Analysing the Post-Quantum Security of S/MIME

Sayan Das* and Anupam Chattopadhyay

College of Computing and Data Science,
Nanyang Technological University, Singapore
sayan005@e.ntu.edu.sg, anupam@ntu.edu.sg

Abstract. Secure/Multipurpose Internet Mail Extensions (S/MIME) is a standards-based mechanism for certificate-backed email signing and encryption. Its post-quantum migration is now technically actionable: ML-KEM public keys can be represented in X.509 certificates, and CMS can carry ML-KEM recipient information through `KEMRecipientInfo`. These standards solve an encoding problem, but they do not by themselves solve an assurance problem. A mailbox may possess a post-quantum-capable certificate while a generated CMS `EnvelopedData` object still protects the content-encryption key (CEK) through RSA or elliptic-curve key management; a multi-recipient message may mix ML-KEM and classical recipient paths for the same CEK; and archived mail may remain protected only by quantum-vulnerable mechanisms. This paper studies this assurance gap by shifting the unit of analysis from certificates to encrypted messages. We model encrypted S/MIME as a multi-recipient CMS object with certificate-bound paths to a shared CEK and show that post-quantum confidentiality is a universal message-level property: every valid path to the CEK must satisfy the active migration policy. Consequently, the presence of one ML-KEM recipient path is insufficient if another valid classical path can recover the same CEK. We then present `SMIME-PQCheck`, a standards-driven validation framework that combines X.509 profile checks, CMS recipient-structure analysis, policy-driven hybrid handling, and archive-level risk aggregation. The framework classifies S/MIME objects as `pqc-protected`, `hybrid-protected`, `classical-only`, `unsafe-mixed-mode`, `invalid`, or `unknown`. The result is a practical assurance layer for organizations migrating S/MIME deployments from certificate-level PQC readiness to auditable message-level protection.

Keywords: CMS, X.509, Post-Quantum Cryptography, ML-KEM, Migration Assurance

1 Introduction

Secure/Multipurpose Internet Mail Extensions (S/MIME) protects MIME email by encapsulating signing and encryption operations in the Cryptographic Message Syntax (CMS) and binding public keys to mailbox identities through X.509

* Corresponding author: sayan005@e.ntu.edu.sg

certificates. S/MIME version 4.0 defines signatures for authentication, message integrity, and non-repudiation of origin, and encryption for confidentiality [1]; its certificate-handling profile relies on X.509 public-key infrastructure (PKI) validation [2,3]. S/MIME is not merely a legacy specification. A recent large-scale measurement study collected more than 41 million unique X.509 certificates from public LDAP address books, of which 38 million were suitable for S/MIME clients [4]. This deployment footprint matters because encrypted email is persistent: messages may remain in mailboxes, enterprise archives, backup systems, and legal-discovery repositories for years.

The transition to post-quantum cryptography (PQC) is therefore urgent for S/MIME. Shor's algorithm breaks the hardness assumptions underlying RSA and elliptic-curve public-key cryptography [5], while NIST has standardized ML-KEM for key establishment and ML-DSA for signatures [6,7]. The S/MIME standards ecosystem has also begun to incorporate these algorithms. RFC 9935 specifies ML-KEM algorithm identifiers and public-key representation for X.509 certificates [8]; RFC 9936 specifies the use of ML-KEM in CMS using the `KEMRecipientInfo` structure defined by RFC 9629 [9,10]; and RFC 9881 specifies ML-DSA identifiers for X.509 [11]. The CA/Browser Forum has also introduced specifications for PQC algorithms in the S/MIME Baseline Requirements to support experimentation with publicly trusted S/MIME certificates [12,13].

These developments address what we call the *encoding problem*: how to represent post-quantum or hybrid cryptographic mechanisms inside existing protocol objects. In S/MIME, this means encoding an ML-KEM public key in an X.509 certificate, representing the corresponding algorithm identifiers, and carrying an ML-KEM recipient mechanism inside CMS. Encoding support is necessary for migration, but it does not by itself provide security assurance for a delivered message.

The missing step is the *assurance problem*: deciding whether the produced application object actually satisfies a security policy once all usable cryptographic paths are considered. In S/MIME, the protected MIME content is encrypted once under a content-encryption key (CEK), and each CMS recipient entry is an alternative mechanism for recovering that same CEK. Therefore, the presence of one correctly encoded PQC recipient path does not imply message-level post-quantum confidentiality. For example, a CMS object may contain an ML-KEM recipient path for the intended mailbox and an RSA recipient path for an enterprise archive. The object is PQC-encoded for one recipient, but it does not satisfy strict PQC assurance because the RSA path can independently recover the same CEK. A quantum adversary, or an attacker who compromises the archive key, can recover the CEK through the classical path and decrypt the encrypted content.

This encoding–assurance gap is not unique to S/MIME. Browser deployments, email clients, VPNs, secure messaging systems, code-signing workflows, and document-signing systems all face a related challenge: protocol-level support for PQC does not automatically prove that a particular session, object, or artifact satisfies a migration policy. For example, Chrome has deployed hybrid

post-quantum key exchange and has moved from experimental Kyber-based negotiation toward ML-KEM-based hybrid key exchange [14,15]. The IETF PQC migration use-case work discusses deployment settings such as PKI, S/MIME certificates, code signing, and identity systems [16]. In the email ecosystem, hybrid PQC S/MIME prototypes and discussions around Outlook, Thunderbird, and certificate migration further show that end-user applications require validation evidence, not merely new encodings [17]. We use S/MIME as a concrete case study because CMS explicitly exposes multiple recipient paths to the same CEK, making the assurance gap precise and operationally testable. Against this background, this paper studies the following research question:

> *Given an S/MIME encrypted message, its CMS recipient structures, and the relevant certificate evidence, how can an organization decide whether the message is actually post-quantum protected under an explicit migration policy?*

The question is intentionally narrower than designing a new S/MIME protocol: we validate produced artifacts, including X.509 certificates, CMS `EnvelopedData` objects, recipient-path algorithms, and policy decisions. This scope matches practical migration workflows in which operators must classify existing mail, detect unsafe fallback, and produce evidence that a message or archive satisfies a migration policy. Our core observation is that the confidentiality of an S/MIME object is bounded by the weakest valid path to the CEK. Thus, a message with both ML-KEM and standalone RSA/ECDH recipient paths is not fully PQC-protected, because the classical path can still recover the same CEK. We call this an *unsafe mixed-mode* object. Hybrid mechanisms are acceptable only when explicitly allowed by policy and when no independent classical fallback remains. This paper makes the following contributions:

C1. Encoding–assurance distinction for PQC S/MIME migration. We identify the gap between syntactically encoding PQC mechanisms in X.509/CMS artifacts and obtaining message-level assurance that every usable CEK-recovery path satisfies an explicit migration policy.

C2. Message-level formulation of PQC S/MIME assurance. We formalize encrypted S/MIME as a multi-recipient CMS object with certificate-bound paths to a shared CEK, and we state a universal acceptance rule: every valid path to the CEK must satisfy the active PQC migration policy.

C3. Policy-aware treatment of mixed and hybrid migration. We distinguish between an approved hybrid recipient path whose derived KEK depends on policy-recognized components and an unsafe mixed-mode object in which a standalone classical path can independently recover the CEK.

C4. A validation framework for deployed artifacts. We propose SMIME-PQCheck, a verifier that combines X.509 profile checks, CMS recipient-structure analysis, OID-registry based classification, and versioned policy evaluation to classify messages as `pqc-protected`, `hybrid-protected`, `classical-only`, `unsafe-mixed-mode`, `invalid`, or `unknown`.

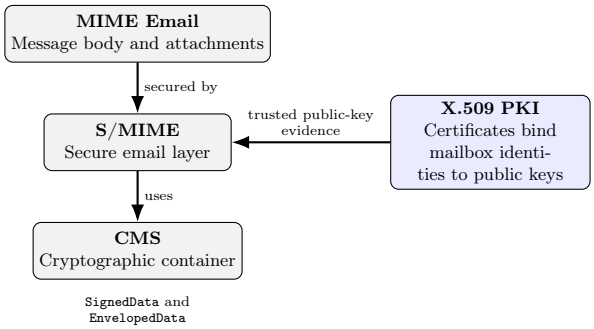

Fig. 1. Layering of MIME email, S/MIME, CMS, and X.509 PKI. S/MIME protects MIME content; CMS provides the cryptographic object format; and X.509 certificates provide identity-bound public keys for signature verification and encryption/key management.

Organization. Section 2 reviews S/MIME, CMS, X.509, and related PQC migration approaches. Section 3 formalizes the validation problem, defines the message-level acceptance rule, and presents the SMIME-PQCheck framework, prototype implementation, and archive-analysis logic. Section 4 concludes.

2 Background and Migration Context

CMS is a general syntax for digitally signing, digesting, authenticating, and encrypting arbitrary content [18]. S/MIME uses CMS objects inside MIME email. Figure 1 summarizes the relation between MIME, S/MIME, CMS, and X.509 PKI. The CMS content type most relevant to this paper is `EnvelopedData`. In `EnvelopedData`, the sender encrypts the content with a symmetric CEK and stores one or more recipient-specific structures that allow authorized recipients to recover that CEK.

Let Alice send an encrypted S/MIME message to Bob. Bob has a certificate Cert_B containing a public key suitable for encryption or key management. In a classical deployment, this key may be RSA or elliptic-curve based. In a PQC deployment, it may be an ML-KEM public encapsulation key encoded according to RFC 9935 [8]. Alice obtains Cert_B from a prior signed message, an enterprise directory, a public address book, LDAP, or manual import. She validates the certificate chain, mailbox identity, validity period, revocation state, key usage, extended key usage, and applicable policy constraints. Alice then samples a fresh CEK K and encrypts the MIME payload M:

$$K \leftarrow \{0,1\}^{\lambda}, \qquad C = \mathsf{Enc}_K(M),$$

where λ is the CEK security parameter and C is the encrypted content stored in CMS.

For an ML-KEM recipient path, Alice computes

$$(ct_B, ss_B) \leftarrow \mathsf{Encaps}(ek_B), \tag{1}$$
$$\mathsf{KEK}_B = \mathsf{KDF}(ss_B, \mathsf{ctx}), \tag{2}$$
$$w_B = \mathsf{Wrap}_{\mathsf{KEK}_B}(K), \tag{3}$$
$$\mathsf{RI}_B = (ct_B, w_B, \mathsf{AlgIDs}, \mathsf{rid}_B), \tag{4}$$

where ek_B is Bob's ML-KEM public encapsulation key, ct_B is the KEM ciphertext, ss_B is the KEM shared secret, ctx denotes context information bound into the KDF, KEK_B is the key-encryption key derived from ss_B, w_B is the wrapped CEK, AlgIDs records the relevant algorithm identifiers, and rid_B identifies the recipient. Bob later uses his decapsulation key dk_B to recover ss_B, derives KEK_B, unwraps K, and decrypts C. RFC 9629 defines the generic `KEMRecipientInfo` pattern, and RFC 9936 instantiates it for ML-KEM in CMS [10,9].

For multiple recipients, CMS normally encrypts the content once and protects the same CEK separately for each recipient. This design is efficient and standard-compatible, but it creates the assurance gap studied in this paper: recipient structures are alternative routes to the same plaintext. Therefore, a PQC migration checker must inspect the CMS recipient structures, not only the certificates. Figure 2 shows the relevant evidence flow.

Standards and Migration Context. The standards landscape now contains the pieces needed to encode post-quantum S/MIME artifacts. RFC 9935 specifies ML-KEM public-key identifiers and X.509 representation, including the ML-KEM public-key byte strings for ML-KEM-512, ML-KEM-768, and ML-KEM-1024 and the associated certificate-usage constraints [8]. RFC 9936 specifies that ML-KEM may be used for one or more recipients in CMS `EnvelopedData`, `AuthenticatedData`, and `AuthenticatedEnvelopedData`, using `KEMRecipientInfo` [9]. RFC 9629 provides the generic KEM interface for CMS and is not limited to ML-KEM [10]. These documents make ML-KEM integration interoperable at the ASN.1 and object-identifier level.

Migration guidance and tooling address a broader problem. NIST's PQC transition guidance identifies quantum-vulnerable public-key algorithms and emphasizes the need to transition systems and services to quantum-resistant alternatives [19]. NIST NCCoE's migration project focuses on cryptographic discovery, inventory, prioritization, and migration roadmaps across hardware, software, and services [20]. CycloneDX CBOM similarly provides a machine-readable way to represent cryptographic assets such as algorithms, keys, certificates, and their relationships to components [21]. These approaches are essential for inventory and planning, but they do not answer the message-level assurance question studied here: whether a specific CMS `EnvelopedData` object has any remaining classical route to the CEK. Section 3 formalizes this question as a recipient-path validation problem.

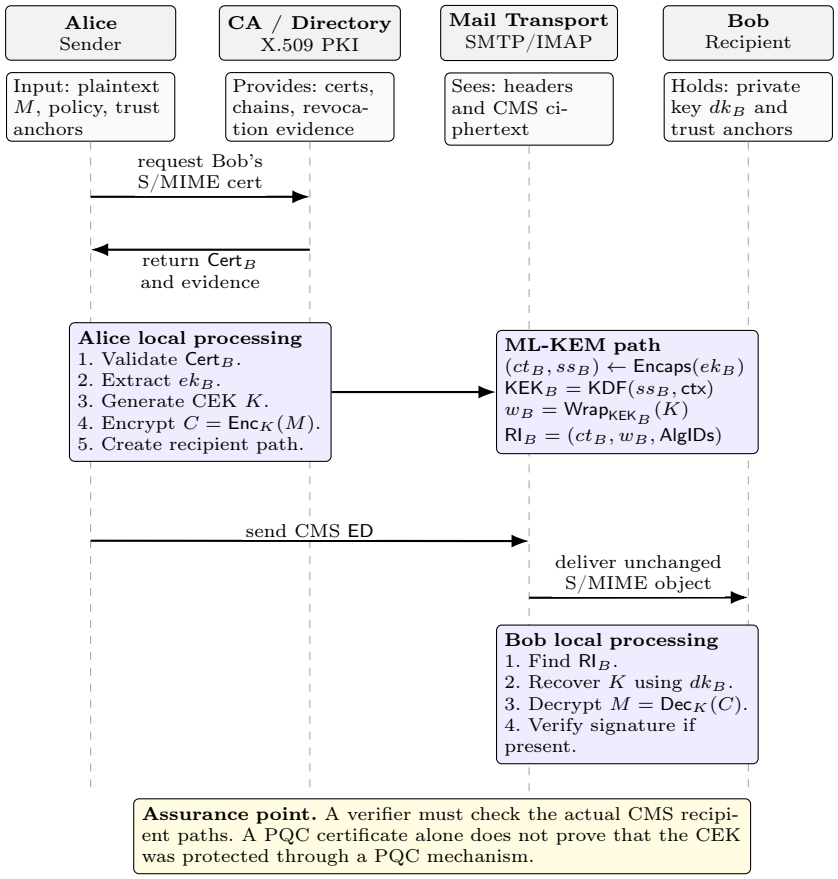

Fig. 2. S/MIME encryption and verification evidence. The certificate establishes an identity-bound key, but the CMS `EnvelopedData` object records which mechanism actually protected the CEK.

3 The Proposed Validation Framework

This section formalizes the validation target and presents the operational stages of the proposed S/MIME PQC validation framework. The object under analysis is a CMS `EnvelopedData` object together with the certificate evidence available to the verifier. The verifier may be an enterprise mail gateway, a compliance scanner, a mailbox migration tool, or an archive auditor. The verifier does not require recipient private keys and does not decrypt email content; instead, it reasons over visible CMS structures, certificate metadata, algorithm identifiers, and policy rules. The framework is therefore designed to solve the assurance problem rather than merely the encoding problem: standardized PQC encodings are treated as necessary evidence, but an S/MIME object is accepted only when all valid CEK-recovery paths satisfy the active policy.

We consider two practical adversarial settings. First, a harvest-now-decrypt-later adversary stores encrypted S/MIME messages and later uses a quantum computer to attack classical RSA or elliptic-curve recipient mechanisms. Second, an operational adversary compromises a recipient's long-term S/MIME decryption or decapsulation key and uses any valid recipient path in an archived message to recover the content-encryption key (CEK). Both settings are relevant because S/MIME ciphertexts are often retained for long periods and because the same encrypted content may be recoverable through multiple recipient-specific paths. The following model instantiates the assurance problem for CMS `EnvelopedData`: the verifier must decide whether every valid recipient path that can recover the CEK satisfies the active migration policy.

Formal Model. Let a CMS `EnvelopedData` object be represented as

$$\mathsf{ED} = (C, \{\mathsf{RI}_i\}_{i \in R}, \mathsf{AlgIDs}, \mathsf{Meta}), \tag{5}$$

where R is the set of recipient entries, M is the plaintext MIME content, K is the CEK, $C = \mathsf{Enc}_K(M)$ is the encrypted content, RI_i is the recipient information for recipient i, AlgIDs is the set of visible algorithm identifiers, and Meta denotes non-secret metadata such as recipient identifiers and content-type information.

For compactness, we denote the certificate-validity decision for recipient i at validation time t by

$$V_i(t) = \mathsf{ValidCert}(\mathsf{Cert}_i, id_i, usage, t, P_t) \in \{0, 1\}, \tag{6}$$

where Cert_i is the certificate associated with recipient i, id_i is the expected mailbox identity, $usage$ is the required key-management or encryption usage, t is the validation time, and P_t is the active policy at time t. This abstraction covers certificate-chain validation, identity binding, key usage, extended key usage, validity period, revocation evidence, and policy constraints.

We separate certificate validity from recipient-path classification: a certificate may be valid while the CMS mechanism using it is classical, unknown, or policy-invalid. Each recipient path is assigned a policy-visible path class. We write this classification as

$$\chi_i(t) = \mathsf{Classify}(\mathsf{RI}_i, \mathsf{Cert}_i, P_t) \in \{\mathsf{PQC}, \mathsf{Hybrid}, \mathsf{Classical}, \mathsf{Unknown}, \mathsf{Invalid}\}. \tag{7}$$

For example, an ML-KEM `KEMRecipientInfo` with an approved OID and a conforming certificate is classified as PQC; RSA key transport and classical ECDH key agreement are classified as $\mathsf{Classical}$; a standardized hybrid KEM path may be classified as Hybrid if the active policy recognizes the construction; and unrecognized or incomplete evidence is classified as $\mathsf{Unknown}$ or $\mathsf{Invalid}$. Let $A_t = \mathsf{Allowed}(P_t)$ denote the set of recipient-path classes permitted by the active policy. The core acceptance rule is universal:

$$\mathsf{Accept}(\mathsf{ED}, P_t) = 1 \iff \forall i \in R,\ \big(V_i(t) = 1 \Rightarrow \chi_i(t) \in A_t\big). \tag{8}$$

This universal condition is necessary because all valid recipient paths may recover the same CEK. Therefore, the existential rule "there exists one PQC recipient" is insufficient: a single classical recipient path remains enough to expose the protected content if that path can recover K.

Proposition 1. Let $\mathsf{ED} = (C, \{\mathsf{RI}_i\}_{i \in R}, \mathsf{AlgIDs}, \mathsf{Meta})$, where $C = \mathsf{Enc}_K(M)$. If there exists a valid recipient path $j \in R$ such that an adversary computes $K \leftarrow \mathsf{RecoverCEK}(\mathsf{RI}_j)$, then the adversary recovers $M = \mathsf{Dec}_K(C)$. Therefore, the confidentiality level of ED is bounded by the weakest valid recipient path protecting K.

Proof. The encrypted content C is protected by a single CEK K. Recipient structures are alternative mechanisms for recovering that same K. Once the adversary obtains K from any valid path, symmetric decryption yields M. Hence, compromising one valid path to K decrypts the whole CMS object. □

Proposition 1 is the basis for the framework. A message containing both an ML-KEM path and an RSA or ECDH path is not fully PQC-protected under a strict confidentiality policy, even though it contains a PQC mechanism.

3.1 Framework Stages

The proposed framework, shown in Figure 3, is a linter and policy engine for S/MIME artifacts. It proceeds in five stages: artifact normalization, evidence extraction, recipient-path classification, policy evaluation, and diagnostic reporting. The framework deliberately separates parsing from policy judgment. This separation is important because an organization may initially audit legacy S/MIME traffic without blocking it, later allow only approved hybrid constructions, and eventually enforce a strict PQC-only policy.

Stage 1: Artifact normalization. The verifier accepts S/MIME artifacts such as `.eml`, `.p7m`, DER-encoded CMS, and PEM-encoded CMS inputs. It decodes the outer MIME or CMS container, identifies supported content types, and rejects malformed objects that cannot be parsed soundly. This stage does not make a PQC judgment. Its purpose is to produce a normalized internal representation of the CMS object and the associated certificate evidence.

Stage 2: Evidence extraction. The verifier enumerates all `RecipientInfo` entries and extracts the visible evidence needed for policy evaluation. At the CMS layer, this includes the recipient type, recipient identifier, KEM or key-transport algorithm OID, KDF, key-wrap algorithm, and content-encryption algorithm. At the certificate layer, it extracts the SubjectPublicKeyInfo algorithm OID, public-key encoding length, key-usage and extended-key-usage extensions, mailbox identity, validity interval, issuer information, and certificate fingerprint. For ML-KEM certificates, the verifier checks whether the parameter set and public-key encoding are consistent with the expected profile.

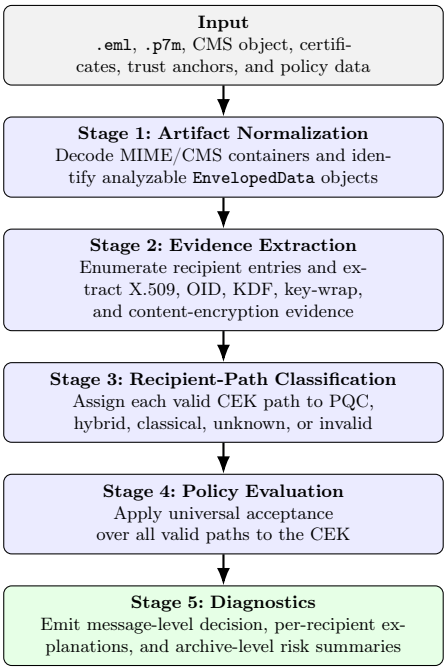

Fig. 3. Architecture of the proposed S/MIME PQC validation framework. The verifier moves from artifact parsing to evidence extraction, recipient-path classification, and policy evaluation over all valid paths to the CEK.

Stage 3: Recipient-path classification. Each recipient path is classified using a versioned OID and policy registry. ML-KEM `KEMRecipientInfo` paths with approved identifiers and conforming certificate evidence are classified as PQC. Recognized hybrid or composite KEM paths are classified as Hybrid only when the policy explicitly permits the construction. RSA `KeyTransRecipientInfo` and classical ECDH `KeyAgreeRecipientInfo` are classified as Classical for key-management purposes. Unsupported OIDs, incomplete evidence, or ambiguous structures are classified as Unknown, while mandatory profile violations are classified as Invalid. Unknown mechanisms fail closed: they are not treated as PQC-safe by default.

Stage 4: Policy evaluation. The policy engine applies Eq. (8) to all valid recipient paths. In *strict-PQC* mode, every valid path to the CEK must be classified as PQC. In *transitional-hybrid* mode, the policy may additionally allow approved hybrid constructions, but it still rejects an independent standalone classical fallback to the same CEK. In *legacy-audit* mode, classical paths are reported for migration planning without being labeled PQC-protected. This distinction allows the same framework to support inventory, transitional deployment, and strict enforcement.

Table 1. Message-level output classes used by the validation framework.

Class	Definition
`pqc-protected`	Every valid recipient path to the CEK is classified as PQC; no valid classical, hybrid, or unknown fallback path exists.
`hybrid-protected`	Every valid path is classified as PQC or as an explicitly policy-approved Hybrid construction; no independent standalone classical fallback exists.
`classical-only`	Every valid path to the CEK uses a classical key-management mechanism such as RSA key transport or classical ECDH.
`unsafe-mixed-mode`	At least one valid PQC or hybrid path coexists with at least one independent valid classical path to the same CEK.
`invalid`	The artifact is malformed or violates a mandatory CMS, certificate, or algorithm-profile requirement.
`unknown`	The verifier cannot soundly classify one or more relevant recipient paths because required evidence is missing or unsupported.

Stage 5: Diagnostic reporting. When multiple conditions appear applicable, the framework applies conservative precedence: malformed artifacts are rejected as `invalid`; unsafe mixed-mode exposure is reported before weaker uncertainty labels; and unknown mechanisms fail closed rather than being treated as PQC-safe. The diagnostic report includes the final class, the active policy identifier, the recipient-path evidence supporting the decision, and an explanation suitable for mail gateways, compliance dashboards, and archive-scanning workflows. Table 1 defines the message-level output classes. These classes are conservative: an object is labeled `pqc-protected` only when every valid CEK-recovery path is positively classified as PQC.

Thus, malformed artifacts are rejected before policy classification, mixed-mode exposure is reported before weaker uncertainty labels, and a message is called `pqc-protected` only when all valid CEK-recovery paths are positively classified as PQC. The diagnostic report includes the final class, the active policy identifier, the recipient-path evidence supporting the decision, and an explanation suitable for mail gateways, compliance dashboards, or archive-scanning workflows.

3.2 Archive-Level Risk Analysis

The same logic extends to stored S/MIME archives. Let an archive be

$$\mathcal{A} = \{\mathsf{ED}_1, \ldots, \mathsf{ED}_n\}. \tag{9}$$

For each ED_j, the verifier records the visible recipient mechanisms, certificate evidence when available, output class, and policy version. This supports aggregate questions such as the fraction of an archive that is `classical-only`, the number of objects that are `unsafe-mixed-mode`, and the set of messages whose status is `unknown` because certificate evidence is missing.

When recipient identifiers are available, the verifier can estimate a key-compromise blast radius:

$$\mathsf{BlastRadius}(id) = \left|\{\mathsf{ED}_j \in \mathcal{A} : \exists \mathsf{RI}_i \in \mathsf{ED}_j \text{ associated with } id\}\right|. \tag{10}$$

This metric does not prove key compromise and does not repair old ciphertexts. It provides prioritization evidence for archive migration, such as re-encrypting selected messages when plaintext access is legally and operationally available, placing legacy objects in a PQC-protected container for future storage, or marking high-value classical-only messages for retention review.

3.3 Prototype Implementation

We implemented the framework as `SMIME-PQCheck`[1], a lightweight command-line verifier for S/MIME and CMS artifacts. The prototype is designed as a validation layer rather than a mail user agent: it does not send mail, decrypt content, or manage recipient private keys. Instead, it parses certificates and CMS objects, extracts policy-relevant evidence, classifies recipient paths, and emits a machine-readable JSON report.

The implementation consists of four compact components. The *certificate analyzer* extracts X.509 metadata, including SPKI algorithm identifiers, key-usage extensions, email identities, validity intervals, and ML-KEM public-key profile evidence. The *CMS analyzer* enumerates `RecipientInfo` structures and records recipient type, algorithm identifiers, KDF, key-wrap, and content-encryption metadata. The *OID registry* separates algorithm recognition from parser logic by mapping OIDs to algorithm families, standards sources, and policy-visible classes. The *policy engine* applies the universal rule in Eq. (8) and generates the final message-level decision.

The prototype accepts individual `.eml`, `.p7m`, DER, and PEM-encoded CMS artifacts, as well as directory inputs for archive scanning.The JSON report includes a top-level decision, the policy identifier, the detected content type, a list of recipient paths, and explanatory evidence. This report format is intended for integration with CI pipelines, mail gateways, compliance dashboards, and archive-scanning workflows. The prototype therefore provides an executable realization of the framework while keeping the core contribution policy-driven and independent of any specific mail client.

4 Conclusion

PQC support for S/MIME is becoming deployable through ML-KEM profiles for X.509 certificates and CMS `KEMRecipientInfo`. These standards address the encoding problem: they define how post-quantum keys, algorithm identifiers, and recipient mechanisms can be represented in existing S/MIME artifacts. This paper argued that encoding support is necessary but not sufficient. The remaining problem is assurance: deciding whether a produced CMS `EnvelopedData` object actually satisfies a migration policy once all usable paths to the content-encryption key are considered.

We addressed this problem by shifting the unit of analysis from certificate-level readiness to message-level protection. We modeled encrypted S/MIME as

[1] `https://github.com/dsayan99/smime-pqcheck`

a multi-recipient CMS object in which the content is encrypted once under a CEK and each recipient structure provides a possible recovery path to that same key. This leads to a universal acceptance condition: an S/MIME object is post-quantum protected only if every valid CEK-recovery path satisfies the active policy. Consequently, the presence of an ML-KEM recipient path is not by itself evidence of post-quantum confidentiality. A single independent RSA or ECDH path can still expose the same encrypted content, yielding an unsafe mixed-mode object.

We then presented `SMIME-PQCheck`, a standards-driven validation framework for deployed S/MIME artifacts. The framework combines X.509 profile checks, CMS recipient-structure analysis, OID-registry based classification, explicit hybrid-policy handling, and archive-level risk aggregation. It classifies artifacts as `pqc-protected`, `hybrid-protected`, `classical-only`, `unsafe-mixed-mode`, `invalid`, or `unknown`, thereby producing evidence that can be consumed by mail gateways, compliance scanners, migration tools, and archive auditors.

The broader lesson is that PQC migration cannot be assessed only by checking whether a system supports post-quantum algorithms. User-facing applications require assurance that the concrete object, session, or workflow produced by the system satisfies a policy under realistic deployment conditions. For S/MIME, this means validating the complete set of CMS recipient paths, not merely the presence of PQC-capable certificates. Message-level validation therefore provides a practical bridge between emerging PQC standards and operational migration decisions such as blocking unsafe outgoing mail, labeling inbound messages, auditing stored archives, and prioritizing remediation of long-lived classical ciphertexts.

Acknowledgment. This research is supported by the National Research Foundation, Singapore, and Cyber Security Agency of Singapore under its National Cybersecurity Research & Development Programme (Development of Secured Components & Systems in Emerging Technologies through Hardware & Software Evaluation < NRF-NCR25-DeSNTU-0001 >). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the view of National Research Foundation, Singapore and Cyber Security Agency of Singapore.

References

1. J. Schaad, B.C. Ramsdell, S. Turner. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification. RFC 8551 (2019). DOI 10.17487/RFC8551. URL `https://www.rfc-editor.org/info/rfc8551`
2. J. Schaad, B.C. Ramsdell, S. Turner. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Certificate Handling. RFC 8550 (2019). DOI 10.17487/RFC8550. URL `https://www.rfc-editor.org/info/rfc8550`
3. S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell, D. Cooper. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (2008). DOI 10.17487/RFC5280. URL `https://www.rfc-editor.org/info/rfc5280`

4. G. Öndarö, J. Kaspereit, S. Umezulike, C. Saatjohann, F. Ising, S. Schinzel, in *34th USENIX Security Symposium (USENIX Security 25)* (USENIX Association, Seattle, WA, 2025), pp. 6737–6756. URL https://www.usenix.org/conference/usenixsecurity25/presentation/oendaroe
5. P.W. Shor, in *Proceedings 35th annual symposium on foundations of computer science* (Ieee, 1994), pp. 124–134
6. NIST. Module-lattice-based key-encapsulation mechanism standard (fips 203). https://csrc.nist.gov/pubs/fips/203/final (2024)
7. NIST. Module-lattice-based digital signature standard (fips 204). https://csrc.nist.gov/pubs/fips/204/final (2024)
8. S. Turner, P. Kampanakis, J. Massimo, B. Westerbaan. Internet X.509 Public Key Infrastructure - Algorithm Identifiers for the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM). RFC 9935 (2026). DOI 10.17487/RFC9935. URL https://www.rfc-editor.org/info/rfc9935
9. P. Julien, M. Ounsworth, D.V. Geest. Use of ML-KEM in the Cryptographic Message Syntax (CMS). RFC 9936 (2026). DOI 10.17487/RFC9936. URL https://www.rfc-editor.org/info/rfc9936
10. R. Housley, J. Gray, T. Okubo. Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS). RFC 9629 (2024). DOI 10.17487/RFC9629. URL https://www.rfc-editor.org/info/rfc9629
11. J. Massimo, P. Kampanakis, S. Turner, B. Westerbaan. Internet X.509 Public Key Infrastructure – Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA). RFC 9881 (2025). DOI 10.17487/RFC9881. URL https://www.rfc-editor.org/info/rfc9881
12. CA/Browser Forum. Ballot SMC013: Enable PQC Algorithms for S/MIME (2025). URL https://cabforum.org/2025/07/02/ballot-smc-013/
13. CA/Browser Forum. Baseline Requirements for the Issuance and Management of Publicly-Trusted S/MIME Certificates (2026). URL https://cabforum.org/working-groups/smime/requirements/
14. D. Adrian, B. Beck, D. Benjamin, D. O'Brien. Advancing Our Amazing Bet on Asymmetric Cryptography. https://blog.google/chromium/advancing-our-amazing-bet-on-asymmetric/ (2024). Accessed 2026-05-25
15. D. Adrian, D. Benjamin, B. Beck, D. O'Brien. A New Path for Kyber on the Web. https://security.googleblog.com/2024/09/a-new-path-for-kyber-on-web.html (2024). Accessed 2026-05-25
16. A. Vaira, H. Brockhaus, A. Railean, J. Gray, M. Ounsworth, Post-quantum cryptography migration use cases. Internet-Draft draft-vaira-pquip-pqc-use-cases-02, Internet Engineering Task Force (2024). URL https://datatracker.ietf.org/doc/draft-vaira-pquip-pqc-use-cases/02/. Work in Progress
17. J. Klaußner. Hybrid PQC E-Mail Communication: Easing Migration Pain. PKI Consortium PQC Conference (2025). URL https://pkic.org/events/2025/pqc-conference-austin-us/WED_BREAKOUT_1430_Jan-Klau%C3%9Fner_Hybrid-PQC-E-Mail-Communication-Easing-Migration-Pain.pdf
18. R. Housley. Cryptographic Message Syntax (CMS). RFC 5652 (2009). DOI 10.17487/RFC5652. URL https://www.rfc-editor.org/info/rfc5652
19. D. Moody, et al., Transition to Post-Quantum Cryptography Standards. Tech. Rep. NIST IR 8547 Initial Public Draft, National Institute of Standards and Technology (2024). URL https://csrc.nist.gov/pubs/ir/8547/ipd
20. National Cybersecurity Center of Excellence. Migration to Post-Quantum Cryptography. https://www.nccoe.nist.gov/applied-cryptography/migration-to-pqc (2026). Accessed: 2026-05-25

21. OWASP CycloneDX. Cryptography Bill of Materials (CBOM). https://cyclonedx.org/capabilities/cbom/ (2026). Accessed: 2026-05-25