Hybrid key exchange in TLS 1.3

draft-ietf-tls-hybrid-design-04

Douglas Stebila, **Scott Fluhrer**, **Shay Gueron**

UNIVERSITY OF WATERLOO

NSERC CRSNG

Motivation

- **Permit simultaneous use of traditional and post-quantum key exchange**
  - Enable early adopters to get post-quantum security without discarding security of existing algorithms
- Reduce risk from break of one algorithm
- Maintain standards compliance during transition

Goals

Define data structures for negotiation, communication, and shared secret calculation for hybrid* key exchange

Non-goals

- Hybrid/composite certificates or digital signatures
  - (LAMPS working group)
- Selecting which post-quantum algorithms to use in TLS
  - (NIST, CFRG)

* Some people use the word “composite” instead of “hybrid”.

Mechanism

Idea: Each desired combination of traditional + post-quantum algorithm & parameter set will be a new (opaque) key exchange “group”

- **Negotiation**: new named groups for each desired combination will need to be standardized
- **Key shares**: concatenate key shares for each constituent algorithm
- **Shared secret calculation**: concatenate shared secrets for each constituent algorithm and use as input to key schedule
  - Concatenation is a NIST-approved combiner [1]

[1] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf#page=10

Is it safe to use concatenation? ss = H(k1 || k2)

Aviram et al.:

If:

a) H is not collision-resistant
   - (and H-collisions can be found within lifetime of TLS session)

b) k_1 is adversary-controlled and variable length

c) ephemeral keys are reused

then it possible to learn k_2.

- Based on attack on APOP (MD5-based challenge response protocol); similar to CRIME attack.

- Possible but significant assumptions:
  - Need long session timeout
  - Ephemeral key reuse
- Assumption (b) not satisfied:
  - k_1 is fixed-length for all standardized TLS 1.3 DH groups
- => No changes made to this draft

- Worthwhile exercise: given long-lived hard-to-upgrade implementations, how robust should our protocol designs be to algorithm failure?

Aviram, Dowling, Komargodski, Paterson, Ronen, Yogev. Concatenating secrets may be dangerous, August 2021.
https://github.com/nimia/kdf_public

Next steps

- No known pending tasks for this draft

- Several interoperable implementations:
  - Open Quantum Safe OpenSSL and BoringSSL forks [1]
  - wolfSSL [2]
  - s2n-tls [3]

- Specific PQ algorithms to be identified outside of this document
  - NIST Round 3 conclusion → CFRG → TLS

- Could move to Working Group Last Call?

[1] https://github.com/open-quantum-safe/openssl • https://github.com/open-quantum-safe/boringssl
[2] https://www.wolfssl.com/hybrid-post-quantum-groups-tls-1-3/
[3] https://github.com/aws/s2n-tls/blob/main/pq-crypto/README.md