TLS WG

Eric Rescorla
Network Resonance
ekr@networkresonance.com

Agenda

1. Agenda bashing (5 minutes) - chairs
   - Bluesheets
   - Agenda changes
   - Scribe for minutes
   - Jabber scribe
2. Document status (5 minutes) - chairs
   - Progress since last IETF
   - IANA considerations reminder
3. TLS 1.2 (45 minutes?) - Eric Rescorla
4. Counter Mode IVs (10 minutes) - Eric Rescorla
5. TLS Record Layer bugs (10 minutes) - Pasi Eronen
6. TLS Evidence Extensions (15 minutes) - Russ Housley
7. KDF (15 minutes) - Tim Polk
8. SPNEGO and TLS (5 minutes) - Stefan Santesson

Document Status

TLS 1.1	RFC 4346 (PS)	Published
Extensions (revised)	RFC 4346 (PS)	Published
Datagram Transport Layer Security	RFC 4347 (PS)	Published
ECC Cipher Suites	RFC 4492 (PS)	Published
Transport Layer Security (TLS) Session Resumption without Server-Side State	RFC 4505 (PS)	Published
TLS User Mapping Extension	RFC 4681	**Published**
TLS Handshake Message for Supplemental Data	RFC 4680	**Published**
Transport Layer Security (TLS) Authorization Extensions	draft-housley-tls-authz-extns-07	RFC Ed Queue
Using OpenPGP keys for TLS authentication	draft-ietf-tls-openpgp-keys-10	**RFC Ed Queue**
Using SRP for TLS Authentication	draft-ietf-tls-srp-12	Editors revising
Pre-Shared Key Cipher Suites with NULL Encryption for Transport Layer Security (TLS)	draft-ietf-tls-psk-null (Proposed Standard)	**RFC Ed Queue**
AES Counter Mode Cipher Suites for TLS and DTLS	draft-ietf-tls-ctr-01.txt	Working...
The TLS Protocol Version 1.2	draft-ietf-tls-rfc4346-bis-02.txt	Working...

TLS 1.2 Status

Eric Rescorla

Network Resonance

ekr@networkresonance.com

TLS 1.2 Status

- New draft (-02)
  - Technical
    - Fixed PRF text (but still need to discuss)
    - Added support for combined authenticated encryption modes (per charter)
    - $verify\_data$ values now computed with $Hash()$
  - Editorial
    - Protocol version fixes (cleanup)

PRF Discussion

Pasi asks mailing list:

> 1) Default PRF is tied to the TLS version number: in other words, ciphersuites that don't specify anything else (including all currently defined ciphersuites) will use the new TLS 1.2 PRF (details of which are TBD) when TLS 1.2 is negotiated.
>
> 2) The new PRF will be used only for new ciphersuites that explicitly say so; all currently defined ciphersuites continue to use the current (TLS 1.0/1.1) PRF even when TLS 1.2 is negotiated.

General consensus on list was for #1. Still need to nail down details.

Original Proposal for default PRF

- $P\_hash()$ using only one hash function
  - This is what -01 was supposed to say but I broke it
- Hash function is tied to HMAC
- Default hash function is SHA-1
  - Nothing weaker should be specified
- Two proposed variants
  - Default hash function is SHA-256
  - Use a fixed hash function not tied to HMAC (probably SHA-256)
- Recommendation: ???

Combined authenticated encryption modes

- One algorithm that provides both encryption and authentication
  - With a single key
  - Examples: GCM, CCM
  - See draft-mcgrew-auth-enc-001
- How do we interface with them?
  - Just make a hole... cipher suites defined in other drafts
  - s/stream, block/stream, block, aead/
  - No separate MAC value to encrypt
  - MAC key no longer necessary
  - Read Section 6.2.3.3

verify_data

- Discussion on list about whether to feed it into PRF directly
  - My read of people's opinions: NO
- Current text: $Hash(handshake\_messages)$

Target Schedule

- Reach consensus on this stuff here and on list
- New version by end of year
- Be done by Prague

TLS Counter Mode

Eric Rescorla
Network Resonance
ekr@networkresonance.com

Document Almost Done... we thought

- Current block format:

```
struct {
   case client:
      uint48 client_write_IV;  // low order 48-bits
   case server:
      uint48 server_write_IV;  // low order 48-bits
   uint64 seq_num;
   uint16 blk_ctr;
} CtrBlk;
```

- Issue raised by Steve Kent
  - Should we use an explicit IV?

Why an explicit IV?

- Unique IVs are a security condition
  - Much more than with CBC or stream ciphers
  - This suggests they need to be within the FIPS-140 evaluation boundary
- Obvious solution: crypto hardware controls sequence number
  - This is a problem with more than one hardware unit
  - ... need to coordinate sequence numbers
- An explicit IV lets each hardware unit generate its own IV
- 64 bits is plenty

Strawman Explicit IV Proposal

- Use a 64-bit explicit IV

```
struct {
   case client:
      uint48 client_write_IV;  // low order 48-bits
   case server:
      uint48 server_write_IV;  // low order 48-bits
   uint64 iv;
   uint16 blk_ctr;
} CtrBlk;
```

- Note: the IV can't be randomly generated
  - Birthday collision problems
  - Use a counter or LSFR
  - Can segment the space between hardware units
- Is this worth paying 8 bytes/packet for?