



PQ algorithms in Secure Transport Protocols and Software Signing

Experiments and Lessons Learned

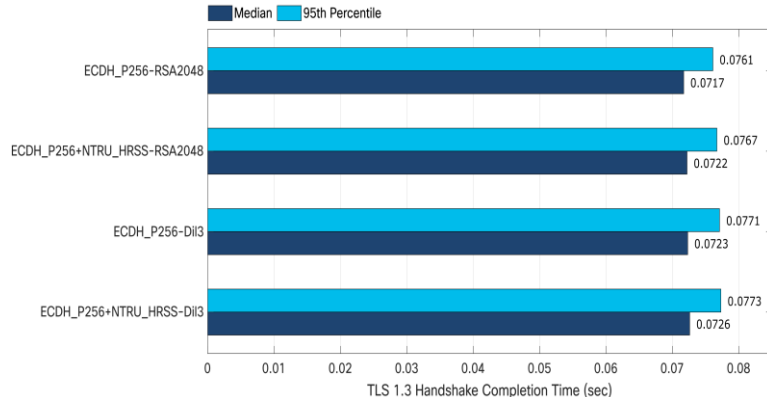
Panos Kampanakis

Cisco Systems

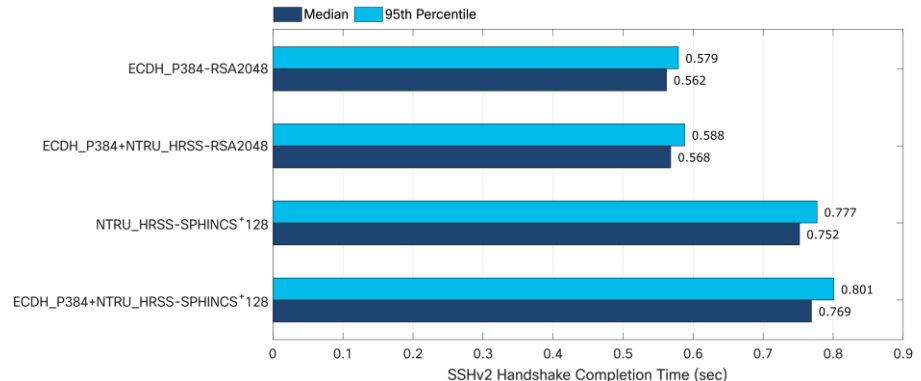
10/7/2020

Summary of Experimental Findings

- (D)TLS, SSH, IKEv2
 - Some lattice schemes seem to perform acceptably compared to classical algorithms.
- Software Signing, Secure Boot
 - HBS signatures (Stateful & Stateless (SPHINCS+)) seem to perform OK.



[From paper to appear in CoNEXT 2020]



[From paper to appear in CoNEXT 2020]

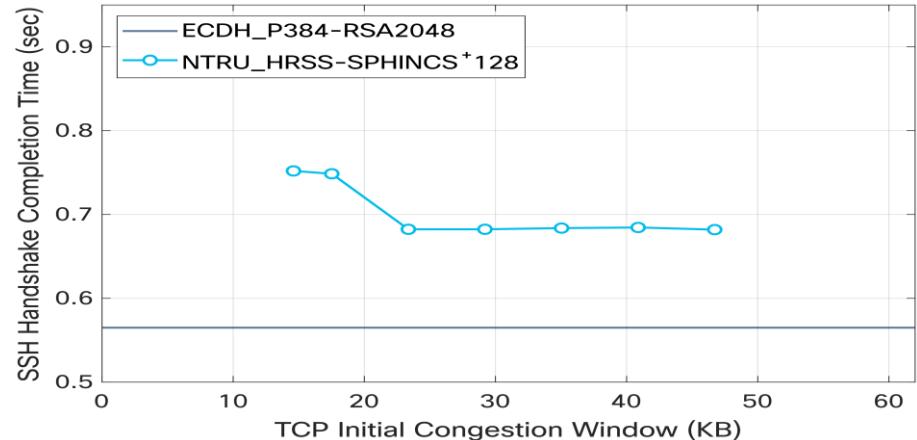
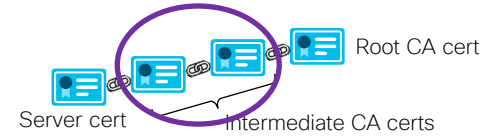
Summary of Experimental Findings

Challenges, Considerations

- Bigger PQ public key, ciphertext, signature sizes
 - can lead to extra round-trips due to the **TCP Initial Congestion Window**.
 - mean more packets, which means **more slowdown** in lossy environments (probability $1 - (1-p)^n$).
- Web: **PKI, OCSP, SCT signatures** increase the transferred handshake data even more.
- Keygen, encaps, decaps, sign, verify **performance is important** for “fast and short”, high-volume connection applications.
- PQ overhead can be amortized of **long tunnel lifetimes**.
- The migration of **long-lived Hardware Roots of Trust** is more urgent.
- **Interdependent standards** X.509, IEEE 802.1AR, TCG. CAB/F, UEFI, IETF.

Potential Optimizations for the data issue

- Intermediate CA caching and suppression in (D)TLS.
- Use small signature size schemes (e.g. Rainbow) where public keys are not transferred (e.g. Root CA cert, SCTs in TLS)
- Increase TCP Initial Congestion Window (after careful standardization).



Preparing for migration

- Predictions
 - **Hybrid Key Exchange** (protects against “store now and harvest later” and FIPS compliant). Probably will be come here to stay.
 - **Hybrid Signatures** have different pros and cons.
 - Based on history, we will probably see a **limited set of algorithms standardized** for various protocols, applications and usecases.
- **Temporary options** for some usecases (e.g. RFC8784, RFC8696)
- NIST standardization needs to finish. **Other standards** (IETF, IEEE, UEFI) **need to work in parallel**.
- **Experimentation and public discussion** to reach consensus (e.g. NCCoE, mailing lists, workgroups, conferences etc).

Thank you.

