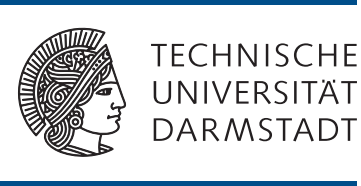
A Cryptographic Analysis of the TLS 1.3 Handshake Protocol Candidates

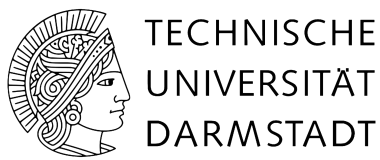
The main modes, 0-RTT, and replays



Felix Günther

Technische Universität Darmstadt, Germany

joint work with Benjamin Dowling, Marc Fischlin, and Douglas Stebila

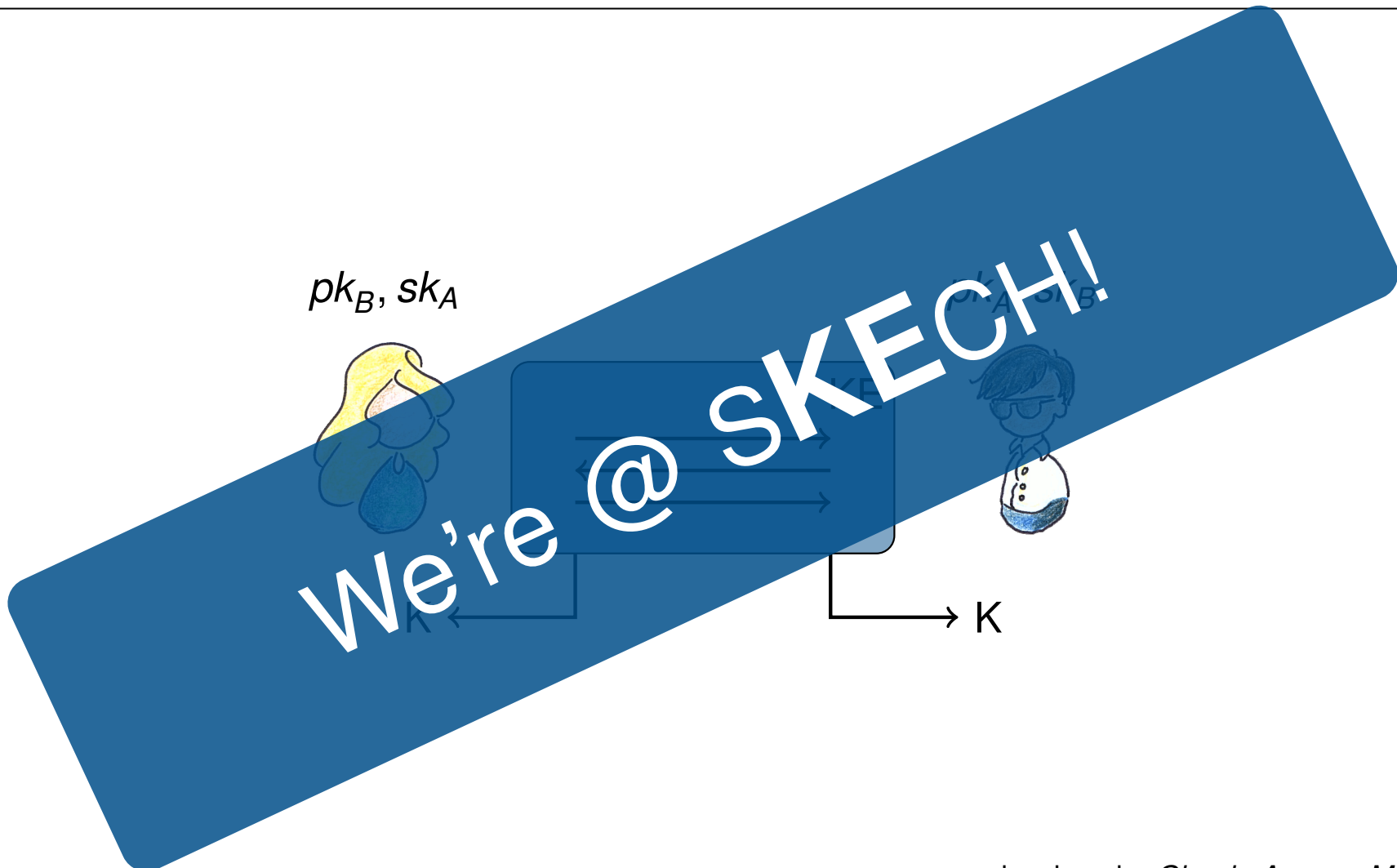


Queensland University of Technology

Australian Government
Australian Research Council

Key Exchange

TECHNISCHE UNIVERSITÄT DARMSTADT



drawings by *Giorgia Azzurra Marson*

TLS History

TECHNISCHE UNIVERSITÄT DARMSTADT

The [TLS] prot *lications*
to commu
prevent ea ge forgery.
LS 1.2 [RFC 5246]

70% of global Inter
expected to be encryp
(Sandvine: Internet Traffic Encryptio

	Monday	Tuesday	Wednesday
07:30–08:45	Breakfast	Breakfast	Breakfast
09:00–09:40	Cedric Fournet *Modelling TLS 1.3 in F**	Hugo Krawczyk *Client Authentication in TLS*	Karthik Bhargavan *Secure messaging protocols: Telegram, Signal, & CryptoCat*
09:40–10:20	Britta Hale *Secure Channels and Termination: The Last Word on TLS*	Hugo Krawczyk *Recent Results on PAKE*	Cas Cremers *TBA*
10:20–10:50	Coffee Break	Coffee break	Coffee break
10:50–11:30	Giorgia Azzurra Marson *Security Notions for Stream-based Channels*	Sasha Boldyreva *Human Computing for Handling Strong Corruptions in Authenticated Key Exchange*	
11:30–12:10	Douglas Stebila *On Writing Key Exchange Models*	Vlad Kolesnikov *Attribute-based Key Exchange with General Policies*	
12:30–13:30	Lunch	Lunch	Lunch
14:30–15:10	Workshops (Part I)	Felix Günther *A Cryptographic Analysis of the TLS 1.3 Handshake Protocol Candidates*	
15:10–15:50		Tibor Jager *0-RTT Key Establishment with Full Forward Secrecy*	
15:50–16:30		Ben Dowling *TLS 1.3 and the miTLS Crypto Model*	Coffee Break
16:30–17:00	Coffee Break	Coffee Break	
17:00–18:20	Workshops (Part II)		Panel discussion
20:00	Dinner	Dinner	Dinner

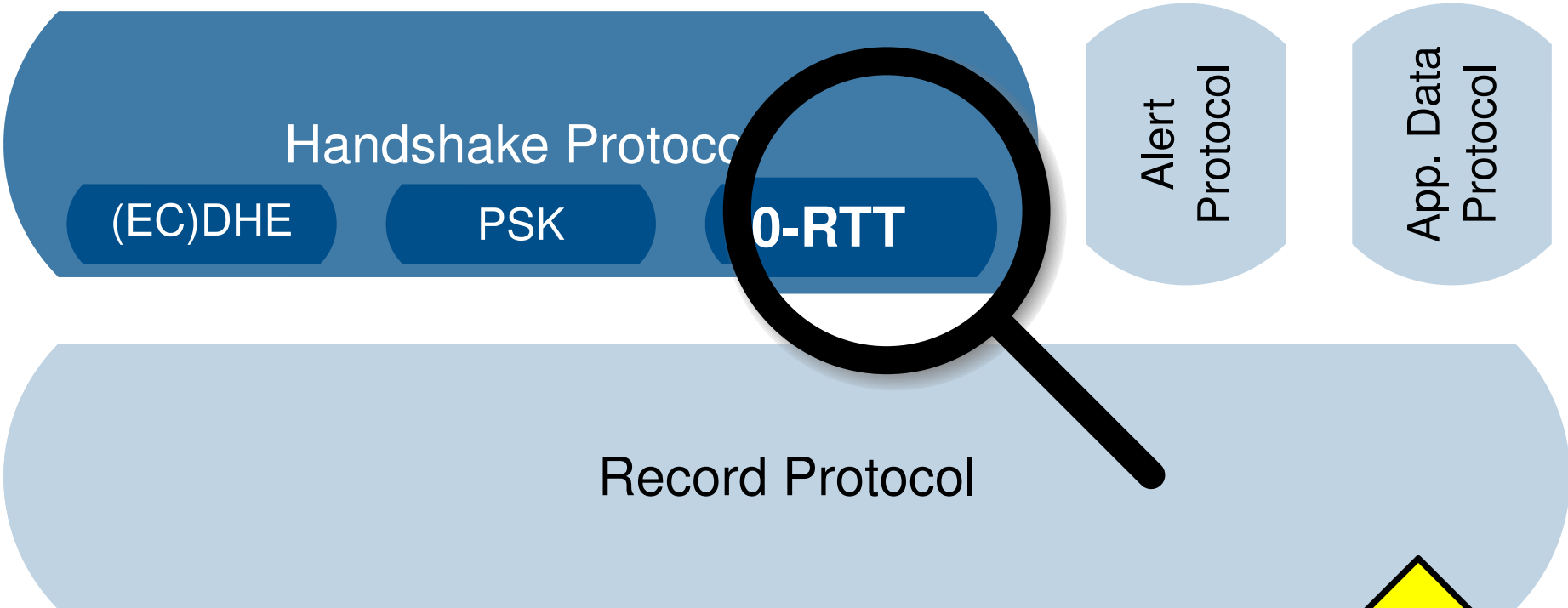
TLS 1.3

TECHNISCHE UNIVERSITÄT DARMSTADT

- next TLS version, currently being specified (latest: `draft-13`, May 2016)
- several substantial cryptographic changes (compared to TLS 1.2), incl.
  1. encrypting some handshake messages with intermediate session key
  2. signing the entire transcript when authenticating
  3. including handshake message hashes in key calculations
  4. generating `Finished` messages with separate key
  5. deprecating some crypto algorithms (RC4, SHA-1, key transport, MtEE, etc.)
  6. using only AEAD schemes for the record layer encryption
  7. switch to HKDF for key derivation
  8. providing reduced-latency **0-RTT handshake**

TLS Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

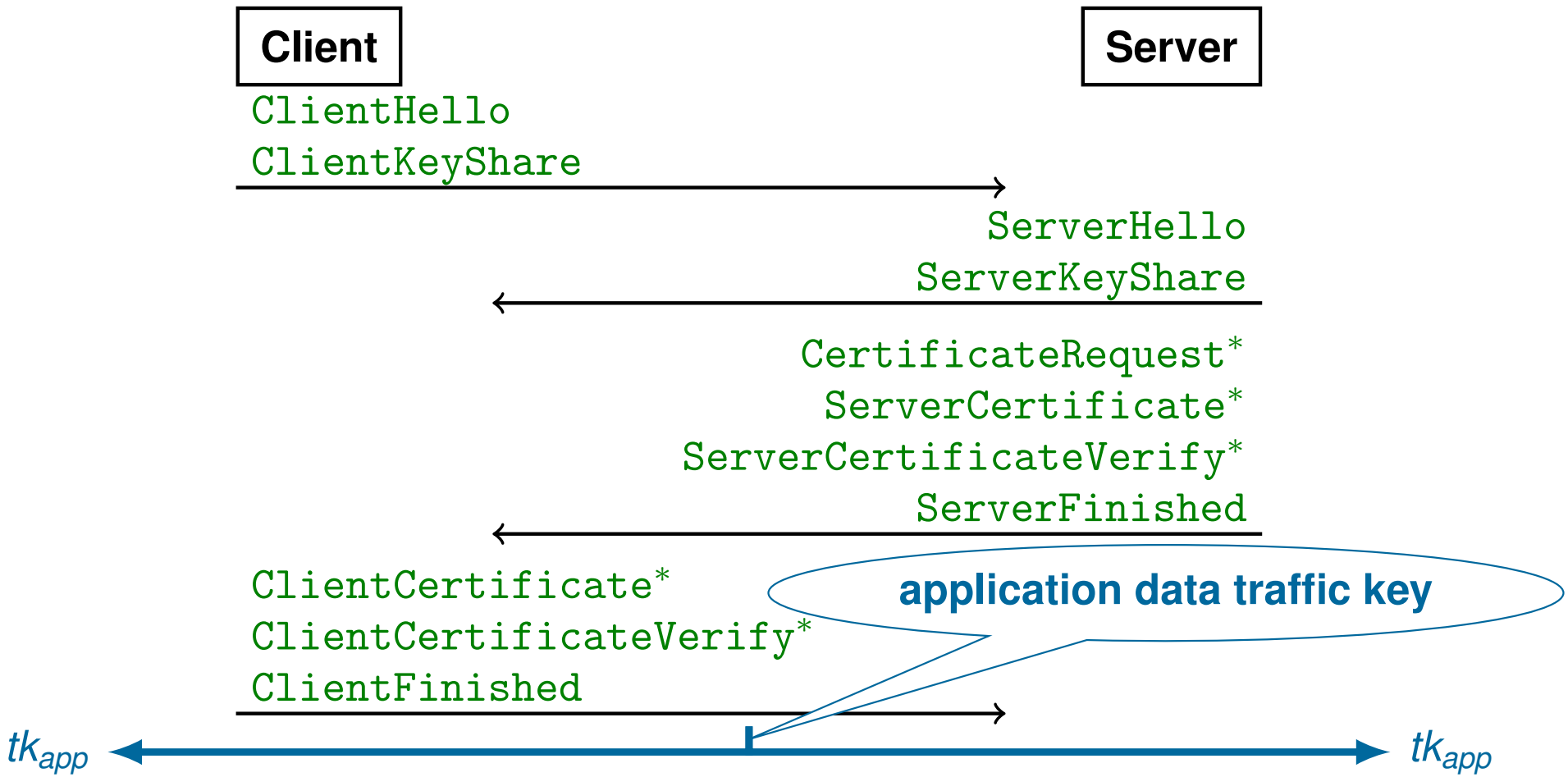


- we analyze the **handshake protocol candidates** for TLS 1.3

STANDARD UNDER CONSTRUCTION

TLS 1.3 Full/(EC)DHE Handshake
(simplified)

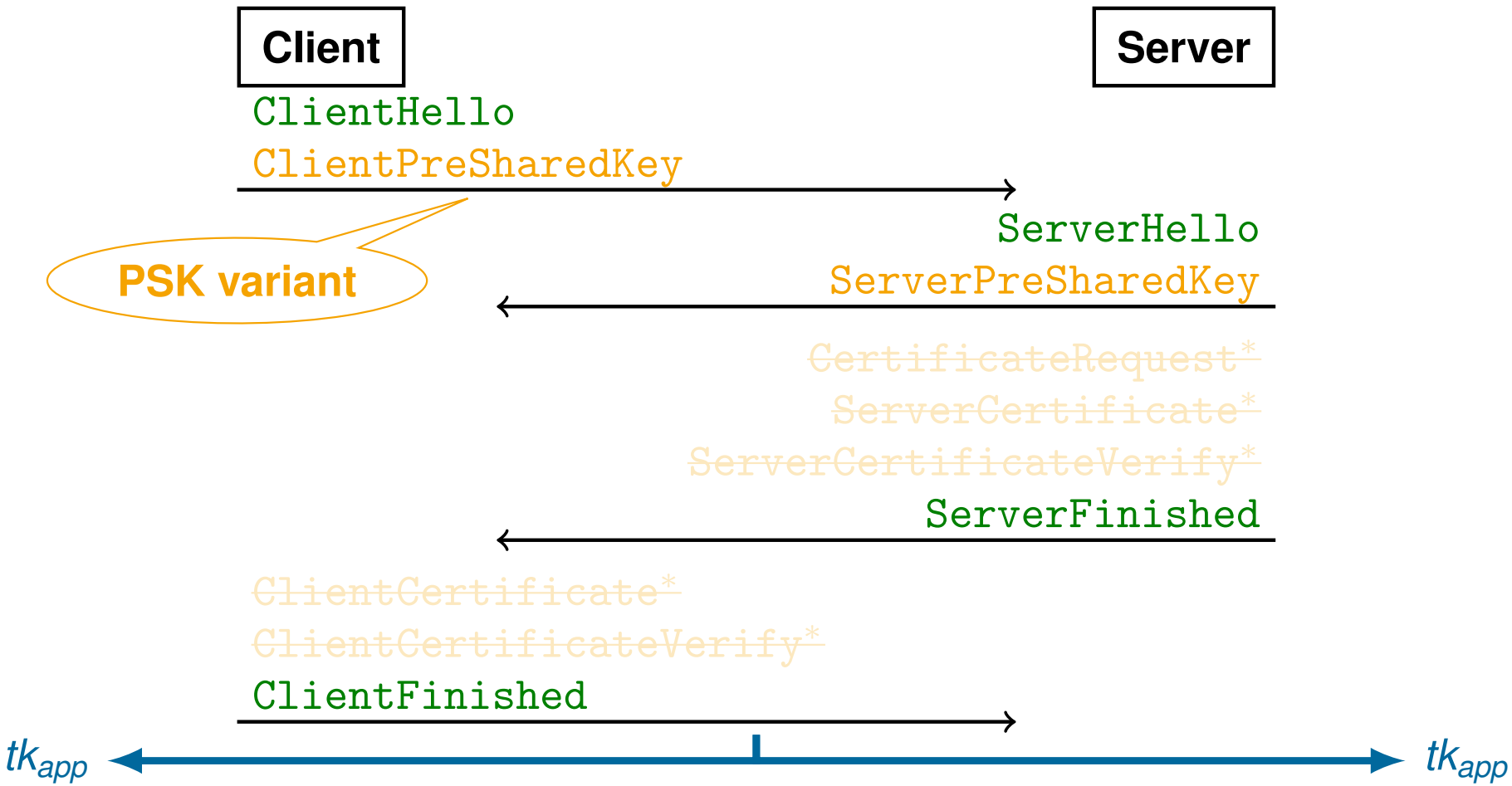
TECHNISCHE UNIVERSITÄT DARMSTADT



. . . actually, there is more . . .

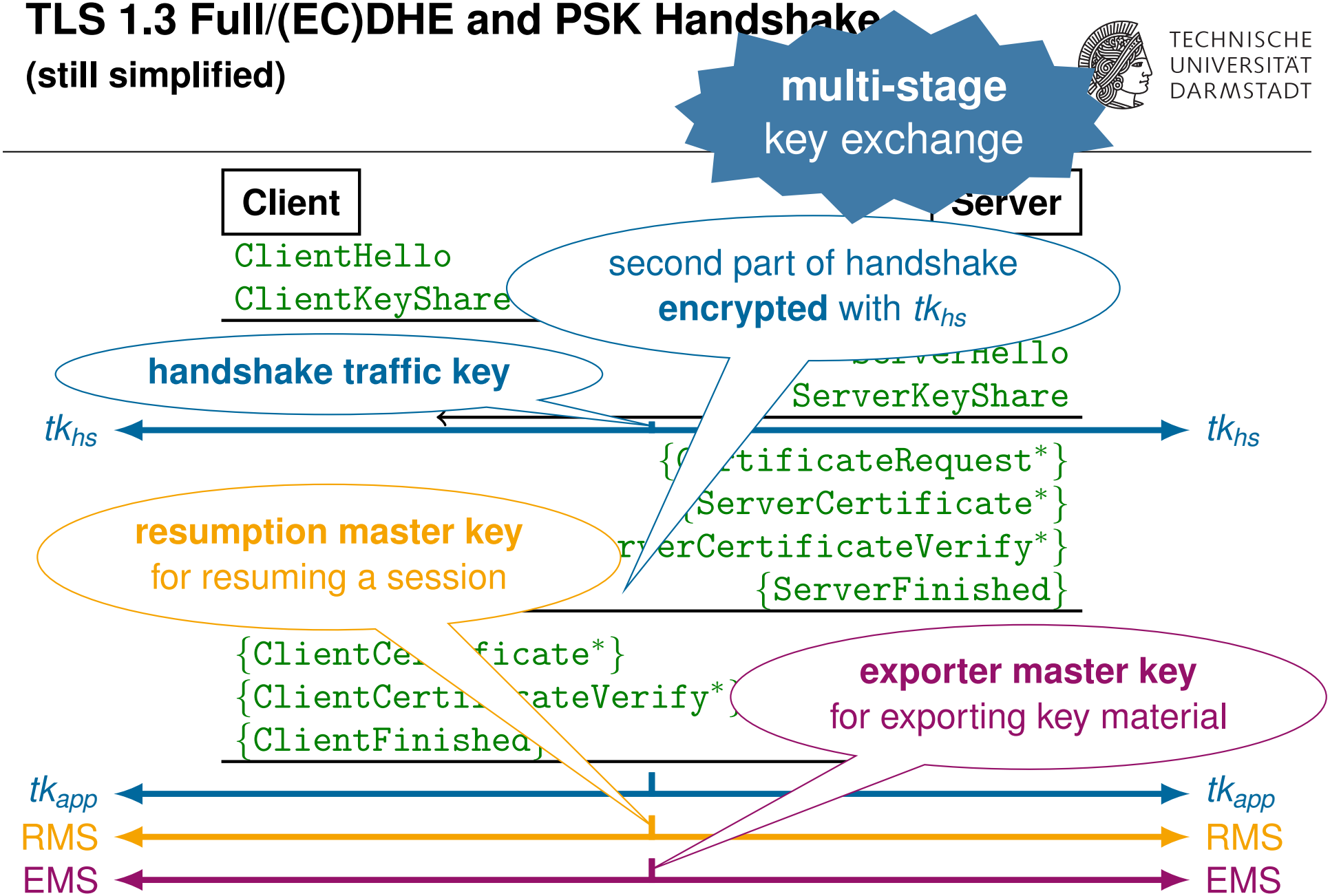
TLS 1.3 Full/(EC)DHE and PSK Handshake
(simplified)

TECHNISCHE UNIVERSITÄT DARMSTADT



TLS 1.3 Full/(EC)DHE and PSK Handshake

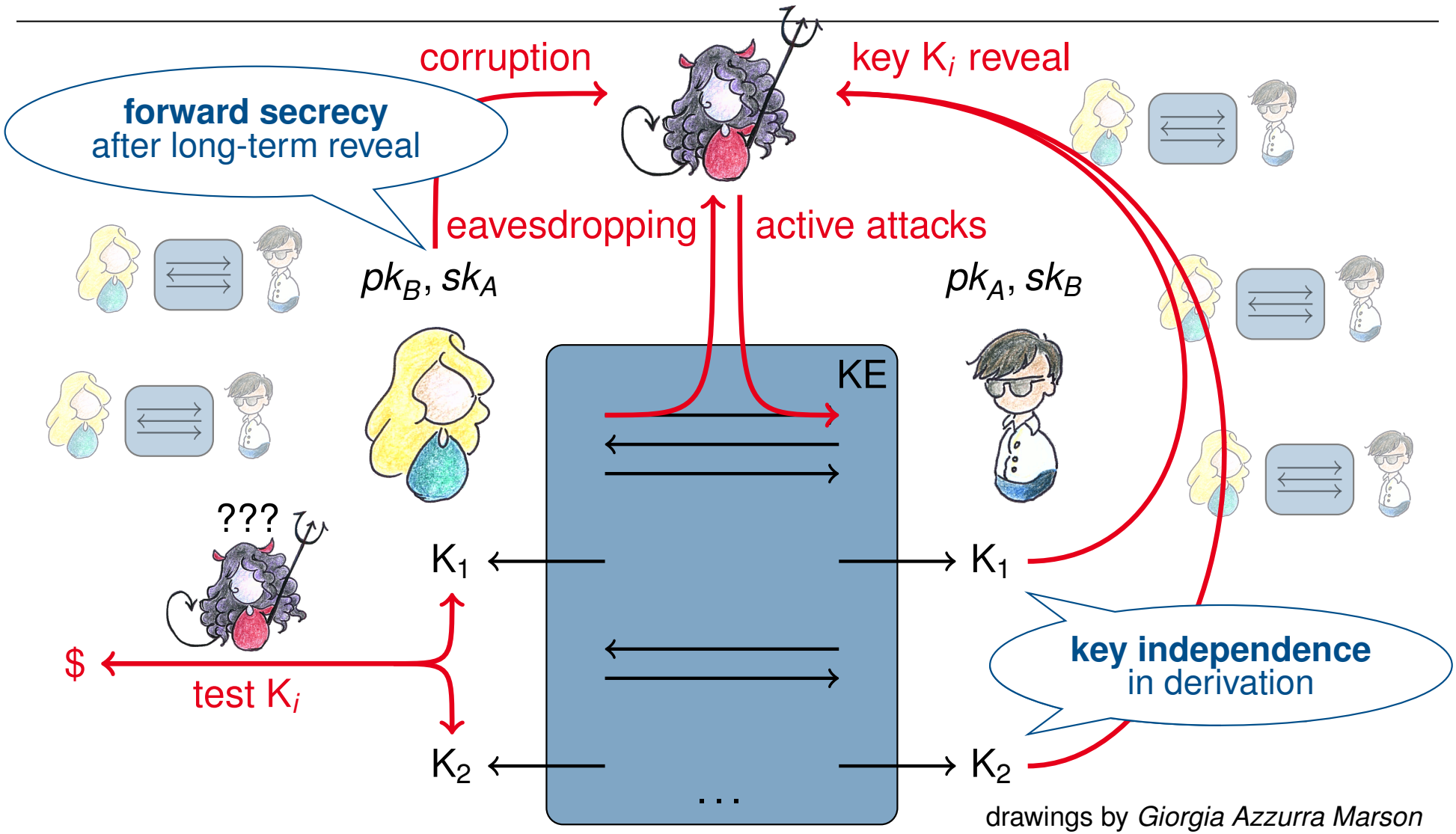
(still simplified)



Multi-Stage Key Exchange (Security)

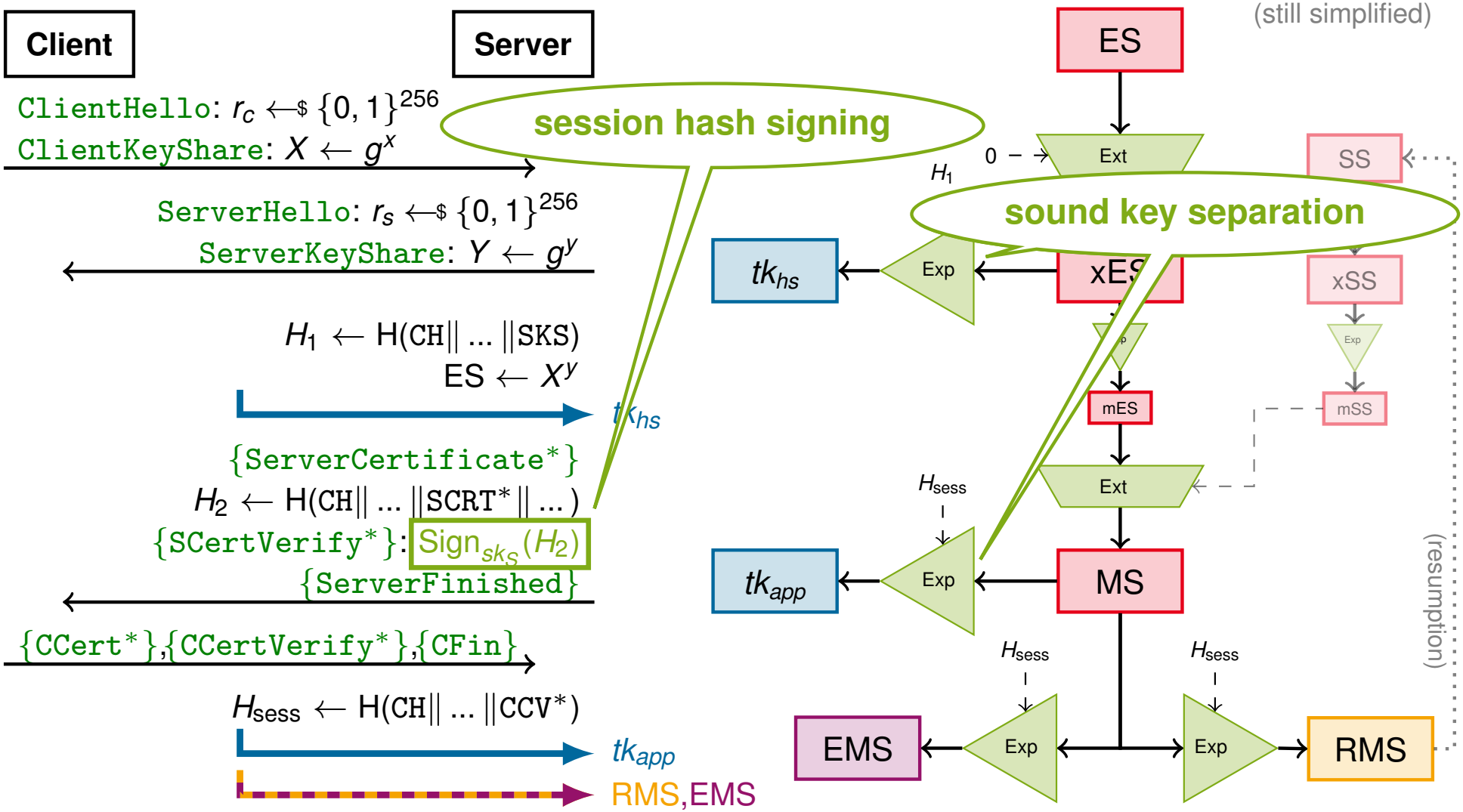
(Fischlin, Günther @ ~~CCS 2014~~ SKECH 2014)

TECHNISCHE UNIVERSITÄT DARMSTADT



Security of the `draft-10` (EC)DHE Handshake

TECHNISCHE UNIVERSITÄT DARMSTADT



Security of the `draft-10` (EC)DHE Handshake

TECHNISCHE UNIVERSITÄT DARMSTADT

We show that the `draft-10` full (EC)DHE handshake establishes

- random-looking keys (tk_{hs}, tk_{app}, RMS, EMS)
  with adversary allowed to corrupt other users and reveal other session keys
- forward secrecy for all these keys
- concurrent security of anonymous, unilateral, mutual authentication
- key independence (leakage of traffic/resumption/exporter keys in same session does not compromise each other's security)

assuming

- collision-resistant hashing
- unforgeable signatures
- HKDF is pseudorandom function
- PRF-ODH assumption holds

} **standard key exchange security** under **standard(-model) assumptions**

Security of the `draft-10` PSK Handshakes

TECHNISCHE UNIVERSITÄT DARMSTADT

PSK

- random-looking keys (tk_{hs}, tk_{app}, EMS)
- mutual authentication (down to RMS)
- key independence
- no forward secrecy

PSK-DHE

- random-looking keys (tk_{hs}, tk_{app}, EMS)
- mutual authentication (down to RMS)
- key independence
- forward secrecy for all keys

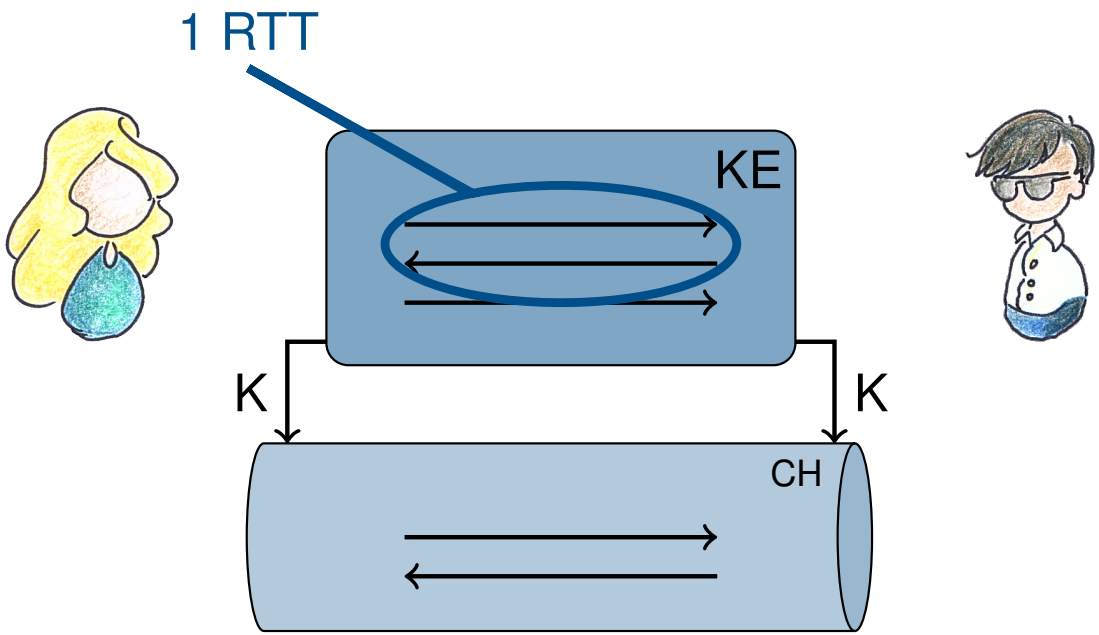
Under similar standard(-model) assumptions:

- collision-resistant hashing
- HKDF is pseudorandom function

- collision-resistant hashing
- HKDF is pseudorandom function
- HMAC is unforgeable
- PRF-ODH assumption holds

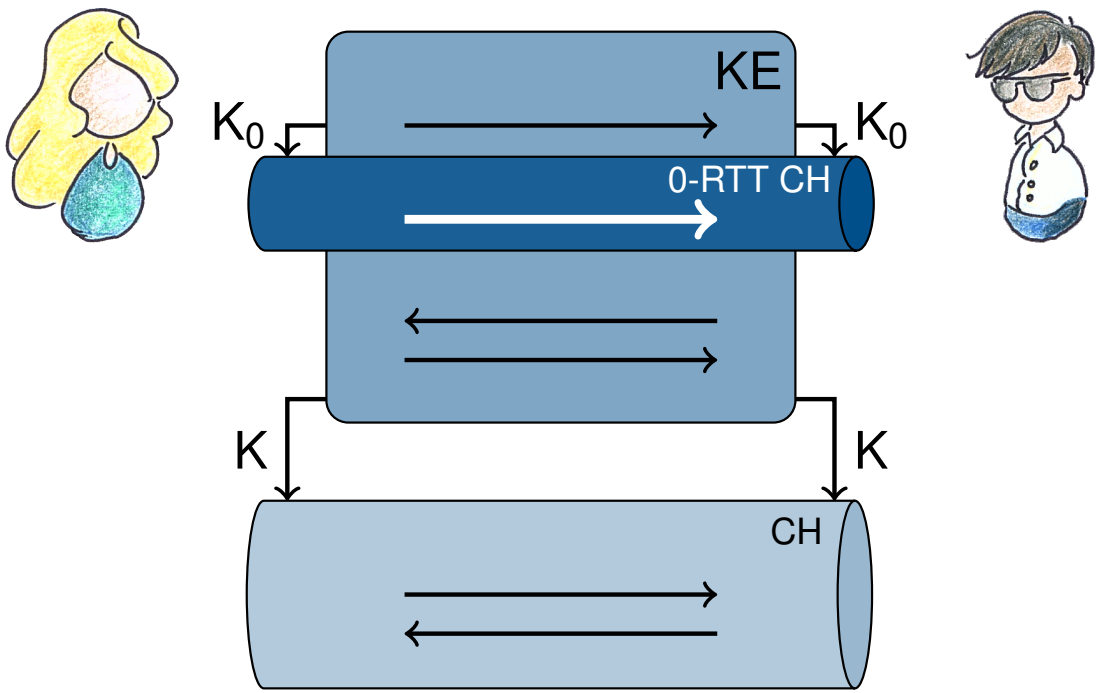
Zero Round-Trip Time (0-RTT)

TECHNISCHE UNIVERSITÄT DARMSTADT



Zero Round-Trip Time (0-RTT)

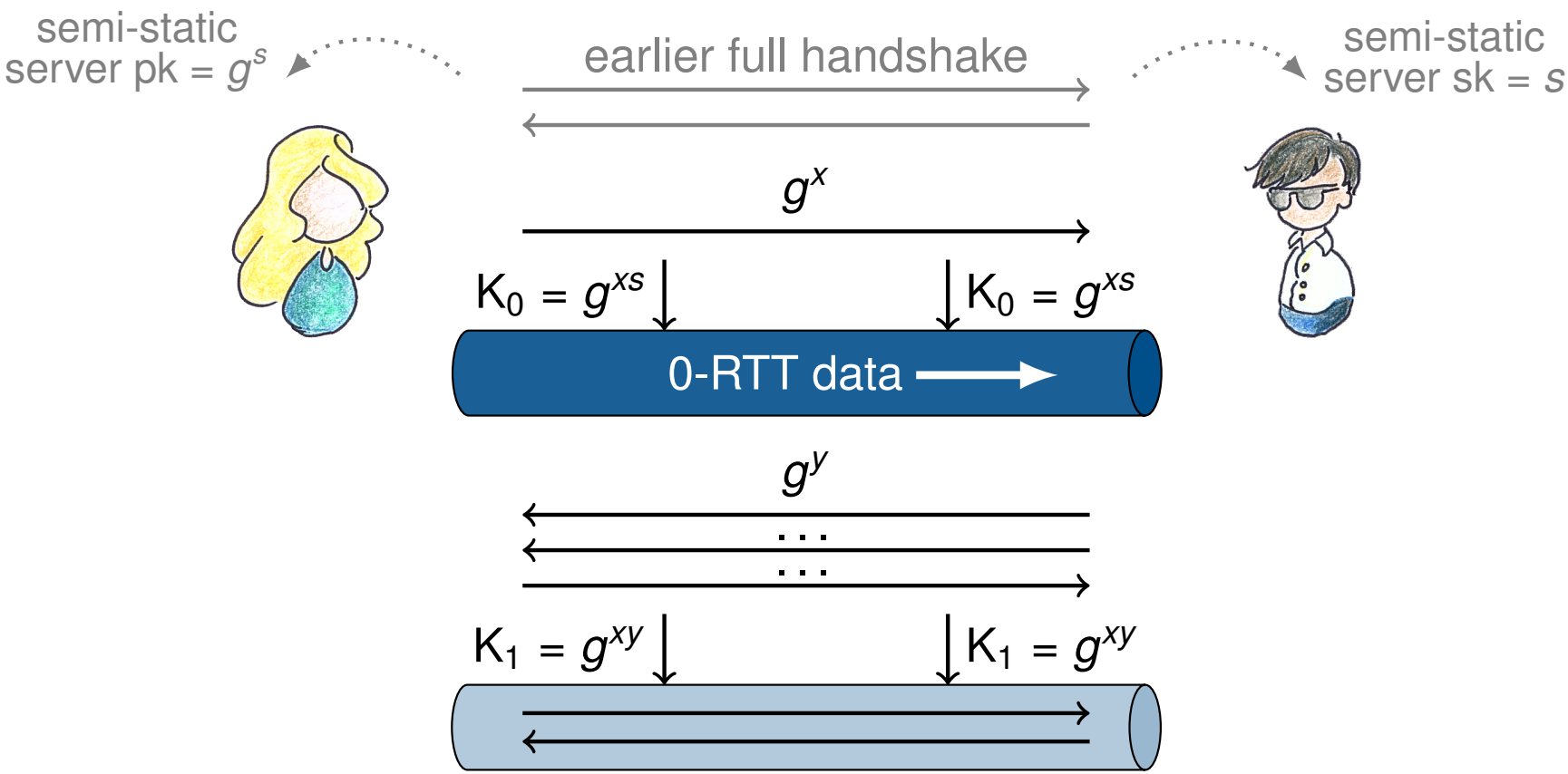
TECHNISCHE UNIVERSITÄT DARMSTADT



Diffie–Hellman-based 0-RTT

TECHNISCHE UNIVERSITÄT DARMSTADT

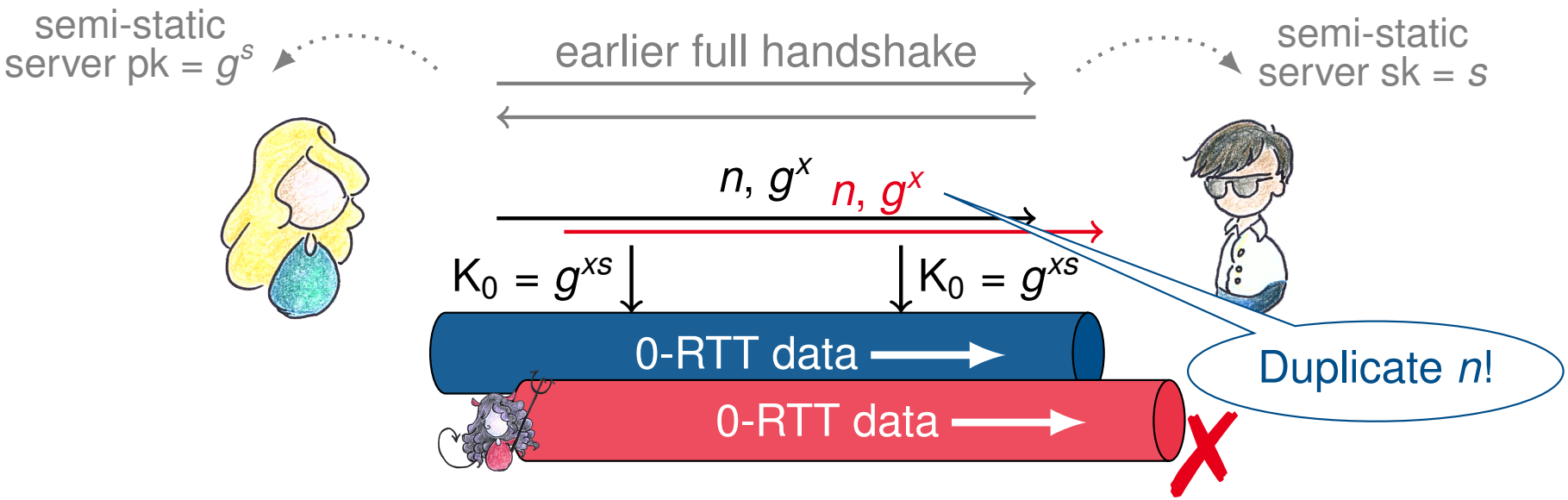
- à la QUIC, but also TLS 1.3 DH-based 0-RTT mode



Diffie–Hellman-based 0-RTT

TECHNISCHE UNIVERSITÄT DARMSTADT

- à la QUIC, but also TLS 1.3 DH-based 0-RTT mode

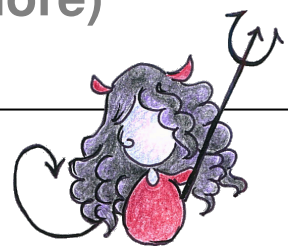


- what about **replays**?
- QUIC: remember nonces in “strike register” (restricted by “orbit”+time)
- effectively prevents same key is derived twice

Generic Replay Attack on 0-RTT

(by Daniel Kahn Gillmore)

TECHNISCHE UNIVERSITÄT DARMSTADT



0-RTT KE msg (n, g^x)

0-RTT data "/buy-something"

process "/buy-s..

accept 0-RTT, KE response (g^y)

enforce state loss (e.g., reboot)

simpler in real world:
send to two
distributed servers

replay 0-RTT KE msg (n, g^x)

replay "/buy-s..

rej. after state loss
for security reasons

reject 0-RTT, KE response msg $(g^{y'})$

complete KE

(resend) "/buy-something" under final key

process "/buy-s..
again

Generic Replay Attack on 0-RTT

What's going on?

TECHNISCHE UNIVERSITÄT DARMSTADT

- attack applies to QUIC —vs.— security proofs for QUIC [FG'14, LJBN'15]

- standard answer: "out of model"
- actually sth. beyond KE: it's conscious replay on application level

> *"This isn't that odd, since, as AGL observes, browsers already* routinely retry *some HTTP requests that appear to fail* even for ordinary TLS *[...] but* of course that's different from having TLS give up those guarantees*."*
>
> Eric Rescorla @ TLS mailing list

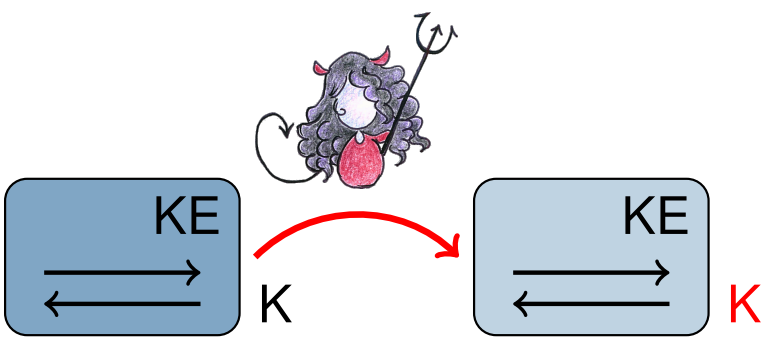
QUIC's Response

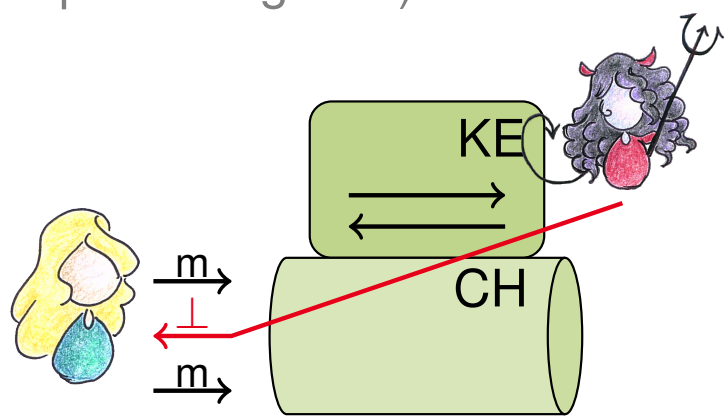
TECHNISCHE UNIVERSITÄT DARMSTADT

"The QUIC crypto protocol is destined to die*."*

Langley, Chang / QUIC Crypto, Revision 20150720

- we claim: actually provides some replay protection, just on a different level
- distinguish between replay @ KE level and replay @ application level (latter fundamentally beyond what KE can protect against)





TLS 1.3's Response

TECHNISCHE UNIVERSITÄT DARMSTADT

- can't protect against replays anyway (on application level) . . .
- . . . so give up any replay protection for 0-RTT

i.e.

- don't check for duplicate nonces, allow keys to be "replayed"
- don't retransmit automatically on 0-RTT reject, but let application decide

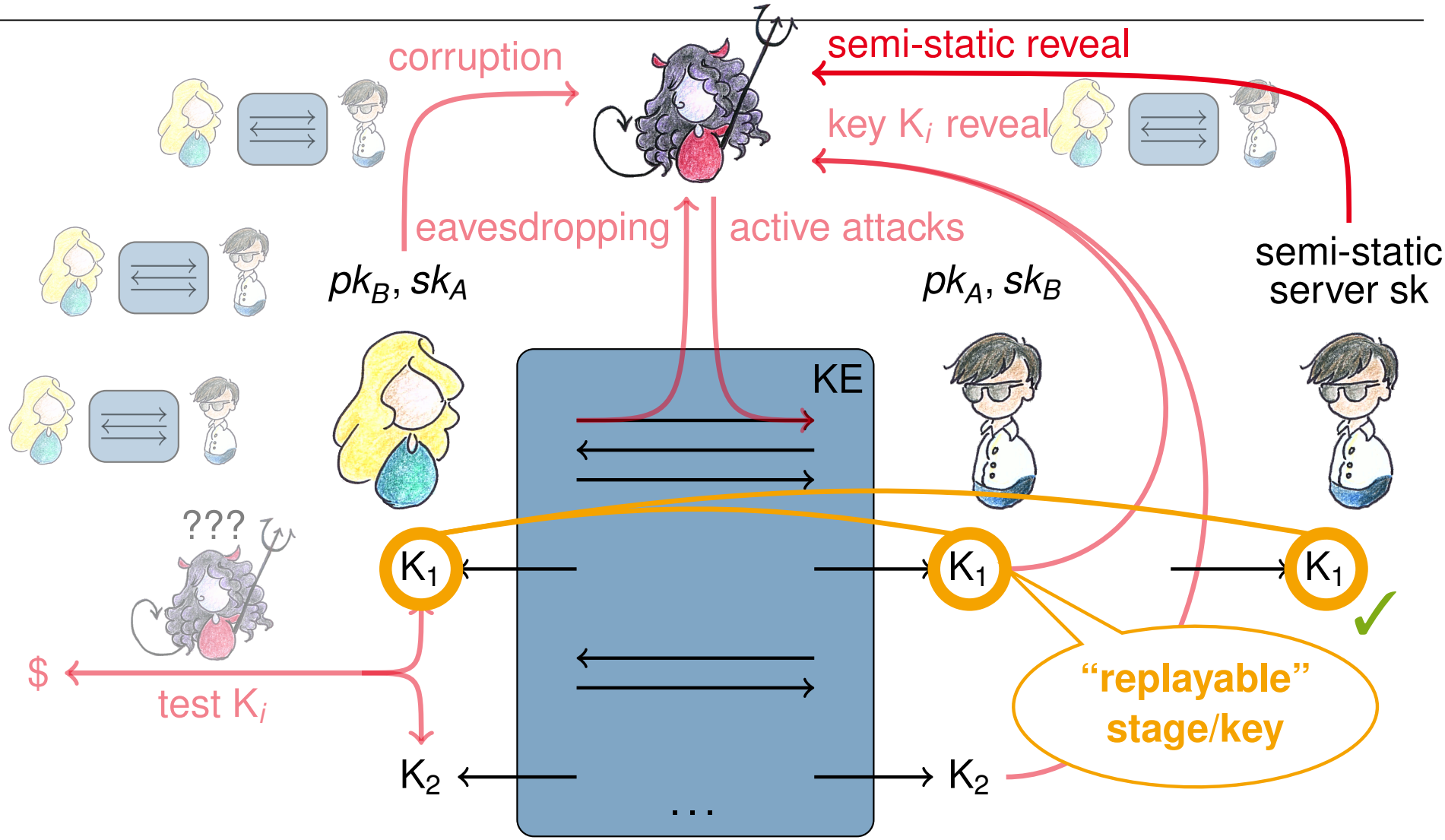
- in theory: can be okay for some requests? (HTTP GET?)
- in practice: unclear / will have to see. . .

"browsers already routinely retry some HTTP requests"

Multi-Stage Key Exchange (Security)

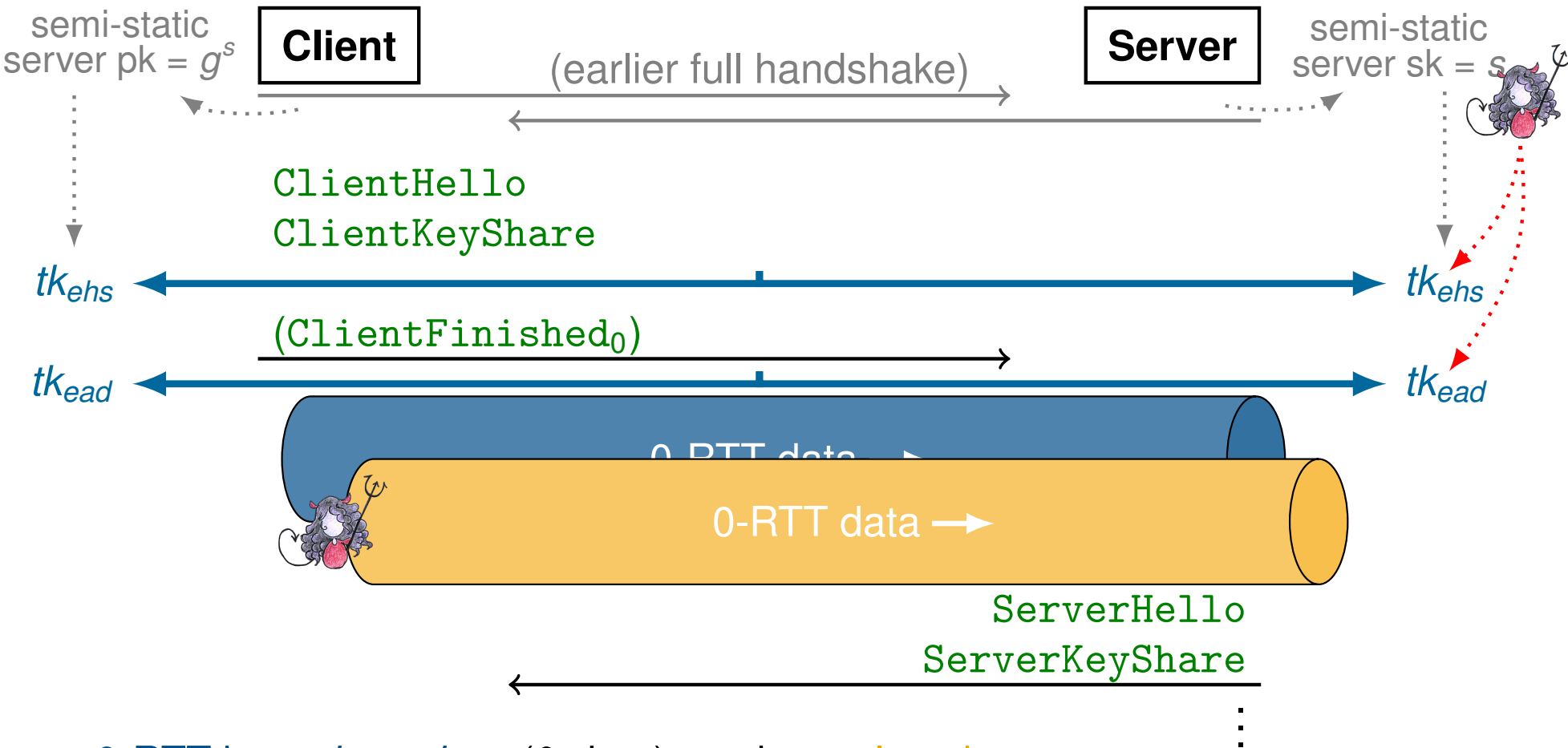
with replayable stages/keys

TECHNISCHE UNIVERSITÄT DARMSTADT



Security of the `draft-12` (EC)DHE 0-RTT Handshake

TECHNISCHE UNIVERSITÄT DARMSTADT



- 0-RTT keys tk_{ehs}, tk_{ead} (& data) can be replayed
- weaker forward secrecy guarantees

(Still) Not the End of the Story

TECHNISCHE UNIVERSITÄT DARMSTADT

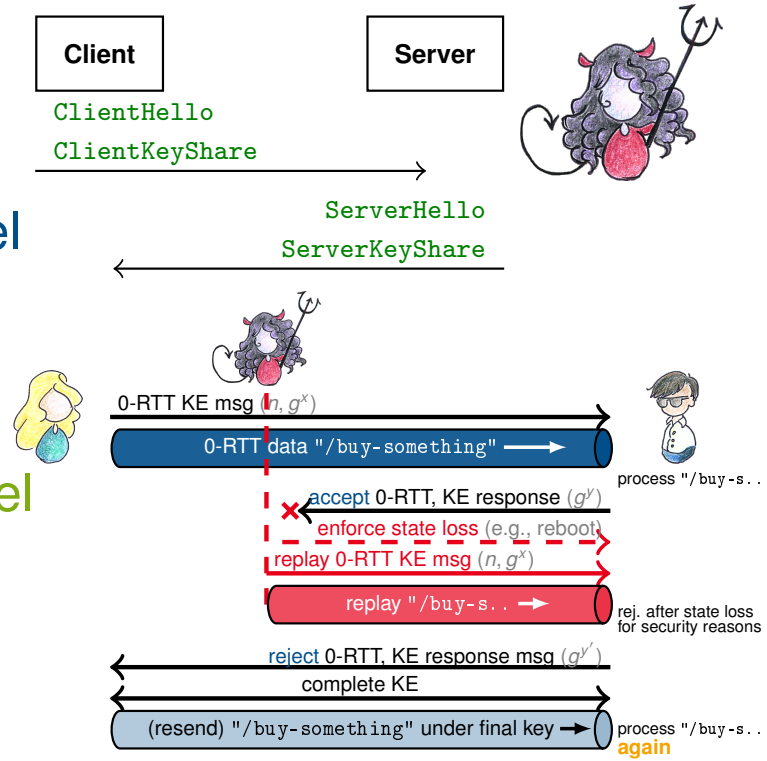
- TLS WG decided (in April) to only support PSK-based 0-RTT
- . . . but (EC)DHE-based 0-RTT might come back as extension, esp. for better forward secrecy
- our model can serve as stepping stone for understanding 0-RTT & replays
- . . . and can be applied to PSK-based 0-RTT as well
  (we currently look into that, security results appear to be similar)

Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

We

- analyze TLS 1.3 (`draft-10`)
  full (EC)DHE, PSK, and PSK-DHE handshake
  in an extended multi-stage key exchange model



- for 0-RTT, distinguish replays @ KE level
  from (unpreventable) replays @ application level

- establish TLS 1.3 (`draft-12`) (EC)DHE 0-RTT handshake is secure multi-stage KE with replayable 0-RTT keys

- are looking into TLS 1.3 PSK-based 0-RTT handshake

Thank You!