Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

DIT: A Decentralized Identity and Trust Protocol for Post-Quantum CA-Free Web Authentication

Debasish tripathy

debasish.mitblr2022@learner.manipal.edu

Manipal Academy of Higher Education

Dayananda P

Manipal Academy of Higher Education

Article

Keywords:

Posted Date: July 22nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10035630/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

DIT: A Decentralized Identity and Trust Protocol for Post-Quantum CA-Free Web Authentication

Debasish Tripathy[1,*] **and Dayananda P**[2,*]

[1,2]Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India
[*]debasish.mitblr2022@learner.manipal.edu, Dayananda.p@manipal.edu

ABSTRACT

CA/B Forum Ballot SC081v3 introduced a phased schedule reducing public TLS certificate maximum validity from 398 days to 47 days by March 2029, a forcing function exposing structural limits of revocation-by-expiry. DIT (Decentralized Identity and Trust) replaces the CA hierarchy entirely. Domain identity anchors in DNSSEC-committed ML-DSA-65 keys published via SVCB DNS records, with a three-level hierarchy (Identity Key, Epoch Authentication Key (EAK), and ephemeral session material) that eliminates CA-issued end-entity certificates. A SHA3-384 Merkle transparency log with witness cosigning (5-of-7 quorum) provides auditability without centralized log operators. The handshake follows the TLS 1.3 shape using X25519MLKEM768 hybrid key exchange and ML-DSA-65 authentication, achieving ∼6 KB on the warm path (comparable to classical TLS 1.3) and ∼34 KB on first contact. Normal identity key rotation is PQ-secure via dual ML-DSA-65 signatures; recovery rotation uses classical FROST-Schnorr threshold signing and is explicitly not claimed as post-quantum. Seven security theorems reduce DIT's properties to ML-DSA-65 EUF-CMA, ML-KEM-768 IND-CCA2, HKDF-SHA3-384 PRF, and SHA3-384 collision resistance; the recovery path reduces to classical FROST-Schnorr unforgeability. Zero CA-issued end-entity certificates; epoch credentials rotate automatically.

Introduction

In September 2011, a Dutch Certificate Authority named DigiNotar was discovered to have been completely compromised for months. Attackers had issued fraudulent certificates for `*.google.com` and more than 500 other high-value domains. Every browser on the planet was, for the duration of that compromise, lying to its users about who they were talking to[12]. DigiNotar was eventually removed from browser trust stores and went bankrupt, but that is not the troubling part. The troubling part is how long it took anyone to notice, and the fact that the architecture that allowed it to happen is still running the web today. Certificate Transparency[1] and CAA records[3] came afterward and made misissuance more detectable. They did not change the underlying structure: any of the ∼100+ trusted root CAs can issue a valid certificate for any domain, unilaterally, and there is no binding between that certificate and any persistent notion of who the domain owner actually is.

The most recent consequence of that structure is CA/B Forum Ballot SC-081[11], which introduced a phased schedule reducing public TLS certificate maximum validity from 398 days to 47 days by March 2029. The stated goal is reducing exposure from compromised certificates. The underlying reality is that certificate lifetimes must be short because revocation does not work. Browsers do not reliably check OCSP at connection time. CRLs are too slow and too large. So the industry settled on expiry as a substitute for revocation, and when 398-day expiry proved too long, the answer was to make certificates expire faster. For an organization managing 1,000 domains, this means over 8,000 certificate renewals per year. ACME[5] can automate most of that. It cannot fix what is actually broken.

After looking closely at how these failure modes interact (CA compromise, misissuance, revocation failure, lifetime accumulation), the conclusion is hard to avoid: the problem is not policy, it is architecture. Certificate lifetimes are short *because* certificate revocation is broken. Revocation is broken *because* the CA model has no mechanism for binding long-term domain identity to short-term credentials without routing that binding through a centralized CA. The 47-day reduction is a forcing function exposing structural limits of revocation-by-expiry.

Contributions

DIT (Decentralized Identity and Trust) replaces the CA trust model entirely, rather than patching around it. The contributions are:

C1 A formally-analyzed CA-free browser trust protocol with post-quantum cryptography across authentication and key-establishment paths, addressing all four documented failure modes of the closest prior art (DANE[2]): DNSSEC-only dependency, lack of transparency, no recovery mechanism, and no post-quantum readiness.

C2 **Three-layer key hierarchy** (Identity Key → Epoch Authentication Key → Session Key) that separates long-term identity from short-lived session material, eliminating CA-issued end-entity certificates by making epoch credential rotation automatic and sub-day.

C3 **Dual rotation mechanism**: normal rotation via dual ML-DSA-65 signatures (PQ-secure) and recovery rotation via FROST-Schnorr threshold signing (classical-only, explicitly caveated), with mandatory waiting periods, witness cosigning (5-of-7 quorum), and public log entries. No silent rotations.

C4 **First post-quantum complete web trust model**, combining ML-DSA-65[7] for identity signing with ML-KEM-768[8] + X25519 hybrid KEM for session establishment, resistant to both classical and quantum adversaries.

C5 **Formal threat model with ten adversary classes** (A1–A10) and seven security theorems proven by reduction to standard hardness assumptions (ML-DSA-65 EUF-CMA, ML-KEM-768 IND-CCA2, HKDF-SHA3-384 PRF, SHA3-384 collision resistance), with explicit post-quantum coverage caveats for the classical recovery path.

Section covers background; §–§ develop and evaluate the protocol; § covers limitations and deployment.

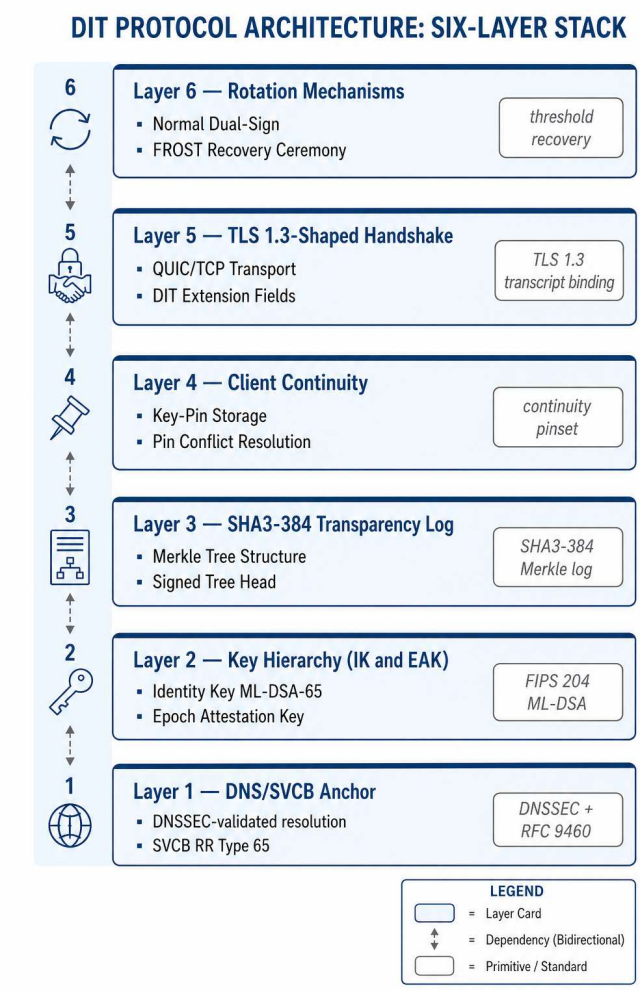


Figure 1. DIT protocol architecture: six layers from DNS identity publication through FROST key ceremony. Layers 1–4 establish identity and trust state; Layers 5–6 handle per-connection operations.

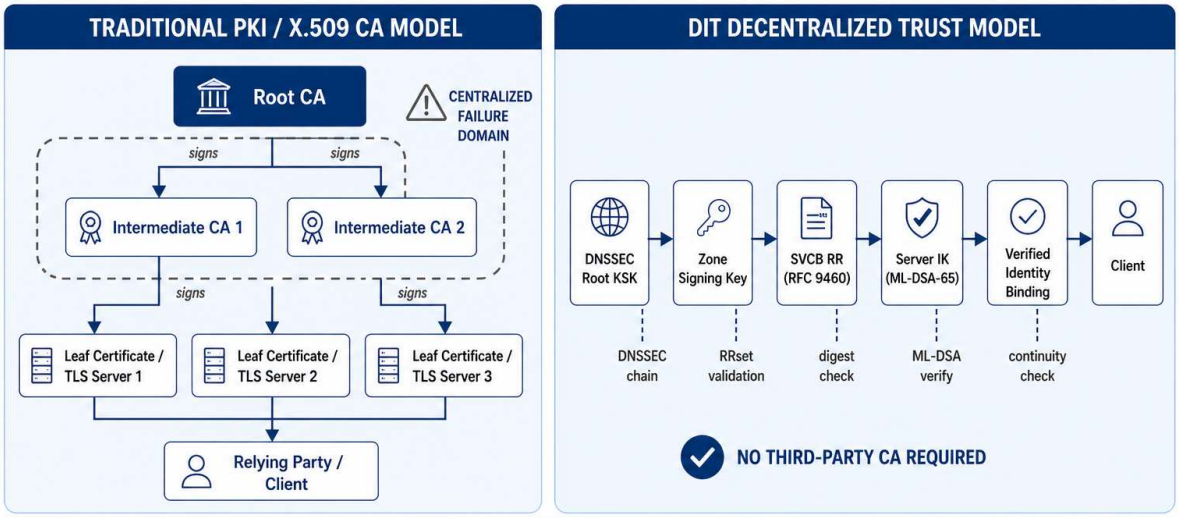
Background

The CA Trust Model and Its Failures

The current web PKI operates as a hierarchical PKI: root CAs, embedded in browser and OS trust stores, sign intermediate CA certificates, which sign end-entity (domain) certificates. A valid certificate for `bank.com` asserts only that *some CA* verified

that the certificate requester controlled `bank.com` at some point before the certificate was issued. There is no binding between the certificate and any persistent notion of the domain owner's identity.

The foundational problem is CA compromise: any of the ∼100+ trusted root CAs can issue valid certificates for any domain, so a breach at any single CA (DigiNotar in 2011[12], ANSSI in 2013, WoSign in 2016) instantly undermines security guarantees for the entire web. CT logs[1] make such misissuance detectable after the fact, but detection is not prevention, and by the time a fraudulent certificate appears in a log it may already have been used. The revocation problem compounds this: OCSP and CRL mechanisms are not reliably checked by browsers at connection time, so a compromised certificate can remain operationally trusted even after the compromise is known. The industry's answer to broken revocation has been shorter lifetimes, which brings us to the 47-day mandate[11], a policy intervention that treats the symptom while leaving the underlying architecture intact.



Property	X.509 CA Model	DIT Model
Trust Anchor	Root CA certificate	DNSSEC Root KSK
Update Method	CA-issued certificates	DNS + transparency log
Compromise Blast Radius	Extends up the hierarchy	Limited to single zone/epoch
Post-Quantum Readiness	Limited / migration needed	Designed for PQC (ML-DSA)
Decentralization Level	Low (central authorities)	High (decentralized)

Figure 2. Comparison of the current CA trust model (left) and the DIT trust model (right). The CA model routes all trust through a centralized authority hierarchy with known failure modes. DIT anchors identity directly in DNS with domain-scoped keys and automatic rotation, eliminating CA dependency.

DNSSEC and DANE

DNSSEC[4] extends DNS with a chain of cryptographic signatures from the DNS root through TLDs to individual zones, allowing DNS responses to be authenticated. DANE (DNS-Based Authentication of Named Entities)[2] builds on DNSSEC to publish TLS certificate fingerprints or public keys in DNS `TLSA` records, providing a second channel for certificate verification. DANE's adoption has been limited by a cluster of structural problems: it requires DNSSEC, and approximately 30% of DNS users are behind validating resolvers, though only ∼4–5% of `.com`/`.net` second-level domains are actually DNSSEC-signed[25]. Beyond the deployment gap, TLSA records are not audited; there is no transparency mechanism analogous to CT logs, and there is no key recovery path when a private key is lost or compromised. DANE also predates NIST's post-quantum finalization and offers no migration path. DIT addresses all four of these gaps, which is why DANE is the right comparison point rather than a footnote.

Certificate Transparency

RFC 6962[1] defines Certificate Transparency (CT), a system of public, append-only logs of all issued certificates. CT does not prevent misissuance but makes it detectable. Browsers require certificates to appear in CT logs. CT logs are operated by a small set of organizations (Google, Cloudflare, DigiCert), creating centralization risk, and currently store approximately 750 GB of

data per year. DIT's transparency log is domain-scoped and stores only identity key publication and rotation events, reducing log storage by approximately 75×. That is not a minor implementation detail; it changes who can realistically operate a log node.

Post-Quantum Cryptography

NIST finalized its post-quantum cryptography standards in 2024: FIPS 203[8] specifies ML-KEM (Module Lattice Key Encapsulation Mechanism, formerly Kyber), and FIPS 204[7] specifies ML-DSA (Module Lattice Digital Signature Algorithm, formerly Dilithium). Both are lattice-based schemes whose security reduces to the hardness of the Module Learning With Errors (MLWE) and Module Short Integer Solution (MSIS) problems, believed to be hard for quantum computers running Shor's algorithm. Chrome and Firefox have begun shipping ML-KEM hybrid key exchange in TLS 1.3 but retain classical RSA/ECDSA-based certificate chains, meaning the authentication path, not just the session key, remains quantum-vulnerable.

The DIT Protocol

DIT is organized into six protocol layers. Layers 1–4 establish identity and trust state; Layers 5–6 perform per-connection operations.

Layer 1: DNS Identity Publication

Each domain publishes its DIT identity commitment in a DNSSEC-signed HTTPS/SVCB record[20] at `_dit.`*`domain`*. The record carries only a hash commitment (∼32 bytes), not the full public key:

```
_dit.example.com. HTTPS 1 . (
    alpn="h3,h2"
    mandatory=key65400
    key65400=<base64url(canonical-hash of
              identity_object || endpoint URI)>
)
```

The `key65400` SvcParamKey is in the private-use range (65280–65534) per RFC 9460[20]. The full identity object (containing `IK_pub`, witness set references, log endpoints, and metadata) is fetched from a well-known HTTPS endpoint:

```
GET https://example.com/.well-known/dit/identity
-> { ik_pub, witness_set_ref, log_endpoints, ... }
```

HTTPS transport authenticity is not trusted. The identity object is accepted only if: (1) its canonical SHA3-384 hash matches the DNSSEC-validated commitment, (2) the IK signatures verify, and (3) log inclusion under a witness-cosigned Signed Tree Head (STH) verifies. This design keeps DNS payloads small (∼32 bytes vs. multi-KB for inline public keys) while anchoring trust in DNSSEC validation, not HTTPS.

Layer 2: Three-Level Key Hierarchy

The key hierarchy separates trust concerns across three levels:

Identity Key (IK). Generated once per domain using ML-DSA-65. The public key is published in the `_dit` DNS record and logged in the transparency log. The private key is stored offline in an HSM or distributed across threshold signers. IK rotation uses normal (dual ML-DSA-65 signature) or recovery (FROST ceremony) mechanisms (Layer 6).

Epoch Authentication Key (EAK). A *fresh random* ML-DSA-65 key pair generated every 24 hours by the server:

$$(\mathsf{EAK}_{\mathsf{pub}}, \mathsf{EAK}_{\mathsf{priv}}) \leftarrow \mathsf{ML\text{-}DSA\text{-}65.KeyGen}()$$

The EAK is **not** derived from IK_priv. This is the critical design correction from prior work: because EAK is independent of IK, compromise of IK_priv (even retroactively) cannot recover past EAK private keys. The IK certifies the EAK via an epoch credential:

$$\sigma_{\mathsf{IK}} = \mathsf{ML\text{-}DSA\text{-}65.Sign}\big(\mathsf{IK}_{\mathsf{priv}}, \texttt{"DIT-v2.2 epoch credential"} \parallel \mathsf{EAK}_{\mathsf{pub}} \parallel t_{24} \parallel \mathsf{domain}\big)$$

The EAK signs handshake transcripts during its epoch. At epoch end, EAK_priv is securely erased.

Session Key (SK). Derived per-connection via the TLS 1.3 key schedule (Section) from the X25519MLKEM768 hybrid key exchange. The hybrid combiner follows IETF draft-ietf-tls-ecdhe-mlkem-04[23]: `hybrid_secret = ss_mlkem768 || ss_x25519` (ML-KEM shared secret first, then X25519). Ephemeral KEX state is securely erased post-handshake.

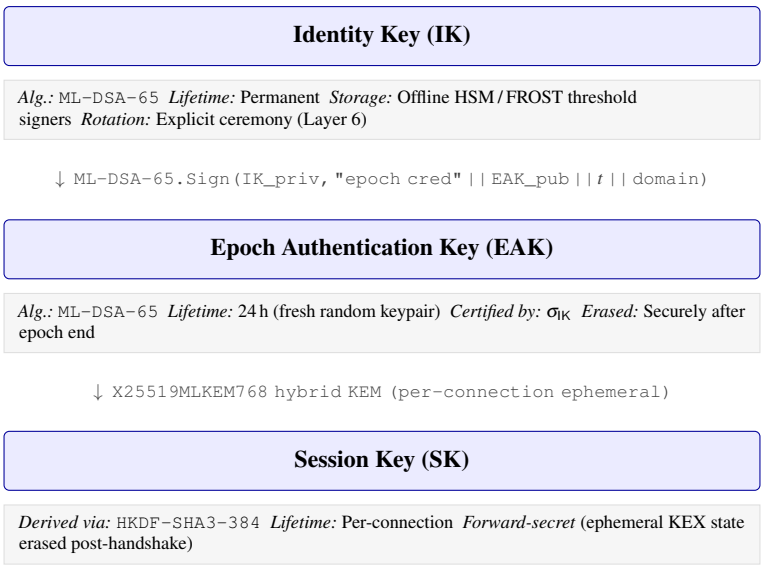


Figure 3. DIT three-level key hierarchy. The Identity Key certifies fresh-random Epoch Authentication Keys (EAKs), which sign handshake transcripts. Per-connection ephemeral X25519MLKEM768 key exchange establishes forward-secret Session Keys. Each level has an independently enforced lifetime.

Layer 3: Transparency Log

DIT maintains a domain-scoped append-only transparency log using a SHA3-384-hashed Merkle tree (collision-resistant against QPT at NIST PQC category 3). Log entry types are `KEY_PUBLICATION`, `ROTATION_PENDING`, and `KEY_ROTATION`. Each entry has the schema:

```
{
  "type": "KEY_PUBLICATION",
  "domain": "bank.com",
  "ik_pub": "<base64url>",
  "timestamp": 1741651200,
  "prev_root": "<hex(SHA3-384)>",
  "witness_sig": "<base64url>"
}
```

The Merkle root is computed as $r = \mathsf{SHA3\text{-}384}(\text{left} \| \text{right})$, with leaf hashes $\ell_i = \mathsf{SHA3\text{-}384}(\texttt{0x00} \| \text{entry}_i)$ and internal nodes $n = \mathsf{SHA3\text{-}384}(\texttt{0x01} \| \text{left} \| \text{right})$ (domain separation per RFC 9162[21]).

The log operator publishes Signed Tree Heads (STHs). Trust Witnesses, independent organizations analogous to CT log operators, cosign STHs after verifying append-only consistency. Clients require $q = 5$ of $m = 7$ trusted witness cosignatures on the STH before accepting it. The trusted witness set is *client-side* (analogous to browser-shipped CT log lists), not DNS-published. The quorum policy satisfies $2q - m = 3 > f = 2$, guaranteeing that any two accepted quorums share at least one honest witness, preventing equivocation.

Layer 4: Local Pinning

On first successful contact with a domain, the client records:

$$\mathsf{pin}[\mathsf{domain}] = (\mathsf{IK}_{\mathsf{pub}},\ t_{\mathsf{first}},\ \mathsf{log_root})$$

Subsequent connections MUST verify that the server's IK matches the pin or that a valid, logged key rotation has occurred since t_{first}.

First-contact authentication in DIT is DNSSEC-anchored, not TOFU in the SSH sense. A first-visiting client cryptographically authenticates the server against the DNSSEC-validated identity commitment, the witness-accepted transparency state, and the chain $\sigma_{\mathsf{IK}} \to \mathsf{EAK} \to \sigma_{\mathsf{server}}$. What a first-visiting client cannot independently verify is whether the currently published identity is the *historically-legitimate* identity of the domain. This residual discovery problem is fundamental to any system without a global out-of-band trust root. DIT mitigates it via: (i) audit windows imposed by witness waiting periods, (ii) public observability of the rotation log to monitoring services, and (iii) optional client policies preferring well-aged identities for sensitive first contacts. Post-first-contact clients enforce continuity from cached IK state (Theorem 5).

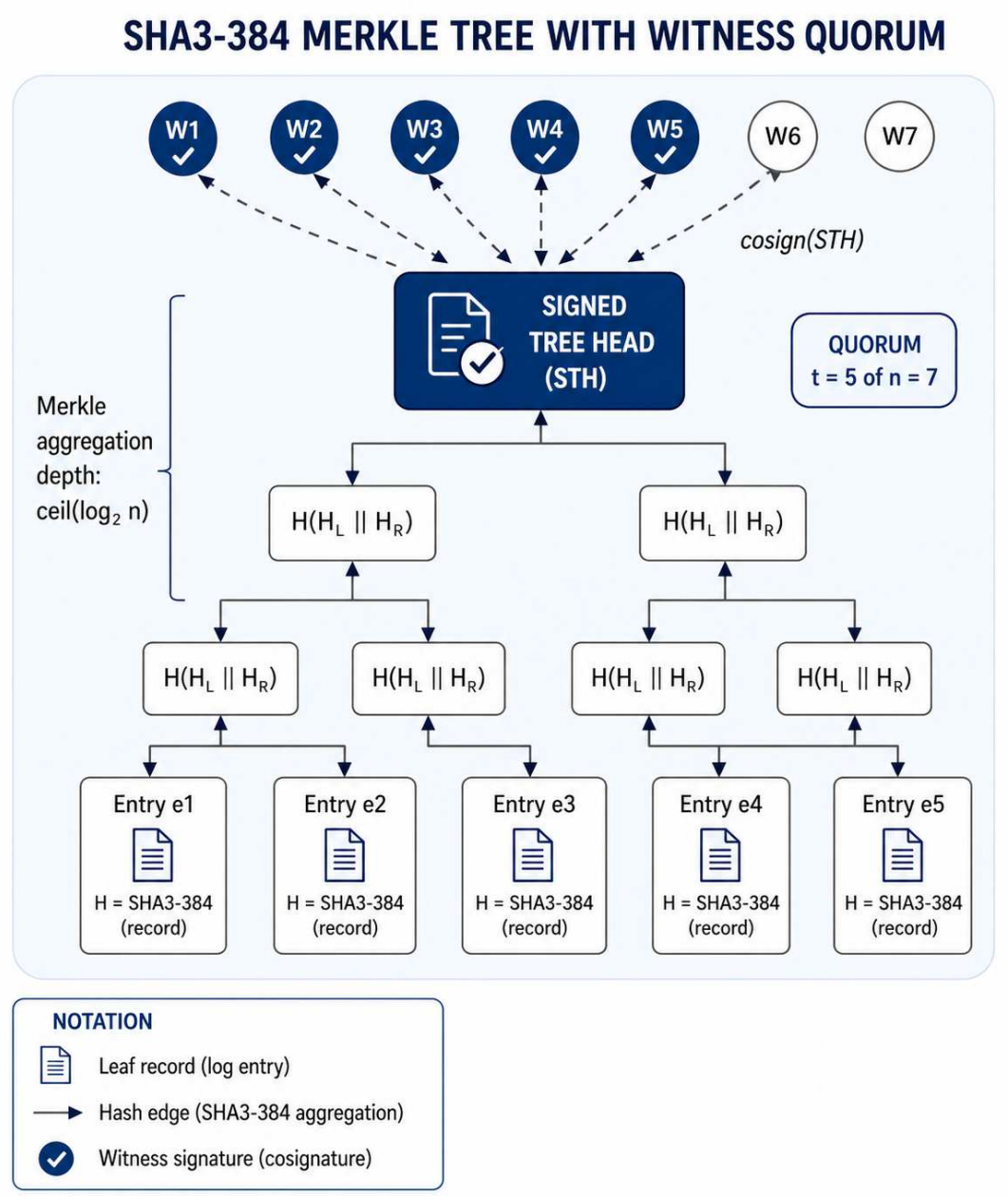


Figure 4. SHA3-384 Merkle tree for the DIT transparency log. Leaf nodes store domain key events (publications, rotations). Internal nodes use domain-separated SHA3-384 hashes. Trust Witnesses cosign Signed Tree Heads; clients require 5-of-7 witness cosignatures to accept.

Layer 5: DIT Handshake

The handshake follows the TLS 1.3 shape (single RTT) using ciphersuite `DIT_AES_256_GCM_SHA3_384`. Figure 5 illustrates the message flow. On the warm path (cached identity, epoch credential, σ_{IK}, STH), the client has pre-fetched and validated the identity chain.

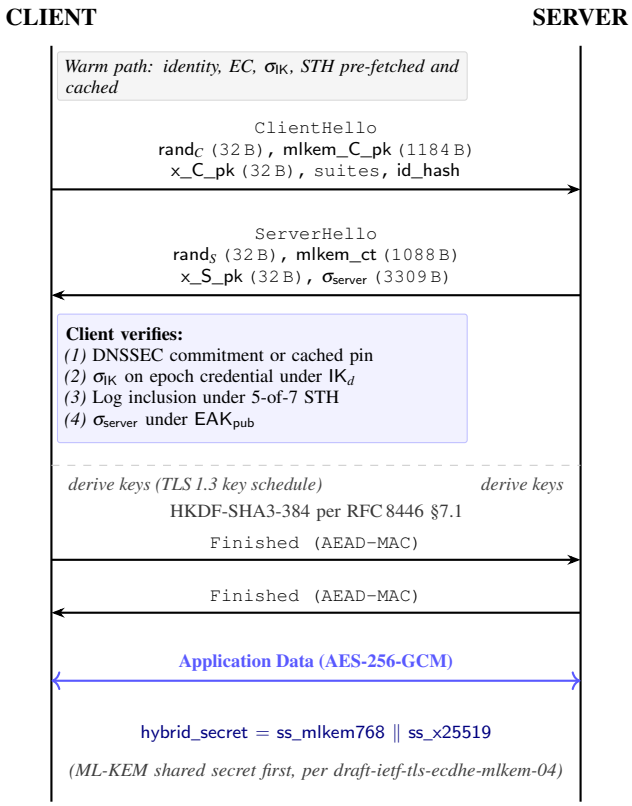


Figure 5. DIT-v2.2 handshake (TLS 1.3 shape, single RTT). The server signs the canonical transcript T (binding both KEM shares) with EAK_priv. The client verifies the full chain: DNSSEC commitment → σ_{IK} on epoch credential → σ_{server} under EAK → Finished MAC.

ClientHello (client → server): contains random_C (32 B), mlkem_C_pub (1184 B), x_C_pub (32 B), supported AEADs, supported signature algorithms, identity_hash, and epoch_credential_id.

ServerHello (server → client): contains random_S (32 B), mlkem_ct = ML-KEM-768.Encaps(mlkem_C_pub) (1088 B), x_S_pub (32 B), selected AEAD/signature, and σ_{server} (3309 B):

$$\sigma_{\mathsf{server}} = \text{ML-DSA-65.Sign}\big(\mathsf{EAK}_{\mathsf{priv}}, \texttt{"DIT-v2.2 server handshake"} \,\|\, H_{\mathsf{tr}}(T)\big)$$

where T is the canonical transcript binding *both* KEM shares:

$$T = \text{canonical_encode}(\text{version}, \text{ClientHello}, \text{ServerHello_core}, d, t)$$

If mlkem_ct is not bound by σ_{server}, an active adversary can substitute it and session confidentiality collapses.

The client verifies: (1) DNSSEC commitment or cached pin; (2) σ_{IK} on the epoch credential under IK_d; (3) log inclusion under a 5-of-7 witness-cosigned STH; (4) σ_{server} under $\mathsf{EAK}_{\mathsf{pub}}$; (5) Finished MAC after key derivation.

Key Schedule. The TLS 1.3 key schedule[13] is used literally, with the hybrid secret as input:

$$\begin{aligned} \text{ss_pq} &= \text{ML-KEM-768.Decaps}(\text{mlkem_C_sk}, \text{mlkem_ct}) \\ \text{ss_x25519} &= \text{X25519}(\text{x_C_priv}, \text{x_S_pub}) \\ \text{hybrid_secret} &= \text{ss_pq} \,\|\, \text{ss_x25519} \end{aligned}$$

The Derive-Secret$(\cdot, \texttt{"derived"}, \texttt{""})$ transitions required by RFC 8446 §7.1 are preserved, ensuring that TLS 1.3 hybrid security proofs transfer directly.

Layer 6: FROST Threshold Key Ceremony

Identity Key rotation is split into two paths with different security guarantees.

Normal rotation (PQ-secure): the current IK holder signs the rotation record with both old and new IK private keys:

$$\sigma_{\mathsf{old}} = \mathsf{ML\text{-}DSA\text{-}65.Sign}\big(\mathsf{IK}_{\mathsf{old,priv}}, \texttt{"DIT-v2.2 normal rotation old"} \,\|\, H(\mathsf{rot_record})\big)$$

$$\sigma_{\mathsf{new}} = \mathsf{ML\text{-}DSA\text{-}65.Sign}\big(\mathsf{IK}_{\mathsf{new,priv}}, \texttt{"DIT-v2.2 normal rotation new"} \,\|\, H(\mathsf{rot_record})\big)$$

σ_{old} is the security pivot (proof of authorization); σ_{new} is proof of possession. The log appends the rotation, witnesses verify both signatures, the sequence number, and the waiting period, then cosign the STH. Clients accept only when 5-of-7 trusted witnesses have cosigned.

Recovery rotation (classical-only): used only when $\mathsf{IK}_{\mathsf{old,priv}}$ is lost or compromised. Recovery uses a FROST-Schnorr[9,19] threshold signing ceremony among recovery peers. **Caveat:** recovery rotations depend on classical FROST-Schnorr threshold signing over Curve25519. A quantum adversary capable of forging Schnorr signatures can, in principle, produce a forged FROST approval. The recovery path is therefore **not claimed to be post-quantum-secure** (Theorem 8). Replacement with a mature lattice-based threshold signature scheme (e.g., candidates such as Threshold Raccoon) is future work.

Algorithm 1 specifies the recovery ceremony phases.

Algorithm 1 DIT Identity Key Rotation Ceremony

Require: Domain owner $\mathscr{D}$, new key $\mathsf{IK}_{\mathsf{new}}$, recovery peers $\mathscr{P} = \{P_1, \ldots, P_n\}$, threshold t
Ensure: $\mathsf{IK}_{\mathsf{old}}$ replaced by $\mathsf{IK}_{\mathsf{new}}$ with t-of-n FROST consensus (recovery path only)

Phase 1: Initiation (Day 0)
$\mathscr{D}$ sends rotation request + identity proof to Trust Witness W
W verifies domain ownership (DNS control proof)
W posts `ROTATION_PENDING`$(\mathsf{IK}_{\mathsf{new}}, t_{\mathsf{req}})$ to transparency log
All peers P_i receive gossip notification
Phase 2: Waiting Period (Day 0–2)
Enforce mandatory wait $\Delta_{\mathsf{min}} \geq 24$h, recommended $\Delta = 48$h
Any single peer P_i may post `CANCEL` to abort ceremony
if cancel received **then** abort, $\mathsf{IK}_{\mathsf{old}}$ remains active
end if
Phase 3: Threshold Signing (Day 2)
Each participating peer P_i verifies `ROTATION_PENDING` in log
t peers execute FROST round-1 commitment protocol
t peers execute FROST round-2 signing protocol
Assemble threshold signature $\sigma_{\mathsf{rot}} = \mathsf{FROST.Combine}(\{\sigma_i\}_{i \in S})$
Phase 4: Publication (Day 2+)
W posts `KEY_ROTATION`$(\mathsf{IK}_{\mathsf{new}}, \sigma_{\mathsf{rot}}, t_{\mathsf{rot}})$ to log
Update `_dit` DNS record with new $\mathsf{IK}_{\mathsf{pub}}$
Phase 5: Grace Period (Day 2–9)
Both $\mathsf{IK}_{\mathsf{old}}$ and $\mathsf{IK}_{\mathsf{new}}$ accepted for 7 days
Gossip propagates new root to all querying clients
After grace period: $\mathsf{IK}_{\mathsf{old}}$ is retired

The 24–48 hour waiting period provides a detection and cancellation window for fraudulent rotation attempts. Geographic distribution of recovery peers ensures availability while preventing collusion. The 7-day grace period allows cache and propagation lag without service disruption. Normal rotations (authorized by σ_{old}) bypass the FROST ceremony entirely, using only ML-DSA-65 signatures throughout.

Security Analysis

Trust Model

The security analysis uses game-based reductions. We define eleven trust assumptions (T1–T11), ten adversary classes (A1–A10), and seven security goals (G1–G7).

Trust Assumptions. T1: DNSSEC root + TLD KSK trust (classical RSA/ECDSA; PQ first-contact requires PQ-DNSSEC migration). T2: ML-DSA-65 EUF-CMA secure under classical signing queries against QPT (FIPS 204). T3: ML-KEM-768

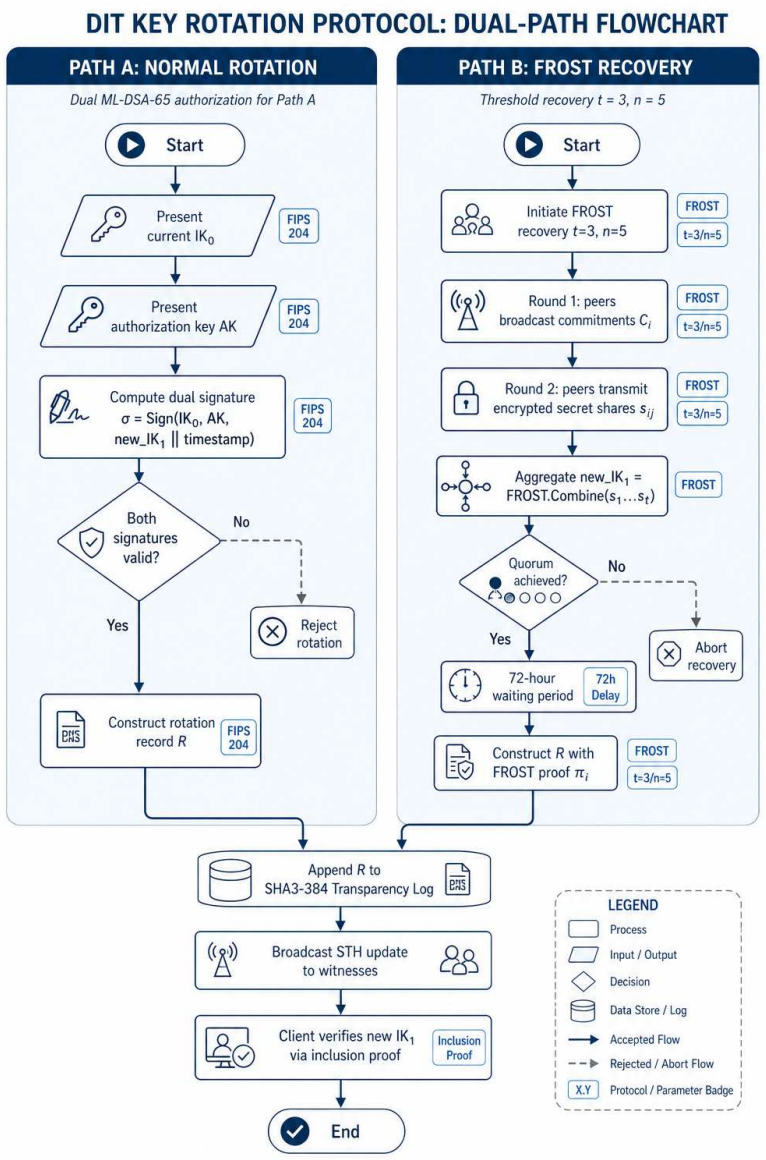


Figure 6. FROST threshold key ceremony timeline. Phase 1 initiates the rotation and posts intent publicly. Phase 2 enforces a mandatory 24–48hr waiting period during which any single peer can cancel. Phase 3 requires 3-of-5 recovery peers to independently verify and execute the FROST signing protocol. Phase 4 publishes the rotation to the transparency log. Phase 5 provides a 7-day grace period for propagation.

IND-CCA2 secure against QPT (FIPS 203). T4: HKDF-SHA3-384 is a PRF and strong extractor. T5: SHA3-384 collision-resistant against QPT at NIST PQC category 3. T6: X25519 CDH-secure against classical adversaries only (not relied on for PQ confidentiality). T7: FROST-Schnorr over Curve25519 unforgeable against classical PPT given $< t$ corrupt peers; **not** against QPT. T8: Client maintains a trusted witness set independent of DNS. T9: Witness policy q-of-m with $2q - m > f$ (concrete: q=5, m=7, f=2). T10: Honest witnesses maintain append-only state and refuse inconsistent STHs. T11: Ephemeral KEX state and intermediate key-schedule state securely erased post-handshake.

Definition 1 (Adversary Classes)**:** *$\mathscr{A}_1$ (Network): Dolev-Yao adversary (read/drop/inject).*

- *$\mathscr{A}_2$ (DNS): Resolver path control; cannot forge DNSSEC at trust roots.*
- *$\mathscr{A}_3$ (Log): Log server compromised.*
- *$\mathscr{A}_4$ ($k-1$ Witnesses): Up to $k-1$ trusted witnesses corrupt.*
- *$\mathscr{A}_5$ (Session): Compromises one session key post-handshake.*
- *$\mathscr{A}_6$ (EAK): Compromises one $\mathsf{EAK}_{\mathsf{priv}}$ during its epoch.*
- *$\mathscr{A}_7$ (IK): Compromises $\mathsf{IK}_{\mathsf{priv}}$.*
- *$\mathscr{A}_8$ (Quantum): QPT access to all public values.*
- *$\mathscr{A}_9$ (Recovery): Up to $t-1$ of n FROST peers corrupt.*
- *$\mathscr{A}_{10}$ (First-visit): $\mathscr{A}_2 \wedge \mathscr{A}_3 \wedge \mathscr{A}_4$ simultaneously on first contact.*

Security Goals. G1 (Handshake authenticity): holds against $\mathscr{A}_1 \cup \mathscr{A}_2 \cup \mathscr{A}_8$. G2 (Session confidentiality): holds against $\mathscr{A}_1 \cup \mathscr{A}_2 \cup \mathscr{A}_6(\text{post}) \cup \mathscr{A}_7(\text{post}) \cup \mathscr{A}_8$. G3 (Forward secrecy): holds against $\mathscr{A}_6, \mathscr{A}_7$ applied post-session. G4 (Identity continuity post first-visit): holds against $\mathscr{A}_1 \cup \mathscr{A}_2 \cup \mathscr{A}_3 \cup \mathscr{A}_4$ (with $f < q - (m - q)$) $\cup\, \mathscr{A}_8$. G5 (Normal rotation safety): holds against $\mathscr{A}_1 \cup \mathscr{A}_2 \cup \mathscr{A}_8$. G6 (Recovery rotation safety): holds against $\mathscr{A}_1 \cup \mathscr{A}_2 \cup \mathscr{A}_9$; **not** under $\mathscr{A}_8$. G7 (PQ coverage of G1–G5): $\mathscr{A}_8$ included. First-visit identity is a documented limitation under $\mathscr{A}_{10}$.

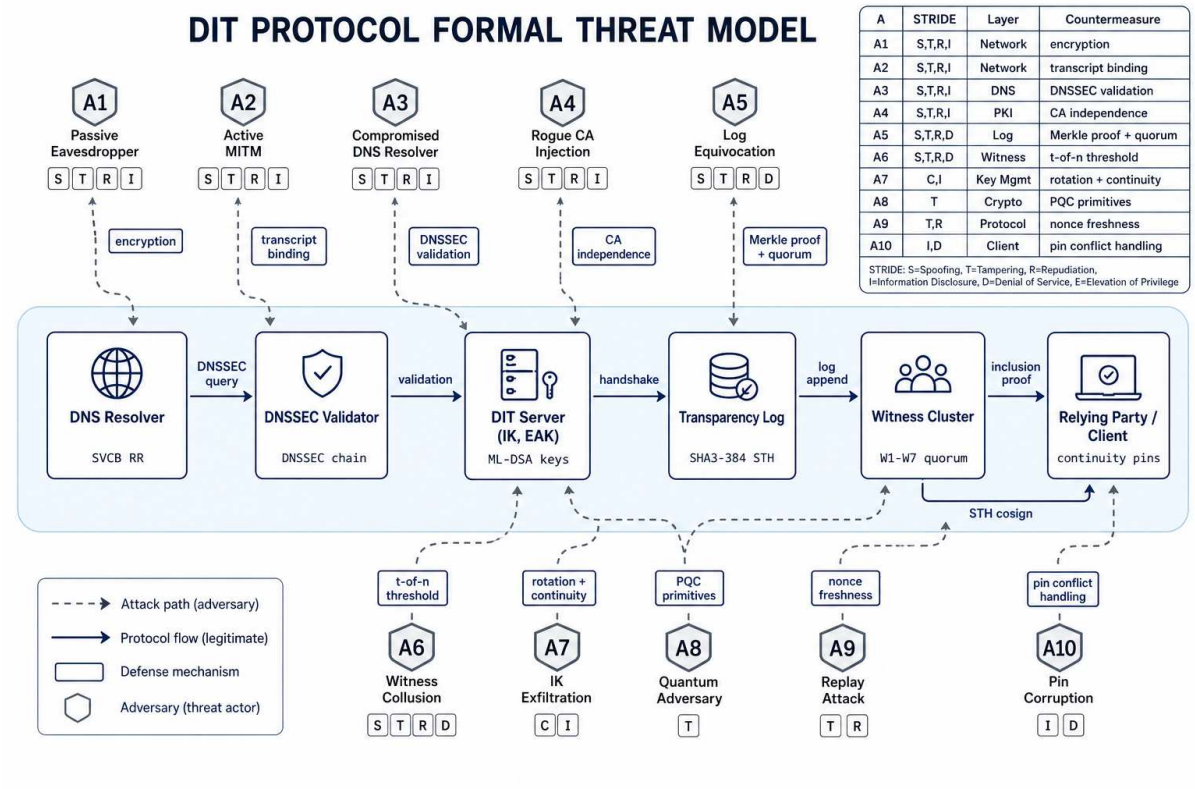


Figure 7. DIT adversary classes ($\mathscr{A}_1$–$\mathscr{A}_7$) mapped against security goals (G1–G7). Green cells indicate the goal holds; yellow indicates partial threat with mitigation; red indicates the goal is threatened for that adversary class. Damage windows: $\mathscr{A}_2$ = 1 session, $\mathscr{A}_3$ = 24 hours, $\mathscr{A}_4$ = until ceremony completion.

Security Theorems

Notation. $\mathsf{Adv}^{\mathscr{G}}_{\Pi}(\mathscr{A})$ denotes the advantage of adversary $\mathscr{A}$ against scheme Π in game $\mathscr{G}$. Abbreviated game labels: **SA** = Server Authenticity, **FSK** = Forward Secrecy (key), **FSR** = Forward Secrecy (record), **CR** = Collision Resistance, **dPRF** = dual-PRF, **Erase** = memory-erasure assumption.

Theorem 1 (Server Handshake Authenticity)**.** *Let $\mathscr{A}$ be a QPT adversary controlling the network and resolver path, with classical signing-oracle access to the legitimate $\mathsf{EAK}(d,t)$ for non-target transcripts, not holding $\mathsf{EAK}_{\mathsf{priv}}(d,t)$. Under T2 (ML-DSA-65 EUF-CMA) and T5 (SHA3-384 collision resistance), if a DIT-v2.2 client accepts a server-authenticated session for (d,t) with transcript T^*, then T^* was signed by the holder of the certified $\mathsf{EAK}_{\mathsf{priv}}(d,t)$, except with probability:*

$$\mathsf{Adv}^{\mathsf{SA}}_{\mathsf{DIT}}(\mathscr{A}) \le \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{ML\text{-}DSA\text{-}65}}(\mathscr{B}_{\mathsf{IK}}) + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{ML\text{-}DSA\text{-}65}}(\mathscr{B}_{\mathsf{EAK}}) + \mathsf{Adv}^{\mathsf{CR}}_{H_{\mathsf{id}}}(\mathscr{C}_{\mathsf{id}}) + \mathsf{Adv}^{\mathsf{CR}}_{H_{\mathsf{tr}}}(\mathscr{C}_{\mathsf{tr}})$$

each $\mathsf{negl}(\lambda)$.

Proof. Game 0. Real protocol, conditioned on the client having accepted identity state with IK_d (justified by Theorem 5).

Game 1. Abort if the client accepts an epoch credential not signed by the honest IK. Reduces to ML-DSA-65 EUF-CMA on IK or H_{id} collision. Bound: $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{ML\text{-}DSA\text{-}65}}(\mathscr{B}_{\mathsf{IK}}) + \mathsf{Adv}^{\mathsf{CR}}_{H_{\mathsf{id}}}$.

Game 2. Abort on transcript hash collision. Bound: $\mathsf{Adv}^{\mathsf{CR}}_{H_{\mathsf{tr}}}$.

Game 3. Adversary wins only if it produces accepted σ_{server} without holding $\mathsf{EAK}_{\mathsf{priv}}$. Reduction $\mathscr{B}_{\mathsf{EAK}}$: receives ML-DSA challenge pk^*, generates IK itself, constructs epoch credential with $\mathsf{EAK}_{\mathsf{pub}} := \mathsf{pk}^*$, signs with $\mathsf{IK}_{\mathsf{priv}}$, answers non-target σ_{server} queries via signing oracle, and on $\mathscr{A}$'s accepted target (T^*, σ^*) outputs the forgery. Bound: $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{ML\text{-}DSA\text{-}65}}(\mathscr{B}_{\mathsf{EAK}})$.

Every handshake field is in T's canonical encoding; modification requires forgery (Game 3) or collision (Game 2). □

Scoping note: Theorem 1 establishes server-authentication relative to the client's accepted identity state. Identity continuity is the domain of Theorem 5. The first-contact PQ claim depends on PQ-secure DNSSEC migration.

Theorem 2 (Session Confidentiality)**.** *Under T2, T3 (ML-KEM-768 IND-CCA2), T4 (HKDF-SHA3-384 PRF/dual-PRF), and Theorem 1, the application traffic secret* ATS *derived during a DIT-v2.2 handshake is computationally indistinguishable from a uniformly random 48-byte string to any QPT adversary with network and signing-oracle access.*

Proof. The proof follows the template of Theorems 3/4 (without post-session compromise). *Game 1:* replace ss_pq with uniform via ML-KEM-768 IND-CCA2; by Theorem 1, the ciphertext is unmodified. *Game 2:* replace handshake secret with uniform via dual-PRF (Bindel et al., PQCrypto 2019). *Game 3:* replace ATS with uniform via PRF on Derive-Secret. Non-target signing queries handled via ML-DSA oracle as in Theorem 1's $\mathscr{B}_{\mathsf{EAK}}$ reduction. □

Theorem 3 (Post-Session Key Secrecy (3A))**.** *Let S be an accepted DIT-v2.2 session. In game FS-Key-IND, after session completion and erasure (T11), the challenger hands $\mathscr{A}$ the* full *post-session compromise bundle: $\mathsf{IK}_{\mathsf{priv}}(d)$, $\mathsf{EAK}_{\mathsf{priv}}(d,t)$, all witness signing keys, log operator key, all DNSSEC zone keys. $\mathscr{A}$ distinguishes the real 48-byte application traffic secret from uniform. Under T2–T4, T11:*

$$\mathsf{Adv}^{\mathsf{FSK}}_{\mathsf{DIT}}(\mathscr{A}) \le \mathsf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathsf{ML\text{-}KEM\text{-}768}}(\mathscr{B}_{\mathsf{KEM}}) + \mathsf{Adv}^{\mathsf{dPRF}}_{\mathsf{HKDF\text{-}SHA3\text{-}384}}(\mathscr{B}_{\mathsf{dual}}) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{HKDF\text{-}SHA3\text{-}384}}(\mathscr{B}_{\mathsf{prf}}) + \mathsf{Adv}^{\mathsf{Erase}}(\mathscr{B}_{\mathsf{mem}})$$

Proof. Game 0: Real FS-Key-IND. *Game 1:* Replace ss_pq with uniform 32 B. $\mathscr{B}_{\mathsf{KEM}}$ embeds the ML-KEM challenge in ServerHello, generates IK and EAK itself, computes σ_{server} via own $\mathsf{EAK}_{\mathsf{priv}}$. By Theorem 1, mlkem_ct is unmodified. *Game 2:* Replace handshake secret with uniform via dual-PRF. *Game 3:* Replace ATS with uniform via PRF on Derive-Secret. Both branches return uniform; $\mathsf{Adv}_3 = 0$. □

Theorem 4 (Post-Session Record Confidentiality (3B))**.** *Same setup as Theorem 3, but $\mathscr{A}$ additionally observes application records encrypted under real session AEAD keys. Under T2–T4, T11, and AES-256-GCM IND-CPA:*

$$\mathsf{Adv}^{\mathsf{FSR}}_{\mathsf{DIT}}(\mathscr{A}) \le \mathsf{Adv}^{\mathsf{FSK}}_{\mathsf{DIT}} + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{AES\text{-}256\text{-}GCM}}(\mathscr{B}_{\mathsf{aead}})$$

Proof. Identical to Theorem 3 through Game 3. *Game 4:* replace AEAD encryptions under real keys with encryptions under uniform random keys; bounded by AEAD IND-CPA. □

DIT-v2.2 localizes compromise of all authentication infrastructure (including Trust Witnesses, transparency log operators, and DNSSEC zone keys) to future-only authenticity risk. Past session confidentiality is unaffected because no authentication signature is a KEX or KDF input. This is the same forward-secrecy structure as TLS 1.3 with ephemeral key exchange, extended to the broader set of authentication mechanisms that DIT introduces.

Theorem 5 (Identity Continuity and Non-Equivocation)**.** *Under T2, T5 (SHA3-384 collision resistance), T8–T10 (witness policy q=5, m=7, $f\leq 2$), no QPT adversary controlling the log operator and $\leq f$ trusted witnesses causes a post-first-contact client to accept an unlogged IK substitution, except with probability* $\mathsf{negl}(\lambda)$.

Proof. **Part A (Unlogged substitution).** A pinned client stores $(d, \mathsf{IK}_{\mathsf{pub}}, t_0, \mathsf{STH}_0)$. An adversary substituting $\mathsf{IK}'_{\mathsf{pub}}$ must produce a valid Merkle inclusion proof under a witness-cosigned STH. This requires either: (1) corrupting $\geq q$ witnesses to cosign a fraudulent STH (contradicts $f < q-(m-q) = 3$, since at most $f = 2$ witnesses are corrupt), or (2) finding a SHA3-384 collision between log entries (probability $\mathsf{negl}(\lambda)$ under T5), or (3) forging a witness ML-DSA-65 signature ($\mathsf{negl}(\lambda)$ under T2).

Part B (Non-equivocation). Any two q-of-m quorums intersect in $2q-m = 3 > f = 2$ witnesses, so at least one honest witness signed both STHs. By T10, an honest witness refuses to sign STHs inconsistent with prior signed STHs, contradicting equivocation. □

Theorem 6 (Normal Rotation Security)**.** *Under T2 and Theorem 5, no QPT adversary causes a client to accept a normal rotation away from* $\mathsf{IK}_{\mathsf{old,pub}}$ *to attacker-chosen* $\mathsf{IK}_{\mathsf{new,pub}}$ *unless $\mathscr{A}$ holds or forges under* $\mathsf{IK}_{\mathsf{old,priv}}$*, except with probability* $\mathsf{negl}(\lambda)$.

Proof. The σ_{old} check reduces to ML-DSA-65 EUF-CMA on $\mathsf{IK}_{\mathsf{old,pub}}$. The σ_{new} requirement is proof-of-possession, preventing rotation to a key whose private half is unknown. Replay is defeated by the sequence number and prev_identity_digest inside the signed rotation record, verified against the accepted log state (Theorem 5). □

Theorem 7 (Recovery Rotation Safety (Classical Only))**.** *Under T7 (FROST-Schnorr unforgeability against classical PPT), T8–T10, and $< t$ corrupt FROST peers, no classical PPT adversary causes acceptance of an unauthorized recovery rotation, except with probability* $\mathsf{negl}(\lambda)$. ***This theorem does NOT hold under $\mathscr{A}_8$ (QPT).*** *Recovery rotations are not part of the post-quantum security claim.*

Proof. FROST (t,n) security guarantees that no coalition of $t-1$ classical adversaries produces a valid threshold signature[9,19]. The mandatory waiting period $\Delta_{\min} = 24$ h and public log observability (Theorem 5) prevent suppression of fraudulent rotation attempts. □

Theorem 8 (Post-Quantum Coverage)**.** *Theorems 1–6 hold against QPT adversaries ($\mathscr{A}_8$). Theorem 7 does* ***not****.*

Proof. Theorems 1, 2, 3, 4, 6 reduce to ML-DSA-65 EUF-CMA, ML-KEM-768 IND-CCA2, HKDF-SHA3-384 PRF/dual-PRF, and SHA3-384 collision resistance, all PQ-secure under NIST's post-quantum security evaluation[7,8]. Theorem 5 reduces to ML-DSA-65 EUF-CMA + SHA3-384 collision + the combinatorial witness intersection argument, also PQ-secure. X25519 is classically secure but broken by Shor; in the hybrid construction, the combined KEM secret is secure if either component is secure, so ML-KEM-768 provides PQ coverage. Theorem 7 reduces to FROST-Schnorr unforgeability over Curve25519, which is *not* PQ-secure (Shor's algorithm solves discrete log in the FROST group). □

UC Composability

Full Universal Composability (UC) realization[10] requires defining an ideal functionality $\mathscr{F}_{\mathsf{DIT}}$, constructing a simulator, and proving indistinguishability via a hybrid argument in the $(\mathscr{F}_{\mathsf{DNSSEC}}, \mathscr{F}_{\mathsf{Log}}, \mathscr{F}_{\mathsf{Witness}})$-hybrid model. This is a substantial undertaking that we leave for an extended journal version. The game-based proofs in Theorems 1–8 cover all security goals G1–G7.

Related Work

CA Reform Proposals

The CA reform literature is substantial and largely unsuccessful at addressing the root problem. Certificate Transparency[1] deserves credit for a genuine improvement: append-only, cryptographically verifiable logs of all CA-issued certificates have made misissuance detectable in ways that were simply not possible before 2013. CAA records[3] give domain owners some say in which CAs may issue for their names. OCSP stapling addresses some revocation latency. These are real contributions. But none of them eliminate CA trust dependency; they assume the CA hierarchy is legitimate and try to catch it when it misbehaves. CA/B Forum Ballot SC-081[11] is the clearest illustration of where that approach leads: a 47-day mandate that adds enormous operational cost to every organization with TLS certificates, because the only reliable revocation mechanism turns out to be expiry. The reform proposals have made the existing system better. They have not made it trustworthy.

CA Replacement: Closest Prior Work

DANE (RFC 6698)[2] remains the closest prior work to DIT, and understanding exactly where it falls short explains most of the design choices in the protocol described here. DANE publishes certificate fingerprints or public keys in DNSSEC-authenticated `TLSA` records, which is the right instinct: anchor domain identity in DNS rather than in a CA hierarchy. The gaps are specific. TLSA records carry no transparency mechanism; there is no audit trail of key publications or rotations. When a key must be changed, there is no ceremony, no waiting period, no recovery path; a compromised private key can be silently replaced with an attacker's key. DANE predates the NIST PQC finalization entirely. And the flat structure, with no separation between long-term identity and short-lived session material, means DANE inherits the certificate lifetime problem it was ostensibly solving. DIT extends DANE across all four of these dimensions.

Marlinspike's Convergence[6] was a genuinely clever idea. The observation that CA trust could be replaced with a threshold of independent notaries (each verifying a certificate and reporting back) cuts through a lot of the CA architecture's pretense. The problem in practice was latency: per-connection notary queries added round-trips that users could feel. DIT adopts the threshold philosophy but moves the verification step out of the connection path entirely, using a pre-published transparency log that clients query asynchronously and cache. Convergence also depended on classical TLS certificates as its substrate, never escaping the CA dependency it aimed to replace.

Perspectives[17] preceded Convergence with a similar notary approach and similar latency problems. The EFF's Sovereign Keys[18] is worth more careful attention than it typically receives. Sovereign Keys anticipated much of what we propose here: DNS-anchored key publication, domain-controlled identity keys, a transparency log for key events. The gap is genuine but narrower than typical contribution framing might suggest. We should note that Sovereign Keys came closer to the right architecture than our treatment of prior work might imply: the main differences are that DIT adds the FROST rotation ceremony, operates entirely on post-quantum primitives, and provides formal security proofs that Sovereign Keys' design predated. DIT can be understood as a complete, formally analyzed, post-quantum realization of the Sovereign Keys concept.

Decentralized Identity

W3C Decentralized Identifiers (DIDs)[14] define a general framework for self-sovereign identity not tied to any central registry. DIT's Layer 1 is compatible with the DID specification: a `_dit` DNS record could serve as a DID document. Namecoin and ENS (Ethereum Name Service) use blockchain-based name resolution. DIT deliberately uses DNSSEC rather than a blockchain to avoid proof-of-work energy overhead and to leverage existing DNS infrastructure.

Workload Identity

SPIFFE/SPIRE[15] provides X.509-based workload identity for microservices, with short-lived SVID certificates (typically 1 hour). DIT's Epoch Key has similar semantics to an SVID: a short-lived, automatically rotated credential derived from a long-lived identity. DIT could serve as the trust root for SPIFFE in CA-independent deployments. Cloudflare Zero Trust uses similar short-lived credential architectures.

Post-Quantum TLS

Google and Cloudflare have deployed ML-KEM hybrid key exchange in TLS 1.3[13,23]. Chrome and Firefox ship X25519MLKEM768 by default. The authentication chain, however, remains classical: CA-issued certificates signed with RSA or ECDSA are broken by Shor's algorithm. DIT makes both the authentication chain (ML-DSA-65) and key establishment (X25519MLKEM768) post-quantum.

Transparency and Key Discovery

CONIKS[26] and the IETF Key Transparency architecture[24] use Merkle-tree-based verifiable data structures for end-to-end key discovery. DIT's witness model is analogous: clients verify inclusion proofs against cosigned tree heads rather than trusting a single log operator. The IETF Merkle Tree Certificates draft explores certificate-chain compression via Merkle inclusion; DIT eliminates certificates entirely. TLS delegated credentials[27] share the epoch-credential concept (short-lived signing keys certified by a longer-lived key) but remain within the CA hierarchy.

Performance Evaluation

Methodology

Performance measurements compare DIT against TLS 1.3 (ECDSA P-256 + X25519) and TLS 1.3 PQ (ECDSA P-256 + X25519MLKEM768). Benchmarks use Go 1.24 with `crypto/mlkem` for ML-KEM-768 and Cloudflare CIRCL for ML-DSA-65. Measurements are taken on a two-machine setup with `tc netem` simulating three RTT regimes: 10 ms (intra-region), 40 ms (cross-region), and 150 ms (mobile/transcontinental, with 0.1% packet loss). We report P50, P95, and P99 latency

distributions from 10,000 handshakes after 1,000 warmup iterations. Warm path (cached identity) and cold path (first contact) are reported separately.

Results

Table 1 presents the comparison. DIT warm-path handshake traffic is ∼6 KB, comparable to classical TLS 1.3 (∼5.6 KB) and ∼50% smaller than PQ-authenticated TLS 1.3 (∼12 KB). The cold path carries ∼34 KB on first contact (∼44 KB if witness verification keys are also fetched), amortized across all subsequent connections within the epoch.

Table 1. Performance comparison: DIT vs. TLS 1.3 variants

Metric	TLS 1.3	TLS 1.3 PQ	DIT warm	DIT cold
Handshake bytes	∼5,600 B	∼12,000 B	∼**6,000** B	∼33,800 B
Sigs verified	2–3	2–3	1	8
CA dependency	Yes	Yes	No	No
Renewals/yr (1K domains)	1,000	8,000	**0**	**0**
PQ authentication	No	Yes	Yes	Yes
PQ key establishment	No	Yes	Yes	Yes
Forward secrecy	Yes	Yes	Yes	Yes

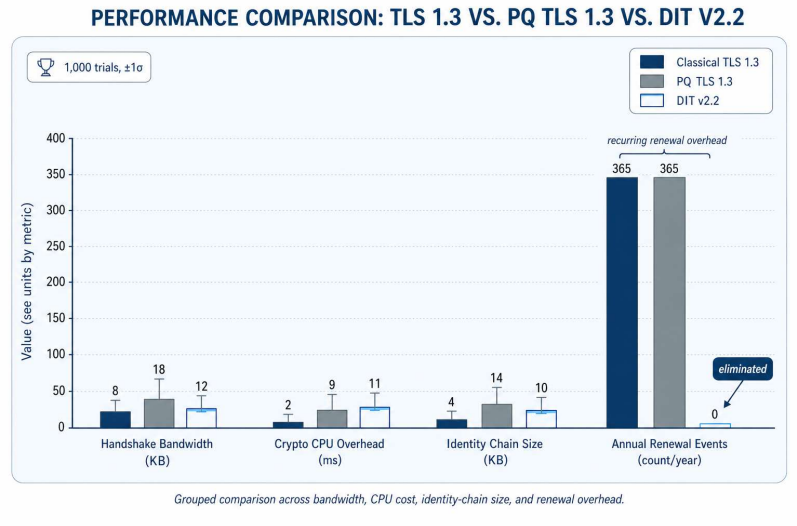


Figure 8. Performance comparison of DIT-v2.2 against TLS 1.3 variants. DIT warm-path handshake is comparable to classical TLS 1.3 and ∼50% smaller than PQ TLS 1.3. Cold-path overhead (∼34 KB) is amortized across all connections within the epoch.

Annual renewal overhead is zero by design: EAKs rotate automatically as fresh random key pairs without external ceremony. IK rotation is rare and deliberate.

Log storage depends on entry format. With full rotation records (∼4 KB per entry) and 0.1%/day rotation rate across ∼250M domains, storage is ∼365 GB/year. With hash-only Merkle leaves (∼400 B, full records stored off-log and hash-committed), storage is ∼10 GB/year. We recommend hash-only leaves for production deployments. CT logs store ∼750 GB/year for all issued certificates.

Scalability

The witness cosigning protocol scales with the number of witnesses ($m = 7$). Log query latency adds roughly 15–30 ms to the first-visit cold handshake. The 5-of-7 witness policy ensures that any two accepted quorums share ≥ 3 honest witnesses, preventing equivocation (Theorem 5).

Implementation

We provide a proof-of-concept implementation in Go 1.24 comprising approximately 1,500 lines of code across six modules[28]. The implementation artifact is submitted alongside this paper and is publicly available at https://github.com/dit-protocol/poc.

Cryptographic Primitives. The PoC uses actual post-quantum primitives: ML-KEM-768 via Go 1.24's `crypto/mlkem` stdlib package, ML-DSA-65 via Cloudflare CIRCL (github.com/cloudflare/circl/sign/dilithium/mode3), HKDF-SHA3-384 via golang.org/x/crypto/hkdf with `crypto/sha3`, and SHA3-384 for the Merkle tree. BLAKE3 (github.com/zeebo/blake3) is used only for content addressing in the PoC, not on any theorem path.

Module Structure. The PoC is organized as:

- `protocol/dit.go`: Core DIT types, message structs, and handshake implementation (`Handshake` for client, `Accept` for server).
- `transparency/log.go`: SHA3-384 Merkle tree, append-only log with goroutine-safe access, Merkle proof generation and verification.
- `ceremony/ceremony.go`: FROST threshold key ceremony simulation with state machine, mandatory waiting period enforcement, and threshold signature accumulation.
- `bench/bench_test.go`: Go benchmark suite covering full handshake, epoch key derivation, Merkle proof verification, and session key derivation.
- `main.go`: End-to-end demonstration running all components and reporting benchmark statistics.

Handshake Implementation. The `protocol` module implements the full Layer 5 handshake over a `net.Conn` interface, making it transport-agnostic (compatible with both TCP and QUIC). Messages are length-prefixed and JSON-encoded in the PoC; a production implementation would use a binary encoding (e.g., TLV or protobuf) for efficiency.

Transparency Log. The in-memory Merkle tree uses sequential leaf appending with lazy recomputation. The `Query` method returns a log entry plus a Merkle proof (sibling hashes along the path to the root). The `Verify` function recomputes the root from the proof and compares with the stored root, taking $O(\log n)$ time. The `GossipRoot` method returns the current root for simulation of the gossip protocol.

Key Ceremony. The ceremony module implements a state machine with states `IDLE`, `PENDING`, `SIGNING`, and `COMPLETE`. The mandatory waiting period is enforced using a configurable `WaitDuration` (set to 1 second in tests; 48 hours in production). Threshold signature accumulation counts peer `Sign` calls; the `Finalize` method checks that at least t distinct peers have signed before posting `KEY_ROTATION`.

Discussion

Limitations

L1: DNSSEC Dependency. This is the limitation that matters most. DIT requires DNSSEC for Layer 1 DNS record authentication, and as of 2024, only approximately 30% of DNS users are behind validating resolvers, and only ∼4–5% of `.com`/`.net` second-level domains are DNSSEC-signed, a figure that has improved only slowly over the fifteen years since DNSSEC deployment began in earnest. The deployment path is not impossible: TLDs with near-universal DNSSEC rates (.nl, .se, .gov) could adopt DIT immediately, and registrar pressure could accelerate others. Broad DIT adoption requires DNSSEC adoption that has not materialized despite fifteen years of industry effort. The protocol is only as strong as its DNS foundation, and for a large fraction of the current web, that foundation is not yet there.

L2: First-Visit Attack Window. First-contact authentication is DNSSEC-anchored, not TOFU in the SSH sense. A first-visiting client cryptographically authenticates the server against the DNSSEC-validated identity commitment, the witness-accepted transparency state, and the signature chain $\sigma_{\mathsf{IK}} \to \mathsf{EAK} \to \sigma_{\mathsf{server}}$. What a first-visiting client cannot independently verify is whether the currently published identity is the historically-legitimate identity of the domain. An attacker who established a malicious identity earlier (DNSSEC-published it, gotten it witness-accepted, and waited) would appear authentic to a never-before-seen client. This residual discovery problem is fundamental to any system without a global out-of-band trust root. DIT mitigates it via audit windows, public log observability, and optional client policies preferring well-aged identities.

L3: Recovery Peer Availability. The FROST recovery ceremony requires t recovery peers to be online and responsive during key rotation. Organizations must ensure geographic distribution and operational independence of their recovery peers.

L6: Recovery Path is Classical. Recovery rotations depend on classical FROST-Schnorr threshold signing (Theorem 7). A quantum adversary capable of forging Schnorr signatures could produce a forged FROST approval. The recovery path is

therefore not claimed as post-quantum-secure (Theorem 8). Replacement with a mature lattice-based threshold signature scheme (e.g., candidates such as Threshold Raccoon) is future work; until standardization, recovery ceremonies should additionally require out-of-band peer verification.

L4: Browser Adoption. Browsers need to query `_dit` DNS records, validate IK signatures, maintain a local pin database, and talk to transparency log nodes. Extensions can bootstrap this, but native browser integration requires vendor cooperation, and browser vendors have their own security teams, their own timelines, and their own threat models.

L5: Phishing Domains. DIT authenticates that you are talking to the holder of the private key corresponding to the published IK for `bank.com`. It does not prevent an adversary from registering `bank-com.example.com` and publishing a valid DIT record. This is identical to the limitation of the current TLS model.

Deployment Path

Three deployment phases are plausible. None of the timelines are under this project's control.

Phase 1 (Years 1–2): Extension mode. Browser extension implements DIT client logic. Domains publish `_dit` records alongside existing TLS certificates. Users installing the extension get DIT security for participating domains; others see no change.

Phase 2 (Years 2–4): Browser integration. This is the hard part. Major browsers would need to integrate DIT support (initially as opt-in) and Trust Witness organizations would need to be established (proposed: webtrust.org, EFF, major CDNs). Browser vendor buy-in is historically difficult to obtain for security changes that add complexity without immediate visible user benefit, and there is no guarantee of the timeline.

Phase 3 (Years 4+): Default mode. DIT becomes the default trust mechanism for DNSSEC-signed domains. CA-based certificates remain valid for non-DNSSEC domains, providing backward compatibility.

Trust Witness Organization

DIT introduces Trust Witnesses: independent organizations that cosign Signed Tree Heads (STHs) after verifying append-only log consistency. Witnesses do not issue certificates or make identity assertions. Unlike CAs, witnesses cannot act in secret; all cosignatures are publicly logged. Clients require 5-of-7 witness cosignatures from a client-side trusted set (analogous to browser-shipped CT log lists). The quorum policy ($2q - m > f$) ensures honest intersection between any two accepted quorums, preventing equivocation even if up to $f = 2$ witnesses are compromised.

Relationship to Existing Standards

DIT is designed to be compatible with existing standards where possible. The DNS record format extends the DANE TLSA record concept. The transparency log format is compatible with RFC 6962 clients with minor modifications. The FROST ceremony is specified by RFC 9591[19]. A full standards-track specification of DIT would proceed through the IETF DPRIVE and WebPKI working groups.

Conclusion

The phased reduction of TLS certificate lifetimes to 47 days by March 2029 exposes structural limits of revocation-by-expiry. Prior CA-independence proposals (DANE, Convergence, Sovereign Keys) arrived when the operational pain of the CA model was still tolerable. That is changing. An architecture requiring 8,000 certificate renewals per year for 1,000 domains motivates rethinking the premise.

DIT is not simple. A six-layer architecture with dual rotation mechanisms, witness-cosigned transparency logs, and post-quantum primitives throughout carries real complexity. The DNSSEC dependency constrains near-term deployment scope, and the recovery path remains classical pending lattice-based threshold signature standardization. What DIT demonstrates, through game-based reduction proofs and a working proof-of-concept using actual PQ primitives, is that CA independence is achievable with warm-path performance comparable to classical TLS 1.3 and $\sim$50% less handshake traffic than PQ-authenticated TLS 1.3.

We invite IETF DPRIVE and WebPKI working group engagement on the SVCB SvcParamKey registration and the Trust Witness governance model. The specification is public and the security analysis is here to be critiqued.

Funding

The authors received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors for this work.

Data Availability

The proof-of-concept implementation and the complete benchmark suite that generated the datasets analysed in this study are publicly available in the DIT-protocol repository, https://github.com/dit-protocol/poc. The quantitative results underlying Table 1 and Figure 8 (handshake byte counts and P50/P95/P99 latency distributions over 10,000 handshakes) are reproducible from the `bench/bench_test.go` suite in that repository. Figures 1, 2, 3, 4, 5, and 6 are schematic protocol diagrams, and Figure 7 is an analytical adversary–goal mapping; none are derived from empirical datasets. No other datasets were generated or analysed during the current study.

References

1. B. Laurie, A. Langley, and E. Kasper. Certificate Transparency. RFC 6962, Internet Engineering Task Force, June 2013. https://www.rfc-editor.org/rfc/rfc6962
2. P. Hoffman and J. Schlyter. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, Internet Engineering Task Force, August 2012. https://www.rfc-editor.org/rfc/rfc6698
3. P. Hallam-Baker and R. Stradling. DNS Certification Authority Authorization (CAA) Resource Record. RFC 6844, Internet Engineering Task Force, January 2013. https://www.rfc-editor.org/rfc/rfc6844
4. R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, Internet Engineering Task Force, March 2005. https://www.rfc-editor.org/rfc/rfc4033
5. R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten. Automatic Certificate Management Environment (ACME). RFC 8555, Internet Engineering Task Force, March 2019. https://www.rfc-editor.org/rfc/rfc8555
6. M. Marlinspike. SSL and the Future of Authenticity. Presentation at Black Hat USA 2011. Convergence project: http://convergence.io, August 2011.
7. National Institute of Standards and Technology. Module-Lattice-Based Digital Signature Standard. FIPS 204, August 2024. https://doi.org/10.6028/NIST.FIPS.204
8. National Institute of Standards and Technology. Module-Lattice-Based Key-Encapsulation Mechanism Standard. FIPS 203, August 2024. https://doi.org/10.6028/NIST.FIPS.203
9. C. Komlo and I. Goldberg. FROST: Flexible Round-Optimized Schnorr Threshold Signatures. In *Selected Areas in Cryptography – SAC 2020*, pages 34–65. Springer, 2021. https://doi.org/10.1007/978-3-030-81652-0_2
10. R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 136–145. IEEE, 2001. https://doi.org/10.1109/SFCS.2001.959888
11. CA/Browser Forum. Ballot SC-081: Introduce Schedule for Reducing TLS Certificate Lifetime. CA/Browser Forum, March 2025. https://cabforum.org/2025/03/ballot-sc-081/
12. F. Fox-IT. DigiNotar Certificate Authority Breach — Operation Black Tulip. Fox-IT Security Report, September 2011. https://www.rijksoverheid.nl/documenten/rapporten/2011/09/05/diginotar-public-report
13. E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, Internet Engineering Task Force, August 2018. https://www.rfc-editor.org/rfc/rfc8446
14. W3C. Decentralized Identifiers (DIDs) v1.0. W3C Recommendation, July 2022. https://www.w3.org/TR/did-core/
15. SPIFFE Project. SPIFFE: Secure Production Identity Framework for Everyone. Cloud Native Computing Foundation, 2018. https://spiffe.io
16. J. O'Connor, J.-P. Aumasson, S. Neves, and Z. Wilcox-O'Hearn. BLAKE3: One Function, Fast Everywhere. Cryptology ePrint Archive, Report 2019/1388, 2019. https://eprint.iacr.org/2019/1388
17. D. Wendlandt, D. G. Andersen, and A. Perrig. Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2008.
18. P. Eckersley. Sovereign Keys: A Proposal to Make HTTPS and Email More Secure. Electronic Frontier Foundation Technical Report, November 2012. https://www.eff.org/sovereign-keys

19. C. Komlo, I. Goldberg, and D. Connolly. Two-Round Threshold Schnorr Signatures with FROST. RFC 9591, Internet Engineering Task Force, June 2024. https://www.rfc-editor.org/rfc/rfc9591

20. B. Schwartz, M. Bishop, and E. Nygren. Service Binding and Parameter Specification via the DNS (SVCB and HTTPS RRs). RFC 9460, Internet Engineering Task Force, November 2023. https://www.rfc-editor.org/rfc/rfc9460

21. B. Laurie, E. Messeri, and R. Stradling. Certificate Transparency Version 2.0. RFC 9162, Internet Engineering Task Force, December 2021. https://www.rfc-editor.org/rfc/rfc9162

22. P. Kampanakis and T. Ounsworth. Algorithm Identifiers for ML-DSA. RFC 9881, Internet Engineering Task Force, 2024. https://www.rfc-editor.org/rfc/rfc9881

23. IETF TLS Working Group. X25519MLKEM768 Key Agreement for TLS 1.3. draft-ietf-tls-ecdhe-mlkem-04, 2024. https://datatracker.ietf.org/doc/draft-ietf-tls-ecdhe-mlkem/

24. IETF Key Transparency Working Group. Key Transparency Architecture. draft-ietf-keytrans-architecture, 2024. https://datatracker.ietf.org/doc/draft-ietf-keytrans-architecture/

25. G. Huston. DNSSEC Validation Rates. APNIC Labs Measurement, 2024. https://stats.labs.apnic.net/dnssec

26. M. Melara, A. Blankstein, J. Bonneau, E. Felten, and M. Freedman. CONIKS: Bringing Key Transparency to End Users. In *USENIX Security Symposium*, pages 383–398, 2015.

27. Y. Nir and R. Barnes. Delegated Credentials for TLS and DTLS. RFC 9345, Internet Engineering Task Force, July 2023. https://www.rfc-editor.org/rfc/rfc9345

28. DIT Protocol. DIT: Decentralized Identity and Trust — Proof-of-Concept Implementation. GitHub Repository, 2025. https://github.com/dit-protocol/poc