# What's new in TLS 1.3 (and OpenSSL as a result)

Rich Salz

# Timeline

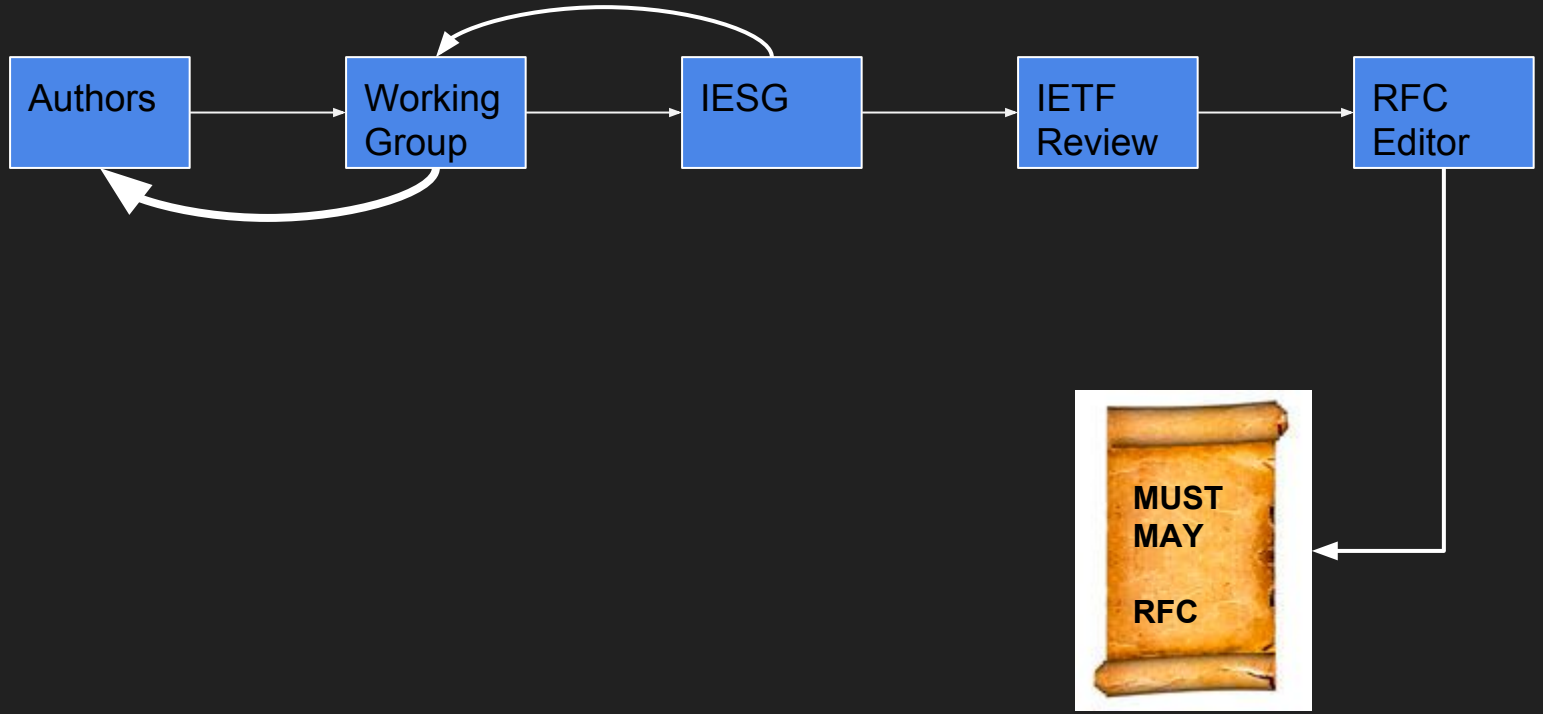
- TLS 1.0 RFC 2246 January 1999
- TLS 1.1 RFC 4346 April 2006
- TLS 1.2 RFC 5246 August 2008
  - … a whole bunch of RFC’s for new crypto, secure renegotiation, etc.
- Charter #5, approved October 2015

# IETF Structure and Workflow

- IETF is divided into *areas*: Security, Operations, DNS, etc.
- Each Area as 1-3 area directors
- IESG is all the AD’s
- AD’s approve working groups, appoint chairs
- Working group chairs run the discussions, appoint doc editors
- Doc editors/authors write the docs

- Document flow is bottom-up

# IETF Structure and Workflow



# TLS 1.3 Goals

- Encrypt as much of the handshake as possible
- Reduce handshake latency, ideally to one or zero roundtrip for repeated handshakes.  The aim is also to maintain current security features.
- Update record payload protection cryptographic mechanisms
- Reevaluate handshake contents, e.g., is time needed in client hello?
- consider the privacy implications of TLS1.3 and where possible (balancing with other requirements) will aim to make TLS1.3 more privacy-friendly,

# Crypto Improvements: handshake

- Only use ECDH (or DHE, but nobody will); all connections have PFS
  - Except 0RTT, about which more anon
- Most things are encrypted
  - SNI is not
- Gory details: NIST P-256 or X25519 will be most common, especially on the web

# Crypto Improvements, bulk encryption

- There are three cipher types and two variations:
  - Chacha/Poly; AESGCM128 (256); AESCCM 128 (256)
- Cipher suite no longer specifies key exchange or authentication mechanism
  - Those are negotiated options
- The bulk encryption ciphers are all modern, secure, and AEAD-only -- in addition to confidentiality, you also get integrity (authenticity)
- The dozens (!!!) of ciphers allowed in TLS 1.2 and earlier are not specified.
  - The IANA registry will say “recommended” or “no comment”

# Crypto Improvements, authentication

- DSA removed
- RSA kept, prefer RSA-PSS over PKCS 1.5
- Ed25519 and Ed448 added
- It will take the CA world time to catch up to this

- PSK cipher-suites now part of handshake (resumption), not special ciphers any more

# Other cleanup and generalizations (OC&G)

- Renegotiation is gone
  - Server can ask client for its cert; rekey is separate
- Almost every message has extensions
  - OCSP or CertficateTransparency stapling
  - On server and client side
- Padding at the TLS layer
  - Slightly controversial
- There is only one hash mechanism, HKDF (RFC 5869) and it’s used consistently (and correctly)

# OC&G: TLS as a Service

- Generating session keys and exporting them is useful
- NTPSecurity, QUIC, TokenBinding
- It’s fairly straightforward to do this with TLS 1.3 (API’s permitting)

# OC&G: Renegotiation is gone

- Main use-cases for it:
  - Request client cert
  - “Step-up” crypto algorithm
  - Re-key
- It was buggy and unclean and a source of problems
- Special “certificate request” message
  - Maybe H2 will do something here at their layer
- Strong crypto; document key usage limits (sic); KeyUpdate message

# OC&G: Session resumption

- Prefer session over tickets
- Server can send session information at any time
- Session is like PSK
- PSK is like session resumption
- If you’re reconnecting, send data with the resumption
- … bingo, 0RTT

# 0-RTT, early data

- Client connects, C&S do the ECDHE dance.
- Client remembers the server’s key share
- Next time, client reconnects and sends data encrypted with that key
- Avoid an extra round-trip; less latency
  - Web is faster
  - We all make money quicker

# 0RTT has a big butt

- There is no PFS with early data
- That data can be replayed elsewhere
  - GET is idempotent
- Nobody really listened until Colm@Amazon spoke and posted two weeks ago
- Now everybody’s trying to figure out how to “save” it with liberal use of RFC2119 keywords.

# TLS 1.3 Status

- “>This close<” to IESG review
- Chrome and FF and some others are at Draft-18
- Draft 19 made some minor changes, mainly wording
- Draft 20 changed some hash inputs; is incompatible
- OpenSSL has branches for 18 and 19; and master is always latest

```
https://github.com/openssl/openssl
git clone \
    git@github.com/openssl/openssl.git
 git co tls1.3-draft-18
```

# OpenSSL and TLS 1.3

- Blog entries: OpenSSL and Akamai
  https://www.openssl.org/blog/blog/2017/05/04/tlsv1.3/
  https://blogs.akamai.com/2017/01/tls-13-ftw.html

- Based on 1.1.0; it will be 1.1.1
  - *Binary* and *source* compatible

- As previous slide said, it works and interoperates now :)

# Thanks

- For your time and attention
- And for flooding the TLS@IETF.ORG mailing list with messages of support.  I mean opinions.