Pseudorandom cTLS

TLSWG, IETF 112 (November 2021)
Authors: Ben Schwartz, Chris Patton
Intended status: Experimental
Slides v00

What

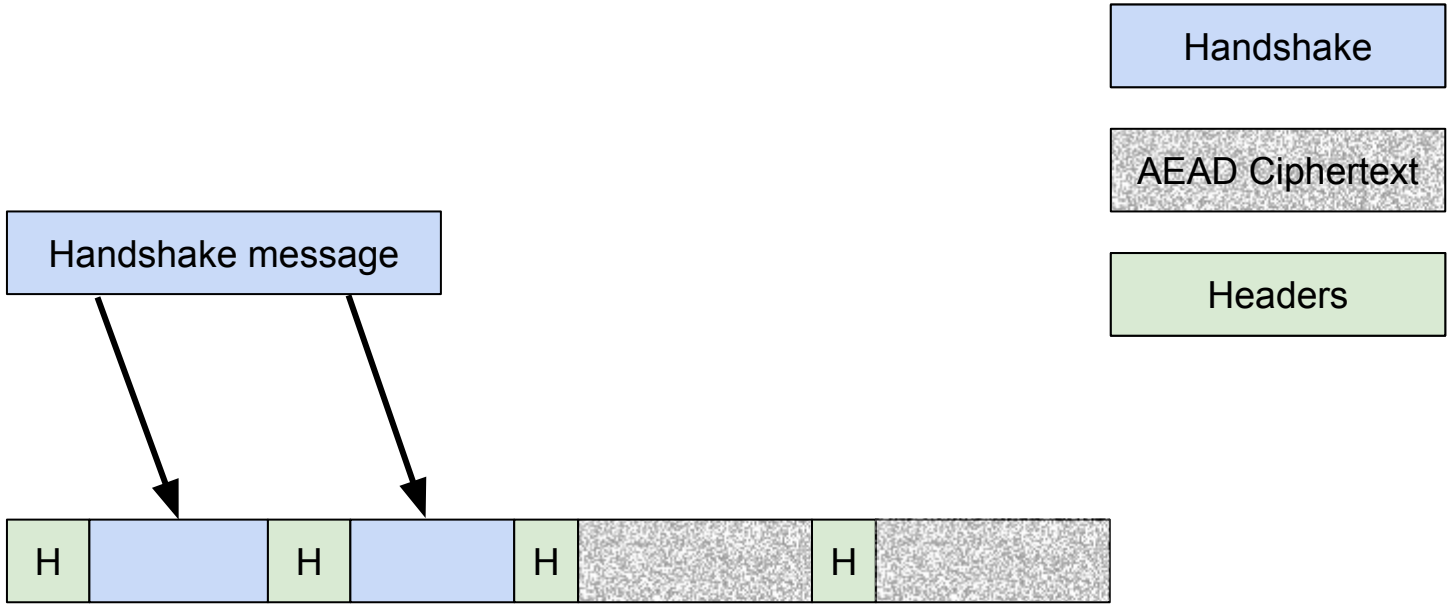
- An experimental extension for cTLS that makes the wire image purely pseudorandom, i.e. indistinguishable from random to an observer who doesn't know the template.
- This is possible because:
  - cTLS requires a pre-shared "template" that client and server agree on out-of-band.
  - cTLS is already unparseable to anyone who doesn't know the template.
- Layered between cTLS (unmodified) and the transport.

Why

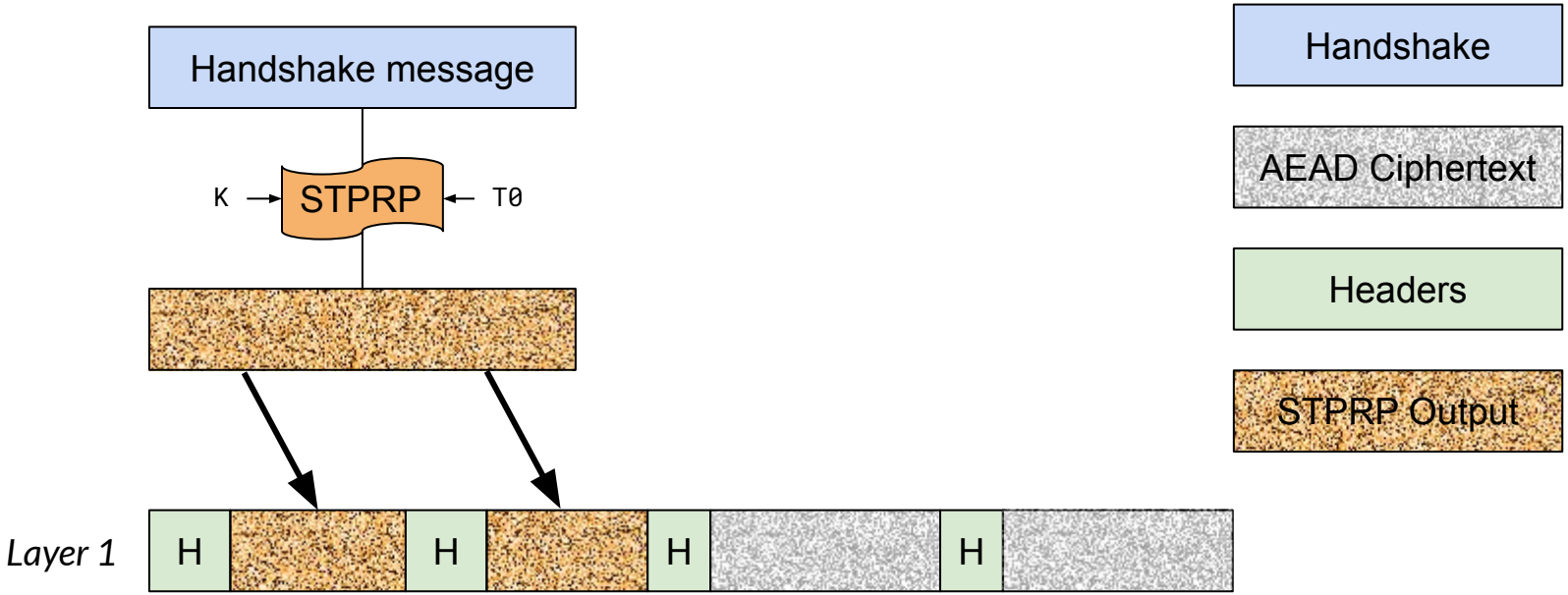
- Security - Prevents protocol confusion attacks (e.g. NAT Slipstream) by ensuring that neither party can influence the other's output.
- Privacy - Conceals which cTLS template is in use.
- Protocol Agility - Ensures that the bitstream is only parsed by authorized parties.

How

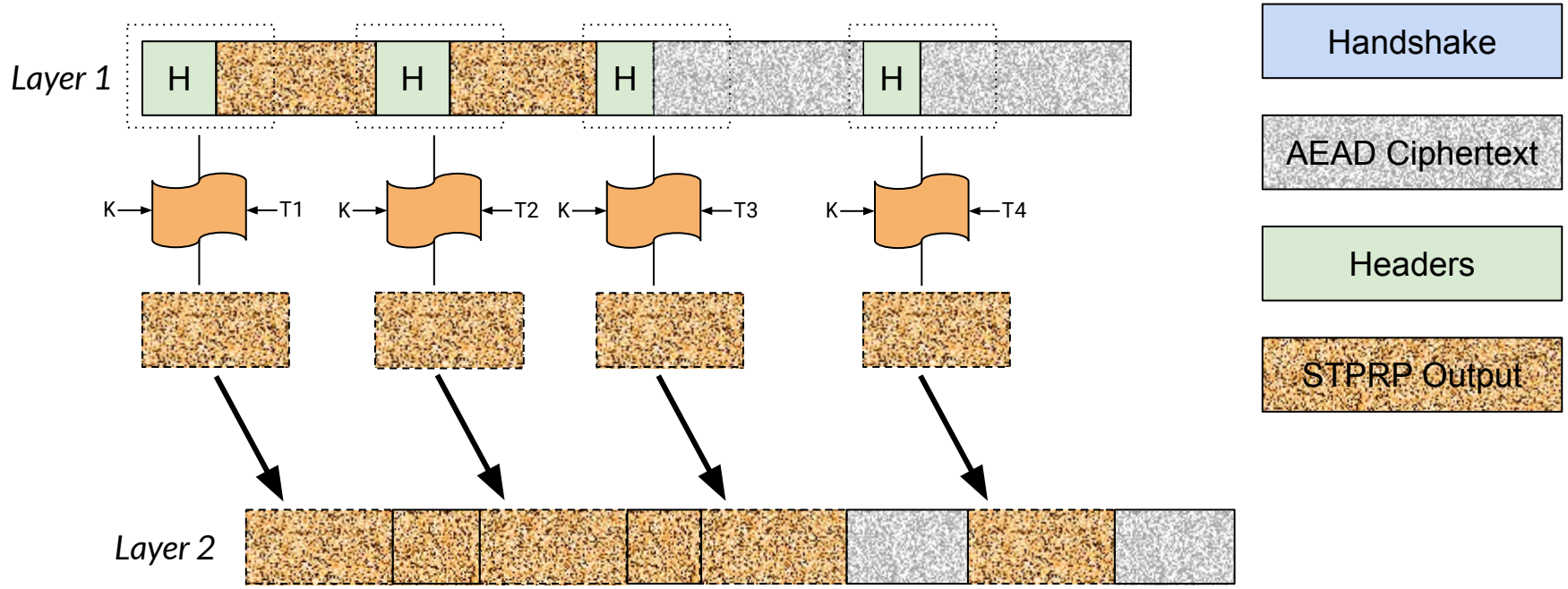
- Uses a Strong Tweakable Pseudorandom Permutation (STPRP).
  - Also known as a “wide” or “variable-input-length” block cipher.
  - Can be constructed from conventional primitives, e.g. AES.
- Applies the STPRP to
  - plaintext messages
  - segments containing headers and at least 16 bytes of ciphertext
- No ciphertext expansion (zero overhead)
  - Relies on TLS’s integrity checks (AEAD and Finished)



Baseline cTLS Example stream



Pseudorandom cTLS Example stream (First Layer)



Pseudorandom cTLS Example stream (Second Layer)

Next steps

- Seeking WG input on
  - Cryptographic primitives (currently STPRP)
    - STPRP enables easiest analysis but may not be necessary
    - Standard AEADs can be used if we allow overhead > 0
  - Construction of the “tweak” (analogous to nonce, may repeat)
  - Strategy for formal analysis
  - Best way to integrate with cTLS
  - Preferred document status (currently Experimental)
- Some requested changes to the cTLS draft
- May pursue WG adoption at some future point
  - What would you like to see in an adopted version?

Bonus: Privacy threat model

- Passive adversaries
  - All Pseudorandom cTLS streams should look the same, despite different templates and keys
  - Timing and size leaks are out of scope (but should be addressed by some future draft…)
- Active adversaries
  - A probing adversary shouldn't be able to tell which template is in use.
  - Preventing adversaries from measuring any aspect of the template requires very carefully tuned error responses (fragile!).

Bonus: Dirty details

- Handshakes messages need to contain good entropy, and each fragment needs to be at least 16 bytes long.
  - TLS Alerts have to be padded with randomness.
    - We could also add 16 zeros as a MAC…
- cTLS/UDP headers have unpredictable lengths, which can lead to overlapping STPRP segments if a datagram contains multiple short records. Encoding must proceed back-to-front, starting with the last record.
  - This could be fixed in cTLS!