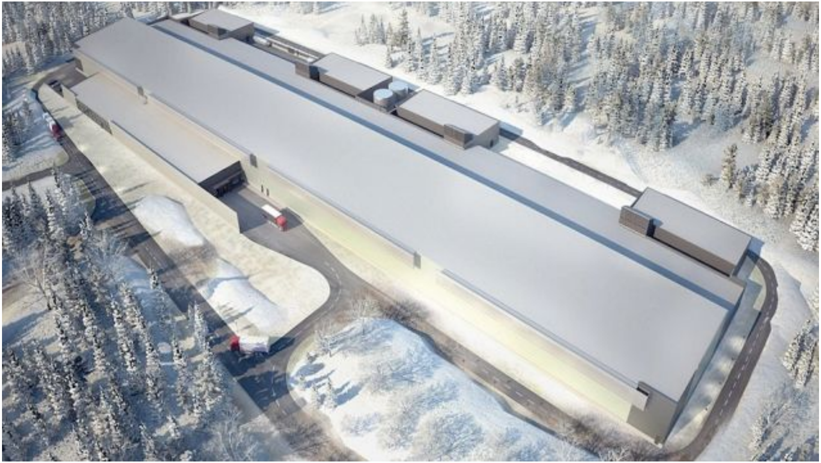
A Perfect Memory: Key Compromise in an Efficiency-centric World

Britta Hale

NTNU, Norwegian University of Science and Technology

A Perfect Memory....

Facebook Google

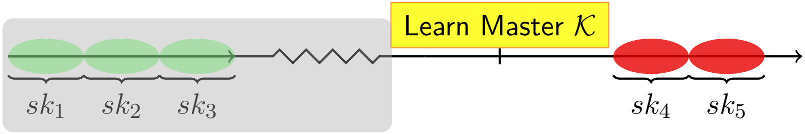


Luleå, Sweden

Threat Landscape:

Always present adversary

Long-term adversary

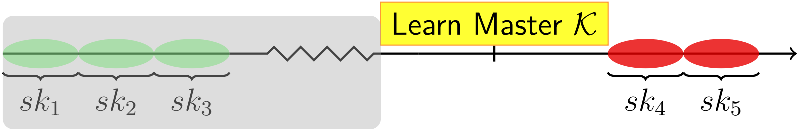


Are past session keys secure?

Perfect Forward Secrecy:

Long-term key compromised

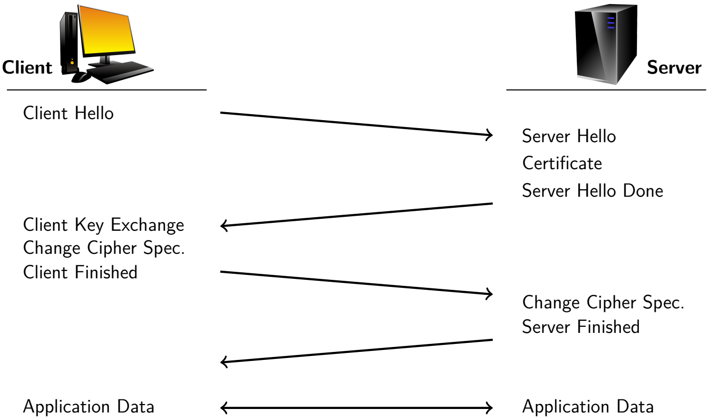
Past session keys remain secure



*Günther, C. G. Eurocrypt '89

FORWARD SECRECY IN PRACTICE

- **TLS... ?**
  - DHE-RSA / ECDHE-RSA / ...
  - TLS 1.2 vs. TLS 1.3
  - TLS 1.3 0-RTT ***... What?***



Client

Server

Client Hello

Server Hello

Certificate

Server Hello Done

Client Key Exchange

Change Cipher Spec.

Client Finished

Change Cipher Spec.

Server Finished

Application Data

Application Data

Simplified TLS Handshake Protocol

The story of low-latency / 0-RTT protocols...

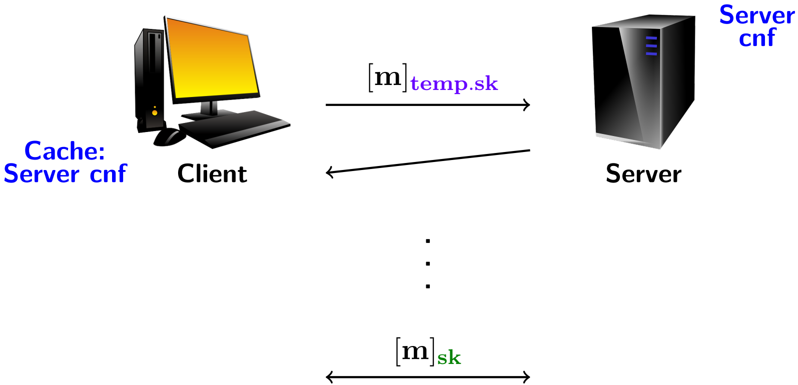
Data is sent encrypted ***immediately***

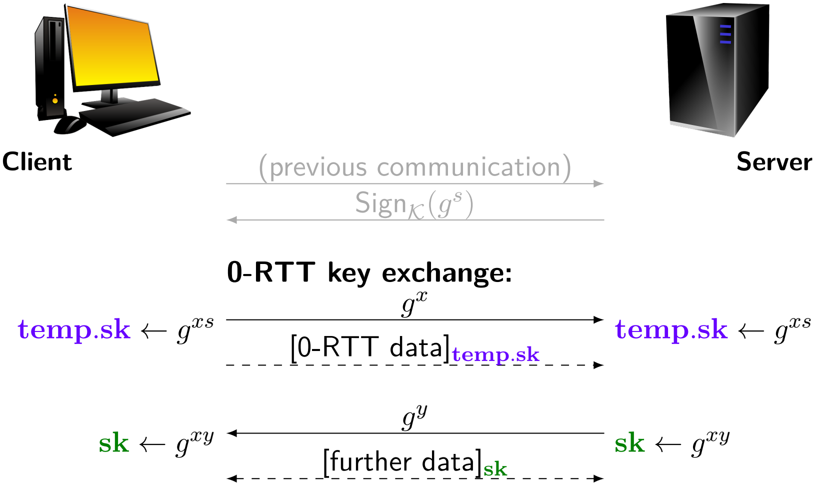
- **QUIC** by ...

Google

(Quick UDP Internet Connections)

LOW-LATENCY KEY EXCHANGE





Client

Server

(previous communication)

$\mathrm{Sign}_{\mathcal{K}}(g^s)$

0-RTT key exchange:

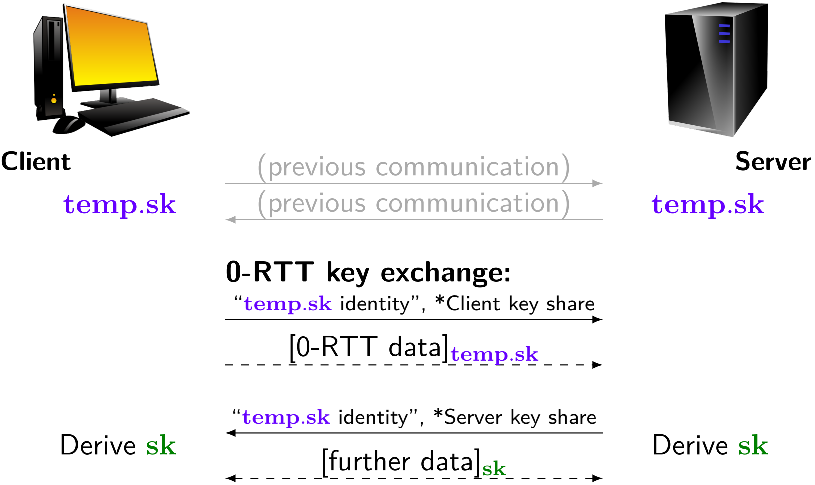
$\textbf{temp.sk} \leftarrow g^{xs}$ $\xrightarrow{g^x}$ $\textbf{temp.sk} \leftarrow g^{xs}$

$[\text{0-RTT data}]_{\textbf{temp.sk}}$

$\textbf{sk} \leftarrow g^{xy}$ $\xleftarrow{g^y}$ $\textbf{sk} \leftarrow g^{xy}$

$[\text{further data}]_{\textbf{sk}}$

- **QUIC**
  - Presented in 2013
  - Encrypted data can be sent in the first flow
  - **To be replaced by TLS 1.3**
- **TLS 1.3 draft (version 18): 0-RTT variant**
  - based on a pre-shared key
  - new forward secrecy concerns



Client

Server

temp.sk

(previous communication)

(previous communication)

temp.sk

0-RTT key exchange:

"temp.sk identity", *Client key share

[0-RTT data]$_{\text{temp.sk}}$

"temp.sk identity", *Server key share

Derive sk

[further data]$_{\text{sk}}$

Derive sk

"This data is not forward secret, as it is encrypted solely under keys derived using the offered PSK." – TLS 1.3 Draft

0-RTT folklore

For 0-RTT, there is an **"upper bound on the forward security of the connection"**

– QUIC Crypto Specification

Forward secrecy **"can't be done in 0-RTT"**

– TLS 1.3 mailing list

0-RTT Key Exchange with Full Forward Secrecy

Felix Günther[1] Britta Hale[2] Tibor Jager[3] Sebastian Lauer[3]

[1] TU Darmstadt [2] NTNU, Trondheim [3] Ruhr-University Bochum

- Server has public/secret key pair (PK, SK), where SK is updated
- Puncturable FS Key Encapsulation Mechanism (PFS-KEM)
- Built from a HIBKEM and One-Time Signatures

FINAL COMMENTS

- **Forward secrecy is a serious problem**
  in a world with indefinitely stored data

- **0-RTT encrypted data is a growing demand**:
  traffic increase, IoT, ...

- **Current 0-RTT solutions do not address forward secrecy,**
  or have simply changed the context

- **Forward secrecy *is* possible for 0-RTT data,
  despite all previous claims**

Questions