Part V
Zero Round-Trip Time (0RTT)

8th BIU Winter School on Key Exchange, 2018

Marc Fischlin
Zero Round-Trip Time

- Client delivers data without waiting for response.
- Semi-static server key sent here.
- Later, "weaker" session key from semi-static server key.
- Switch to "stronger" session key.
**0RTT in QUIC**

semi-static server key $g^s$
ephemeral key $e, g^e$

$K_1 = KDF(DH(e, g^s))$

semi-static server key $s$
ephemeral key $t, g^t$

$K_1 = KDF(DH(g^e, s))$

$K_2 = KDF(DH(e, g^t))$

$K_2 = KDF(DH(g^e, t))$
Replay Attacks on QUIC?

semi-static server key $g^s$
ephemeral key $e, g^e$

$K_1 = \text{KDF}(\text{DH}(e, g^s))$

server stores previously received keys $g^e$

(not anymore in latest version of QUIC)
Replay Problems are Inherent (even with State)

Daniel Kahn Gillmor in IETF TLS 1.3 discussions

Request 0RTT → \{data\} → Request 0RTT → \{data\} → Process data

Reject 0RTT

Confirm 0RTT → Enforce reboot → Request 0RTT → Reject 0RTT

Resend data (new session) → Process data again

Reject after state loss (for security)
DH-0RTT in TLS 1.3 draft-12 (I)

ClientHello
ClientKeyShare
{ServerConfiguration*} {ServerCertificate*} {ServerCertificateVerify*} {ServerFinished}

Switched to pre-shared-key 0RTT in later drafts

include semi-static key $g^s$ here

handshake key

channel key

handshake key

channel key
DH-0RTT in TLS 1.3 draft-12 (II)

CH \[ r_c \leftarrow \{0, 1\}^{256} \]
CKS \[ g^x \]
ClientEarlyData

semi-static server key \( g^s \)

\[ \text{early_secret} \leftarrow \text{HKDF}(g^{sx}, \text{trans}) \]

\{ClientCertificate0\}
\{ClientCertificateVerify0\}
\{ClientFinished0\}

\{data\}

run full handshake from ServerHello onwards
0RTT-PSK since draft-19

ClientHello
ClientKeyShare*
early_data
psk_key_exchange_modes
pre_shared_key

{data}*

ServerHello
ServerKeyShare*
pre_shared_key
{EncryptedExtensions}
{Finished}

no forward secrecy: if PSK is comprised, data is available

corresponds to resumption mode (except for data)

only Pre-Shared-Key variant from draft 19 on
0.5RTT: Server Early-Data

ClientHello
ClientKeyShare* early_data
psk_key_exchange_modes
pre_shared_key
{data}*

ServerHello
ServerKeyShare* pre_shared_key
{EncryptedExtensions}
{Finished}
{data}*

PSK
derandawsake_key

"0.5 RTT": under key which is authenticated only later

channel key
Relax security model to replayable stages:

three (or more)
honest sessions can have
same sid in replayable stages

Then, handshake key in PSK-only mode is replayable
and all keys are non-forward secret
(under cryptographic assumptions)
Teaser for the Break

Can you give a (stateful) 0RTT-protocol which is forward secret?

knows server's long-term public key

holds state (e.g., secret key)

secure

CORRUPT

insecure?