When Can Keys Be Destroyed? (#1544)

- **Initial (CRYPTO(ClientHello))**
  - +0-RTT (STREAM)

- **...**
  - +0-RTT (STREAM)
  - +Handshake (CRYPTO(..., Finished))
  - +1-RTT (STREAM)

- **Initial (ACK)**
  - +Handshake (ACK, CRYPTO(..., Finished))
  - +1-RTT (ACK, STREAM)

- **Handshake (ACK)**
  - +1-RTT (ACK, STREAM)

- **0-RTT (STREAM)**
- **1-RTT (STREAM)**
Simple Solution: Timers

Treat each packet number space separately

A space is done when both read and write keys for the next space are ready

Set a timer when done and destroy the keys when it expires

... until then, resend CRYPTO and send ACK as normal

... afterwards, drop packets protected with those keys

The timer can be long-ish (no practical harm in infinite)
Separate Packet Number Spaces

C

Initial (CRYPTO(ClientHello))

Initial (ACK, CRYPTO(ServerHello))

Initial (ACK)

Handshake (CRYPTO(..., Finished))

Handshake (ACK, CRYPTO(..., Finished))

Handshake (ACK)

Handshake Timer Start

S

Initial Timer Start

Handshake Timer Start
Optimization: Implicit Acknowledgment

Receiving Handshake packets implies that all CRYPTO frames from Initial packets were received.

Receiving 1-RTT packets at a server means that all CRYPTO frames in Handshake packets were received by the client.

Receiving acknowledgments for 1-RTT packets at a client means that all CRYPTO frames in Handshake and 0-RTT packets were received by a server.

Stop sending those CRYPTO frames then.

Let the packets with ACK frames that appear afterwards drop.
Alternative: HANDSHAKE_DONE Frame

An explicit signal that an endpoint believes that the handshake is done

On receipt endpoints could destroy all handshake keys

Doesn’t address 0-RTT receive keys at the server
Proposal: Document Timer-based Cleanup

Optimizations are fun, but they don’t need to be standard