SPAKE2

Benjamin Kaduk

15 November 2017

History

- Version -03 submitted in February 2016
- Got comments from Greg Hudson and Alex Elsayed
- No motion on the document afterwards
- Reconstructed -04 submitted last month
- New version coming soon!

Kerberos Pre-Authentication

- “Traditional” Kerberos has the KDC send a ticket to any client that asks.
- Users choose weak passwords
- Modern deployments “‘pre-authenticate”’ ’ users before sending a ticket, but a passive observer still gets a ciphertext to brute-force
- Other options (FAST, PKINIT) are hard to deploy
- PAKE provides protection against offline attacks and also enables second-factor protection without independent attacks on a single factor

Kerberos Pre-Authentication

Why is SPAKE2 good for Kerberos?

- consistent with EC crypto
- computes the shared key after just one message from each side
- small number of group operations

Please review `draft-ietf-kitten-krb-spake-preauth`!

Other use cases

Authenticate file transfer via a password exchanged over the phone:
`https://github.com/warner/magic-wormhole`

Greg Hudson's Review

`https://www.ietf.org/mail-archive/web/cfrg/current/msg07928.html`

- "SPAKE2+ doesn't use w0 or w1 in the derivation of K'" — closer to SPAKE1 than SPAKE2?
- M and N generation is inconsistent between text and code: non-overlapping vs. overlapping output from the hashing chain
- cofactor check: prime order quotient vs. multiply-by-cofactor
- (formatting and editorial nits)

Alex Elsayed's review

Points out that this PRF+-like scheme for arbitrary-length output via repeated hashing is a little silly. HKDF instead?

As AGL notes, this is just for M and N generation, so it's not really important how elegant it is, just that it's reproducible.

Older issues

- Dan Harkins pointed out that we need to be precise about how many bytes we're taking as the "initial sequence of bytes" and whether we preprend or overwrite with 0x02/0x03
- OIDs have both text and binary representations — we use text, but should more explicitly say so
- Nail down interaction between point format and picking group elements from the iterative hashing scheme for M and N
- irtf-cfrg-curves support (ed448goldilocks and ed25519?): when that comment was made, neither had a point format that admits addition; the kitten document includes M and N for ed25519

Any general PAKE topics to consider?

`https://www.ietf.org/mail-archive/web/cfrg/current/msg08365.html`

Stanislav notes that we might consider moving up a level of abstraction, to consider what use cases and requirements there can be for PAKE algorithms.

How would SESPAKE/SPAKE/etc. compare — how many PAKEs do we need?

Do we need to consider the interaction of key confirmation and the surrounding protocol (e.g., final K' derivation for SPAKE2), or just a raw primitive that could be used for TLS/IKE/etc.?

Open Questions

Any other concerns about the document?
More review needed?