TEAPv2

EMU - IETF 123

InkBridge Networks

WHAT'S WRONG WITH TEAPv1?

- Key derivation is complex. Start with

  $\texttt{key}_0 = \texttt{TLS-Exporter(…)}$

- Then, for each round:

  $\texttt{MSK-key}_n = \texttt{mash}(\texttt{MSK}_n,\ \texttt{key}_{n-1})$ $\texttt{EMSK-key}_n = \texttt{mash}(\texttt{EMSK}_n,\ \texttt{key}_{n-1})$

  $\texttt{key}_n = \texttt{choose???}(\texttt{MSK-key}_n \texttt{ or } \texttt{EMSK-key}_n)$

- MSK and/or EMSK might not exist, and MSK is sometimes set to zero.
- Everyone did something different for the key derivation.

InkBridge Networks

EVEN MORE "WHAT'S WRONG WITH TEAPv1"

- Crypto-Binding is tied to authentication methods
  - Except for Basic-Password
  - and not for other inner exchanges (CSR, etc.)
- There is a lot of uncertainty around portions of the protocol which have not been well tested
  - It would be good to have interoperable implementations for all of TEAP before any new specification is published.

HOW TO FIX IT (THEORY)

- Define a new derivation, which is simpler.

  `Start with` key_0 `= TLS-Exporter(…)`

  key_n `= TLS-Exporter(`MSK_n`,` $EMSK_n$`,` key_{n-1}`)`

- Always uses both MSK and EMSK.
- If EMSK isn't available, it's set to zero
- If MSK isn't available, define a pseudo-MSK which is based on the data being exchanged: `pseudo-MSK = TLS-Exporter(TLV-Value)`

HOW TO FIX IT (PRACTICE)

- Write reference implementations which everyone can test against
  - Ideally with full protocol functionality
- Once verified, update the draft

~~rough consensus and running code~~

rough consensus and running code

QUESTIONS

- Derivation is

  $\texttt{key}_n = \texttt{TLS-Exporter(MSK, EMSK, key}_{n-1}\texttt{)}$

- Maybe use `TLS-PRF()` instead?
  - I'm not familiar enough with the differences to make a judgement call

- Anything else?