draft-AuthKEM

IETF 113 update

Presentation at IETF 111

- Introduction of the draft
- Not a lot of space in RFC draft for “why?”
- Some questions on the mailing list and in chat during the meeting

This presentation

1. What do we need from you?
2. `CHANGELOG`
3. AuthKEM Abridged

What do we need from you?

What we would like help with

1. Feedback on draft and open problems
2. RFC-ness still need some work
3. NIST PQ finalists announced soon
    a. Investigate if they fit in your TLS usage
    b. Could AuthKEM solve problems for you?

CHANGELOG

CHANGELOG (github: claucece/draft-celi-wiggers-tls-authkem)

- Restructure of the draft
  - Splits protocol diagrams from implementation details
  - Hopefully much more readable

CHANGELOG (github: claucece/draft-celi-wiggers-tls-authkem)

- Use HPKE context separation (PR#18)
- Added cert_req_ctxt to KemEncapsulation messages (PR#14)

AuthKEM abridged

AuthKEM Abridged

- Feedback very welcome!
- Goals:
  - Provide intuition
  - Prevent confusion
  - Answer questions
  - Don’t shy away from open questions

https://wggrs.nl/docs/authkem-abridged/

Why KEMs for auth?

- Single algorithm for KEX / Auth
- Soon™ (EoM): PQ Signature schemes; they are:
  - Huge (Dilithium / ~~Rainbow~~), or
  - Require special hardware support for reasonable performance (Falcon)
  - NIST has announced it expects to standardize at most one of Dilithium / Falcon.
- PQ KEMs are generally fast AND small (relatively)
- Bonus: Offline Deniability

Why not draft-semistatic?

- (EC)DH semi-static is *beautiful*
  - But not available with PQ KEMs
  - Generally, any protocol that does $((g^x)^y)^s$ doesn't work with KEMs/HPKE

PQ commutative group actions (e.g. CSIDH) are not a solved problem yet.

Why now?

- Standardizing new auth is very hard
- Good opportunity to reconsider authentication mechanisms
- Last meeting already identified rough edges
  - #16 No explicit CertificateRequest authentication
- Filing down all the rough edges before this is put in production will take time

Is this finished enough for TLSWG?

- Research on security & performance on-going
  - Pen-and-paper proof
  - Various experiments both done and still planned
  - Working on Tamarin proof
- Research has a hard time identifying practical or deployment problems
- Draft & discussion help drive both forward

Other things to read in AuthKEM Abridged

- Why we added a new handshake secret
- Why we think the extra half round-trip doesn't matter for performance
- Why we extend `signature_algorithms` with KEMs
- Why we think sending Client Data to the server "early" is fine.

We would like to hear from you

If you…

- … want to help with / have feedback on the draft
- … want to comment on the open issues
- … have any questions that are unanswered
- … have an interesting (embedded?) use case that we should investigate