TLS Working Group Document Status

Eric Rescorla

Published RFCs

- The TLS Protocol Version 1.0 (RFC 2246)
- Addition of Kerberos Cipher Suites to Transport Layer Security (TLS) (RFC 2712)
- Upgrading to TLS Within HTTP/1.1 (RFC 2817)
- HTTP Over TLS (RFC 2818)
- AES Ciphersuites for TLS (RFC 3268)
- Transport Layer Security (TLS) Extensions (RFC 3546)
- Transport Layer Security Protocol Compression Methods (RFC 3749)
- Addition of Camellia Cipher Suites to Transport Layer Security (TLS) (RFC 4132)

In RFC-Ed Queue

- The TLS Protocol Version 1.1 (draft-ietf-tls-rfc2246-bis-13)
- Pre-Shared Key Ciphersuites for Transport Layer Security (TLS) (draft-ietf-tls-psk-09)
- Transport Layer Security (TLS) Extensions (draft-ietf-tls-rfc3546bis-02)

Last Call Requested

- ECC Cipher Suites for TLS (draft-ietf-tls-ecc-12.txt)

Hash Functions, MACs, and PRFs, oh my!

Eric Rescorla

Background: the status of SHA-1 and MD5

- Demonstrated collisions in MD5
  - With desktop-level computing power
- Theoretical collisions in SHA-1
  - Current work factor 2^64
- TLS still fairly safe
  - Collisions not really controllable
  - We depend mostly on preimages anyway
  - No reason to believe this extends to an attack on HMAC
- But it's still time to think about transition

Uses of Hash Functions in TLS

- Individually and negotiable
  - Certificates
  - Per-record MAC
- MD5 and SHA-1 hardwired
  - Digitally-signed element (for ServerKeyExchange and CertificateVerify)
  - KDF
  - Finished message

Certificate Selection

- Problem: I have certs signed with different algorithms
  - Need to somehow select one
- General principle: this is somehow negotiable.
  - As a side effect of cipher-suite?
  - Separately negotiated

Digitally-signed

- RSA
  - Sign a concatenated MD5/SHA of handshake messages
- DSA/ECC
  - Sign a SHA-1 hash
- Idea: replace with a single hash function
  - Again, how to negotiate this?

KDF

- HMAC-based PRF construction
  - XOR SHA-1 and MD5 values
- Idea: switch to a negotiated single hash function
  - Retain basic PRF structure

Finished Message

- Uses the same PRF as for the KDF
  - Current structure: $PRF(H(Handshake\_messages))$
  - This avoids the need to buffer (key is first imput to PRF)
  - Do we retain this structure?
- Additional problem: the Finished messages provide downgrade protection
  - Only as strong as weakest common hash function
  - We're now in the business of approving/disapproving algorithms
    * Hard to get around this
    * Reminder: it's mostly preimages we care about

Other things people have asked for

- Non-HMAC-based integrity check modes (GOST)
- Longer nonce values (Housley)

What next?

- These changes would imply a TLS 1.2
- ... which means a charter change
- Is this something the TLS WG wants to do?