HYBRID TRANSPORT / MUTUAL TRUST

# Post-quantum transport, certificate exchange, and mTLS

A technical white paper connecting hybrid TLS, certificate signatures, bidirectional identity validation, and enterprise PKI migration.

6 September 2026 / Technical edition

## Inside this paper

# 01 / Two cryptographic jobs, one connection

Key establishment creates shared secrets. Authentication establishes who is at the other end. ML-KEM is a key encapsulation mechanism; ML-DSA is a signature algorithm. A hybrid session can still use classical certificate authentication. A post-quantum certificate alone does not establish a post-quantum session secret. Track both dimensions in your acceptance criteria.

References: NIST FIPS 203 / NIST FIPS 204

| Layer | Question to answer | Evidence to retain |
| --- | --- | --- |
| Key establishment | Which group protected this session? | Negotiated group and protocol version. |
| Server identity | Did the intended service prove its identity? | Trust path, hostname validation, and peer signature scheme. |
| Client identity | Did the enrolled device prove key possession? | Client trust path and device-to-identity binding. |
| Authorization | May this identity reach this resource now? | Policy version, posture state, decision, and session scope. |

Antara's integration design treats these controls as independent requirements. Enrollment, transport negotiation, certificate validation, and resource authorization have separate failure paths. That separation makes migration observable and prevents a green connection indicator from hiding a weak identity or an unintended fallback.

# 02 / Hybrid key establishment

RFC 10024 standardizes three TLS 1.3 groups combining ECDHE and ML-KEM. The outputs feed the TLS key schedule. The hybrid construction aims to preserve confidentiality if one component remains secure; it does not compensate for compromised endpoints or incorrect implementations.

References: Hybrid TLS: RFC 10024

From ephemeral contributions to protected traffic

| | | |
|---|---|---|
| 01 | Client contribution | Offers a supported hybrid group and its key share. |
| 02 | Gateway contribution | Selects the group and returns its key share. |
| 03 | Key schedule | Both peers derive handshake and application secrets. |
| 04 | Protected session | Record protection uses the negotiated symmetric cipher. |

The enterprise policy should define whether hybrid negotiation is required, which exceptions are permitted, and how long exceptions remain valid. Record the selected group rather than inferring it from a gateway configuration setting. An unsupported peer should produce a diagnosable compatibility result; silently falling back defeats an enforced migration objective.

## 03 / Client and server certificate exchange

In a full TLS 1.3 mutual-authentication handshake, the server requests a client certificate and presents its own certificate chain. CertificateVerify proves possession of the corresponding private key over the handshake transcript. Finished messages bind the exchange to the derived secrets. Certificates are not the session encryption keys.

References: TLS 1.3: RFC 8446

Full handshake with client authentication


Before releasing application traffic, the client must validate the gateway's chain and intended service name. The gateway must validate the device's chain and bind that credential to the correct tenant and enrolled identity. Each side checks validity and policy for key usage, certificate purpose, algorithm support, and revocation. A trusted certificate from another tenant is not sufficient authorization.

- Keep server and client trust bundles independently governed. A public web trust store should not automatically enroll a device.
- Validate service identity using the intended DNS name and subject alternative names; do not substitute a successful TCP connection for identity verification.
- Require a device credential to map to an active enrollment record and an authorized session. Certificate possession and user authentication solve different problems.
- Define revocation behavior explicitly for unavailable responders, cached status, disconnected devices, and urgent credential compromise.

# 04 / Post-quantum certificate trust

RFC 9881 specifies identifiers for ML-DSA in X.509. End-to-end certificate migration also depends on the issuing CA, peer signature negotiation, trust stores, hardware key support, enrollment tooling, and every intermediate certificate in the path. A leaf with a new algorithm does not automatically modernize its entire chain.

References: ML-DSA in X.509: RFC 9881

| Migration track | Start here | Promotion gate |
| --- | --- | --- |
| Hybrid transport | Inventory TLS termination points and peer groups. | Required groups negotiated through production paths. |
| Server authentication | Test service certificates, names, and issuing chains. | Clients validate the full chain and peer proof. |
| Client authentication | Test enrollment, key custody, and device renewal. | Gateways validate and map clients to active identities. |
| Trust rotation | Distribute new roots before issuing new leaf certificates. | Overlap and retirement tested without bypassing validation. |

Use a compatibility matrix by client release, operating system, gateway release, CA, and key storage provider. Gate rollout on a supported combination. An algorithm appearing in a library does not establish hardware support or a FIPS-validated module boundary. Procurement evidence should identify the exact module, version, operating environment, and applicable validation.

# 05 / Resumption, rotation, and failure

Session resumption changes which messages appear on the wire. Reconcile resumed-session policy with the current enrollment, credential status, and authorization context. Set ticket lifetime and reauthentication policy to match the resource sensitivity. Disable early data for actions that must not be replayed; 0-RTT is not equivalent to a fresh, fully authenticated application request.

Rotate keys and trust in stages: distribute trust, canary issuance, validate enrollment and renewal, expand issuance, then retire old trust when dependencies are gone. An emergency compromise may require immediate session termination; a routine certificate renewal can follow a different policy. Publish both procedures before the first fleet rollout.

- Unsupported group: block when hybrid is required and surface a compatibility reason.
- Expired or revoked credential: deny new access and apply the documented active-session policy.

- Unknown issuing root: repair trust distribution through management channels, never by disabling peer verification.
- Control-plane outage: apply a time-bounded cached-policy rule chosen for the resource, with visible expiry.
- Large certificate chains: test fragmentation, middleboxes, loss, and constrained links instead of assuming LAN behavior.

## 06 / What a successful evaluation produces

The outcome is an evidence package: negotiated-group coverage, authenticated client and server identities, unsupported-peer inventory, renewal results, failure traces, and an approved exception register. Measure full and resumed handshakes separately. Include CPU, bytes transferred, connection success, and tail latency under realistic packet loss.

Cryptographic acceptance record

```
connection_evidence:
  protocol: TLSv1.3
  required_group: X25519MLKEM768
  verify_server_name: gateway.example
  require_client_certificate: true
  check_client_enrollment: true
  record_peer_signature_scheme: true
  record_policy_version: true
  exception_owner: security-architecture
  release_gate: peer-and-pki-compatibility
```

This record is a review worksheet for your deployment team. Map it to the configuration interfaces available in your selected client, gateway, and PKI releases. Keep the generated negotiation and validation evidence with the change record.

References and continued reading

NIST FIPS 203

NIST FIPS 204

Hybrid TLS: RFC 10024

TLS 1.3: RFC 8446

ML-DSA in X.509: RFC 9881

Product overview

Integration guide

Operations and evaluation