A Cryptographic Perspective on TLS 1.3

Modeling Advanced Security Aspects of Key Exchange and Secure Channel Protocols

UC San Diego

Felix Günther

STM PhD Award Talk

15th International Workshop on Security and Trust Management (STM 2019)

September 26, 2019

TECHNISCHE UNIVERSITÄT DARMSTADT

Cryptoplexity

Cryptography & Complexity Theory
Technische Universität Darmstadt
www.cryptoplexity.de

UC San Diego

Secure Connections

UC San Diego

Security goals:

- confidentiality
- authentication
- integrity

Transport Layer Security (TLS)

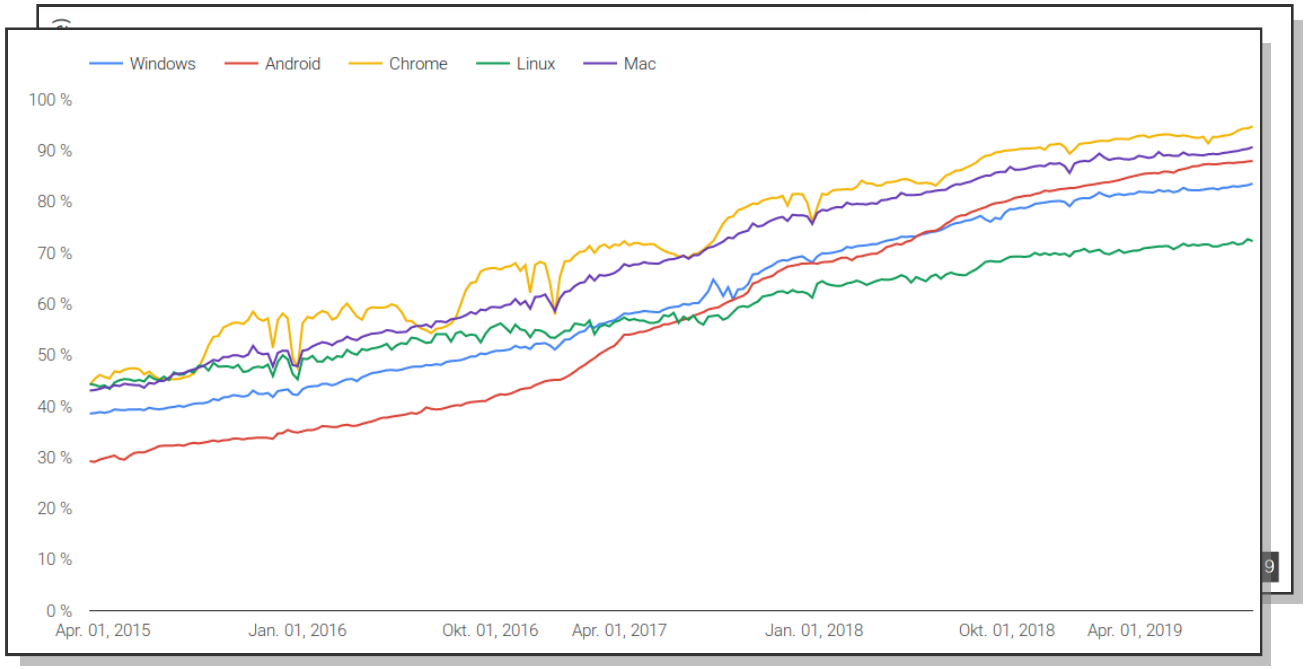
UC San Diego

TLS allows client/server applications to communicate over the Internet in a way that is designed to prevent eavesdropping, tampering, and message forgery.

TLS 1.3 [RFC 8446]

HTTPS-protected traffic

- 50%+ Sandvine (Oct 18)
- 72% Fortinet (Q3 18)
- 79% Firefox (Sep 19)
  (Telemetry)
- 90% Chrome (Sep 19)
  (Transparency Report)

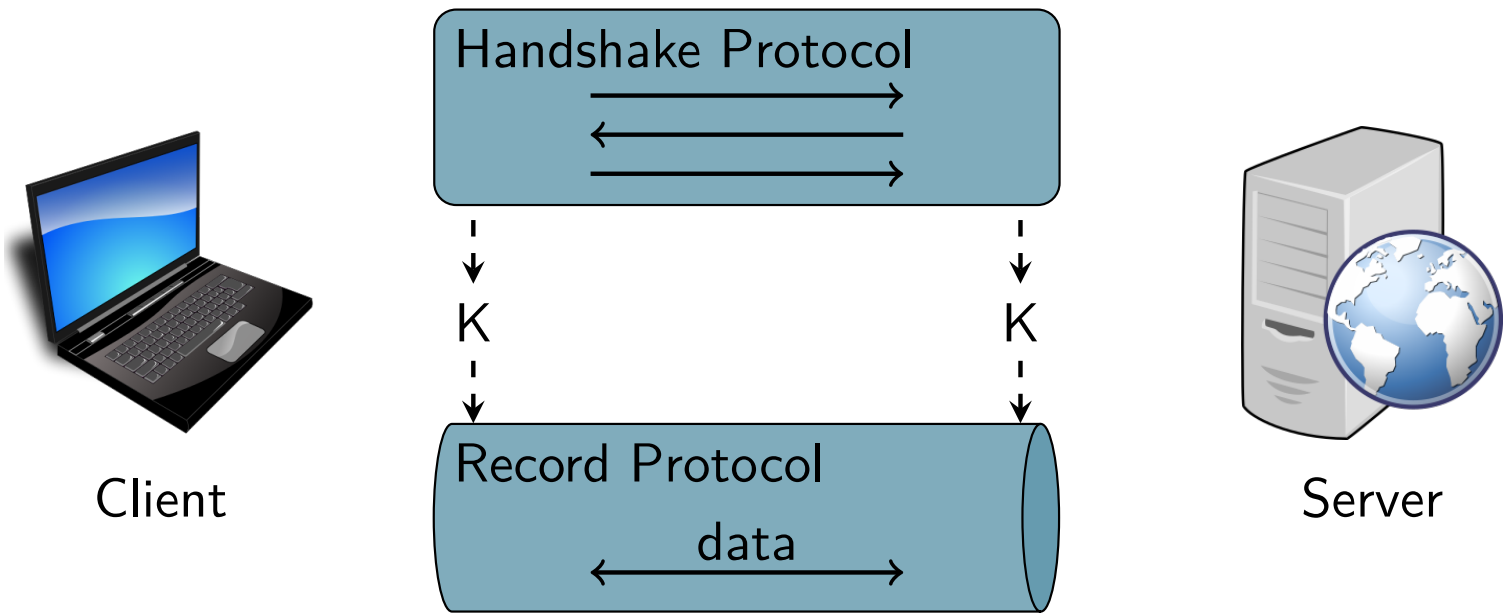


Transport Layer Security (TLS)

UC San Diego

Handshake Protocol:

- negotiate security parameters ("cipher suite")
- authenticate peers
- establish key material for data protection



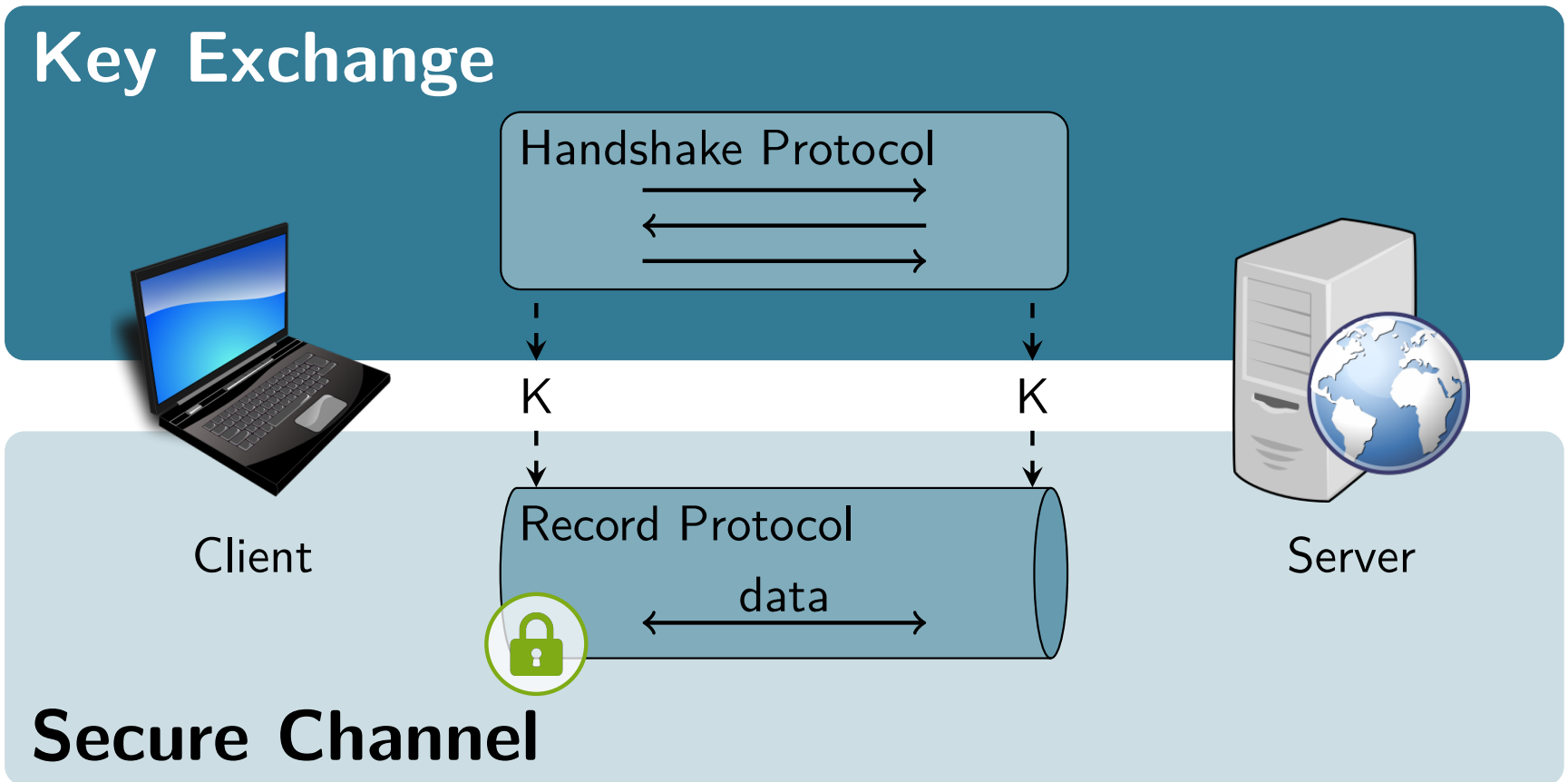
Record Protocol:

- protect data using key material from handshake
- ensuring confidentiality and integrity

Transport Layer Security (TLS)

Cryptographic Core

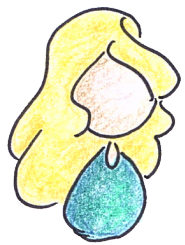
UC San Diego

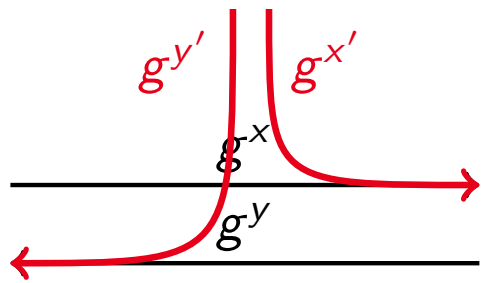


Key Exchange à la Diffie–Hellman (1976)

UC San Diego

knows x





knows y

$$g^{xy} = (g^y)^x$$
$$K = KDF(g^{xy})$$
$$K' = KDF(g^{xy'})$$

KDF: Key Derivation Function

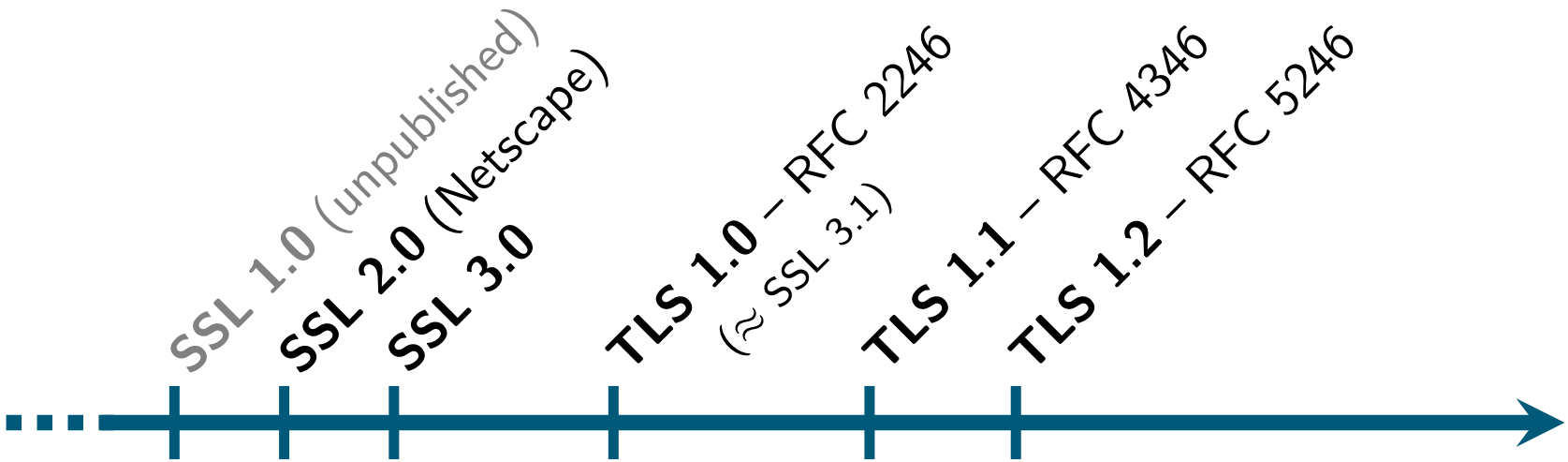
$$g^{xy} = (g^x)^y$$
$$K = KDF(g^{xy})$$
$$K'' = KDF(g^{x'y})$$

- key secrecy: given only g^x and g^y, key K remains secret
- just one building block (no security against MitM, . . .) $\Rightarrow$ need full protocol

Transport Layer Security (TLS)

The SSL/TLS history . . .

UC San Diego



today all considered seriously broken

small improvements & some new crypto

I E T F®

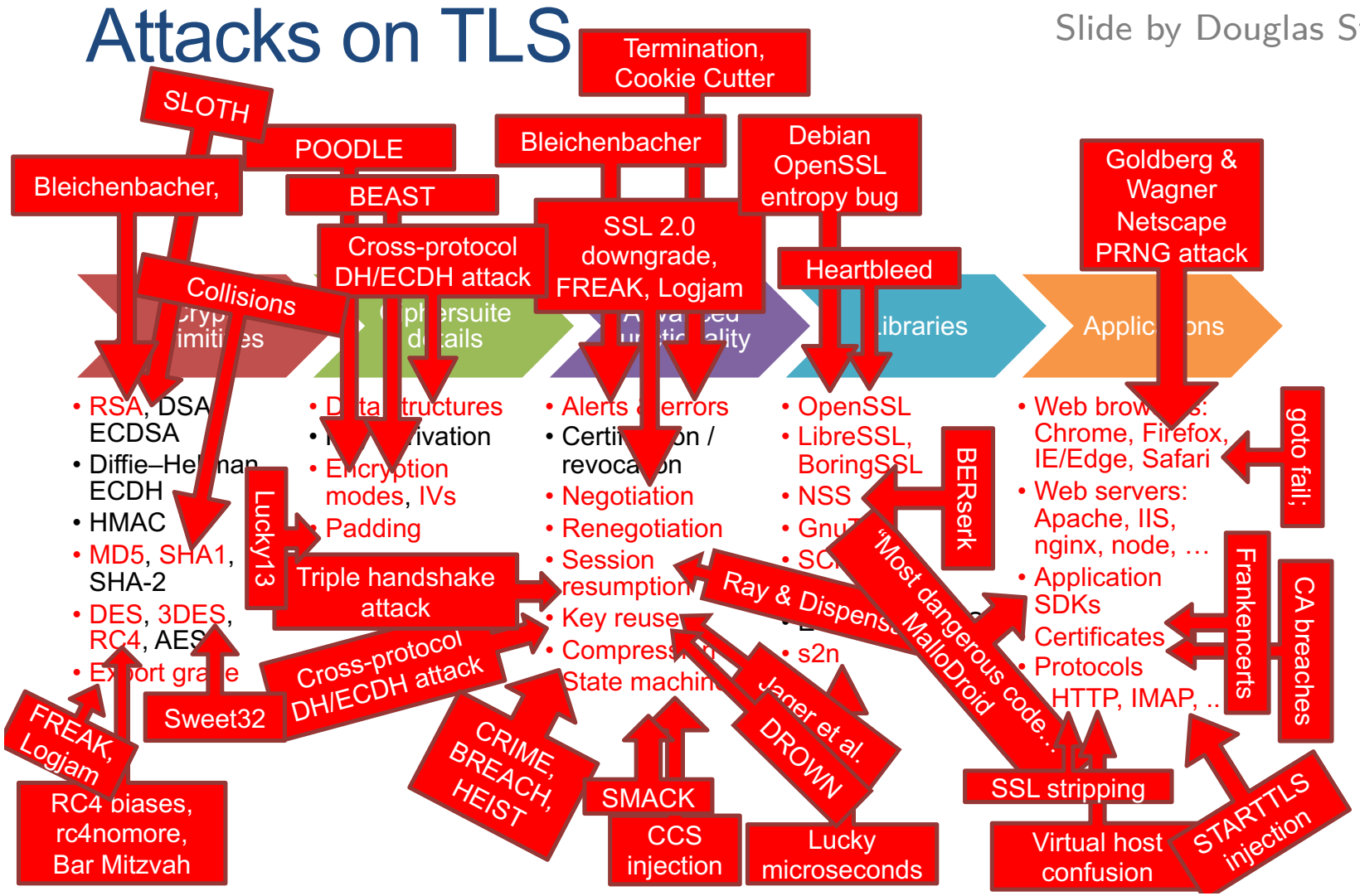
adopted by IETF TLS working group

main editor: Eric Rescorla

Transport Layer Security (TLS)

The SSL/TLS history . . . of attacks

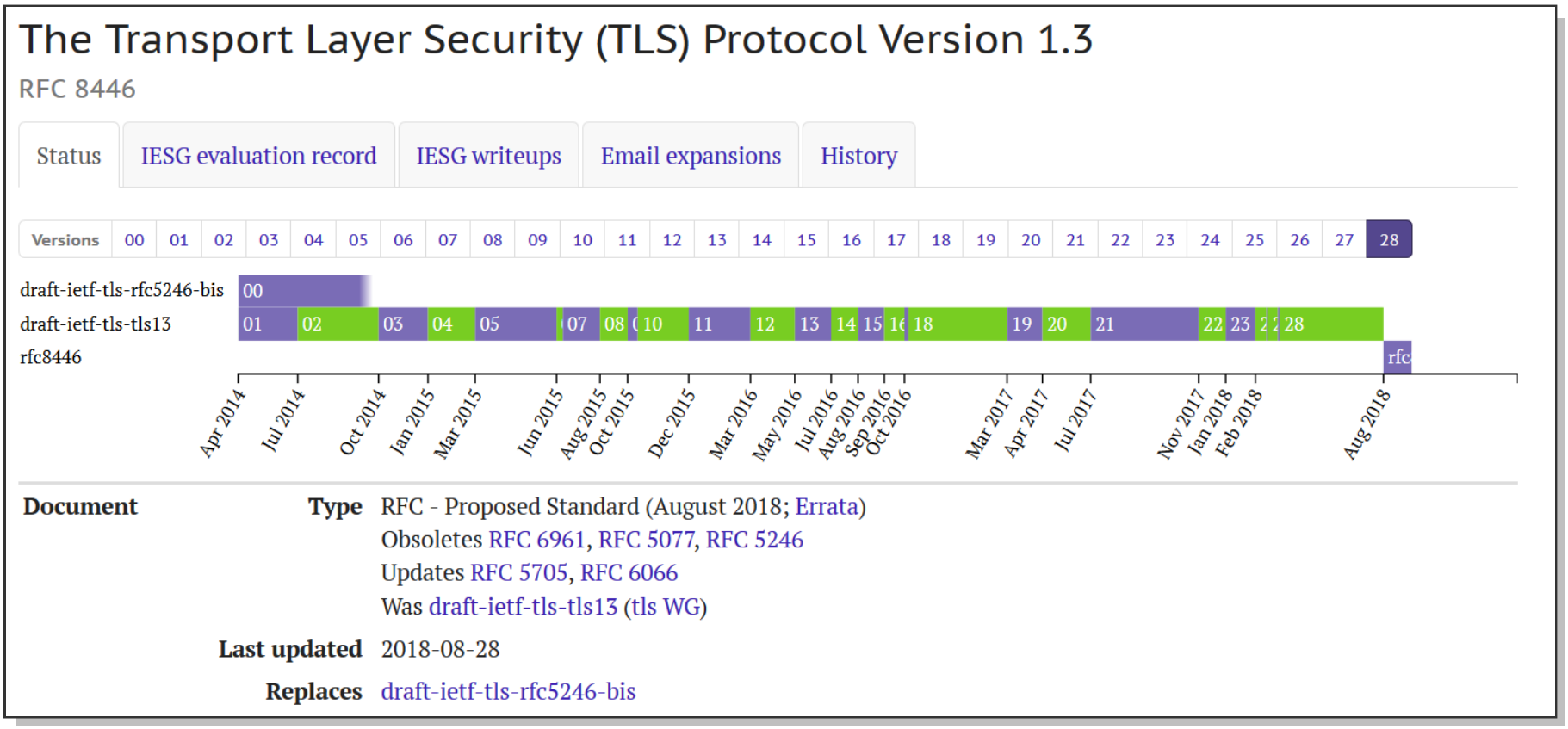
UC San Diego



Transport Layer Security (TLS)

The road to TLS 1.3 = RFC 8446

UC San Diego

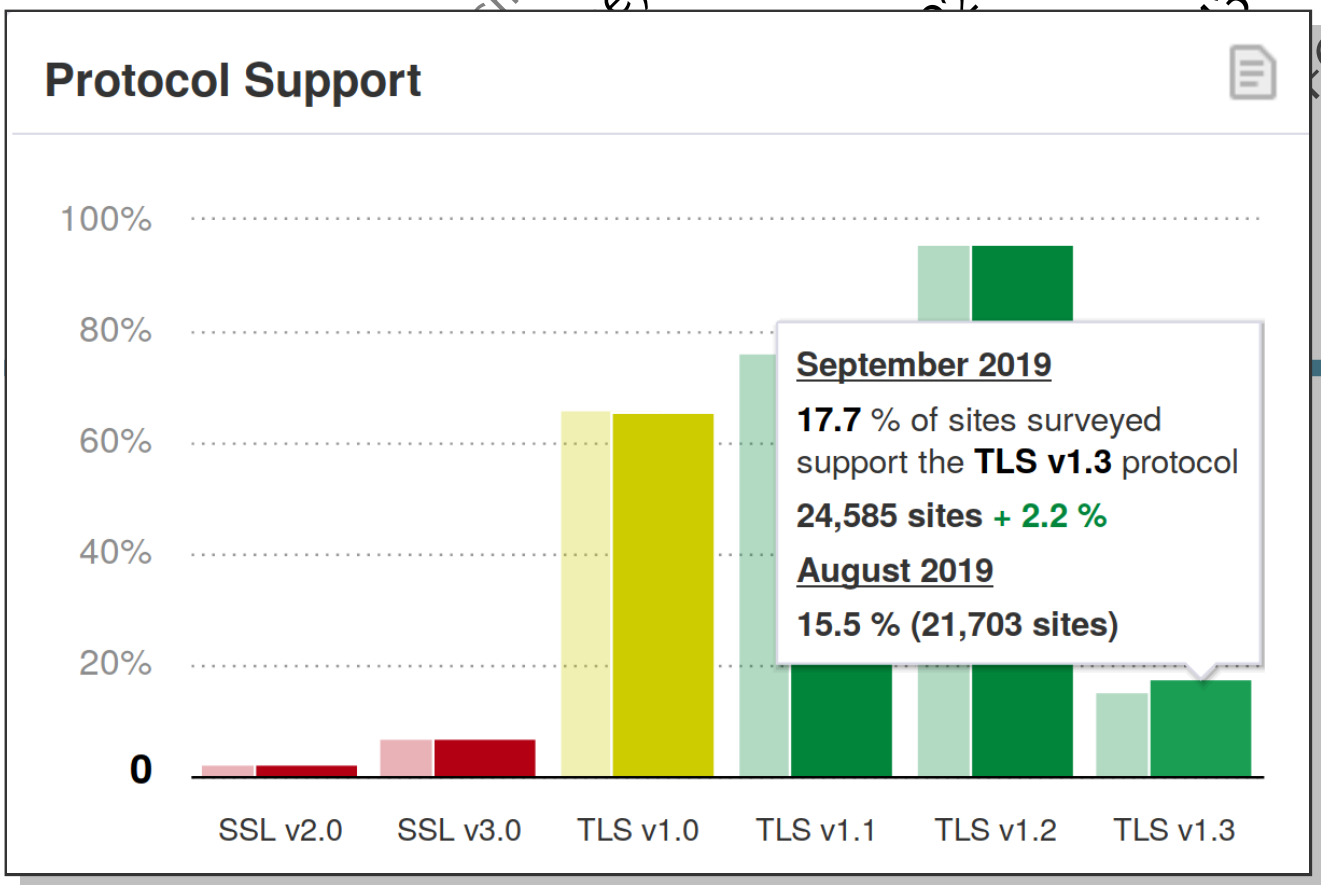


https://datatracker.ietf.org/doc/rfc8446/

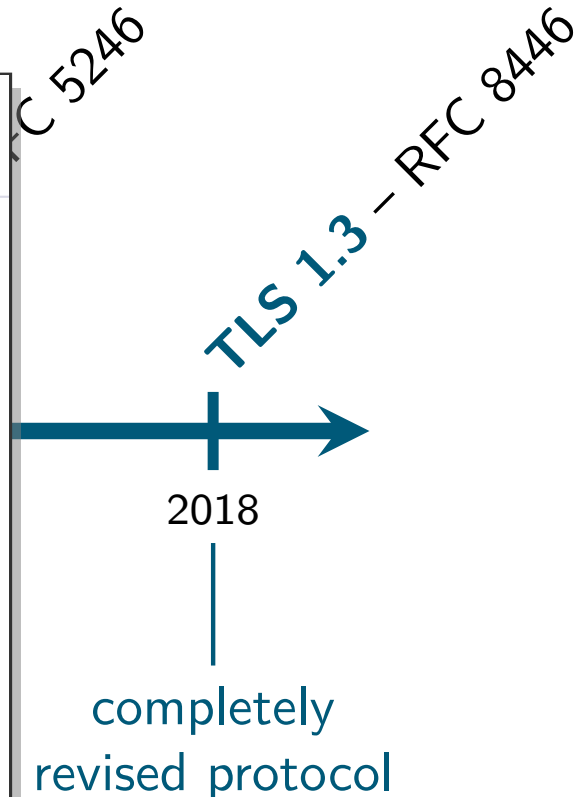
Transport Layer Security (TLS)

A new chapter: TLS 1.3

UC San Diego



https://www.ssllabs.com/ssl-pulse/, 150K top sites (Sep 2, 2019)



TLS 1.3

Design Goals

UC San Diego

- **Clean up:** get rid of flawed and unused crypto & features
- **Improve latency:** for main handshake and repeated connections (while maintaining security)
- **Improve privacy:** hide as much of the handshake as possible
- **Continuity:** maintain interoperability with previous versions and support existing important use cases
- **Security Assurance (added later):** have supporting analyses for changes

TLS 1.3

Main changes (from TLS 1.2)

UC San Diego

Clean up

- removed **legacy and broken crypto**
  - ciphers: (3)DES, RC4, . . . , MtEE (CBC & generally) — only AEAD remains
  - hash functions: MD5, SHA1
  - authentication: Kerberos, RSA PKCS#1v1.5 key transport
  - custom (EC)DHE groups
- removed **broken features**
  - compression
  - renegotiation (but added key updates + late client auth)
- removed **static RSA/DH**: public-key crypto = forward secrecy
- **hardened negotiation** of version/cipher suite against downgrades

quite some resistance from enterprises doing passive inspection

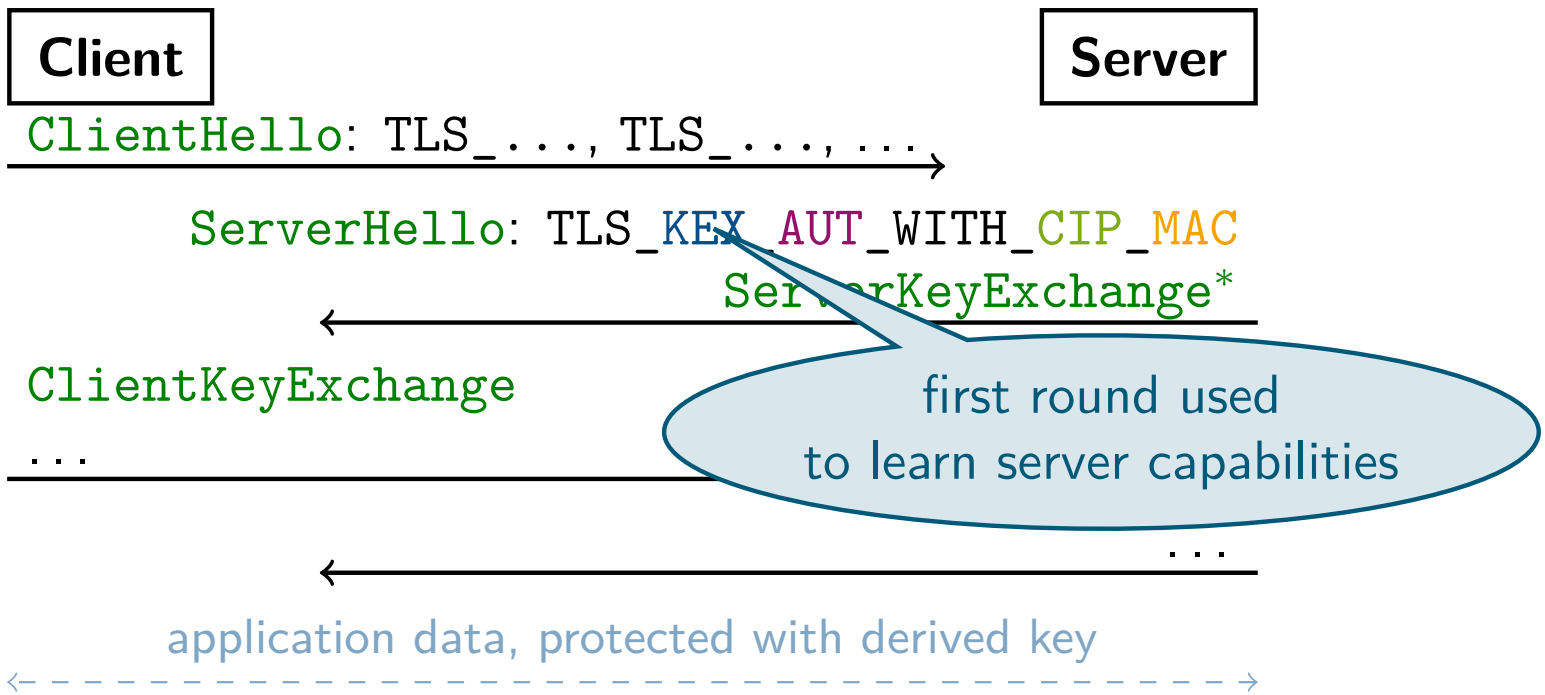
TLS 1.3

Main changes (from TLS 1.2)

UC San Diego

Improve latency

- TLS $\leq$ 1.2 is slow: 2 round trips before client can send data



TLS 1.3

Main changes (from TLS 1.2)

UC San Diego

Improve latency

- TLS $\leq$ 1.2 is slow: 2 round trips before client can send data
- TLS 1.3: **full handshake in 1 round trip**
  - feature reduction $\rightarrow$ we always do (EC)DHE
  - client speculatively sends several DH shares in supported groups
  - server picks one, replies with its share, and key can be already derived
- **0-RTT handshake** when resuming previous connection
  - client+server keep shared resumption secret (PSK)
  - client derives a key from that and can immediately encrypt data
  - but: 0-RTT *sacrifices* certain security properties (we'll get to that)

TLS 1.3

Main changes (from TLS 1.2)

UC San Diego

Improve privacy

- TLS $\leq$ 1.2: complete handshake in the clear (incl. certificates, extensions)
- TLS 1.3: **encrypts almost all handshake messages**
  - derive separate key early to protect handshake messages
  - provides security against passive/active attackers (for server/client)

Continuity

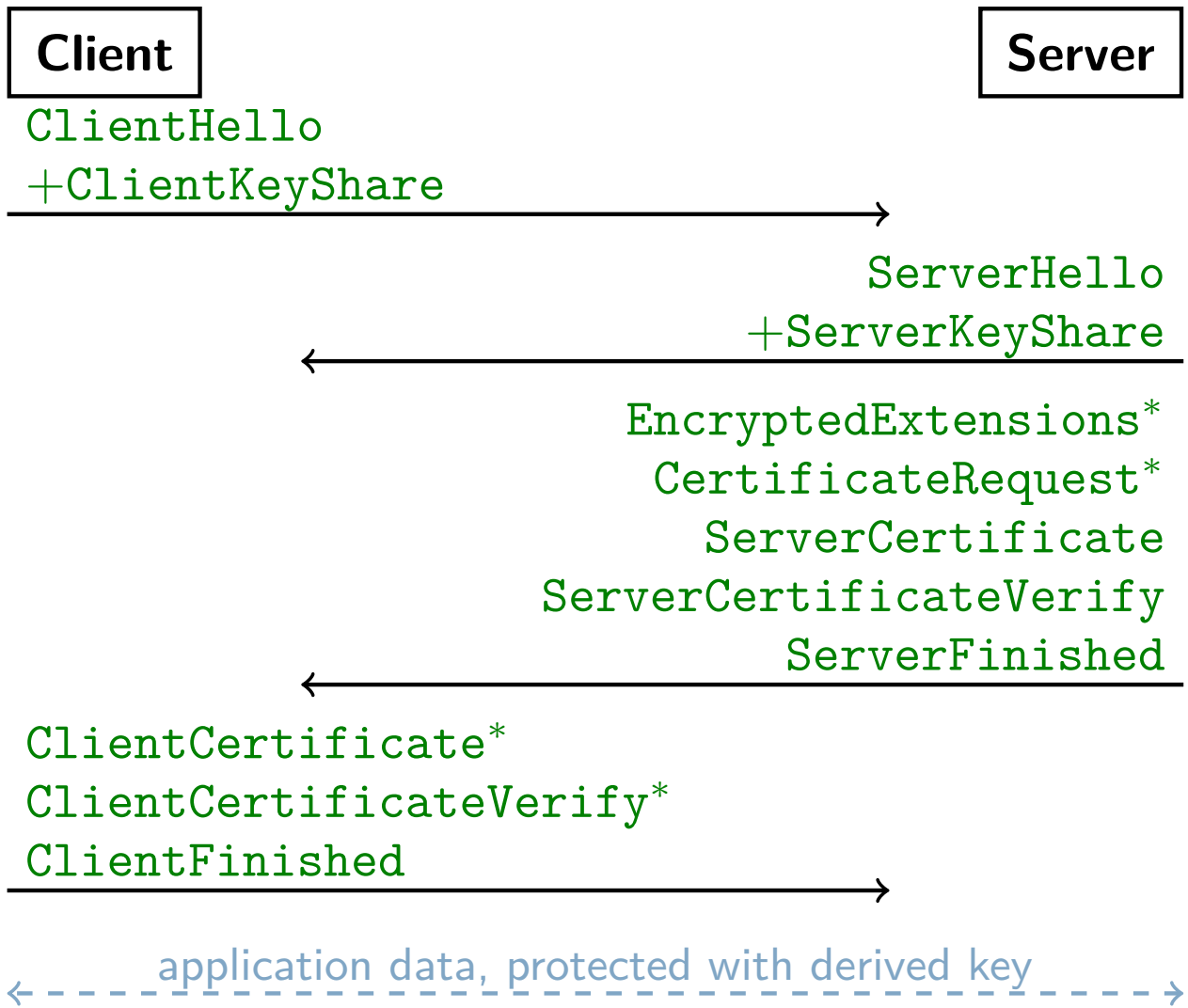
- e.g.: remove complex renegotiation, but keep features (key update + client auth)
- interoperability (idea): let `ClientHello` look like TLS $<$ 1.3

The TLS 1.3 Handshake

Full (EC)DHE Mode

UC San Diego

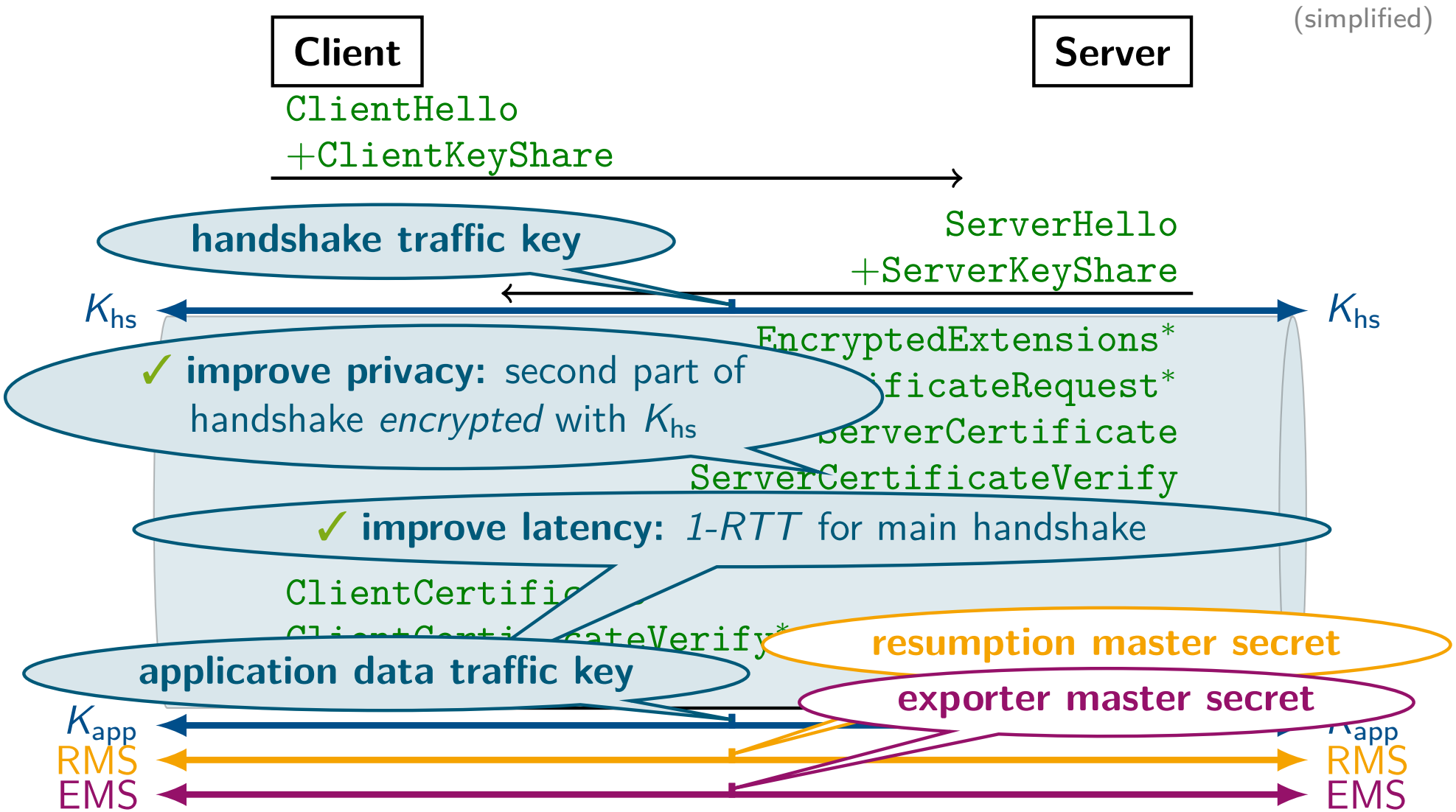
(simplified)



The TLS 1.3 Handshake

Full (EC)DHE Mode

UC San Diego



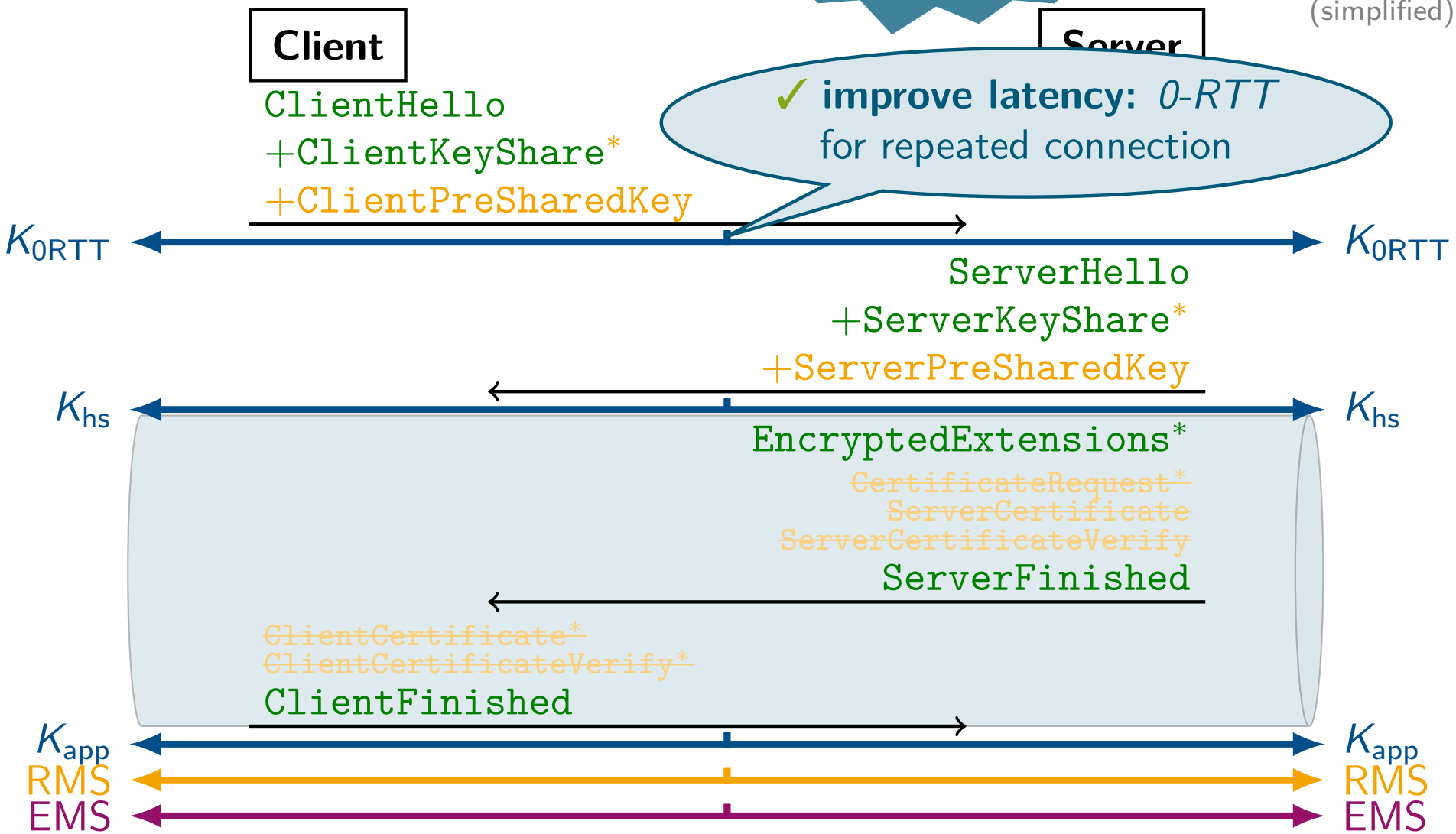
The TLS 1.3 Handshake

PSK / PSK-(EC)DHE Resumption Mode

multi-stage key exchange

UC San Diego

(simplified)

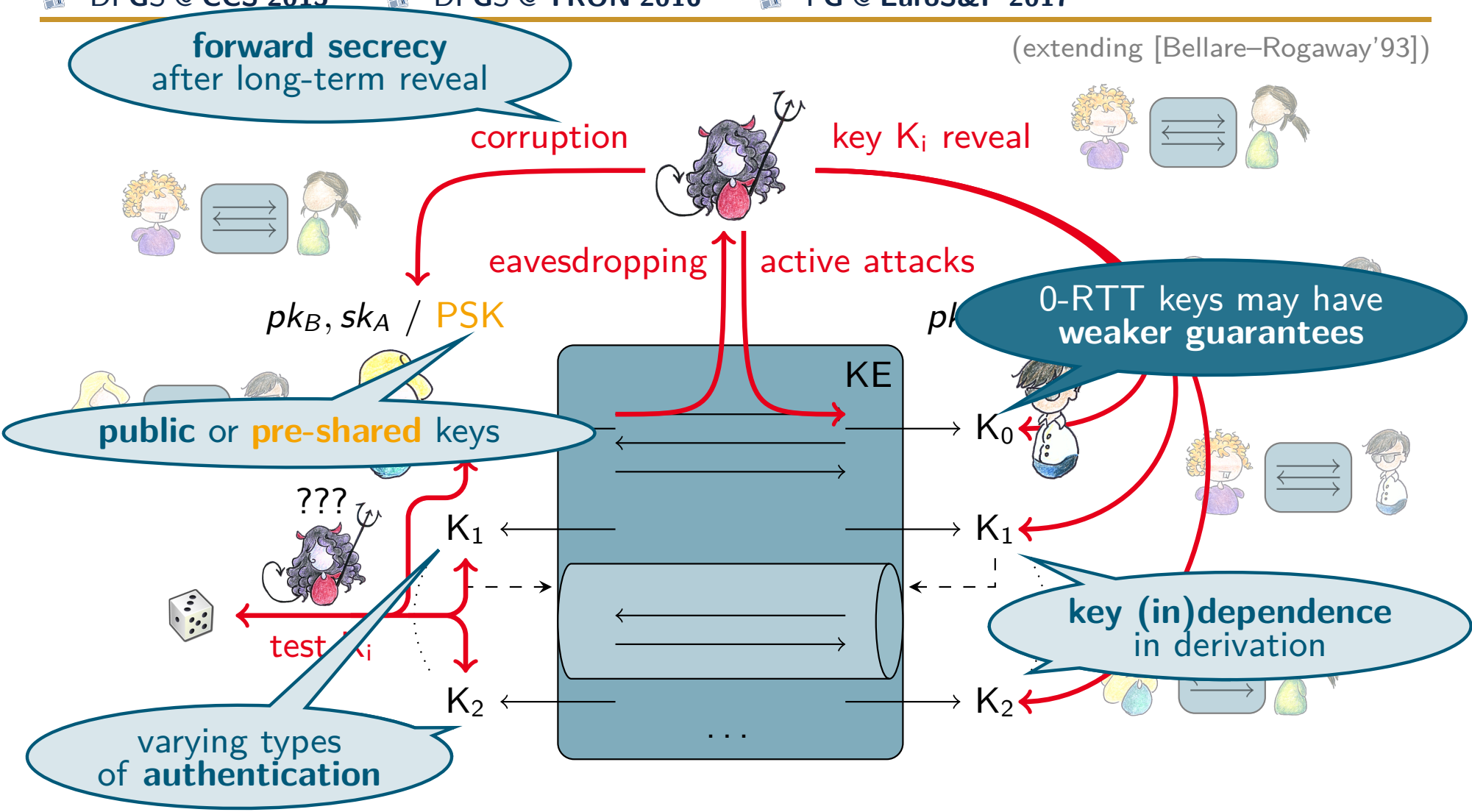


Multi-Stage Key Exchange

Fischlin, **Günther**. Multi-Stage Key Exchange and [..] QUIC [..]. **CCS 2014**
DFGS @ **CCS 2015** DFGS @ **TRON 2016** FG @ **EuroS&P 2017**

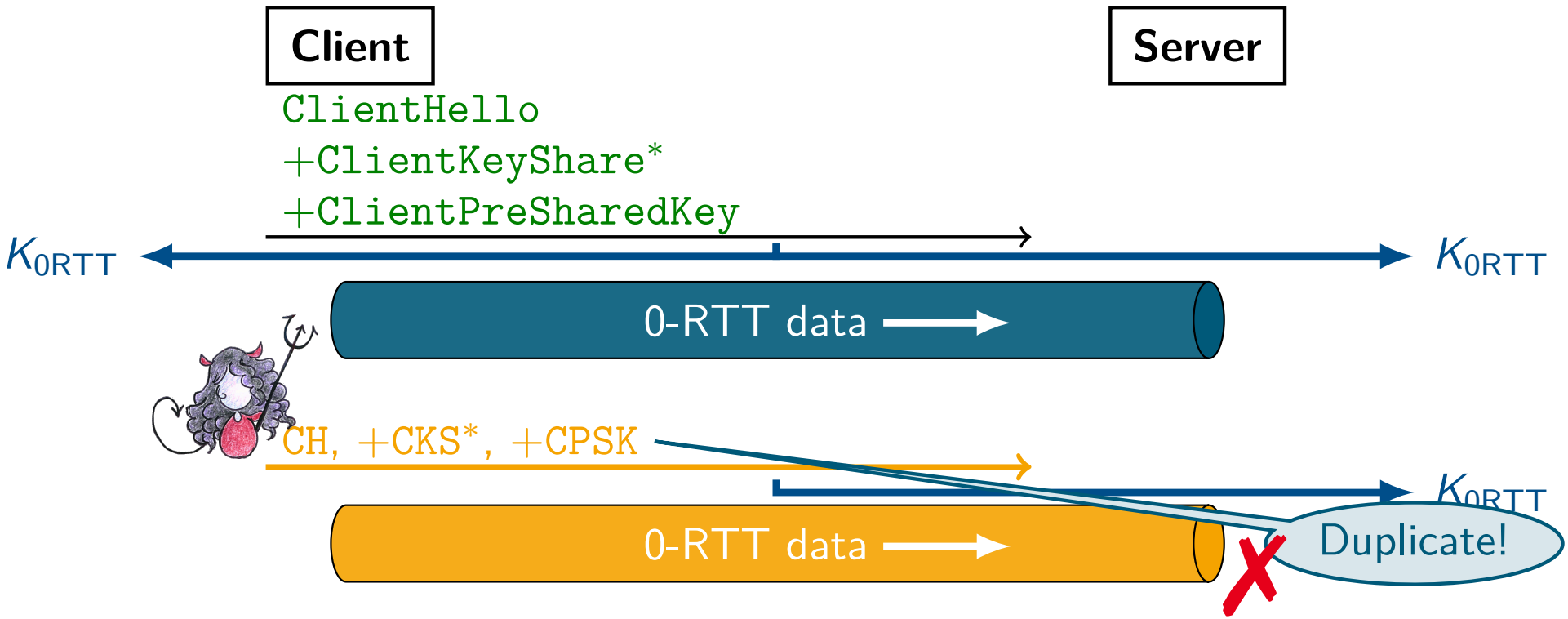
UC San Diego

(extending [Bellare–Rogaway'93])



0-RTT and Replays

UC San Diego

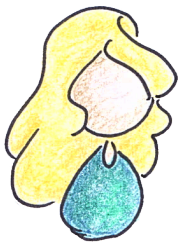


- allows client to send data without waiting for server reply
- but without server input, how does server know the request is fresh?
- adversary can **replay** `ClientHello` together with 0-RTT data
- idea: remember `ClientHello` identifier and reject duplicates

0-RTT and Replays

Generic Replay Attack on 0-RTT (by Daniel Kahn Gillmore)

UC San Diego



0-RTT KE msg

0-RTT request

process request

simpler in real world:
send to two
distributed servers

accept 0-RTT, KE response

enforce state loss (e.g., reboot)

replay 0-RTT KE msg

replay 0-RTT request

rej. after state loss
for security reasons

reject 0-RTT, KE response msg

complete KE

(resend) request under final key

process request
again

0-RTT and Replays

TLS 1.3's take on replays

UC San Diego

TLS does not provide inherent replay protection for 0-RTT data.

[Simple duplicates] can be prevented by sharing state to guarantee that the 0-RTT data is accepted at most once.

Servers SHOULD provide that level of replay safety by implementing one of the methods described in this section [...] [RFC 8446, Section 8]

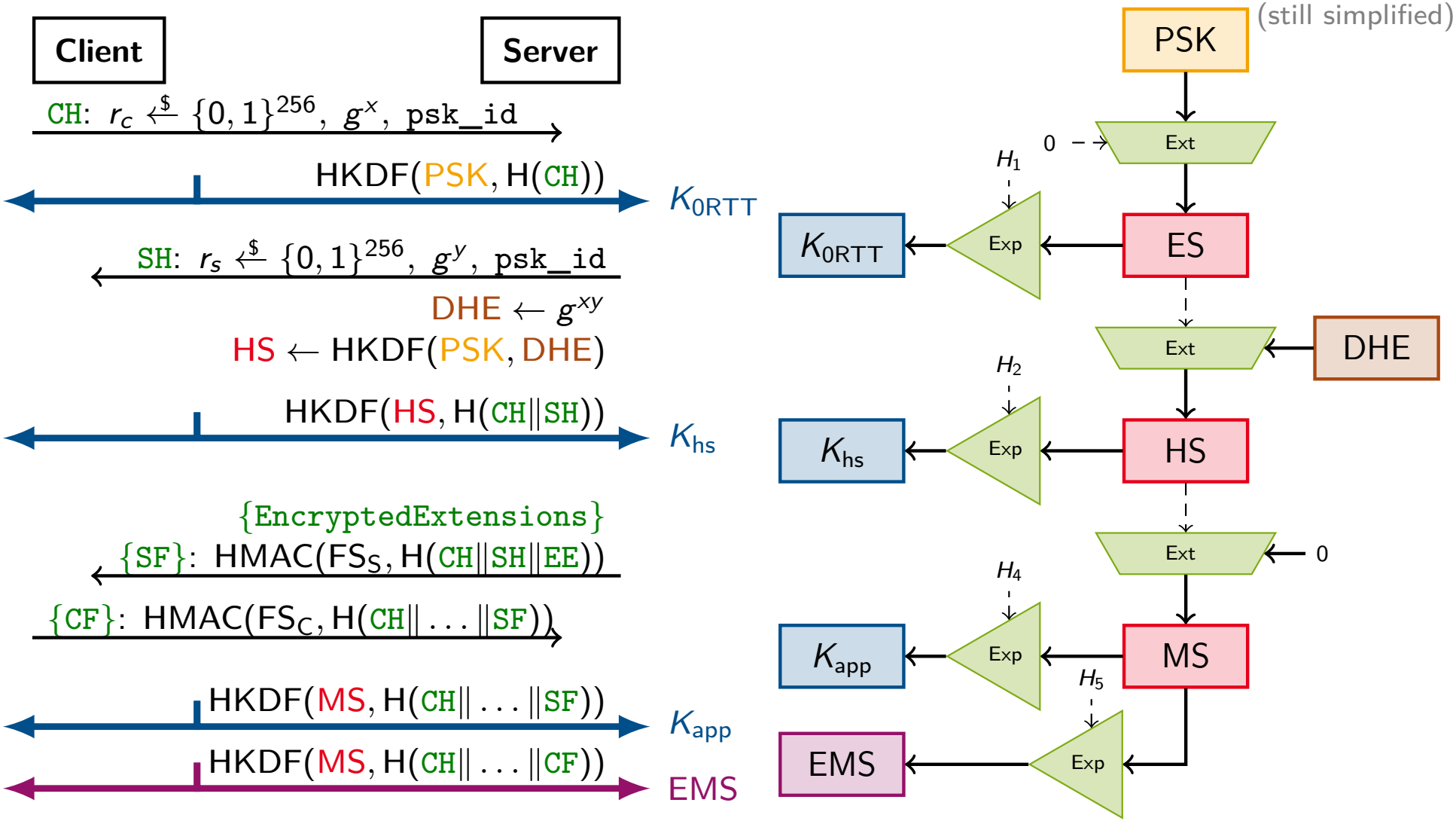
- **suggested mechanisms**
  - single-use tickets: allow each RMS to be used only once (simplest)
  - `ClientHello` recording: reject by unique identifier
  - freshness checks: reject based on `ClientHello` time

- "SHOULD" $\rightarrow$ treat 0-RTT keys generally as **replayable in MSKE model**

The TLS 1.3 Handshake

draft-14 PSK-(EC)DHE 0-RTT

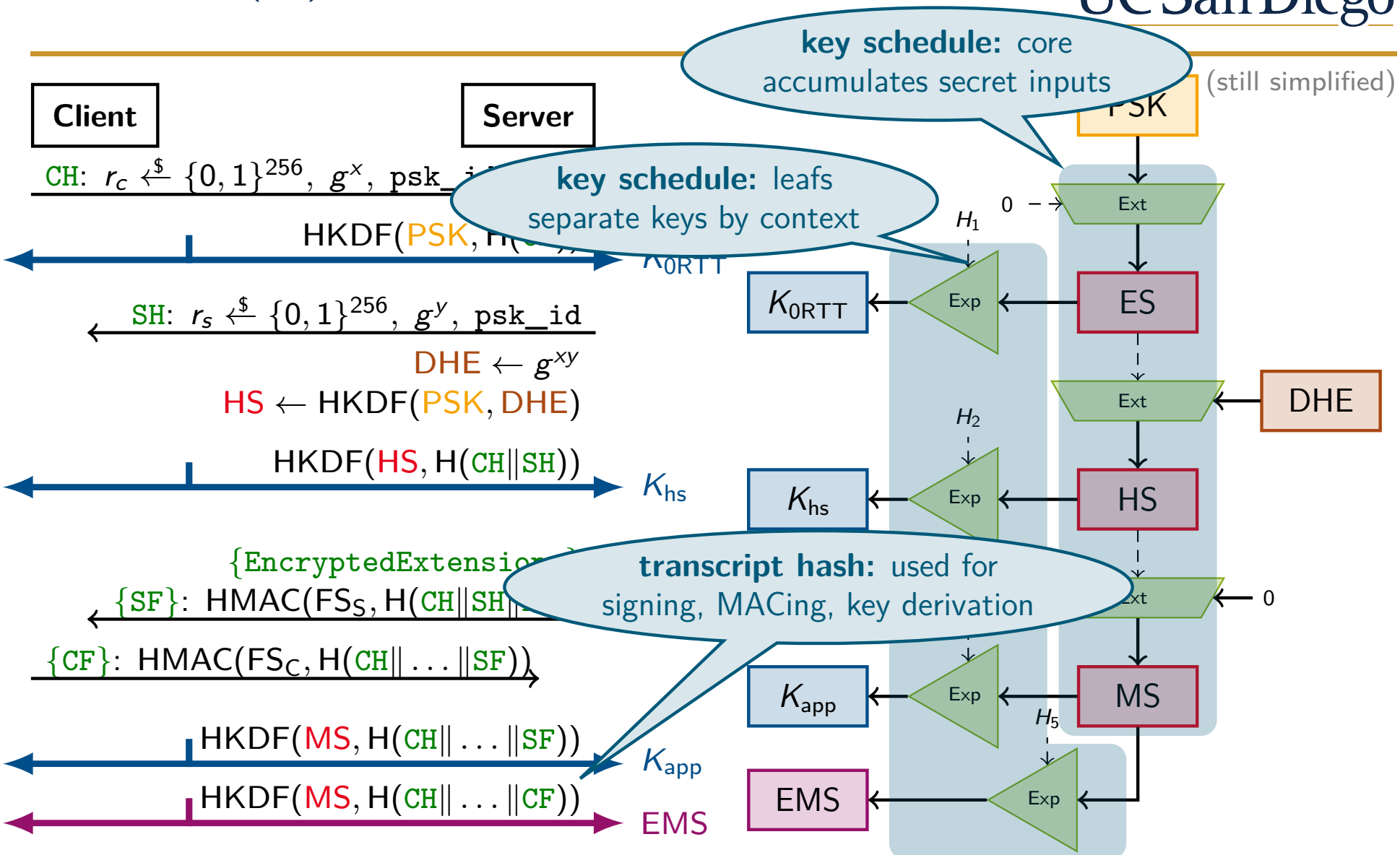
UC San Diego



The TLS 1.3 Handshake

draft-14 PSK-(EC)DHE 0-RTT

UC San Diego



TLS 1.3 Handshake Security

draft-14 PSK-(EC)DHE 0-RTT as Multi-Stage KE


Fischlin, **Günther**. Replay Attacks on Zero Round-Trip Time [..]. **EuroS&P 2017**

UC San Diego

The **TLS 1.3 PSK-(EC)DHE 0-RTT** handshake provides

- random-looking secret keys
- forward secrecy
  for non–0-RTT keys
- mutual authentication wrt. PSK
- key independence
- replayable 0-RTT keys

assuming . . .

Theorem 7.4. *The TLS 1.3* `draft-14` *PSK-(EC)DHE 0-RTT handshake is* **Multi-Stage-secure** *in a* ***key-independent*** *and* ***stage-3-forward-secret*** *manner with properties* (M, **AUTH**, USE, **REPLAY**).

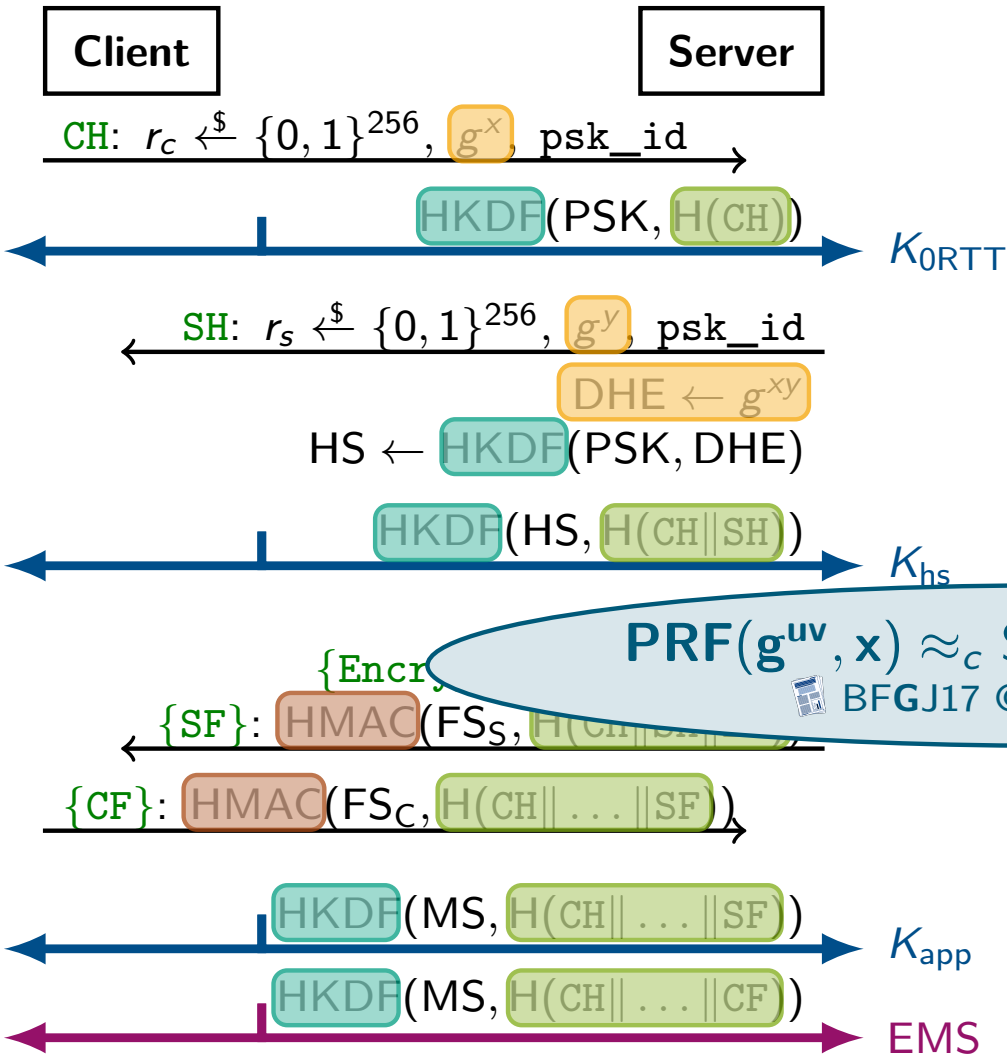
$$
\begin{aligned}
\mathsf{Adv}^{\text{Multi-Stage},\mathcal{D}}_{\texttt{draft-14-PSK-(EC)DHE-0RTT},\mathcal{A}} \leq\ & 5n_s \cdot \Big(\mathsf{Adv}^{\text{COLL}}_{\mathsf{H},\mathcal{B}_1} \\
& + n_p \cdot \Big(\mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HKDF.Expand},\mathcal{B}_2} + \mathsf{Adv}^{\text{HMAC(0,\$)-\$}}_{\mathsf{HMAC},\mathcal{B}_3} \\
& \qquad + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HMAC},\mathcal{B}_4} + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HKDF.Expand},\mathcal{B}_5} \Big) \\
& + n_s \cdot n_p \cdot \Big(\mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HKDF.Expand},\mathcal{B}_6} + \mathsf{Adv}^{\text{HMAC(0,\$)-\$}}_{\mathsf{HMAC},\mathcal{B}_7} \\
& \qquad + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HMAC},\mathcal{B}_8} + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HMAC},\mathcal{B}_9} \\
& \qquad + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HKDF.Expand},\mathcal{B}_{10}} + \mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{HMAC},\mathcal{B}_{11}} \Big) \\
& + n_s \cdot n_p \cdot \Big(\mathsf{Adv}^{\text{snPRF-ODH}}_{\mathsf{HKDF.Extract},\mathbb{G},\mathcal{B}_{12}} + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HMAC},\mathcal{B}_{13}} \\
& \qquad + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HKDF.Expand},\mathcal{B}_{14}} + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HKDF.Expand},\mathcal{B}_{15}} \\
& \qquad + \mathsf{Adv}^{\text{PRF-sec}}_{\mathsf{HKDF.Expand},\mathcal{B}_{16}} \Big) \Big).
\end{aligned}
$$

TLS 1.3 Handshake Security

draft-14 PSK-(EC)DHE 0-RTT as Multi-Stage KE

Fischlin, **Günther**. Replay Attacks on Zero Round-Trip Time [..]. **EuroS&P 2017**

UC San Diego



PRF(g^{uv}, x) $\approx_c$ \$, given oracle **PRF($\cdot^u$, $\cdot$)**

BFGJ17 @ **CRYPTO 2017**

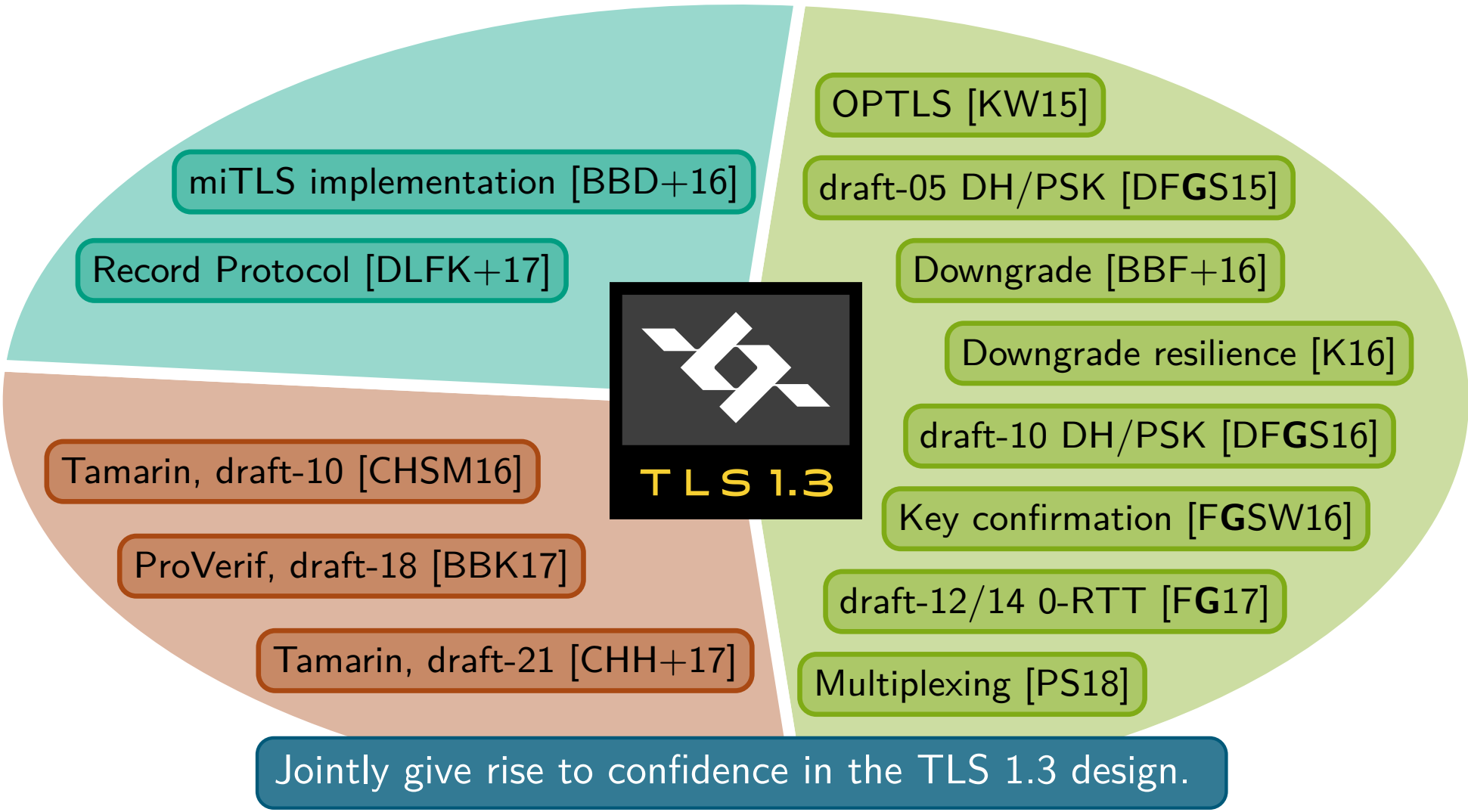
Theorem 7.4. *The TLS 1.3* `draft-14` *PSK-(EC)DHE 0-RTT handshake is* **Multi-Stage-secure** *in a* ***key-independent*** *and* ***stage-3-forward-secret*** *manner with properties* (M, **AUTH**, USE, **REPLAY**).

$$\begin{aligned}
\mathrm{Adv}^{\text{Multi-Stage},\mathcal{D}}_{\texttt{draft-14-PSK-(EC)DHE-0RTT},\mathcal{A}} \leq\ & 5n_s \cdot \Big(\mathrm{Adv}^{\text{COLL}}_{H,\mathcal{B}_1} \\
& + n_p \cdot \Big(\mathrm{Adv}^{\text{PRF-sec}}_{\text{HKDF.Expand},\mathcal{B}_2} + \mathrm{Adv}^{\text{HMAC(0,\$)-\$}}_{\text{HMAC},\mathcal{B}_3} \\
& \quad + \mathrm{Adv}^{\text{PRF-sec}}_{\text{HMAC},\mathcal{B}_4} + \mathrm{Adv}^{\text{PRF-sec}}_{\text{HKDF.Expand},\mathcal{B}_5} \Big) \\
& + \mathrm{Adv}^{\text{HMAC(0,\$)-\$}}_{\text{HMAC},\mathcal{B}_7} \\
& + \mathrm{Adv}^{\text{PRF-sec}}_{\text{HMAC},\mathcal{B}_9} \\
& + \mathrm{Adv}^{\text{PRF-sec}}_{\text{HKDF.Expand},\mathcal{B}_{10}} + \mathrm{Adv}^{\text{EUF-CMA}}_{\text{HMAC},\mathcal{B}_{11}} \Big) \\
& + n_s \cdot n_p \cdot \Big(\mathrm{Adv}^{\text{snPRF-ODH}}_{\text{HKDF.Extract},\mathbb{G},\mathcal{B}_{12}} + \mathrm{Adv}^{\text{PRF-sec}}_{\text{HMAC},\mathcal{B}_{13}} \\
& \quad + \mathrm{Adv}^{\text{PRF-sec}}_{\text{HKDF.Expand},\mathcal{B}_{14}} + \mathrm{Adv}^{\text{PRF-sec}}_{\text{HKDF.Expand},\mathcal{B}_{15}} \\
& \quad + \mathrm{Adv}^{\text{PRF-sec}}_{\text{HKDF.Expand},\mathcal{B}_{16}} \Big) \Big).
\end{aligned}$$

TLS 1.3 Security Analysis

Many more analyses

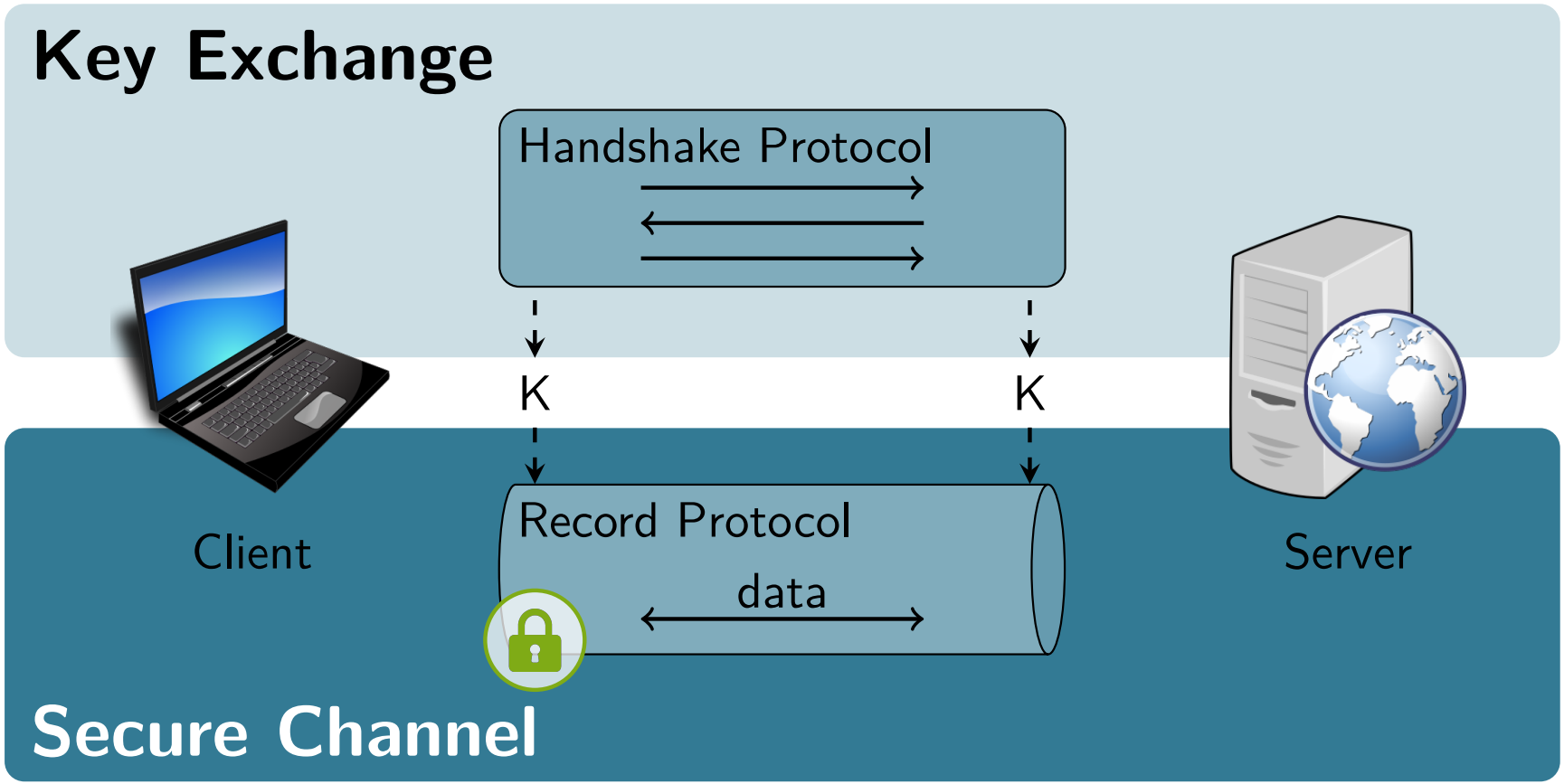
UC San Diego



Transport Layer Security (TLS)

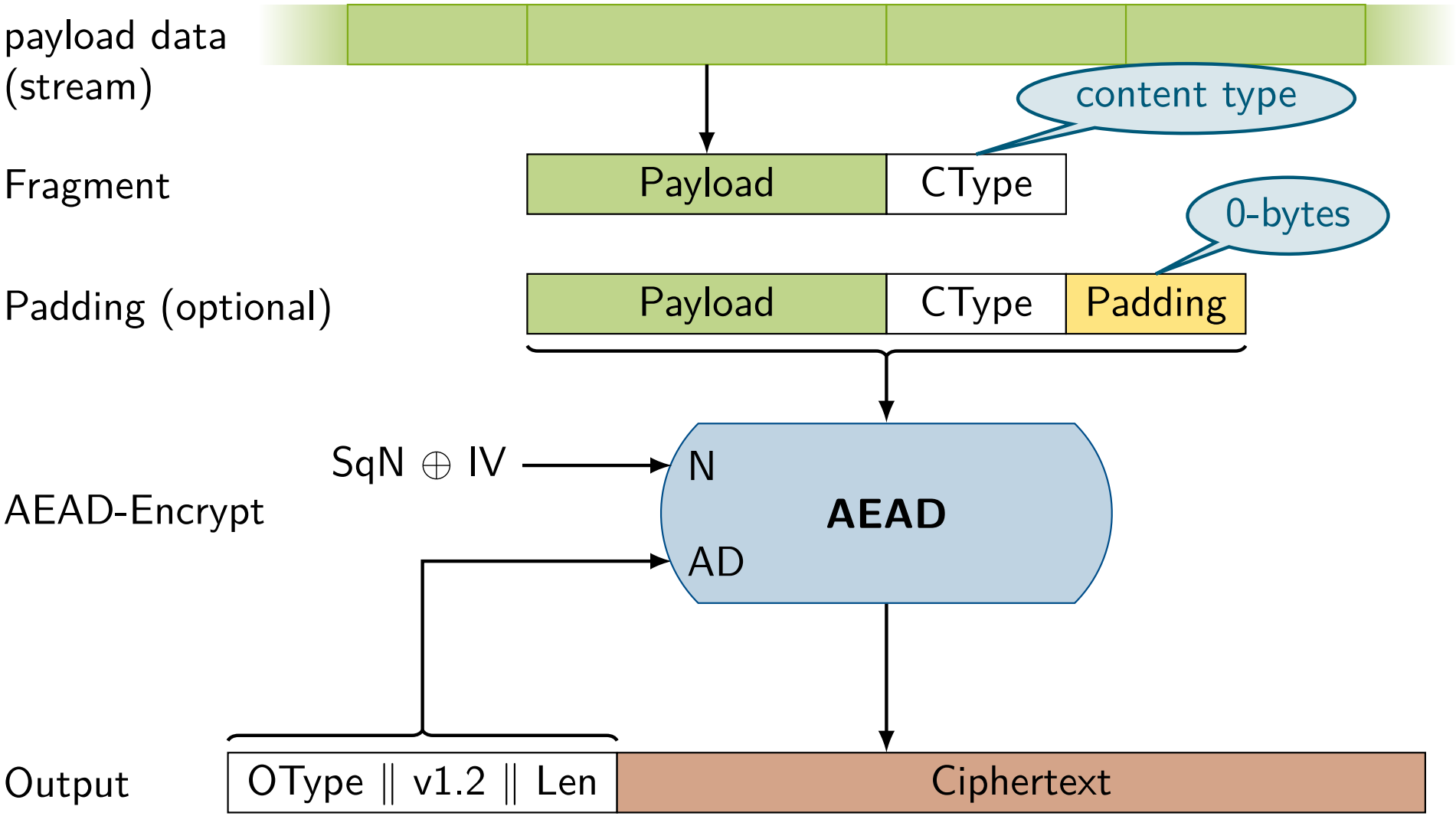
So... what about the Record Protocol?

UC San Diego



The TLS 1.3 Record Protocol

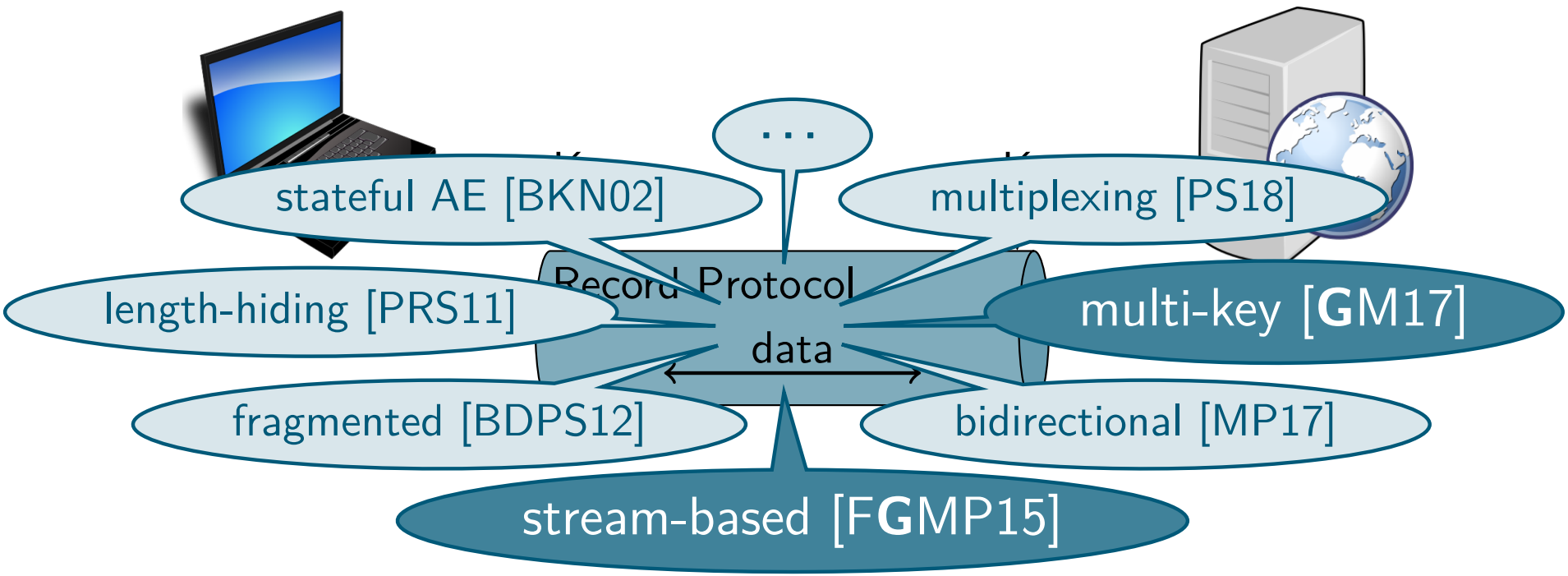
UC San Diego



TLS 1.3 Record Protocol Security

UC San Diego

- AEAD-based design looks sound. . .
- but the crypto community hasn't really conclusively ventilated the question: **What is a secure channel protocol?**



Example: Fragmentation

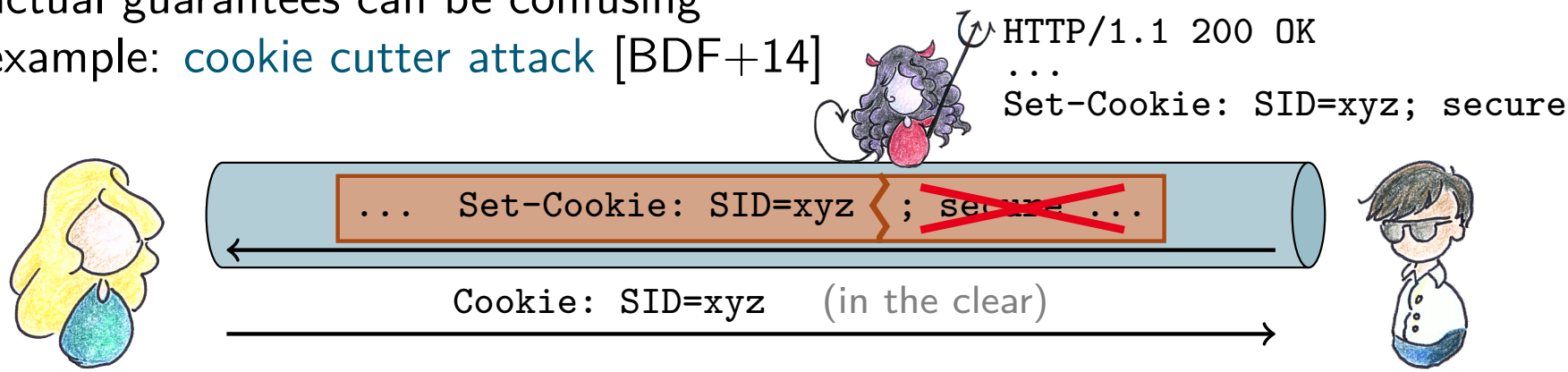
UC San Diego

5.1 Record Layer
*The record layer **fragments** information blocks into [...] records carrying data in **chunks of 2^14 bytes or less**. [...] Application Data fragments MAY be **split** across multiple records or **coalesced** into a single record.*

TLS 1.3 [RFC 8446]

- common crypto notions assume atomic messages / ciphertexts
  ... but channels are more than just AEAD

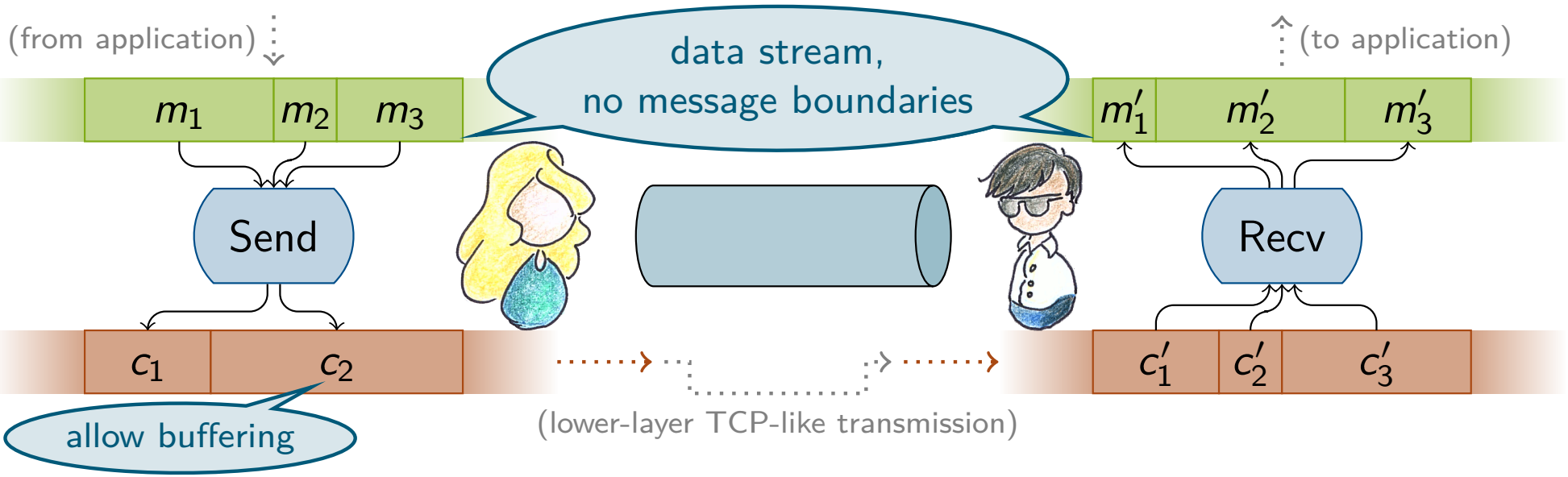
- actual guarantees can be confusing
- example: cookie cutter attack [BDF+14]



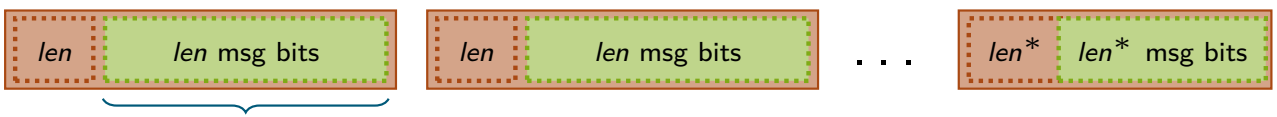
Example: Fragmentation
Stream-based Channels

Fischlin, **Günther**, Marson, Paterson. Data Is a Stream [...]. **CRYPTO 2015**

UC San Diego



- TLS 1.3–like construction of a streaming channel . . .



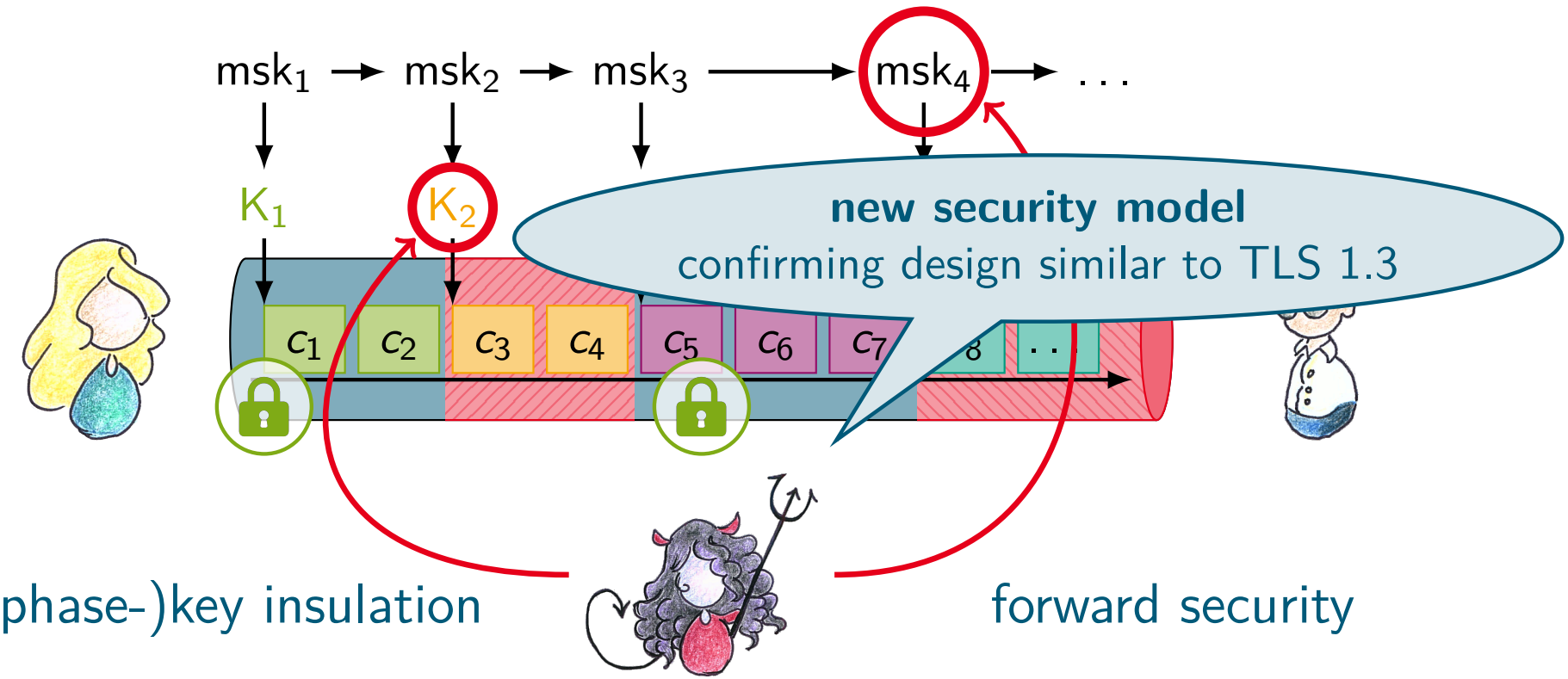
AEAD $\leq$ *len* bits with AD = seqno

- . . . achieves security against chosen ciphertext-fragment attacks (assuming secure AEAD scheme)

Example: Multi-Key Channels

Günther, Mazaheri. A Formal Treatment of Multi-key Channels. **CRYPTO 2017**

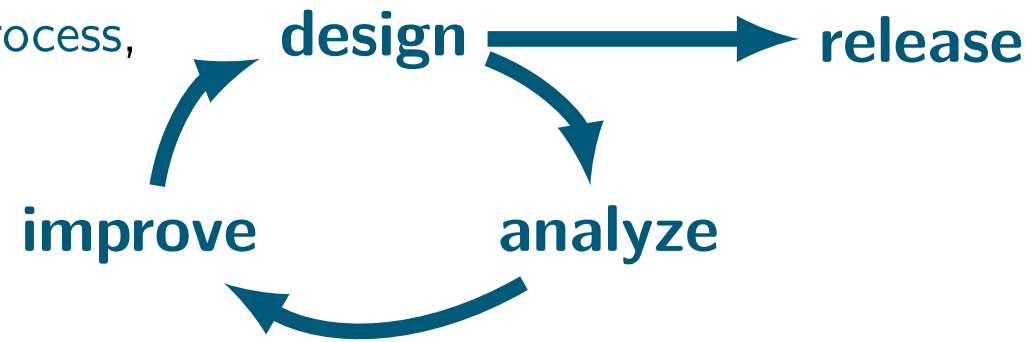
- classically: 1 key ✓
- **TLS 1.3**, **QUIC**, **Signal**, . . . : keys updated during channel operation



UC San Diego

A Summary

- commendable standardization process, highly interactive



- sound cryptographic design
  - improving substantially over prior versions
  - yet with possibly "dangerous" 0-RTT mode

- wide-spread deployment after only about 1 year





Google
CLOUDFLARE
Java
. . .

Conclusions & Where to?

UC San Diego

- **advanced crypto & security modeling contributes to secure standards**
- best to integrate formal analysis into the process
  - **Message Layer Security** (MLS) for secure messaging
    - Workshop on Secure Messaging @ Eurocrypt 2019
  - **QUIC**: UDP-based secure transport
    - **QUIPS'20**: QUIC Privacy and Security Workshop @ NDSS 2020
      Submission deadline: December 13, 2019
- security standards need you!

Thank You!

mail@**felixguenther.info**

References I

UC San Diego

[BKN02] M. Bellare, T. Kohno, and C. Namprempre. "Authenticated Encryption in SSH: Provably Fixing The SSH Binary Packet Protocol". In: *ACM CCS 2002*. Ed. by V. Atluri. ACM Press, Nov. 2002, pp. 1–11.

[BR94] M. Bellare and P. Rogaway. "Entity Authentication and Key Distribution". In: *CRYPTO'93*. Ed. by D. R. Stinson. Vol. 773. LNCS. Springer, Heidelberg, Aug. 1994, pp. 232–249.

[BDF+14] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Pironti, and P.-Y. Strub. "Triple Handshakes and Cookie Cutters: Breaking and Fixing Authentication over TLS". In: *2014 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2014, pp. 98–113.

[BDPS12] A. Boldyreva, J. P. Degabriele, K. G. Paterson, and M. Stam. "Security of Symmetric Encryption in the Presence of Ciphertext Fragmentation". In: *EUROCRYPT 2012*. Ed. by D. Pointcheval and T. Johansson. Vol. 7237. LNCS. Springer, Heidelberg, Apr. 2012, pp. 682–699.

[BFGJ17] J. Brendel, M. Fischlin, F. Günther, and C. Janson. "PRF-ODH: Relations, Instantiations, and Impossibility Results". In: *CRYPTO 2017, Part III*. Ed. by J. Katz and H. Shacham. Vol. 10403. LNCS. Springer, Heidelberg, Aug. 2017, pp. 651–681.

[DFGS15] B. Dowling, M. Fischlin, F. Günther, and D. Stebila. "A Cryptographic Analysis of the TLS 1.3 Handshake Protocol Candidates". In: *ACM CCS 2015*. Ed. by I. Ray, N. Li, and C. Kruegel. ACM Press, Oct. 2015, pp. 1197–1210.

[DFGS16] B. Dowling, M. Fischlin, F. Günther, and D. Stebila. *A Cryptographic Analysis of the TLS 1.3 draft-10 Full and Pre-shared Key Handshake Protocol*. Cryptology ePrint Archive, Report 2016/081. `http://eprint.iacr.org/2016/081`. 2016.

References II

UC San Diego

[FG14] M. Fischlin and F. Günther. "Multi-Stage Key Exchange and the Case of Google's QUIC Protocol". In: *ACM CCS 2014*. Ed. by G.-J. Ahn, M. Yung, and N. Li. ACM Press, Nov. 2014, pp. 1193–1204.

[FG17] M. Fischlin and F. Günther. "Replay Attacks on Zero Round-Trip Time: The Case of the TLS 1.3 Handshake Candidates". In: *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017*. Paris, France: IEEE, 2017, pp. 60–75.

[FGMP15] M. Fischlin, F. Günther, G. A. Marson, and K. G. Paterson. "Data Is a Stream: Security of Stream-Based Channels". In: *CRYPTO 2015, Part II*. Ed. by R. Gennaro and M. J. B. Robshaw. Vol. 9216. LNCS. Springer, Heidelberg, Aug. 2015, pp. 545–564.

[GM17] F. Günther and S. Mazaheri. "A Formal Treatment of Multi-key Channels". In: *CRYPTO 2017, Part III*. Ed. by J. Katz and H. Shacham. Vol. 10403. LNCS. Springer, Heidelberg, Aug. 2017, pp. 587–618.

[MP17] G. A. Marson and B. Poettering. "Security Notions for Bidirectional Channels". In: *IACR Trans. Symm. Cryptol.* 2017.1 (2017), pp. 405–426.

[PRS11] K. G. Paterson, T. Ristenpart, and T. Shrimpton. "Tag Size Does Matter: Attacks and Proofs for the TLS Record Protocol". In: *ASIACRYPT 2011*. Ed. by D. H. Lee and X. Wang. Vol. 7073. LNCS. Springer, Heidelberg, Dec. 2011, pp. 372–389.

[PS18] C. Patton and T. Shrimpton. "Partially Specified Channels: The TLS 1.3 Record Layer without Elision". In: *ACM CCS 2018*. Ed. by D. Lie, M. Mannan, M. Backes, and X. Wang. ACM Press, Oct. 2018, pp. 1415–1428.