evidence token is generated before the relevant classical algorithm becomes practically forgeable for the adversary considered.

We assume that the post-quantum signature scheme used to protect $T_{PQ}$ remains existentially unforgeable against quantum-capable adversaries, under the security assumptions and attack model applicable to the chosen ML-DSA or SLH-DSA instantiation [National Institute of Standards and Technology 2024a, National Institute of Standards and Technology 2024b].

### 3.3. Security Goals

The goals below are stated for the preserved package $P_i$ and evidence record $ER_{t_p}$, not for the original classical signature as a standalone object. This follows from the migration objective: the first post-quantum step protects an observed validation state and its evidence,

while the document-signing infrastructure may still be classical. Each goal therefore has to be read together with the timeline condition $t_s < t_p < t_q$ and the assumptions stated above.

**Preserved-state integrity.** After $t_p$, an adversary should not be able to replace $P_i$ with a different package $P_i'$ that verifies against the same preserved evidence record, except by breaking the hash commitment, Merkle binding, or post-quantum timestamp token.

**Existence-before-time evidence.** A verifier should obtain evidence that the committed package state existed no later than the time asserted by $T_{PQ}$, subject to the trust assumptions of the timestamping service. A plain post-quantum signature over a metadata field containing $t_p$ would bind the root to that metadata, but would not by itself establish a trustworthy time assertion.

**Validation-state preservation.** The evidence layer should preserve the validation result observed at $t_p$, including the certificate chain, revocation material, validation policy, and status information needed to interpret that result later.

**Migration non-disruption.** The first PQC migration step should not require immediate replacement of the original document signature, certificate hierarchy, client software, or signing workflow.

**Algorithmic agility.** The evidence layer should support later renewal or parallel evidence chains when algorithms, parameters, or policy requirements change.

### 3.4. Non-Goals and Residual Risks

The construction does not make $\sigma_{C,i}$ post-quantum secure. A verifier after $t_q$ should not interpret the classical document signature alone as fresh quantum-safe authentication. The relevant protected object is the preserved package and its validation state at $t_p$.

The construction also does not detect a classical signature that was already forged before $t_p$, an incorrect validation result produced by a faulty validator, an inadequate preservation policy, or evidence created only after the classical algorithm was already practically compromised. If ML-DSA or another post-quantum credential used for $T_{PQ}$ is later weakened, the affected evidence chain must be renewed or complemented by another chain. Likewise, hash functions and timestamping credentials remain lifecycle dependencies that require monitoring and renewal.

## 4. Post-Quantum Evidence-Layer Construction

This section defines the target construction. The document-signing system may continue to produce classical signatures, while the evidence layer captures and protects the validation state with post-quantum evidence.

Fig. 1 separates the preservation-time construction from later lifecycle operations. The solid arrows describe the capture performed at $t_p$: the classical signature is validated, the resulting package is hashed, the digest is included in a Merkle tree, and the root is protected by a post-quantum timestamp token. The dashed arrows indicate post-preservation operations rather than additional intake steps: a renewal at $t_r > t_p$ appends fresh timestamp evidence before token expiration or algorithm/policy deprecation, while a verifier at

$t_v \geq t_p$ recomputes the package digest, inclusion path, root, and token chain. Thus, later verification occurs at $t_v$, but the reported validation state remains the one preserved at $t_p$.

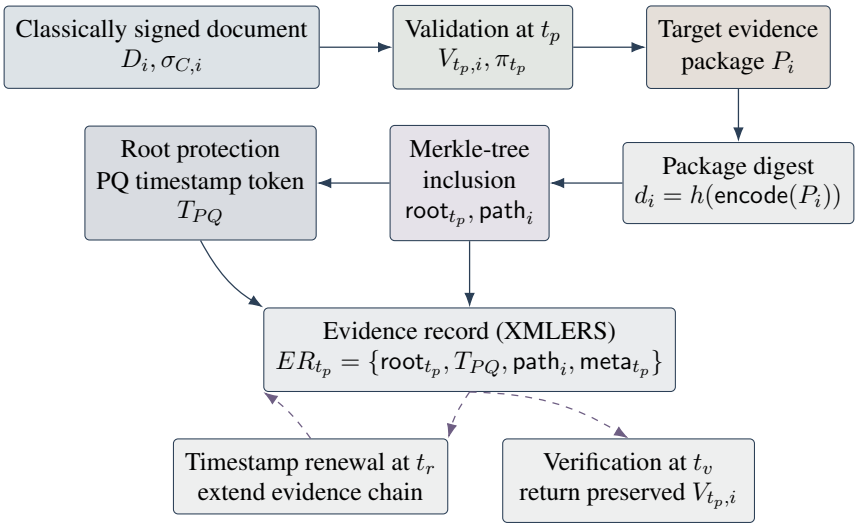


**Figure 1. Conceptual workflow of the post-quantum evidence layer.**

### 4.1. Evidence Capture and Commitment

For each signed document, the preservation service validates the classical signature under the policy in force at $t_p$, unambiguously encodes $P_i$, and computes $d_i = h(\mathsf{encode}(P_i))$. The package binds the document digest and classical signature to the certification material, revocation evidence, validation result, and policy under which that result was obtained.

For a batch of $n$ packages, the evidence layer computes

$$\mathsf{root}_{t_p} = \mathsf{MerkleRoot}(d_1, \ldots, d_n).$$

Each package receives an inclusion path $\mathsf{path}_i$ sufficient to recompute $\mathsf{root}_{t_p}$ from $d_i$. The root is the compact commitment that separates document-level legacy signatures from preservation-layer post-quantum protection.

### 4.2. Post-Quantum Evidence Generation

The preservation service protects $\mathsf{root}_{t_p}$ with a post-quantum timestamp token

$$T_{PQ} = \mathsf{Timestamp}_{PQ}(\mathsf{root}_{t_p}).$$

The SHA-512 root digest is placed in the `hashedMessage` field of the RFC 3161 `MessageImprint`, alongside the SHA-512 identifier. The token also carries responder-generated time, a TSA policy, and signer information; document-validation policy is protected through its inclusion in the committed package. A plain post-quantum signature would provide binding and signer authentication but not trustworthy existence-before-time, so the model requires an independently trusted time assertion.

The construction also allows parallel evidence chains over the same root. For example, a high-assurance deployment may store both $T_{\text{ML-DSA}}$ and $T_{\text{SLH-DSA}}$ for the same $\mathsf{root}_{t_p}$, trading storage and processing cost for diversity of assumptions. The experiments in this paper evaluate single-chain costs and use the measured token and evidence-record sizes to discuss this trade-off.

### 4.3. Verification Procedure

Given $(P_i, \mathsf{path}_i, ER_{t_p})$, a verifier performs the following checks:

1. recompute $d_i = h(\mathsf{encode}(P_i))$;
2. use $\mathsf{path}_i$ to recompute the committed root;
3. verify that the recomputed root matches $\mathsf{root}_{t_p}$ in $ER_{t_p}$;
4. verify that $T_{PQ}$ carries the root digest as the `hashedMessage` of its SHA-512 `MessageImprint`, then evaluate its signature, asserted time, TSA policy, and trust chain;
5. check that $t_p$ belongs to an acceptable migration window, in particular that the preserved validation happened before the relevant classical algorithm was considered practically compromised;
6. output the preserved validation state $V_{t_p,i}$ and its policy context.

The last step is essential. After $t_q$, the verifier's output is the preserved validation state and its evidence context. The evidence supports the claim that a particular validation state existed and was committed before the transition point.

### 4.4. Assumption-Based Security Rationale

The security argument follows by composition. Modifying any field of an unambiguously encoded package changes its leaf unless hash binding fails; a different leaf no longer reconstructs the sealed root through the recorded Merkle path; and replacing that root requires a valid post-quantum timestamp token. Under the hash, Merkle, credential, and trusted-time assumptions and the window $t_s < t_p < t_q$, the resulting representation is therefore tamper-evident. The guarantee applies to the evidence accepted and committed at intake, so the completeness and correctness of $P_i$ remain preservation-policy requirements.

## 5. Experimental Methodology

The experiments measure application-level CMS validation, hashing of received CMS bytes, one-leaf root/token issuance, XMLERS extraction, a separate 1–32-leaf proof check, and direct CMS signing and verification. Table 2 distinguishes the complete target construction from the scope implemented and evaluated in the prototype.

Accordingly, the results characterize the implemented intake, commitment, and token-generation steps; they do not constitute a complete implementation of the enriched $P_i$ or of a production trusted-time service. In this implementation, the preserved leaf was $d_i = \mathrm{SHA\text{-}512}(A_i^{\mathrm{CMS}})$, where $A_i^{\mathrm{CMS}}$ is the encapsulated CMS object received by PreserveEU. The CMS bytes bind their encapsulated content, signature, and certificate material, while the target package $P_i$ additionally models the validation result, policy, preservation identifier, and separately obtained evidence required by a complete deployment.

All experiments were executed locally on a single x86_64 machine with an Intel Core i7-9750H CPU at 2.60 GHz (6 physical cores and 12 threads), 32 GB of RAM, under Linux 6.17.0-14-generic. This environment bounds the interpretation of the reported measurements.

**Table 2. Target construction versus evaluated implementation.**

| Stage | Target construction | Evaluated implementation |
|---|---|---|
| Intake validation | Validate each classical signature under the applicable policy and record its validation state | RSA CMS intake validation to `TOTAL_PASSED` |
| Evidence package $P_i$ | Encode the document digest, signature, certification and revocation material, validation result, and policy | Commit received CMS bytes; the enriched $P_i$ encoding is not implemented |
| Merkle commitment | Batch package digests and retain their inclusion paths | SHA-512 one-leaf records and a separate 1–32-leaf proof-growth check |
| PQ timestamp token | An independently trusted TSA protects the root with a PQC signature and time assertion | A local RFC 3161-compatible issuer measures issuance, encoding, and XMLERS size, without an independent time source |

### 5.1. Prototype Basis and Implementation Scope

The prototype extends PreserveEU, an open-source evidence-record service [PreserveEU 2025], from public revision `11129c3` (version 1.0.1-SNAPSHOT, Java 17, DSS 6.3) at `https://code.europa.eu/dit/preserve-eu`. PreserveEU receives signed objects, orchestrates DSS validation, builds SHA-512 Merkle trees, routes RFC 3161 requests, persists the resulting data, and extracts XMLERS records. We use PreserveEU as the experimental testbed and preservation substrate, not as the conceptual contribution of this paper.

The experimental adaptation added ML-DSA and SLH-DSA algorithm identifiers and mappings, PQC-capable dependencies, PKCS#12-backed local token generation, experimental trust chains, and measurement instrumentation for issuing and encoding post-quantum-protected root tokens. Direct CMS signing and verification were performed with a separate experimental implementation developed at LabSEC, the Computer Security Laboratory at UFSC, and adapted to support the same classical and post-quantum algorithm configurations. The cited public PreserveEU revision provides the reference baseline for the preservation workflow, while the measurements reported here characterize the adapted experimental configuration described in this section.

The system was provisioned with the X.509 chains and revocation artifacts required by the seven configurations. PreserveEU validated all inputs; only the RSA subset is reported as the legacy-intake budget. The CMS inputs were not CAdES and no T/LT/LTA profile augmentation was performed during intake, so the working document remained the received CMS object.

### 5.2. Algorithms and Certificates

The corpus used pure ML-DSA-44/65/87 signatures with SHA-512 as the CMS digest, SLH-DSA-SHA2-128F with SHA-256, SLH-DSA-SHAKE-256F with SHAKE-256, and 4096-bit RSA with SHA-256 or SHA-512. These are cryptographic artifact classes, not document formats. Independently of these document-level choices, all Merkle nodes and RFC 3161 root imprints used SHA-512.

Each configuration signed the same four PDFs ten times each. The input sizes were 18.4, 139.4, 458.5, and 1017.7 KiB, yielding 40 encapsulated CMS objects per

configuration and 280 objects overall. Their raw bytes were hashed with SHA-512 and used as leaves in the evidence-record workflow; in the issuance workload, each object formed a separate one-leaf root. The complementary multi-leaf proof check used binary grouping, with each parent computed as $p = \text{SHA-512}(c_{\min} \parallel c_{\max})$, where child digests were ordered lexicographically as unsigned byte strings; an unpaired node was carried to the next level unchanged.

The two SLH-DSA configurations represent different parameter and security profiles; they were selected to expose contrasting implementation costs, not to isolate the effect of SHA-2 versus SHAKE or to provide an equal-security comparison.

### 5.3. Legacy Signature Intake Validation

All 280 CMS objects were submitted four times, producing 1,120 validation records. The legacy-intake analysis uses 160 observations for each RSA digest configuration. The measured interval covers application-level CMS validation and report generation, excluding storage access, initial processing setup, and instrumented delays caused by external resource retrieval. `TOTAL_PASSED` is the DSS/ETSI indication that all applicable validation constraints passed. These measurements characterize the application-level intake stage before evidence capture.

### 5.4. Local Root-Token Issuance Experiment

Each token-credential run targeted 280 separate one-leaf roots, one for each CMS object, yielding $n = 280$ observations per algorithm. Root creation and token issuance were triggered for each object. The measured interval covers local request processing, token and certificate-chain construction, cryptographic signing, and response encoding. It excludes Merkle construction, network transport, persistence, and external TSA behavior. Because PQC timestamping services with the required experimental profiles were unavailable, the evaluation used a locally operated RFC 3161-compatible issuer. The resulting measurements isolate token-generation and encoding costs; the trusted-time property in the target model remains a responsibility of an independently operated TSA. One representative one-leaf XMLERS size is reported per algorithm, and a separate 1–32-leaf check measures proof growth.

### 5.5. Direct CMS Signing and Verification Baseline

The direct CMS baseline recorded application-level signing time, verification time, and size variation using the separate experimental implementation. Signing covers the application processing required to produce a CMS object, with 40 observations per configuration (four inputs $\times$ ten repetitions). Verification processes those 40 CMS objects over four cycles, yielding 160 observations per configuration and covering the associated container, certificate, signature, and result processing.

Measurements were collected in warm state after discarded preliminary rounds. Timings are descriptive, single-host measurements. We report the arithmetic mean, standard deviation, and median over all classified observations without post-hoc outlier removal. The median complements the conventional mean and standard deviation because several timing distributions were right-skewed. Repeated observations of the same artifacts are not treated as independent population samples, and no inferential claims are made.

### 5.6. Measured Metrics

The reported metrics therefore separate the legacy intake budget, local token-issuance/XMLERS cost, and direct CMS signing and verification comparison.

## 6. Evaluation

This section follows the measured pipeline: legacy RSA intake, local root-token/XMLERS cost, and direct CMS signing and verification baselines.

### 6.1. Proposed-Flow Step: Legacy RSA Intake Validation

For the RSA subset, DSS report generation reached `TOTAL_PASSED` in $100.3 \pm 42.6$ ms with RSA/SHA-256 (median 91; IQR 68–122; $n = 160$) and $108.2 \pm 46.0$ ms with RSA/SHA-512 (median 96; IQR 76–124; $n = 160$). These values characterize the application-level validation preceding evidence capture, including CMS, certificate, policy, and report processing.

### 6.2. Evidence-Layer Cost: Local Root-Token Issuance

After tree construction, local root-token issuance exercises the operation that adds post-quantum protection to the committed set. Table 3 reports timing summaries and representative measured one-leaf XMLERS sizes.

**Table 3. Local root-token issuance and one-leaf XMLERS size. Time is mean $\pm$ standard deviation (median), in ms, with $n = 280$ per algorithm; XMLERS is one representative measured record per algorithm.**

| Algorithm | Token time | XMLERS (KiB) |
|---|---|---|
| ML-DSA-44 | $5.1 \pm 8.1$ (median 3.0) | 23.00 |
| ML-DSA-65 | $7.2 \pm 5.1$ (median 6.0) | 30.13 |
| ML-DSA-87 | $7.2 \pm 5.6$ (median 6.0) | 39.49 |
| RSA/SHA-512 | $34.7 \pm 14.8$ (median 29.0) | 9.95 |
| SLH-DSA-SHA2-128F | $152.4 \pm 28.1$ (median 147.0) | 94.42 |
| SLH-DSA-SHAKE-256F | $459.5 \pm 88.8$ (median 436.5) | 265.27 |

The large standard deviation for ML-DSA-44 reflects a right-skewed timing distribution: its median was 3.0 ms and its IQR was 2–6 ms. Differences among the three ML-DSA medians are too small to rank those parameter sets under host jitter. At family level, ML-DSA was faster and produced smaller records than the evaluated SLH-DSA variants on this host. In the evaluated XMLERS serialization, the separate 1–32-leaf check added approximately 215 bytes per tree level for inclusion-path hashes.

### 6.3. Comparative Baseline: CMS Artifact Size Overhead

Table 4 reports per-input overhead because a pooled percentage would conceal its dependence on input size.

The relative overhead decreases with input size: ML-DSA falls to 1.1–1.7% for the largest input but is 41.6–71.8% for the smallest; SLH-DSA is larger throughout. These direct-signing values contextualize, rather than form part of, evidence-layer migration.

**Table 4. CMS size overhead (%) by input size, rounded to the nearest KiB.**

| Algorithm | 18 | 139 | 459 | 1,018 |
|---|---|---|---|---|
| RSA/SHA-256 | 16.8 | 2.6 | 1.1 | 0.7 |
| RSA/SHA-512 | 17.0 | 2.6 | 1.1 | 0.7 |
| ML-DSA-44 | 41.6 | 5.8 | 2.0 | 1.1 |
| ML-DSA-65 | 54.4 | 7.5 | 2.6 | 1.4 |
| ML-DSA-87 | 71.8 | 9.8 | 3.3 | 1.7 |
| SLH-DSA-SHA2-128F | 190.7 | 25.5 | 8.0 | 3.8 |
| SLH-DSA-SHAKE-256F | 539.5 | 71.4 | 22.0 | 10.1 |

### 6.4. Direct CMS Signing and Verification

Table 5 summarizes application-level CMS signing and internal verification using all observations pooled across the four input sizes. The median complements the mean and standard deviation for the right-skewed configurations.

**Table 5. Direct CMS times (ms): mean $\pm$ standard deviation (median); $n = 40$ for signing and $n = 160$ for verification.**

| Algorithm | Signing | Verification |
|---|---|---|
| ML-DSA-44 | 28.0 $\pm$ 7.6 (median 22.9) | 9.5 $\pm$ 8.3 (median 6.1) |
| ML-DSA-65 | 24.3 $\pm$ 14.3 (median 19.9) | 9.2 $\pm$ 6.9 (median 7.1) |
| ML-DSA-87 | 23.4 $\pm$ 22.9 (median 20.3) | 9.2 $\pm$ 6.7 (median 7.9) |
| RSA/SHA-256 | 35.7 $\pm$ 31.8 (median 30.2) | 45.0 $\pm$ 25.1 (median 33.7) |
| RSA/SHA-512 | 33.4 $\pm$ 19.3 (median 29.6) | 45.2 $\pm$ 23.3 (median 37.9) |
| SLH-DSA-SHA2-128F | 110.2 $\pm$ 15.0 (median 108.2) | 13.9 $\pm$ 5.7 (median 12.3) |
| SLH-DSA-SHAKE-256F | 406.8 $\pm$ 12.1 (median 402.8) | 23.3 $\pm$ 7.5 (median 22.5) |

The apparently slower RSA verification than signing does not contradict RSA exponentiation costs because the columns measure different application paths. Verification includes CMS parsing, certificate processing, signature verification, and result construction, whereas signing measures CMS generation around the private-key operation. The direct-signing baseline also shows that post-quantum CMS signing is technically feasible in this environment. The evidence-layer rationale is therefore not that classical signatures are always cheaper, but that staged protection avoids immediately replacing deployed PKI, policy, client, and workflow dependencies.

## 7. Security and Deployment Discussion

The security gain is limited to the preserved validation state: evidence generated during the migration window can later be checked through a post-quantum evidence chain, but the original classical document signature is not made post-quantum secure. Table 6 summarizes the corresponding guarantees and limits.

On this software-only host, ML-DSA had the lowest PQC token latency and smaller records than the evaluated SLH-DSA variants, making it the measured efficiency candidate. Existing SHA-2 acceleration, providers, infrastructure compatibility, or HSM support may change this trade-off. SLH-DSA consequently remains relevant as an assumption-diversity option. Parallel chains and renewal can reduce dependence on one family when supported by an appropriate lifecycle policy.

The evaluation covers RSA intake, CMS-byte commitments, and separate one-leaf root-token issuance on one software-only host; the 1–32-leaf experiment measures proof

**Table 6. Security guarantees and limits of the target evidence-layer model.**

| Situation | Guarantee provided | Limit |
|---|---|---|
| $t_s < t_p < t_q$ and correct validation | Preserves integrity and validation state before quantum compromise. | Does not make RSA/ECDSA quantum-safe. |
| Adversary replaces $D$ or $P_i$ after $t_p$ | Detected by package hash, Merkle path, and $T_{PQ}$. | Depends on hash, Merkle, and PQ timestamp-token assumptions. |
| Classical signature breaks after $t_p$ | Preserved state remains protected by the evidence layer. | New classical forgeries may exist outside the repository. |
| Classical signature was forged before $t_p$ | Preserves what was observed by the validator. | Does not establish original authenticity. |
| Validator or policy wrong at $t_p$ | Evidence preserves the recorded result and policy context. | Does not correct validation or policy errors. |
| PQ credential approaches deprecation | Timely renewal can protect prior evidence. | Must precede practical compromise. |
| ML-DSA plus SLH-DSA | Adds diversity of assumptions over the same root. | Increases processing and storage cost. |

growth rather than batched issuance performance. ECDSA intake, CAdES T/LT/LTA elevation, enriched serialization of validation and revocation evidence into $P_i$, independently operated PQC TSAs, timestamp and hash renewal, hardware-assisted execution, distributed operation, and combined-chain extraction remain to be evaluated. Network, queueing, and persistence costs are also outside the measured intervals. These dimensions affect the extrapolation of the reported costs to production long-term-validation deployments.

## 8. Conclusion

This paper specified a post-quantum evidence-layer model for incremental migration of legacy-bound digital signature systems. The model defines a security and deployment boundary around evidence captured before classical compromise and specifies how established evidence-record mechanisms can protect signatures together with their validation context while existing signing workflows remain operational.

The partial PreserveEU-based implementation demonstrated CMS-byte commitment and local PQC root-token/XMLERS generation. In the evaluated software environment, ML-DSA offered the lowest measured PQC root-token latency, while SLH-DSA provided a higher-cost option for diversity of assumptions. Direct post-quantum CMS signing was also technically feasible in this environment, but this baseline does not account for the PKI, policy, client, and workflow changes that motivate evidence-layer migration. The proposed evidence-layer step protects the evidence of a validation state while leaving the original signing workflow unchanged. Extending the implementation with enriched package encoding, independently operated PQC timestamping, and evidence renewal constitutes the next step toward production deployment.

## References

Adams, C., Cain, P., Pinkas, D., and Zuccherato, R. (2001). Internet x.509 public key infrastructure time-stamp protocol (tsp). RFC 3161.

Alnahawi, N., Westerbaan, B., and Stebila, D. (2023). On the state of crypto-agility. *IACR Cryptology ePrint Archive*. Report 2023/487.

Bernstein, D. J., Hülsing, A., Rijneveld, J., Niederhagen, R., Schanck, F. S., and Schwabe, P. (2019). The sphincs+ signature framework. *IACR Cryptology ePrint Archive*. Report 2019/1086.

Blazic, A. J., Saljic, S., and Gondrom, T. (2011). Extensible markup language evidence record syntax (XMLERS). RFC 6283.

Brînzea, A., Leancă, R.-A., Aciobăniţei, I., and Pop, F. (2025). Standard-compliant blockchain anchoring for timestamp tokens. *Applied Sciences*, 15(23):12722.

Clupek, V., Malina, L., and Zeman, V. (2015). Secure digital archiving in post-quantum era. In *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, pages 622–626. IEEE.

Driscoll, M. B., Wallace, C., Gray, J., and Housley, R. (2025). Terminology for post-quantum traditional hybrid schemes. RFC 9794.

Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., and Stehlé, D. (2018). Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238–268.

ETSI (2017). ETSI TS 119 122-3 V1.1.1: CAdES digital signatures; part 3: Incorporation of evidence record syntax mechanisms in CAdES. Technical report, ETSI.

ETSI (2021). ETSI EN 319 122-1 V1.2.1: CAdES digital signatures; part 1: Building blocks and CAdES baseline signatures. Technical report, ETSI.

ETSI (2024a). ETSI EN 319 102-1 V1.4.1: Procedures for creation and validation of AdES digital signatures; part 1: Creation and validation. Technical report, ETSI.

ETSI (2024b). ETSI EN 319 132-1 V1.3.1: XAdES digital signatures; part 1: Building blocks and XAdES baseline signatures. Technical report, ETSI.

ETSI (2024c). ETSI EN 319 142-1 V1.2.1: PAdES digital signatures; part 1: Building blocks and PAdES baseline signatures. Technical report, ETSI.

Fall, A. A. et al. (2025). Sok: Systematizing hybrid strategies for the transition to post-quantum cryptography. *IACR Cryptology ePrint Archive*. Report 2025/2052.

Geihs, M., Demirel, D., and Buchmann, J. (2015). On the security of long-lived archiving systems based on the evidence record syntax. In *C2SI 2015*, pages 27–44. Springer.

Gondrom, T., Brandner, R., and Pordesch, U. (2007). Evidence record syntax (ers). RFC 4998.

Housley, R., Fluhrer, S., Kampanakis, P., and Westerbaan, B. (2025). Use of the SLH-DSA signature algorithm in the cryptographic message syntax (CMS). RFC 9814.

Instituto Nacional de Tecnologia da Informação (2020). DOC-ICP-15: Visão Geral sobre Assinaturas Digitais na ICP-Brasil. Infraestrutura de Chaves Públicas Brasileira.

International Organization for Standardization (2025). Iso 14721:2025 space data system practices — reference model for an open archival information system (oais). International Standard. Identical in content to CCSDS 650.0-M-3, December 2024.

Joseph, D., Misoczki, R., Manzano, M., Campagna, M., Albrecht, M., Tang, A., Khovratovich, D., Schwabe, P., Stebila, D., and Hansen, R. (2022). Transitioning organizations to post-quantum cryptography. *Nature*, 605(7909):237–243.

Kusber, T., Schwalm, S., Korte, U., and Shamburger, K. (2021). Records management and long-term preservation of evidence in DLT. In *Open Identity Summit 2021 (LNI P-312)*, pages 131–142. Gesellschaft für Informatik.

Moody, D., Perlner, R., Regenscheid, A., Robinson, A., and Cooper, D. (2024). Transition to post-quantum cryptography standards. NIST Interagency/Internal Report 8547 (Initial Public Draft), National Institute of Standards and Technology.

National Institute of Standards and Technology (2024a). FIPS 204: Module-lattice-based digital signature standard. Federal Information Processing Standards Publication 204, National Institute of Standards and Technology.

National Institute of Standards and Technology (2024b). FIPS 205: Stateless hash-based digital signature standard. Federal Information Processing Standards Publication 205, National Institute of Standards and Technology.

National Institute of Standards and Technology (2026). Algorithm registration — computer security objects register (CSOR). Web page. Accessed: 2026-03-10.

NIST (2026). Considerations for achieving cryptographic agility: Strategies and practices. Cybersecurity White Paper 39upd1, National Institute of Standards and Technology.

NIST NCCoE (2026). Migration to post-quantum cryptography. Web page. Accessed: 2026-03-11.

Ounsworth, M., Gray, J., Pala, M., Klaußner, J., and Fluhrer, S. (2026). Composite module-lattice-based digital signature algorithm (ML-DSA) for use in x.509 public key infrastructure. Internet-Draft draft-ietf-lamps-pq-composite-sigs-19, Internet Engineering Task Force. Work in progress.

PreserveEU (2025). *ePreservation*. PreserveEU. Project documentation describing the ePreservation architecture, XML Evidence Records, Merkle-tree preservation, and timestamp/hash-tree renewal workflows.

Salter, B. S., Raine, A., and Geest, D. V. (2025). Use of the ML-DSA signature algorithm in the cryptographic message syntax (CMS). RFC 9882.

Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS '94)*, pages 124–134. IEEE.

Shostack, A. (2014). *Threat Modeling: Designing for Security*. John Wiley & Sons.

Thiel, C. and Thiel, C. (2021). Quantum computer resistant cryptographic methods and their suitability for long-term preservation of evidential value. In *34th Bled eConference*, pages 481–493. University of Maribor Press.