

# Post-Quantum Enhancements to EAP-TLS and EAP-TTLS

<https://datatracker.ietf.org/doc/draft-reddy-emu-pqc-eap-tls/>

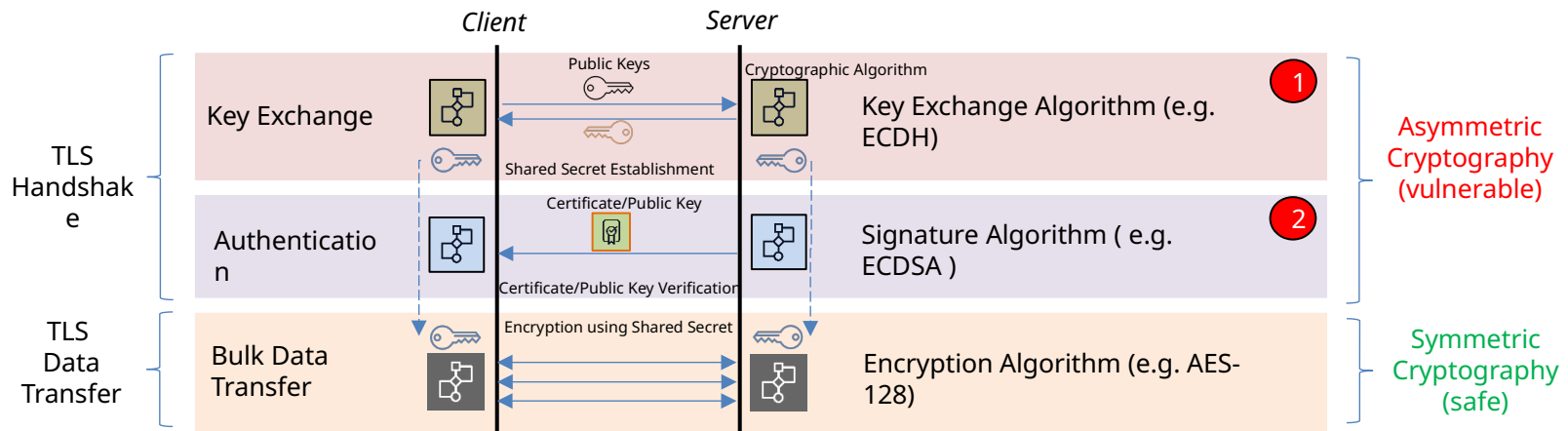
IETF 125 Shenzhen

**K Tirumaleswar Reddy (Nokia)**

# Problem Statement

- Post-quantum threat: CRQCs will break traditional asymmetric key algorithms
- EAP methods (e.g., EAP-TLS, EAP-TTLS) are vulnerable if not quantum-resistant

# Problem Statement



## 1 Key Exchange Algorithms

- **Purpose:** Establish shared secret between two parties
- **Threat:** Harvest Now Decrypt Later i.e. Data Confidentiality
- **Vulnerability Timeframe:** Now!

## 2 Signature Algorithms

- **Purpose:** Authenticate peers
- **Threat:** Server/Client Impersonation (Signature Forgery)
- **Vulnerability Timeframe:** When CRQC available

- Harvest-Now Decrypt-Later attacks threatens long-term confidentiality.
- When CRQC is available, it can spoof the identity of EAP-client and EAP-Server.

# Harvest-Now Decrypt-Later Attack

- TLS 1.3 encrypts most of the handshake using keys derived from the (EC)DHE shared secret via the TLS key schedule
  - HNDL leaks the client identity information used in certificate-based authentication (e.g., usernames, device or organization identifiers).
- In EAP-TTLS, HNDL attacks with CRQCs could expose inner authentication credentials (e.g., MS-CHAPv2 challenge–response), enabling offline password attacks.

# Problem Statement

- PQC Signature Public Key and Signature sizes are much larger than traditional algorithms.
  - Large cert chains can cause fragmentation, latency and EAP implementations can drop EAP-session after large numbers of round-trips (RFC 9191)
  - ML-DSA-65 certificates are approximately 5.5 KB to 7+ KB, depending on extensions; a 3-certificate chain would therefore be roughly 16.5 KB to 21+ KB.

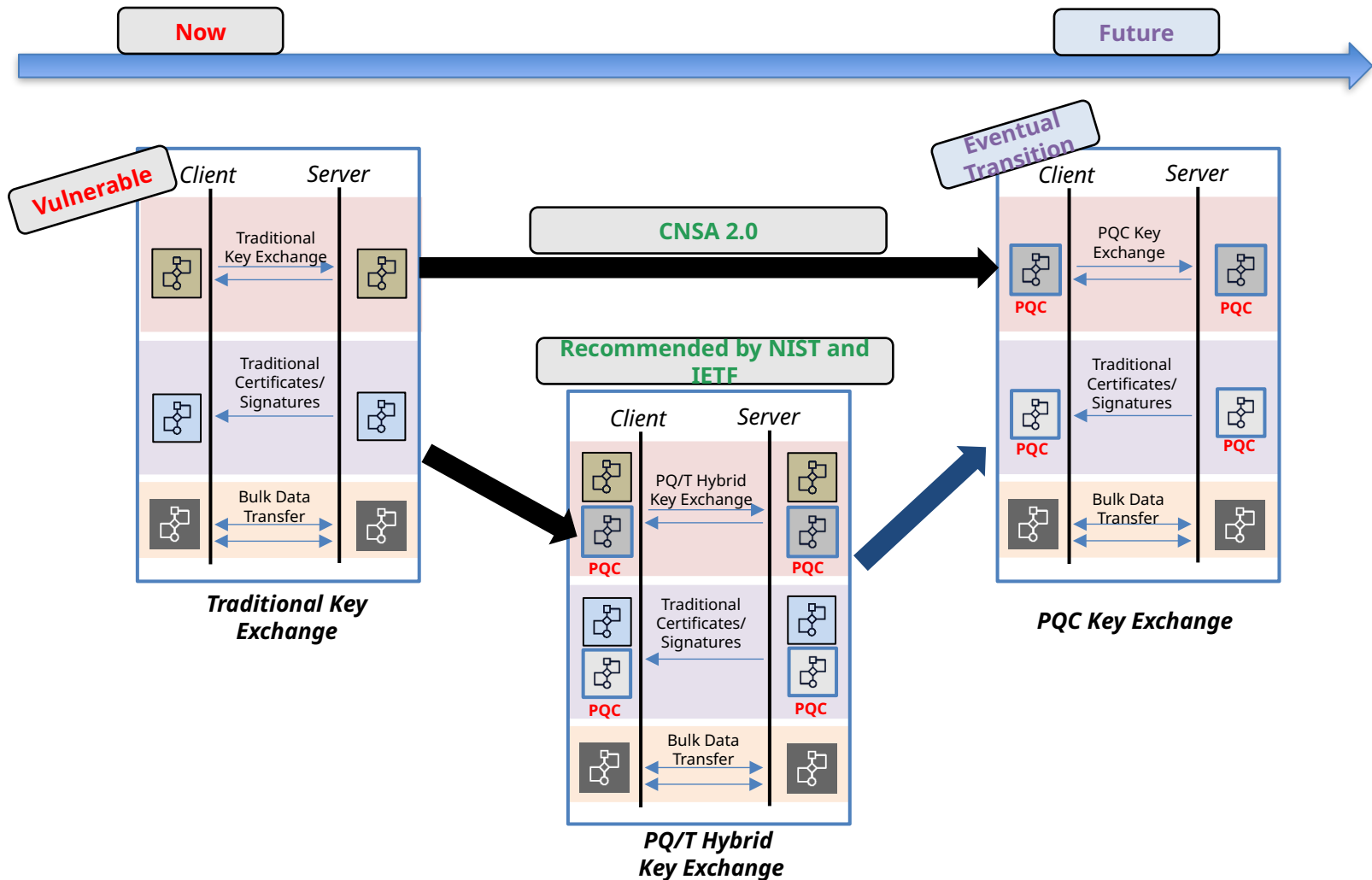
# Why PQC Auth is Critical for EAP-TLS and EAP-TTLS ?

- EAP-TLS deployments rely on X.509 certificates from CAs
- CRQC can compute client/server private keys from public keys in certificates before cert expiry
  - Enables real-time impersonation of access points (APs)
- Risk: user deception, privacy, and credential theft

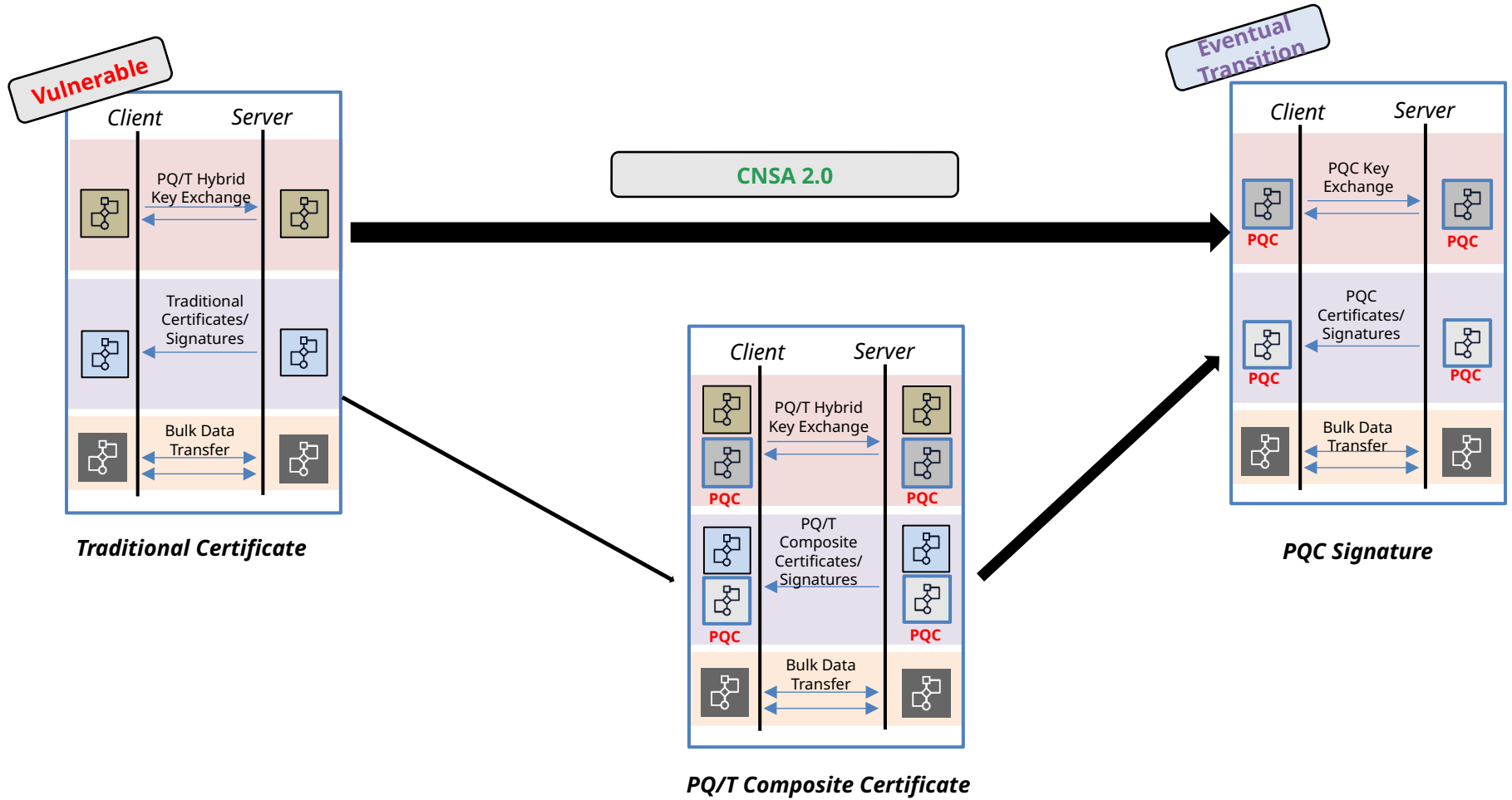
# Recommendations

- Hybrid/PQ for key exchange in EAP-TLS/TTLS
  - "ML-KEM" for the pure PQC key exchange column
  - "ML-KEM + Traditional" for the hybrid key exchange column
- PQ or hybrid certificates for authentication
  - "ML-DSA + Traditional" for the hybrid certificate/signature
  - "ML-DSA or SLH-DSA" for the pure PQC signature column
- Use cert chain optimization to avoid fragmentation

# PQ Key Exchange Migration



# Certificate/Signature Transition



# Use cert chain optimization to avoid fragmentation

- EST Integration for Chain Retrieval
- The draft defines new EST URIs for retrieving EAP certificate chains:
  - GET /.well-known/est/eapservercertchain
  - GET /.well-known/est/eapclientcertchain
- Clients fetch intermediate certs outside the TLS handshake during bootstrapping.
- When needed servers can also retrieve intermediate certificates for client authentication.
- Intermediate certificates are not exchanged during EAP-TLS handshake.

# Next Steps

- Comments and Suggestions are welcome
- Consider for WG adoption