

DTLS Update

draft-ietf-tls-rfc4347-bis-05

IETF 80

Eric Rescorla

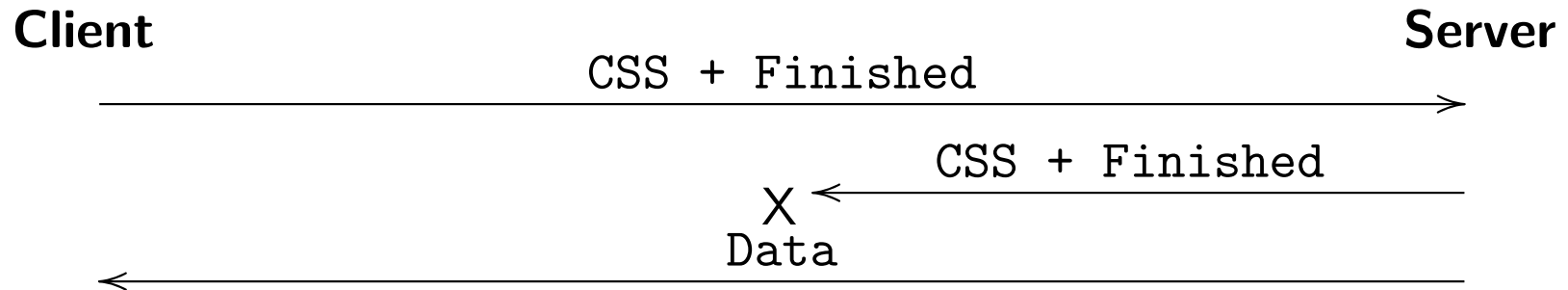
ekr@rtfm.com

Status Overview

- Document taken to IETF LC
- Lots of good comments (both IESG and non-IESG)
- New version (-05)
- A few open issues left

Final Handshake Message Loss

Basic case:



- Server thinks it is done; Client thinks it isn't
- Parallel case with resumption and the client's last message
- Two issues
 - What should the client do?
 - Why doesn't this cause deadlock?
- This was at least unclear and likely unspecified

Behavior Under Loss

- Receiver
 - Epoch allows detection of this case
 - MUST NOT accept non-handshake messages in a new epoch prior to `Finished`
 - MAY buffer those messages and process them later (or not)
 - * MAY shortcut the retransmit timer when receiving unexpected application data messages
- Sender
 - Sender MUST save the last flight for 2MSL
 - Respond to a retransmit of the other side's flight with a retransmit

Epoch Wrapping

- What does epoch wrapping do? [Farrell]
- Prohibit wrapping [MUST rehandshake first]

What about state loss?

- What happens when a client loses state? [Kaufman]
 - It sends a new ClientHello
 - This can be confusing
- New text:
 - Epoch = 0 indicates a new handshake
 - Server **MUST NOT** destroy existing association until reachability established
 - * Cookie exchange
 - * Finished exchange

IANA Considerations

Draft read:

“Upon registration, new TLS cipher suites **MUST** indicate whether they are suitable for DTLS usage and what, if any, adaptations must be made.”

- Unfortunately, there was nothing here about an IANA registry
- Added one (Section 7)

Miscellaneous Mostly Editorial

- Clarifications throughout about DTLS versus IP fragmentation
- Clarification about backward compatibility
- Add reference to v6 Packet Too Big
- Implementations MUST propagate PMTU indications (i.e., ICMP*)
- Silent discard may include logging
- Added a changes list at the end (thanks to Peter Saint-André)

2MSL

- Concern from Miguel Garcia that we referenced TCP MSL
 - Looked like we were expecting DTLS stack to read the TCP MSL
- Intention here is to be referencing the TCP spec
 - So we can benefit if new research/new net conditions
- Will fix in next version

Record Sequence Numbers for Retransmitted Hello Messages

- What happens if a hello message is lost?
 - Client retransmits
- What should the sequence numbers be?
 - Clearly: client should be next sequence
 - Proposal: server echo client [Tuexen]
- Objections?

CCS position hard to determine

- CCS has no handshake sequence number
 - Hard to determine expected position vis-a-vis other messages
 - E.g., CSS arrives, are you expecting a CertificateVerify?
- Processing CSS properly requires knowing handshake state machine
 - But which messages are expected is still deterministic
- Proposal: leave as-is but add note about it

Plan

- Make changes above
- New version by end of April
- IESG approval
- ???
- Profit