TLS Crypto Constructs

draft-stjohns-tls-tls13-crypto-infra-00

Mike StJohns, IETF 90

TLS PRF Now

- TLS 1.2 PRF
  - Uses Recursive HMAC construct to generate pseudo-random byte stream (PRBS)
  - PRF base construct is a HASH construct – defaults to SHA256
  - Could be used with CMAC construct, but needs key length adaptation function.
  - TLS Specific

TLS PRF Proposal

- Use Counter based construct with MAC to generate pseudo-random byte stream (counter-based Keyed PRBG) - basically a well-known Key Derivation Function construct
- Can be used with CMAC or HMAC constructs
- Can be parallelized

TLS MAC Usage Now

- Finished.verify_data is MAC over HASH of handshake
  - Can’t be used with CMAC
  - Uses TLS1.2 PRF as MAC function
- MAC is keyed by master_secret
  - But master_secret also used for deriving session keys
    - Violates good cryptographic usage

TLS MAC Usage Proposed

- Finished.verify_data becomes MAC over handshake (rather than over HASH of handshake)
  - Permits usage of CMAC constructs
  - MAC is HMAC or CMAC, not TLS pseudo-random byte generator construct
- Premaster_secret derivation yields two keys – master_secret and finished_key
  - Finished_key used to key HMAC or CMAC

TLS Key Derivation Now

- Keys derived from pre_master or master_secret plus mixins
- Derivation step uses key with pseudo-random byte stream generator and splits it up into individual keys
  - But!!
    - key stream doesn’t change if derived key lengths change
    - key stream doesn’t change if key types change

TLS Key Derivation Proposed

- Use new counter based keyed PRBG (replacement for current TLS PRF)
- Add key lengths and key types to mixins for the pseudo-random byte stream generation
- Allows secure implementations (e.g. HSMs) to detect and enforce key policies based on key types and lengths
- Causes key stream to change completely if any key length or key type changes
- Also, generate master_secret and other handshake secrets to the native length of the PRF
  - Currently, master_secret has to be compressed to length of underlying PRF hash function (RFC2104, section 3)

TLS IV Generation Now

- IV's generated as side effect of current master_key to session_key derivation
  - IV's are actually part of same key stream used for session keys
  - IV's are public data, but keys are private data
  - Enforcement of public/private data separation isn't possible since generated in same stream

TLS IV Generator Proposed

- Use new Counter-based Keyed PRBG
- Use key of all “0”s
- Use mixins (same as now) of client and server Randoms
- If secret key is required, derive a third key from pre_master specific for this purpose.