QUIC Discard Handshake Keys

IETF 106 – Singapore – 2019-11

David Schinazi, Eric Rescorla, Ian Swett, Jana Iyengar,
Kazuho Oku, Ryan Hamilton, Marten Seemann,
Martin Thomson, Christian Huitema, Martin Duke

We thought we were done, but...

If the wrong ACK gets lost, the client can end up retransmitting its ClientFinished forever because the server has discarded keys

- in some scenarios this can exhaust the congestion window
- in the worst case the connection can get deadlocked

If the client migrates before the server discards handshake keys, the server can end up having to send long headers post migration

Issue by Marten Seemann: https://github.com/quicwg/base-drafts/issues/2863

Analysis by Eric Rescorla: https://github.com/ekr/key-discard-analysis/blob/master/key_discard.pdf

We have a solution (for real this time!)

PR by Marten Seemann: https://github.com/quicwg/base-drafts/pull/3145

On the server, as soon as ClientFinished is received:

- discard handshake keys, confirm handshake
- send new HANDSHAKE_DONE frame
- keep retransmitting this frame until it is acknowledged

On the client, as soon as HANDSHAKE_DONE is received:

- discard handshake keys, confirm handshake, can migrate

If the client receives an ACK of a 1-RTT packet it has sent,
it can confirm the handshake and discard handshake keys

This solution works (for real this time!)

This no longer has a dependency on application data or on recovery draft

Client will always stop sending ClientFinished after finite time

Server is guaranteed to drop handshake keys before the client can migrate

QUIC Discard Handshake Keys

IETF 106 – Singapore – 2019-11

David Schinazi, Eric Rescorla, Ian Swett, Jana Iyengar,
Kazuho Oku, Ryan Hamilton, Marten Seemann,
Martin Thomson, Christian Huitema, Martin Duke