Agenda

Time	Speaker	Topic
5	EKR	Agenda Bash
15	Brian Minard	draft-dugal-tls-ecmqv-00
10	Nagendra Modadugu	draft-ietf-tls-ctr-00
10	Russ Housley	draft-housley-tls-authz-extns-00.txt
10	Yngve Petterson	Interop issues
10	Magnus Westerlund	draft-ietf-mmusic-rfc2326bis-12
60	EKR	draft-ietf-tls-rfc4346bis-00

draft-ietf-tls-rfc4346bis-00

Eric Rescorla

Network Resonance

ekr@networkresonance.com

Background

- RFC 4346 (TLS 1.1) is just waiting for RFC-Ed to push it out
- Recent attacks on MD5 and SHA-1
  - Don't *immediately* threaten TLS, but...
- Rechartered to do a TLS 1.2
  - To do hash function fixes
- Output is draft-ietf-tls-rfc4346

Changes in this draft

- Merged in TLS Extensions and AES Cipher Suites
- Extension for client to indicate which hash functions are supported in certificates
- Replacement of MD5/SHA-1 in the PRF
- Replacement of MD5/SHA-1 in the digitally-signed element.

Digitally-signed

- RSA
  - Sign a concatenated MD5/SHA of handshake messages
- DSA/ECC
  - Sign a SHA-1 hash
- Replaced with hash used to sign the certificate
- ... or SHA-1 for DSA/ECDSA

KDF

- HMAC-based PRF construction
  - XOR SHA-1 and MD5 values
- Retain basic PRF structure
  - based on negotiated hash function in cipher suite
  - What to do about MACs which aren't hash-based?
- And what about other PRFs? GOST, NIST 800-56, etc.

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)
    - But it's less secure
    - Should we move to PRF of the whole handshake
- **But...** 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

Framing the Discussion

- Certificate selection can be done by extension
- The main reason for a TLS 1.2 is to replace the PRF and digitally-signed elements
  - There is no currently known threat to these
  - But it seems ugly to be tied to hashes that don't meet there design goals
- So should we be making a proactive change like this?