Clustrauth: A Post-Quantum Document Authentication Platform with Court-Admissible Evidentiary Output

Ashwin Spencer
Smart Banner Hub LLC
Beaverton, Oregon, USA
ashwin@smartbannerhub.com

March 2026

Abstract

Digital document signing faces two converging threats: the approaching viability of quantum computing against classical cryptographic systems, and the increasing legal scrutiny of digital evidence authenticity in courts worldwide. Existing signing solutions—including industry leaders—produce a single digital signature as proof of authenticity. This model creates a single point of failure: if the signature is compromised, the entire evidentiary chain collapses.

CLUSTRAUTH addresses both threats through a fundamentally different architecture. Rather than producing a single signature, CLUSTRAUTH generates a **four-part forensic evidence package** in which each component is independently authenticated using hybrid classical and post-quantum cryptography. The result is a self-reinforcing evidentiary structure where compromising any single element does not invalidate the others.

This paper describes the threat model, cryptographic design, immutability architecture, and legal admissibility framework behind the CLUSTRAUTH Post-Quantum Authentication Engine and its Encapsulated Multi-Vector Authentication (EMVA) security model.

1. Introduction

1.1 The Quantum Threat Timeline

The National Institute of Standards and Technology (NIST) finalized its first post-quantum cryptographic standards in August 2024, including FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLH-DSA). This standardization was not precautionary—it was a response to concrete advances in quantum computing that place widely deployed cryptographic systems (RSA, ECDSA, Ed25519) at risk within the next decade.

The "harvest now, decrypt later" attack model is particularly relevant to document authentication. An adversary can collect signed documents today and forge or invalidate signatures once quantum computing reaches sufficient capability. Documents with long-term legal significance—contracts, intellectual property filings, regulatory submissions, chain of custody records—require cryptographic protection that extends beyond the classical threat horizon.

1.2 The Single-Signature Problem

Current digital signing platforms produce a single cryptographic signature as proof of document authenticity. This model has three fundamental weaknesses:

1. **Single point of cryptographic failure.** One algorithm, one key, one signature. If the algorithm is broken, or if the key is compromised, the signature provides no residual assurance.
2. **No independent provenance trail.** The signature proves the document was signed, but does not independently prove who submitted it, how it was transmitted, or what happened to it during processing.
3. **No temporal anchoring beyond the signer.** Self-reported timestamps are not independently verifiable. Without a third-party timestamp authority, there is no way to prove *when* a document was signed without trusting the signer.

1.3 Clustrauth's Approach

CLUSTRAUTH replaces the single-signature model with Encapsulated Multi-Vector Authentication (EMVA)—a proprietary security architecture that produces a four-part forensic evidence package where each component carries independent cryptographic authentication. The system is designed so that:

- No single cryptographic failure can invalidate the entire evidence chain
- Provenance and handling metadata are captured and cryptographically locked at signing time
- A third-party timestamp authority provides temporal proof independent of Smart Banner Hub
- Database-level immutability constraints prevent retroactive modification of any evidence record

2. Threat Model

CLUSTRAUTH is designed to withstand the following threat scenarios.

2.1 Quantum Adversary

An adversary with access to a cryptographically relevant quantum computer (CRQC) capable of running Shor's algorithm against classical public-key cryptography. This adversary can forge Ed25519 signatures, recover private keys from classical key exchange, and execute "harvest now, decrypt later" attacks.

Mitigation: Hybrid cryptography combining Ed25519 (classical) with ML-DSA-65 (post-quantum). Both signatures must independently verify. Even if Ed25519 is broken, ML-DSA-65 remains computationally infeasible to forge under known quantum attack models.

2.2 Insider Threat

A privileged administrator or database operator who attempts to modify, delete, or fabricate authentication records after the fact.

Mitigation: Multi-layer immutability architecture including database-level INSERT-only constraints (enforced by PostgreSQL rules that cannot be bypassed without explicitly dropping them), cryptographic hash chains, and third-party timestamp anchoring.

2.3 Infrastructure Compromise

An adversary who gains access to the signing infrastructure, including servers, key material, and databases.

Mitigation: Private keys are managed by Google Cloud KMS at FIPS 140-2 Level 1 protection. The application server never has direct access to private key material—signing operations are performed via authenticated KMS API calls. RFC 3161 timestamps from DigiCert provide proof of document existence that remains valid even if Smart Banner Hub's infrastructure is fully compromised.

2.4 Evidence Tampering

An adversary who modifies a document after authentication and claims the modified version is authentic.

Mitigation: SHA3-256 hash of the complete, unmodified file bytes is computed at signing time and locked into the certificate record, both signatures, the Chain of Custody, and the audit trail hash chain.

2.5 Temporal Dispute

A party who disputes when a document was signed.

Mitigation: RFC 3161 trusted timestamp from DigiCert. The DER-encoded timestamp token can be verified by any party using standard tools (e.g., OpenSSL) without contacting Smart Banner Hub.

3. Cryptographic Model

3.1 Algorithm Selection

Table 1 summarizes the cryptographic primitives and their selection rationale.

Table 1: Cryptographic Algorithm Selection

Layer	Algorithm	Standard
Post-Quantum Sig	ML-DSA-65	FIPS 204
Classical Sig	Ed25519	RFC 8032
Integrity Hash	SHA3-256	FIPS 202
Trusted Timestamp	RFC 3161	IETF RFC 3161
Key Protection	Cloud KMS	FIPS 140-2 L1

ML-DSA-65 (CRYSTALS-Dilithium) is NIST's primary post-quantum digital signature standard, selected after extensive cryptanalysis across multiple competition rounds. It is lattice-based, offering strong security guarantees against both classical and quantum adversaries.

Ed25519 is a widely deployed, efficient, and deterministic elliptic curve signature scheme. It provides immediate classical security while ML-DSA-65 provides quantum resistance.

SHA3-256 (Keccak sponge construction) was selected over SHA-256 for structural diversity—a breakthrough against SHA-2 (Merkle–Damgård) would not affect SHA3. Against Grover's algorithm, SHA3-256 provides 128-bit effective quantum security.

3.2 Hybrid Signature Architecture

CLUSTRAUTH implements a true hybrid signature model: both the classical (Ed25519) and post-quantum (ML-DSA-65) signatures are computed over the same document hash. Verification requires **both signatures to independently pass**. This is not a fallback model—it is a dual-gate model where failure of either signature invalidates the authentication.

This design provides defense in depth:

- **If Ed25519 is broken** (quantum threat): ML-DSA-65 remains secure.
- **If ML-DSA-65 is broken** (unforeseen advance): Ed25519 remains secure.
- **If neither is broken** (expected case): Both signatures provide independent confirmation, strengthening evidentiary weight.

The hybrid model is also forward-compatible. If a future algorithm migration requires replacing either signature scheme, the dual-gate architecture allows one gate to be upgraded while the

other continues to provide continuity—ensuring no disruption to existing verification workflows.

3.3 Key Management

All signing keys are managed by Google Cloud KMS (SOFTWARE protection level, FIPS 140-2 Level 1). This provides:

- Google-managed encryption of key material at rest
- IAM-based access control and audit logging
- Cryptographic key isolation—private keys are never exported or exposed to application code
- Automatic key versioning and rotation support

Signing operations send the document hash to Cloud KMS via authenticated API calls, which returns the signature. At no point does the application server have access to private key material.

All evidence records—certificates, Chain of Custody entries, and audit events—are encrypted at rest using AES-256 via Google Cloud SQL's default encryption layer.

The system enforces a **signing readiness check** before accepting payment. If Cloud KMS is unreachable or unhealthy at checkout time, the payment is blocked with a service unavailability response. This prevents the platform from collecting payment for a signing operation it cannot complete.

3.4 Hash Function Selection

SHA3-256 was selected over SHA-256 for three reasons: (1) *Structural diversity*—Keccak sponge construction is architecturally distinct from Merkle–Damgård; (2) *Quantum resistance*—128-bit effective security against Grover's algorithm; (3) *Standardization*—NIST FIPS 202 ensures broad tooling support and regulatory acceptance.

4. Four-Part Evidence Package

Unlike conventional signing platforms that produce a single certificate, CLUSTRAUTH generates four independent evidence components, each serving a distinct evidentiary purpose. Table 2 provides an overview.

Table 2: Evidence Package Components

Component	Purpose
Certificate of Authenticity	Cryptographic proof of document integrity and authenticity
Chain of Custody Report	Forensic handling record from submission to certification
Audit Trail	Tamper-evident hash-chained event log
RFC 3161 Timestamp	Third-party temporal proof via DigiCert

4.1 Certificate of Authenticity

The primary authentication record containing the unique Certificate ID, SHA3-256 hash of the original document, Ed25519 and ML-DSA-65 digital signatures, signing timestamp, and issuing authority identification. The Certificate of Authenticity cryptographically binds the document content to the signing event.

If the same document (identical SHA3-256 hash) has been previously authenticated by the same user, the system flags the submission as **previously authenticated** and returns the prior Certificate ID and signing timestamp. This provides forensic awareness of duplicate submissions without blocking re-authentication.

4.2 Chain of Custody Report

A forensic record documenting the complete handling of the document from submission to certification:

- **Submitter identity**—Who submitted the document and through which channel
- **Transmission details**—Protocol and encryption information
- **Document characteristics**—Filename, file size, MIME type, cryptographic hash
- **Processing environment**—Infrastructure region, storage backend, encryption at rest
- **Signing details**—Algorithms and key references used
- **Millisecond-precision timeline**—Timestamps for each processing stage

Each Chain of Custody record includes a **SHA3-256 integrity hash** computed at creation time from all record fields in a fixed canonical order. On any subsequent read, the hash is recomputed and compared—any field modification produces a mismatch, providing cryptographic tamper detection independent of the database-level INSERT-only constraints.

Critically, the Chain of Custody Report carries its **own independent Certificate ID** and is authenticated through the same hybrid cryptographic pipeline as the original document. It is not merely descriptive—it is itself a cryptographically authenticated document.

4.3 Audit Trail

A tamper-evident log of every event in the authentication lifecycle, structured as a **cryptographic hash chain**. Each event includes the hash of the previous event, creating a linked chain where modification of any single entry invalidates all subsequent entries. The hash chain uses SHA3-256, ensuring quantum resistance of audit trail integrity.

The Audit Trail, like the Chain of Custody, carries its own

independent Certificate ID and full hybrid cryptographic authentication.

4.4 RFC 3161 Trusted Timestamp

A third-party timestamp token issued by DigiCert's Timestamp Authority, conforming to IETF RFC 3161. This timestamp proves the document hash existed at a specific moment in time, is issued by an independent authority with no affiliation to Smart Banner Hub, can be verified by any party using standard tools, and remains valid even if Smart Banner Hub ceases to operate.

The RFC 3161 timestamp is the only component that does not depend on Smart Banner Hub's continued existence for verification. This is by design—it serves as an external anchor of trust.

To facilitate independent verification, CLUSTRAUTH provides a **downloadable TSA verification kit** for each certificate. The kit contains the raw DER-encoded timestamp response (`.tsr`), the document's SHA3-256 hash, the DigiCert CA certificate chain, and a ready-to-run shell script with OpenSSL commands. Any party can verify the timestamp offline without contacting Smart Banner Hub.

4.5 Self-Reinforcing Evidence Structure

The four components form a self-reinforcing evidence structure:

- The **Certificate** proves the document is authentic
- The **Chain of Custody** proves how it was handled—and is itself authenticated
- The **Audit Trail** proves what happened—and is itself authenticated with a hash chain
- The **RFC 3161 Timestamp** proves when it happened—independently of Smart Banner Hub

An adversary would need to simultaneously forge both classical and post-quantum signatures across three independently authenticated documents, break a SHA3-256 hash chain, bypass database-level immutability constraints, and forge a DigiCert timestamp token.

5. Immutability Architecture

CLUSTRAUTH enforces evidence immutability through multiple independent layers, ensuring that no single point of compromise enables retroactive modification.

5.1 Database-Level Enforcement

Authentication records are stored with INSERT-only constraints enforced at the database level. UPDATE and DELETE operations are intercepted and silently discarded—even when executed by administrators or superusers. These constraints can only be removed by explicitly dropping the database rules, which itself would be an auditable event.

5.2 Cryptographic Hash Chain

Audit trail events are linked in a SHA3-256 hash chain. Each event includes the hash of the previous event in its own hash computation. Modifying any event retroactively changes its hash, invalidating every subsequent event in the chain. The chain can be independently verified by walking from the genesis entry to the most recent event and recomputing each hash. The genesis entry uses a deterministic sentinel value—`SHA3-256("CLUSTRAUTH_CHAIN_GENESIS_V1")`—providing a publicly known, fixed anchor for every certificate's hash chain.

5.3 Third-Party Temporal Anchoring

The RFC 3161 timestamp from DigiCert provides an external, immutable reference point. Even if Smart Banner Hub's entire database were compromised, the timestamp token proves that a specific document hash existed at a specific time, signed by an authority outside Smart Banner Hub's control.

5.4 Dual Cryptographic Binding

Both Ed25519 and ML-DSA-65 signatures bind the document hash to the certificate record. Modifying the document after signing would require forging both a classical and a post-quantum signature—two fundamentally different mathematical problems.

5.5 Record-Level Integrity Hashing

Each Chain of Custody record carries a SHA3-256 integrity hash computed at INSERT time from all fields in a fixed canonical order. On retrieval, the system recomputes this hash and compares it to the stored value. Any modification—even by a database superuser who has dropped the INSERT-only rules—produces a hash mismatch and is flagged as a potential tamper event. This provides defense in depth beyond database-level constraints.

5.6 One-to-One Evidence Records

Each certificate has exactly one Chain of Custody record, enforced by database uniqueness constraints. This prevents fabrication of alternative handling histories for a given authentication.

6. Legal Admissibility Framework

6.1 Evidentiary Standards

CLUSTRAUTH's evidence package is designed to satisfy:

- **FRE Rule 901(b)(9)**—Authentication of system-produced evidence
- **Daubert Standard**—Reliable scientific/technical methodology [7]
- **eIDAS Regulation (EU)**—Electronic signatures and seals framework [8]

- **UETA / E-SIGN Act (US)**—Legal equivalence of electronic signatures [9, 10]

6.2 Why Four Components Matter

Courts increasingly scrutinize not just *whether* a document was signed, but: (1) *Chain of custody*—How was it handled? Was it altered? (2) *Process reliability*—Is the system auditable? (3) *Temporal proof*—When was it authenticated? Can this be independently confirmed? (4) *Tamper evidence*—Is there proof records have not been modified?

A single digital signature addresses only question (2), and only partially. CLUSTRAUTH's four-part evidence package addresses all four with independent, verifiable evidence.

6.3 Third-Party Verifiability

Critical to legal admissibility is the ability for any party—opposing counsel, expert witnesses, courts—to independently verify evidence without relying on Smart Banner Hub. The RFC 3161 timestamp achieves this using publicly available tools and DigiCert's published certificates.

For signature verification, CLUSTRAUTH publishes Ed25519 and ML-DSA-65 public keys via a public API endpoint. Any party can retrieve these keys and verify both signatures independently. However, unlike the RFC 3161 timestamp—where DigiCert serves as a pre-installed trust anchor—public key verification requires trusting that the endpoint serves authentic keys. The RFC 3161 timestamp therefore remains the strongest form of independent verification, while the public key endpoint provides the practical mechanism for classical and post-quantum signature validation.

7. Platform Architecture

7.1 Consumer Access: Quantum Auth Forge

Individual users can authenticate documents through the Quantum Auth Forge (QAF) web interface at $24.99 per document, receiving the full four-part evidence package with the same cryptographic rigor as enterprise deployments.

7.2 API Access

Organizations integrate CLUSTRAUTH via RESTful API across three tiers (Table 3).

Table 3: API Pricing Tiers

Plan	Monthly	Included	Overage
Developer	$249	50 sigs	$5.00/sig
Business	$999	250 sigs	$3.50/sig
Enterprise	Custom	Custom	Custom

7.3 White-Label Studio

The White-Label Studio enables businesses to deploy CLUSTRAUTH as their own branded platform—their logo, colors, domain, and pricing—while Smart Banner Hub operates as invisible infrastructure (Table 4).

Table 4: White-Label Pricing Tiers

Tier	Monthly	Included Sigs
Starter	$2,500	2,000
Growth	$7,500	10,000
Scale	$15,000	50,000
Enterprise	∼$45,000+	100,000+

White-label tenants receive the complete CLUSTRAUTH cryptographic pipeline—the same algorithms, the same evidence package, the same legal admissibility—under their own brand.

7.4 Test and Production Separation

All certificates carry an explicit `is_test` flag distinguishing sandbox from production authentications. Sandbox certificates are watermarked and limited to three free test signatures per API key. This separation ensures that test artifacts cannot be confused with production evidence—a distinction relevant to any forensic or legal proceeding where certificate provenance is scrutinized.

8. Comparison with Conventional Signing

Table 5 compares CLUSTRAUTH against conventional digital signing platforms.

Table 5: Feature Comparison

Capability	DocuSign / Adobe	Clustrauth
Signature	RSA/ECDSA	Ed25519 + ML-DSA-65
PQ Protection	None	ML-DSA-65 (FIPS 204)
Evidence	Single cert	Four-part package
Chain of Custody	Embedded in audit	Separately authenticated
Audit Integrity	App-level logs	SHA3-256 hash chain
DB Immutability	App-level	SQL INSERT-only
Timestamp	Varies	RFC 3161 (DigiCert)
Hash Algorithm	SHA-1	SHA3-256 (FIPS 202)
Key Protection	Varies	Cloud KMS (FIPS 140-2 L1)
White-Label	N/A	Full platform
Indep. Verification	Requires platform	RFC 3161 standalone

9. Conclusion

The transition from classical to post-quantum cryptography is not a future concern—it is a present requirement for any document with long-term legal significance. NIST's finalization of post-quantum standards in 2024 made this explicit.

CLUSTRAUTH does not simply replace one algorithm with another. It replaces the single-signature model with a multi-vector evidence architecture where every component reinforces every other component, and no single point of failure can collapse the evidentiary chain.

The combination of hybrid classical/post-quantum cryptography, independently authenticated evidence components, cryptographic hash chains, database-level immutability, and third-party temporal anchoring creates a document authentication system designed not just for today's threat landscape, but for the post-quantum era.

References

[1] National Institute of Standards and Technology, "FIPS 204: Module-Lattice-Based Digital Signature Standard," August 2024.
[2] National Institute of Standards and Technology, "FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," August 2015.
[3] S. Josefsson and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)," RFC 8032, January 2017.
[4] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)," RFC 3161, August 2001.
[5] National Institute of Standards and Technology, "FIPS 140-2: Security Requirements for Cryptographic Modules," May 2001.
[6] Federal Rules of Evidence, Rule 901(b)(9), "Authentication and Identification."
[7] *Daubert v. Merrell Dow Pharmaceuticals, Inc.*, 509 U.S. 579 (1993).
[8] Regulation (EU) No 910/2014 (eIDAS), "Electronic Identification and Trust Services," July 2014.
[9] Electronic Signatures in Global and National Commerce Act (E-SIGN), 15 U.S.C. ch. 96, 2000.
[10] Uniform Electronic Transactions Act (UETA), Uniform Law Commission, 1999.
