Introduction to TLS

TLS ≤1.2 & Cryptographic Background

UC San Diego

TLS Crypto Seminar

January 17, 2019

Felix Günther
UC San Diego

special thanks to Kenny Paterson

UC San Diego

DFG Deutsche Forschungsgemeinschaft
German Research Foundation

This Seminar

UC San Diego

Goal

- gain (some) understanding of a complex real-world protocol and its crypto
- partially *lecture-style* introduction to protocol and crypto background
- partially *reading-group–style* covering specific results (thanks to presenters!)

Part I TLS ≤1.2

- The Transport Layer Security (TLS) protocol: intro and crypto background.
- Attacks and analyses: understanding past weaknesses and hurdles.

Part II TLS 1.3

- The road to TLS 1.3 & its technical details.
- More analyses: understanding TLS 1.3's security and what drove design.

Schedule

UC San Diego

TLS ≤1.2

Jan 17	**TLS intro** [TLS12] **& crypto backround** [BR94,BKN02]	Felix
Jan 24	**Lucky 13** [AP13]	Nicholas
Jan 31	no seminar	-
Feb 7	**The ACCE model** [JKSS12,KPW13]	Joseph
Feb 14	**Logjam** [ABD+15]	Mark

TLS 1.3

Feb 21	**TLS 1.3** [TLS13] **& some security models** [FG17,GM17]	Felix
Feb 28	**Multiplexing channels** [PS18]	Vivek
Mar 7	**Symbolic Tamarin analysis** [CHH+17]	Baiyu
Mar 14	**Downgrade resilience** [BBF+16]	Ruth

Introduction to TLS

So What Is TLS?

UC San Diego

The Transport Layer Security (TLS) Protocol

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.

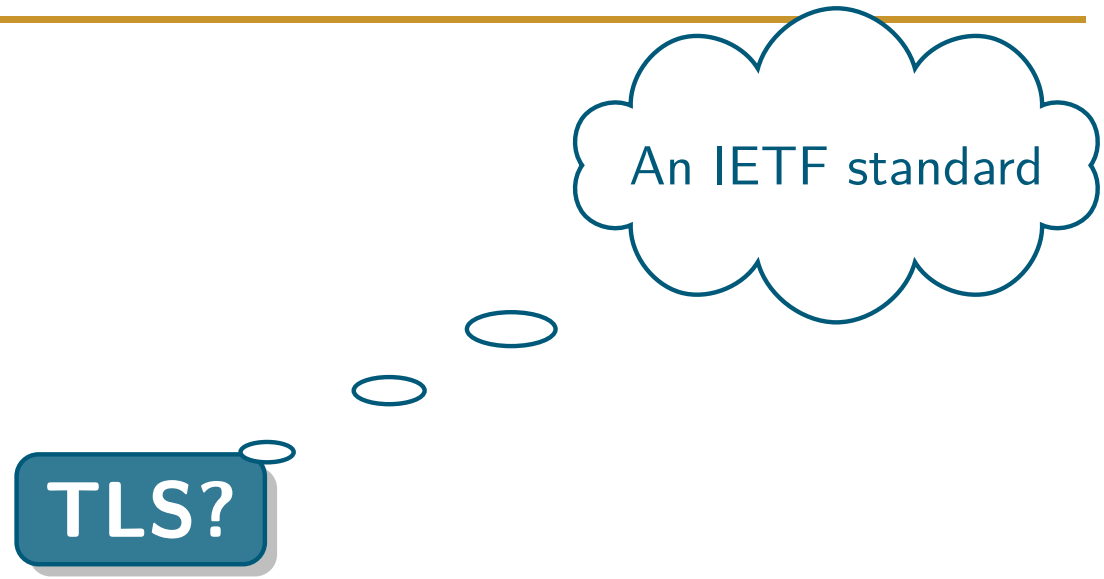
TLS 1.3 [RFC 8446]

1994	**SSL 1.0** (unpublished)	all considered seriously broken today
1995	**SSL 2.0**	
1996	**SSL 3.0**	
1999	**TLS 1.0** – RFC 2246	≈ SSL 3.0, adopted by IETF
2006	**TLS 1.1** – RFC 4346	maintained by IETF TLS working group
2008	**TLS 1.2** – RFC 5246	a team effort, editor: Eric Rescorla
2018	**TLS 1.3** – RFC 8446	

IETF

So What Is TLS?

UC San Diego



The TLS Protocol

A Story of Success . . . and Failures

UC San Diego

- initially introduced by Netscape to enable e-commerce on the WWW
- today: protecting billions of Internet connections every day
  - web, email, messaging, VoIP, banking, payments, e-health, . . .
  - $>80\%$ of web traffic is encrypted[1]
- an exposed target for attacks with a track record of critical flaws
  - structural/design errors
  - weaknesses in cryptographic primitives
  - implementation flaws
  - . . .
- crypto and security research important to understand and improve security
  - finding protocol flaws, guiding design, discussing security trade-offs

[1] e.g., `https://www.f5.com/labs/articles/threat-intelligence/the-2017-tls-telemetry-report`

So What Is TLS?

UC San Diego

The TLS Protocol

High-level Goals

UC San Diego

(from TLS 1.3, RFC 8446)

"The primary goal of TLS is to provide a **secure channel between two peers**"

- only requirement from underlying transport: reliable, in-order data stream
- **Authentication**
  - **server** side of the channel is **always authenticated**
  - **client** side is **optionally authenticated**
  - via **asymmetric crypto** (e.g., signatures) or a symmetric **pre-shared key**
- **Confidentiality**
  - **data** sent over the channel is **only visible to the endpoints**
  - TLS does **not hide the length** of the data it transmits (but allows padding)
- **Integrity**
  - **data** sent over the channel **cannot be modified** without detection
- security in the face of **attacker who has complete control of the network**

So What Is TLS?

UC San Diego

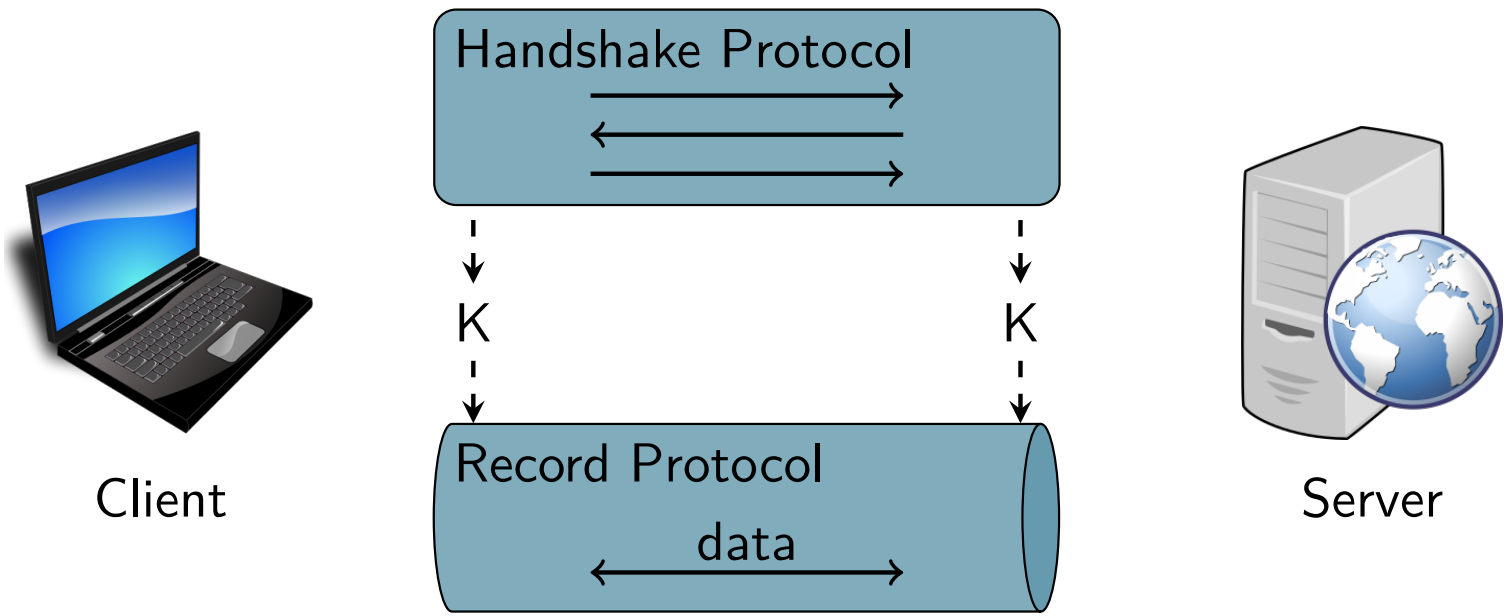
The TLS Protocol

Overly Simplified

UC San Diego

Handshake Protocol:

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

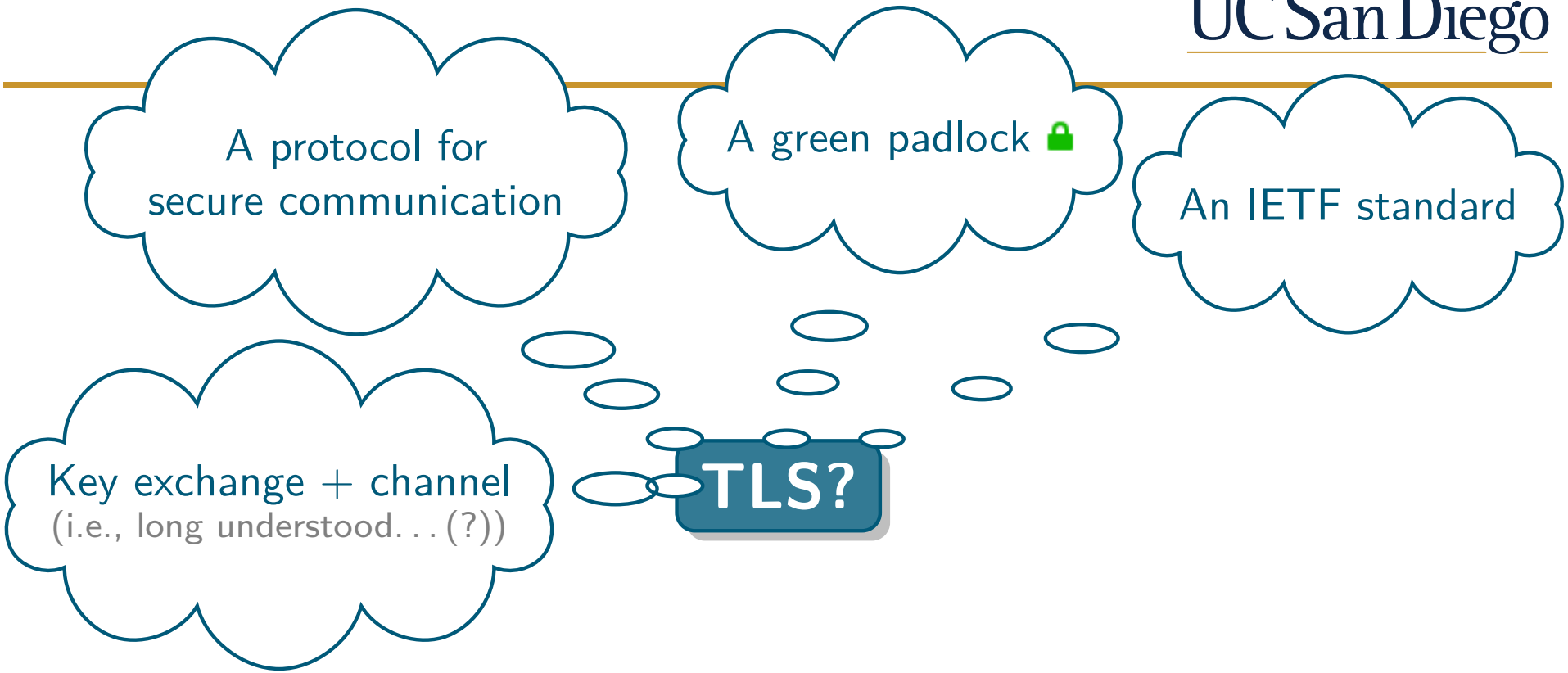


Record Protocol:

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

So What Is TLS?

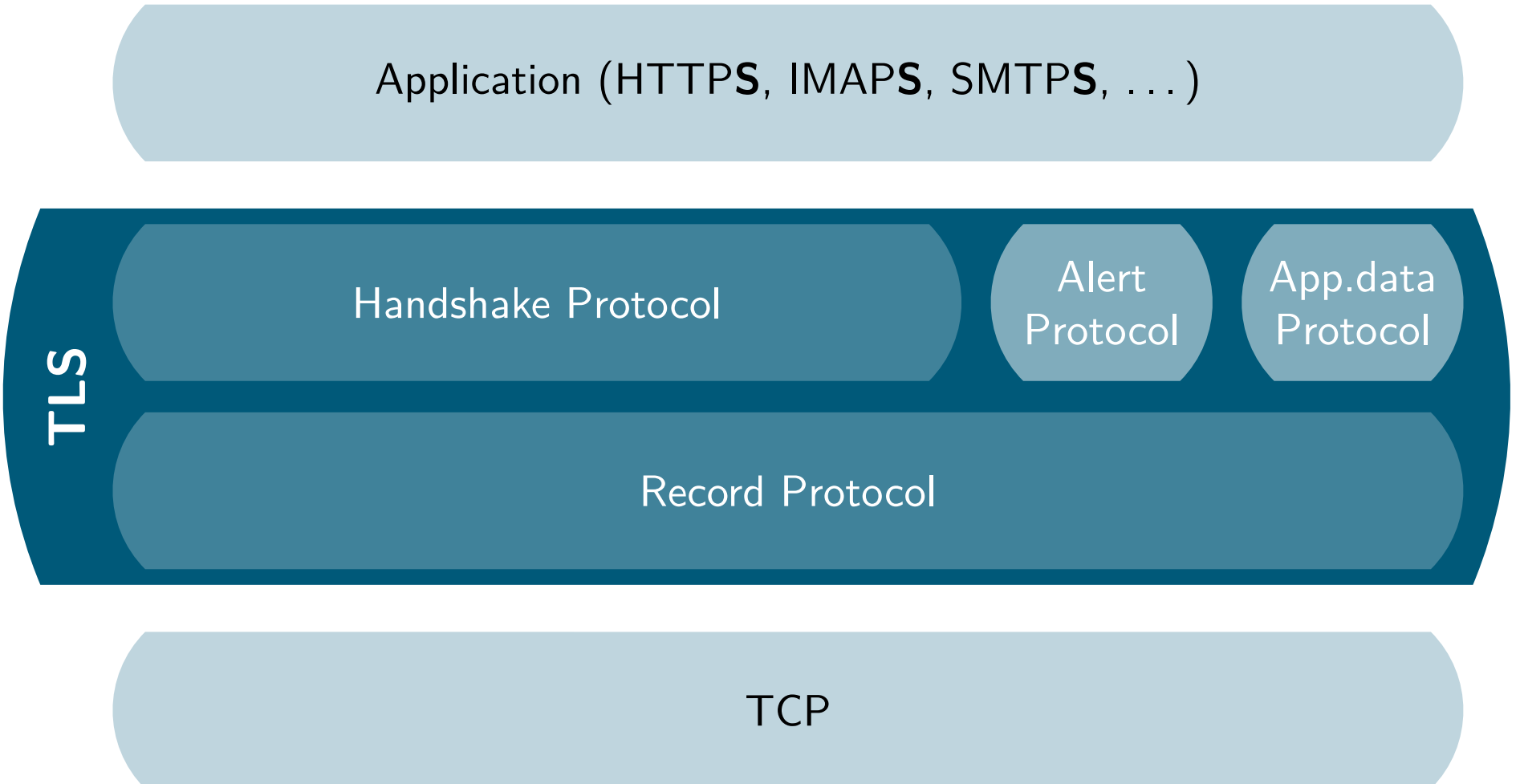
UC San Diego



The TLS Protocol

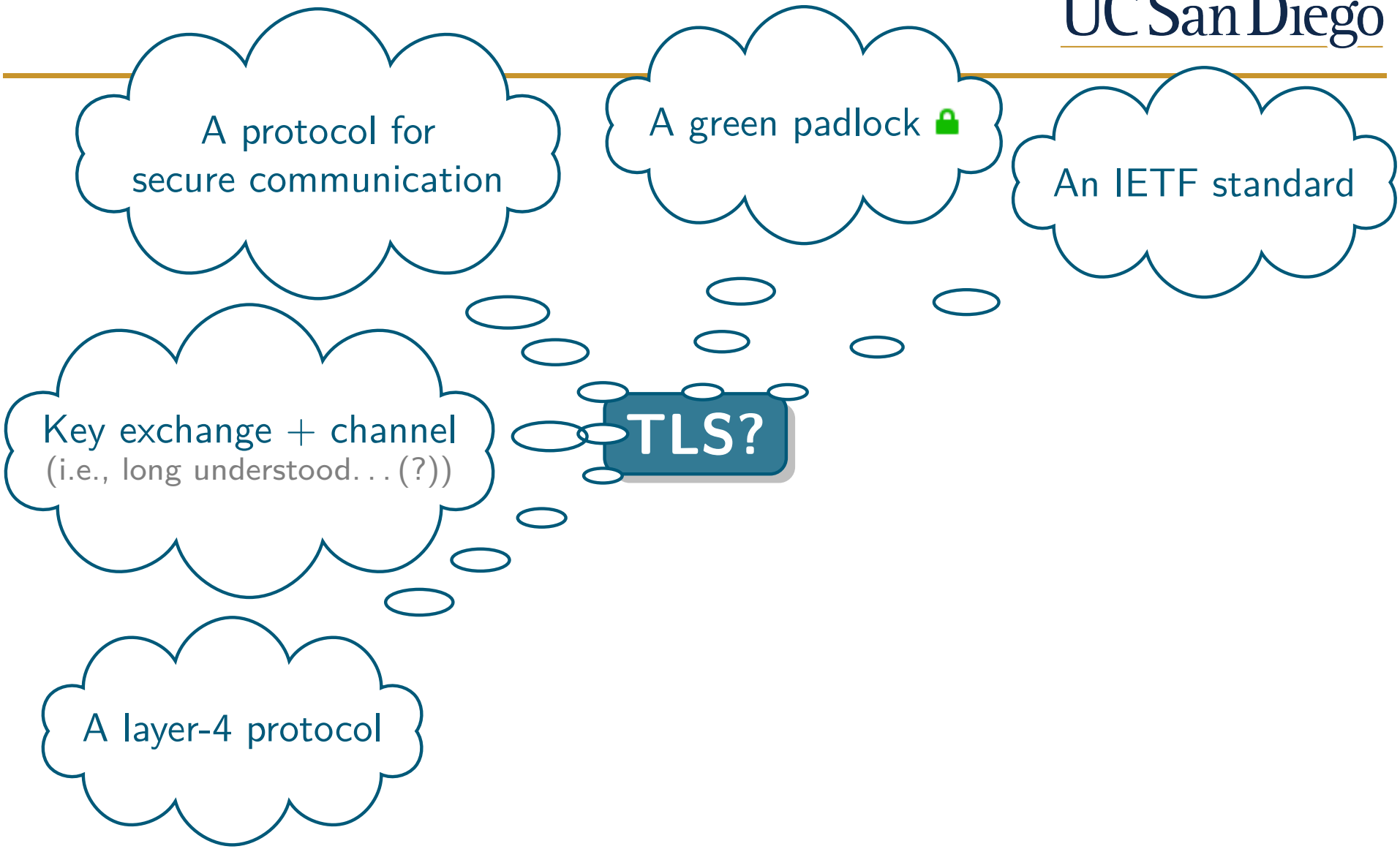
Architecture within Network Stack

UC San Diego



So What Is TLS?

UC San Diego



The TLS Protocol

Actors

UC San Diego

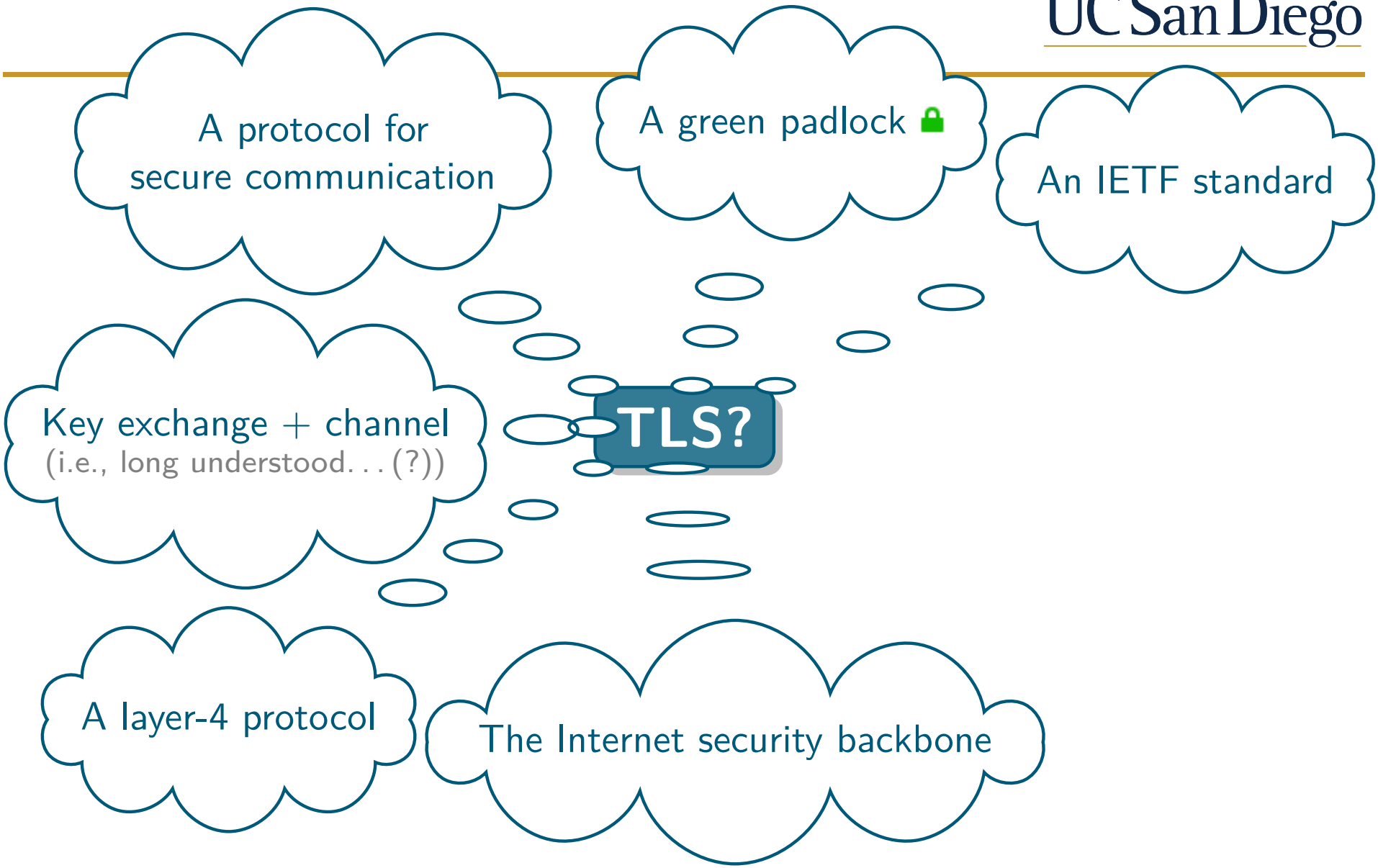
- with billions of users come **billions of devices** (for servers and clients)
- of all types, from *laptop* $\leftrightarrow$ *cloud* to *embedded device* $\leftrightarrow$ *smart home hub*

- running **various implementations** of TLS, in software and hardware
- from widely-used libraries (OpenSSL, Google's BoringSSL, ...)
  to small or even ad-hoc implementations

- authentication via **Certification Authorities** (100+ in standard browser)
- highly trusted and single-point-of-failure

So What Is TLS?

UC San Diego



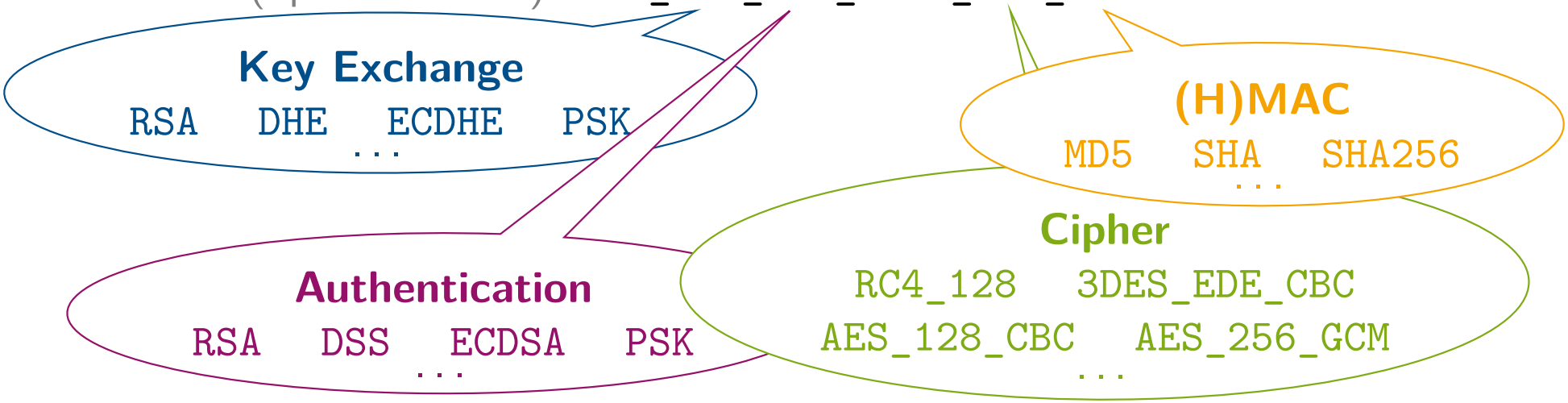
The TLS Protocol

Components

UC San Diego

- TLS is a **"self-negotiating"** protocol
- handshake first of all agrees on TLS version and cipher suite to use

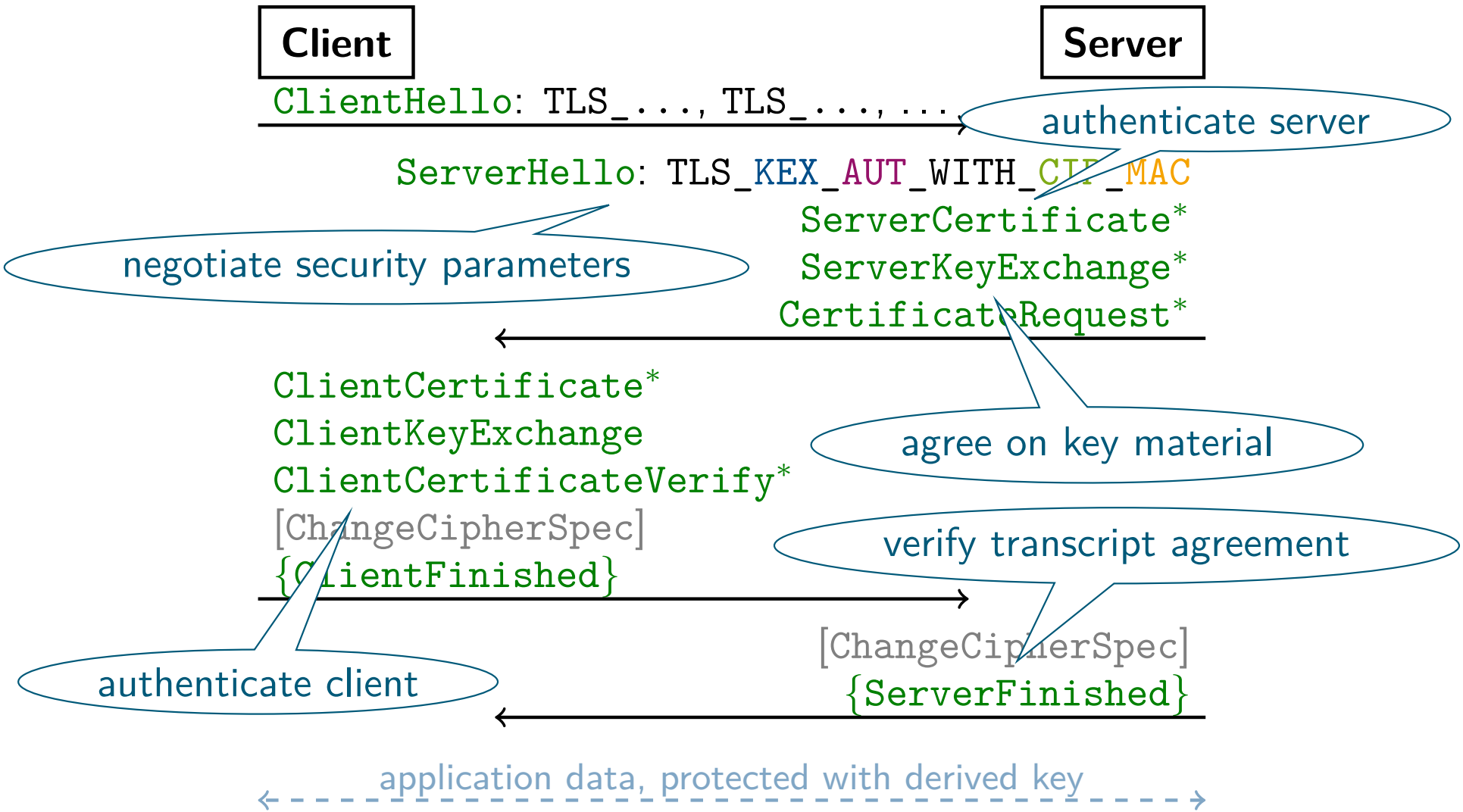
- **Cipher suites:** client proposes list, server picks
- fixes crypto algorithms to be used for that session
- format (up to TLS 1.2): `TLS_KEX_AUT_WITH_CIP_MAC`



The TLS Protocol

Handshake Protocol Structure

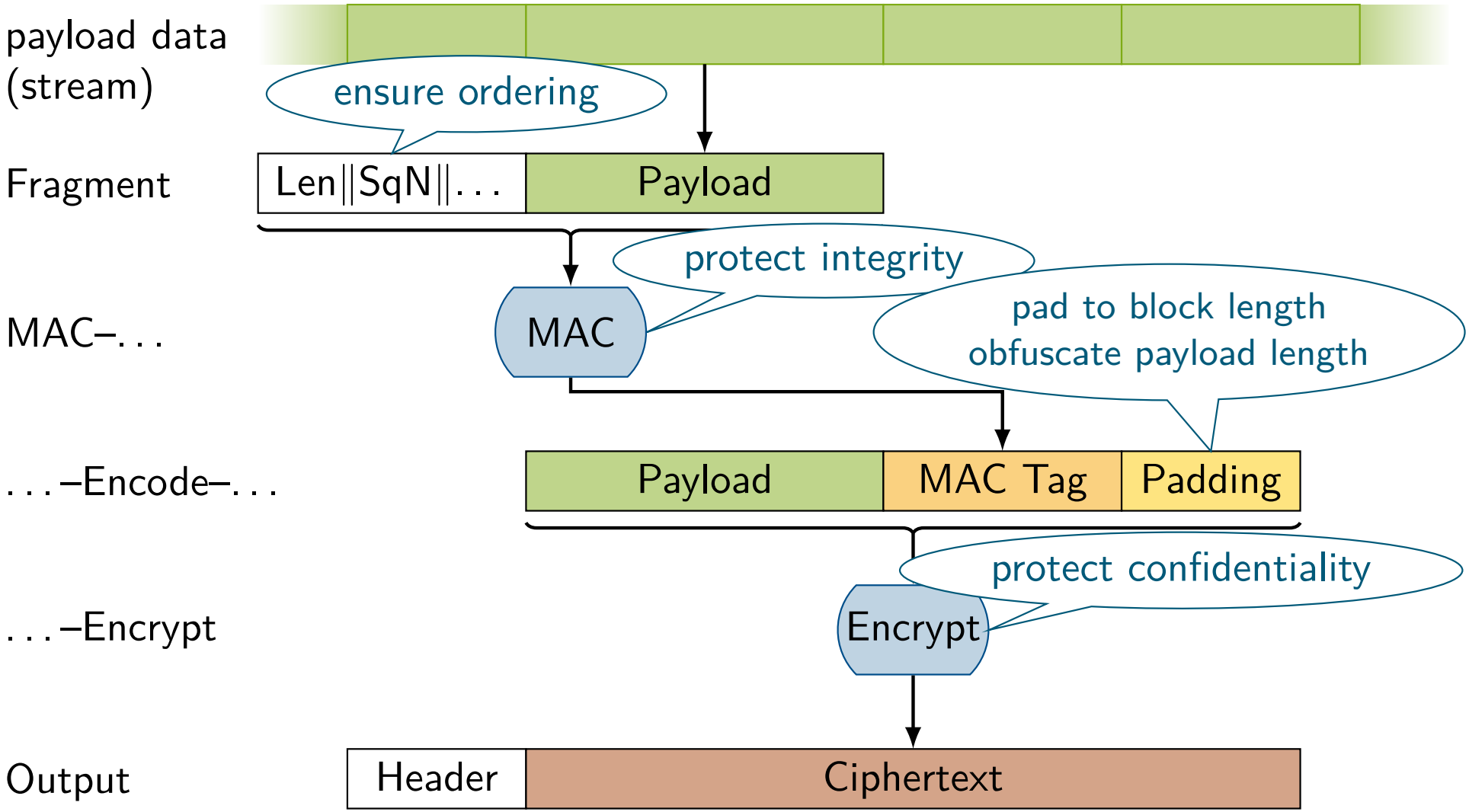
UC San Diego



The TLS Protocol

Record Protocol Structure

UC San Diego

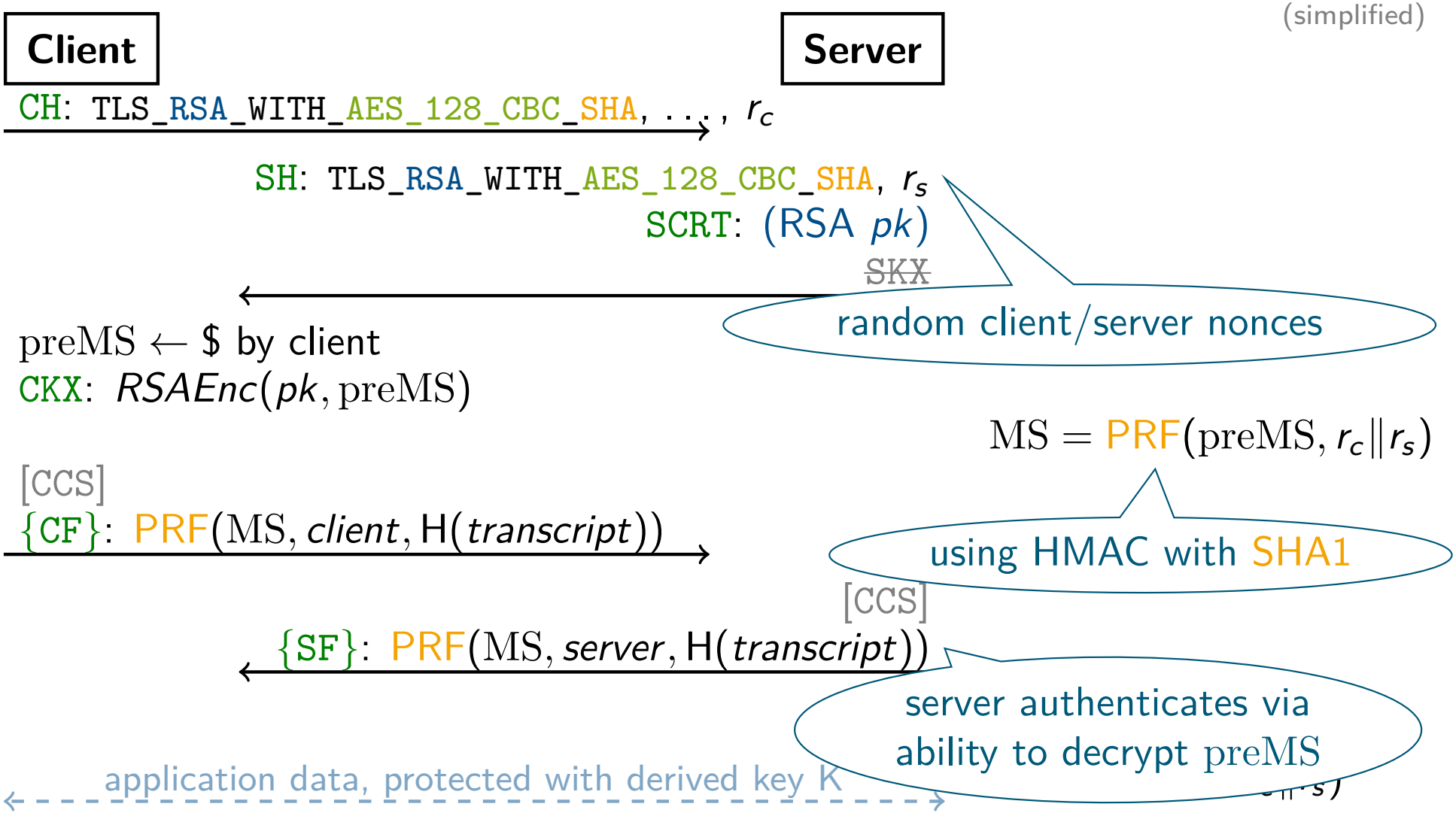


The TLS Protocol

Example: TLS_RSA_WITH_AES_128_CBC_SHA
— Handshake

UC San Diego

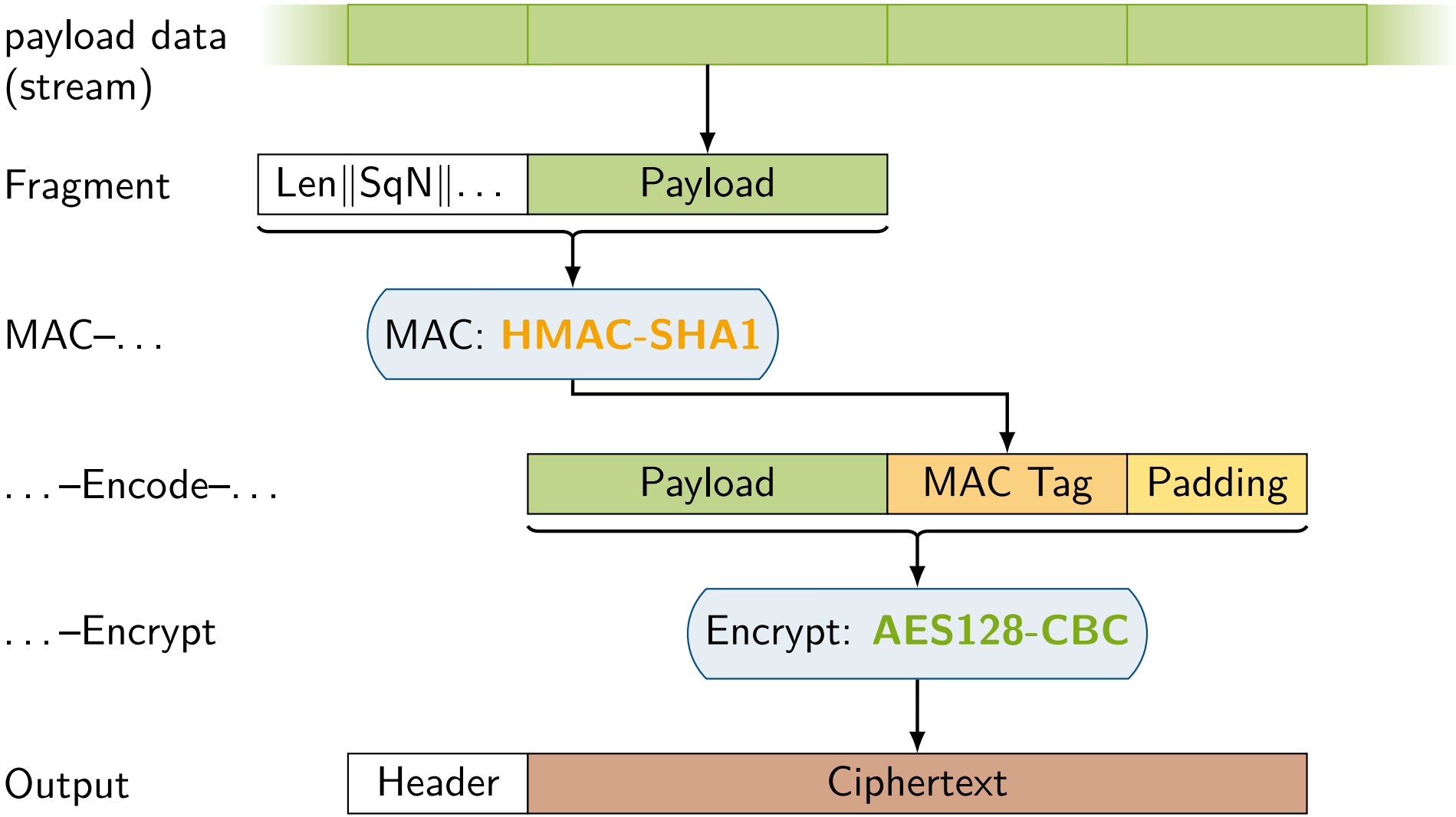
(simplified)



The TLS Protocol

Example: TLS_RSA_WITH_AES_128_CBC_SHA
— Record Protocol

UC San Diego

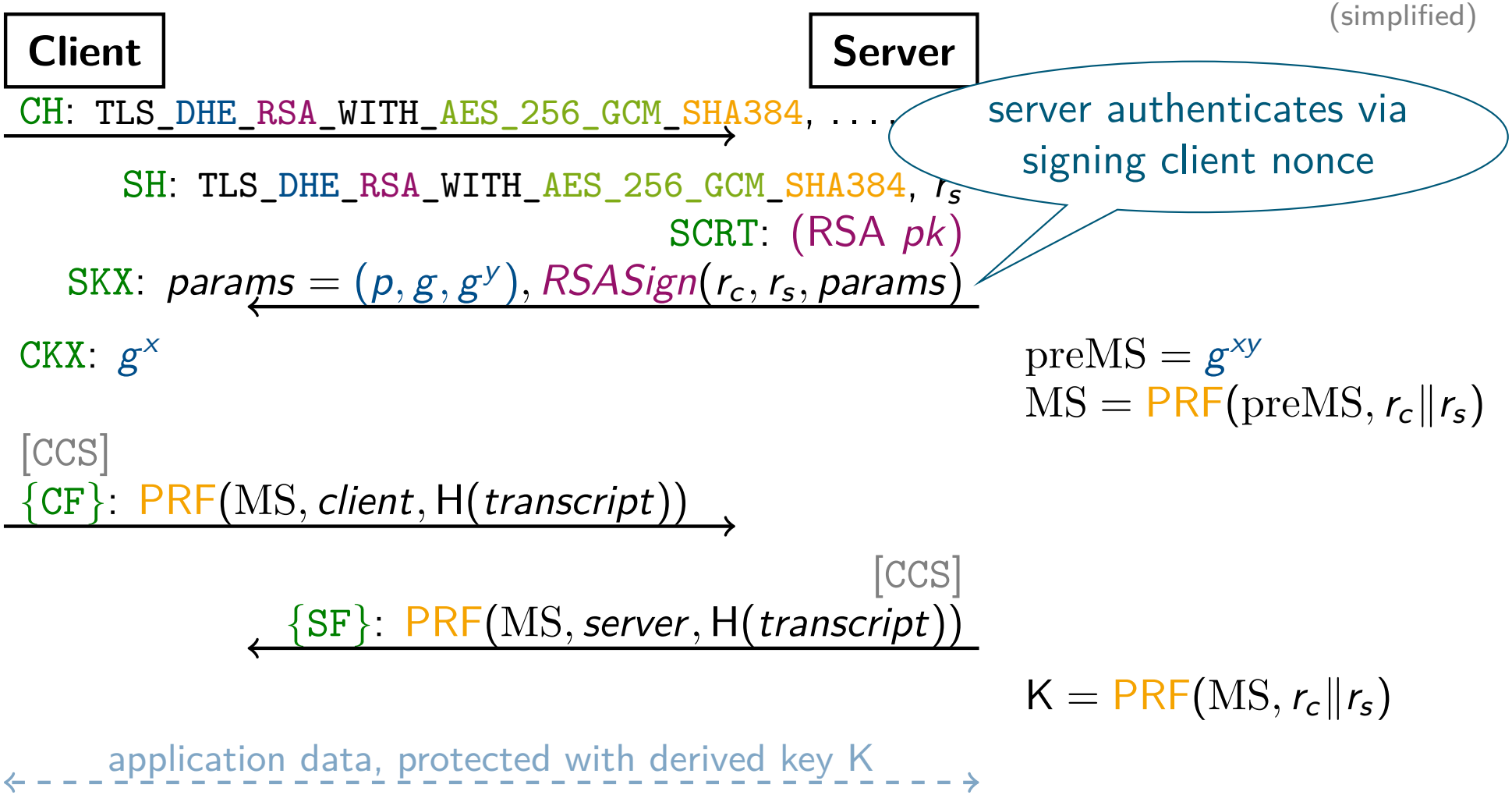


The TLS Protocol

Example: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
— Handshake

UC San Diego

(simplified)

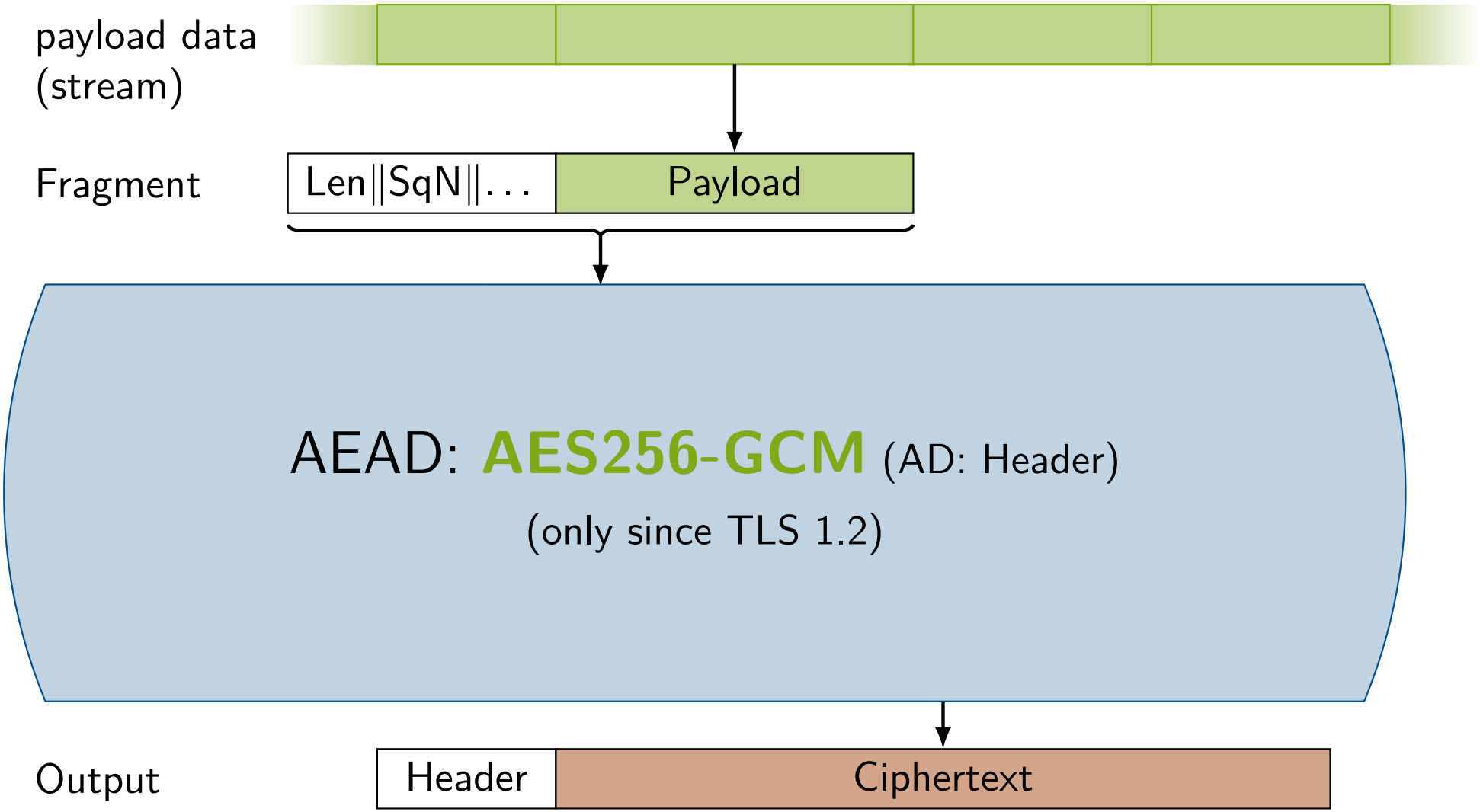


The TLS Protocol

Example: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

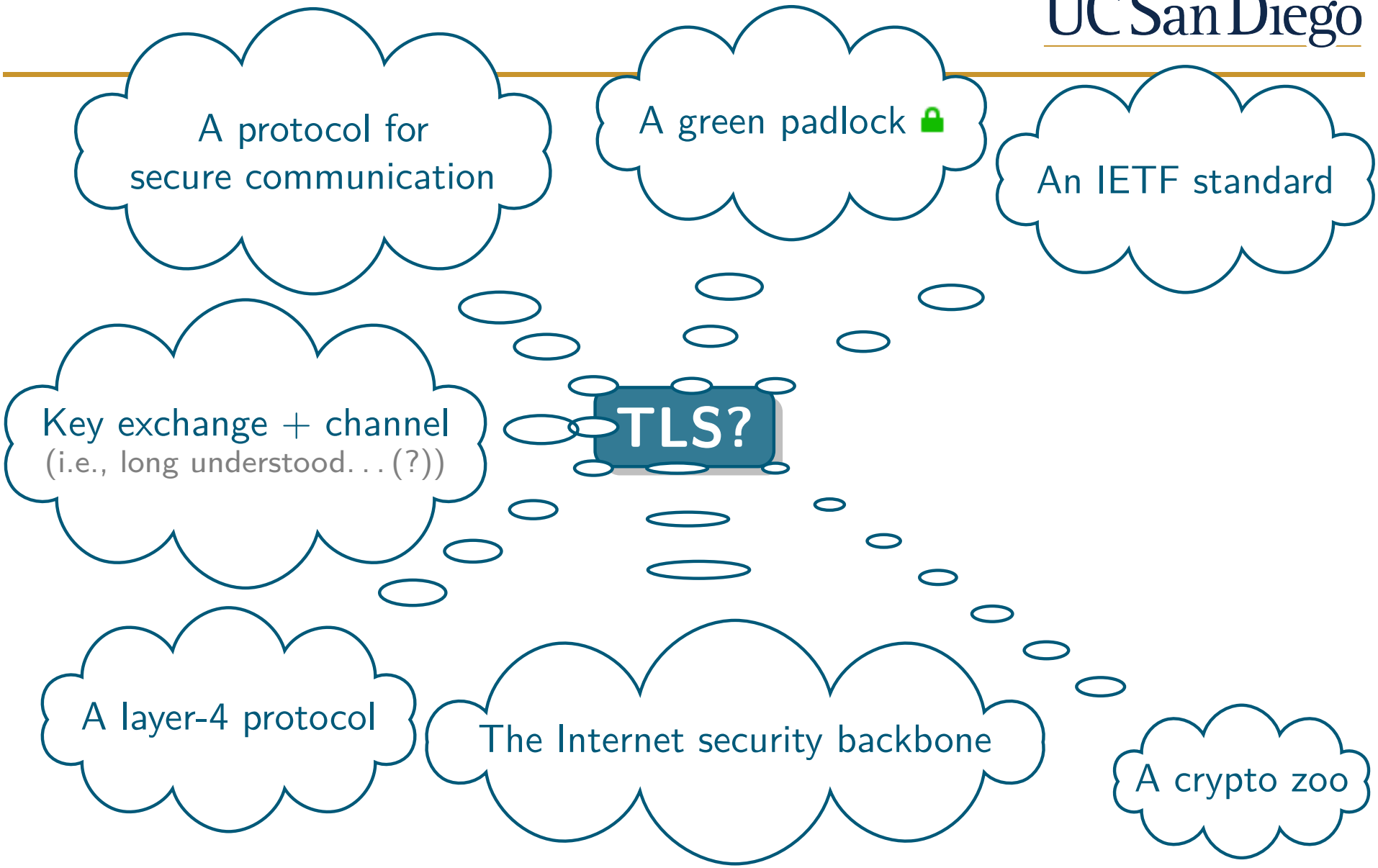
— Record Protocol

UC San Diego



So What Is TLS?

UC San Diego



The TLS Protocol

Resumption, Renegotiation, Extensions, ...

UC San Diego

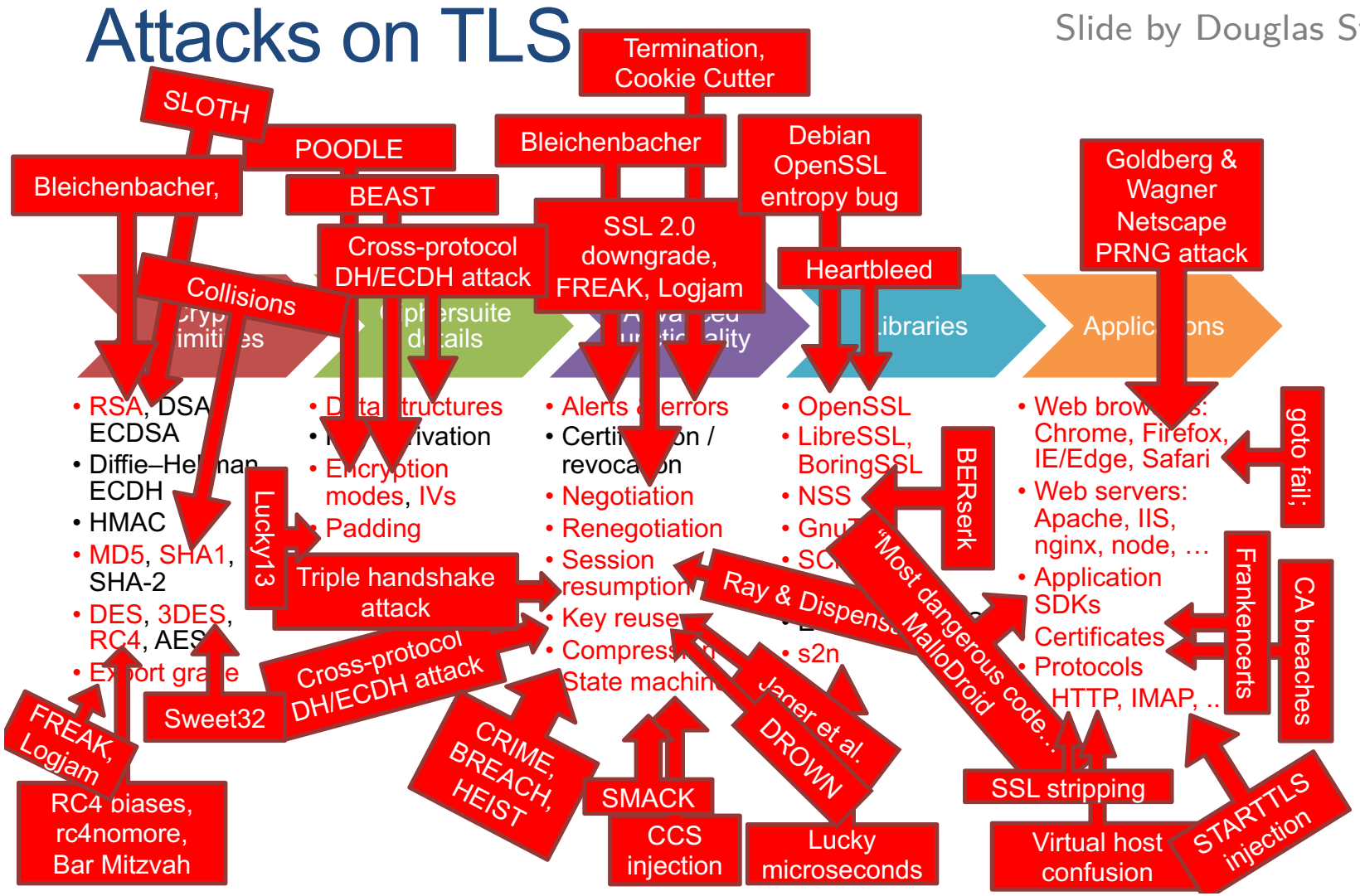
- **(Session) Resumption**
  - abbreviated handshake based on previously established shared secret
  - multiple and possibly parallel connections from same initial secret
- **Renegotiation**
  - change of cipher suite (and keys) within session, protected by Record Protocol
  - used, e.g., for late client authentication (hiding client's identity)
    or key renewal on long-lived connections without re-establishing connection
- **Extensions & Variants**
  - extensions specify additional functionality and/or security features
  - e.g.: AEAD cipher suites, ECC, connections to other protocols, ...
  - some mandatory to implement, some security-critical patches
  - DTLS: variant for TLS over UDP
- **TLS: complex protocol with many subtly interacting sub-components**

"What could possibly go wrong?" :-) (Kenny Paterson)

TLS Security Issues

Well. . .

UC San Diego

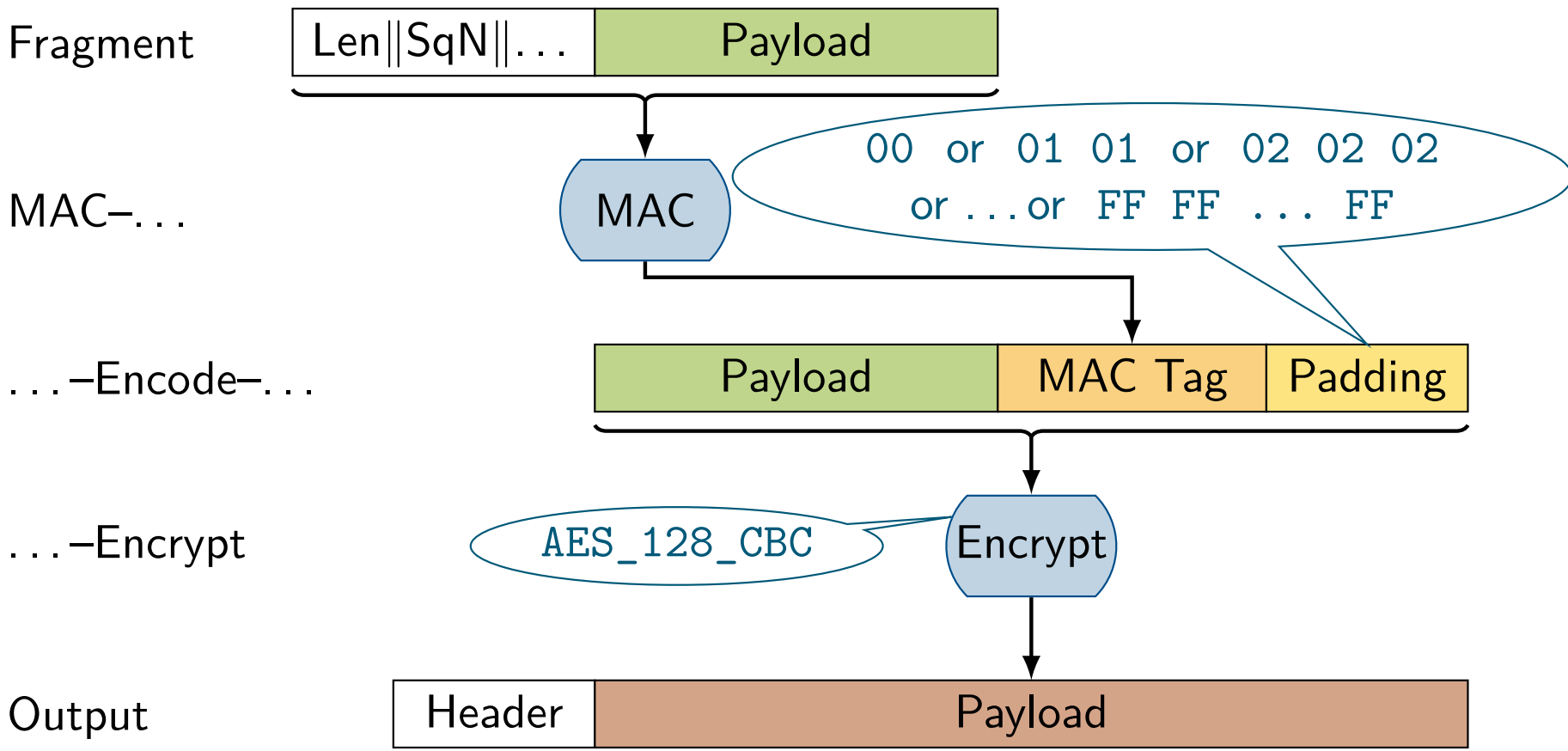


TLS Security Issues

@Crypto: MAC-Encode-Encrypt and Lucky13

UC San Diego

- core issue: (good) MAC –then– (good) Encrypt $\neq$ **CCA-secure** AE [BN00]



TLS Security Issues

@Crypto: MAC-Encode-Encrypt and Lucky13

UC San Diego

- core issue: (good) MAC –then– (good) Encrypt $\neq$ **CCA-secure** AE [BN00]
- **MAC–then–AES-CBC Decryption**
  - decrypt ciphertext to obtain Payload $\|$ MAC Tag $\|$ Padding
  - remove padding — what if padding is incorrect?
  - check MAC
- A padding oracle
  - in a modified ciphertext, either the padding check fails. . .
  - . . . or the MAC check fails
  - if the two are distinguishable: padding oracle
  - can lift a padding oracle to a **decryption oracle** [Vau02] (conditions apply)
- instead of switch to CCA-secure Enc-then-MAC, TLS tried hiding error signal
  - "compute MAC w/ zero padding"
  - "leaves a [non-exploitable] small timing channel"
  - **Lucky13** [AP13]: HMAC timing difference still big enough
  - really need constant time—which is extremely difficult!

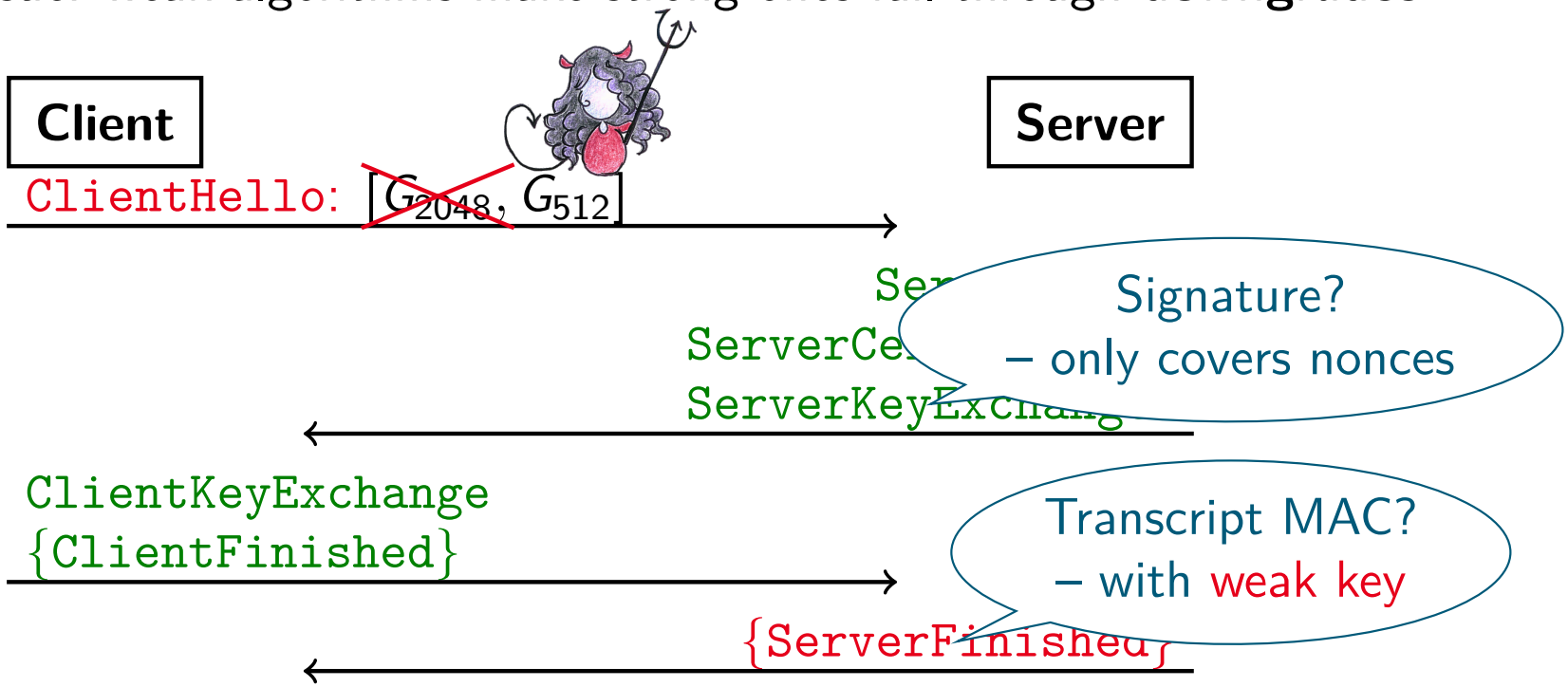
TLS Security Issues

@Protocol Design: Weak DH Negotiation and Logjam

Mark
Feb 14

UC San Diego

- core issue: weak algorithms make strong ones fail through **downgrades**



- **Logjam** [ABD+15]: How Diffie–Hellman Fails in Practice
  - server impersonation through support of (also) weak DH groups

drawings by *Giorgia Azzurra Marson*

TLS Security Issues

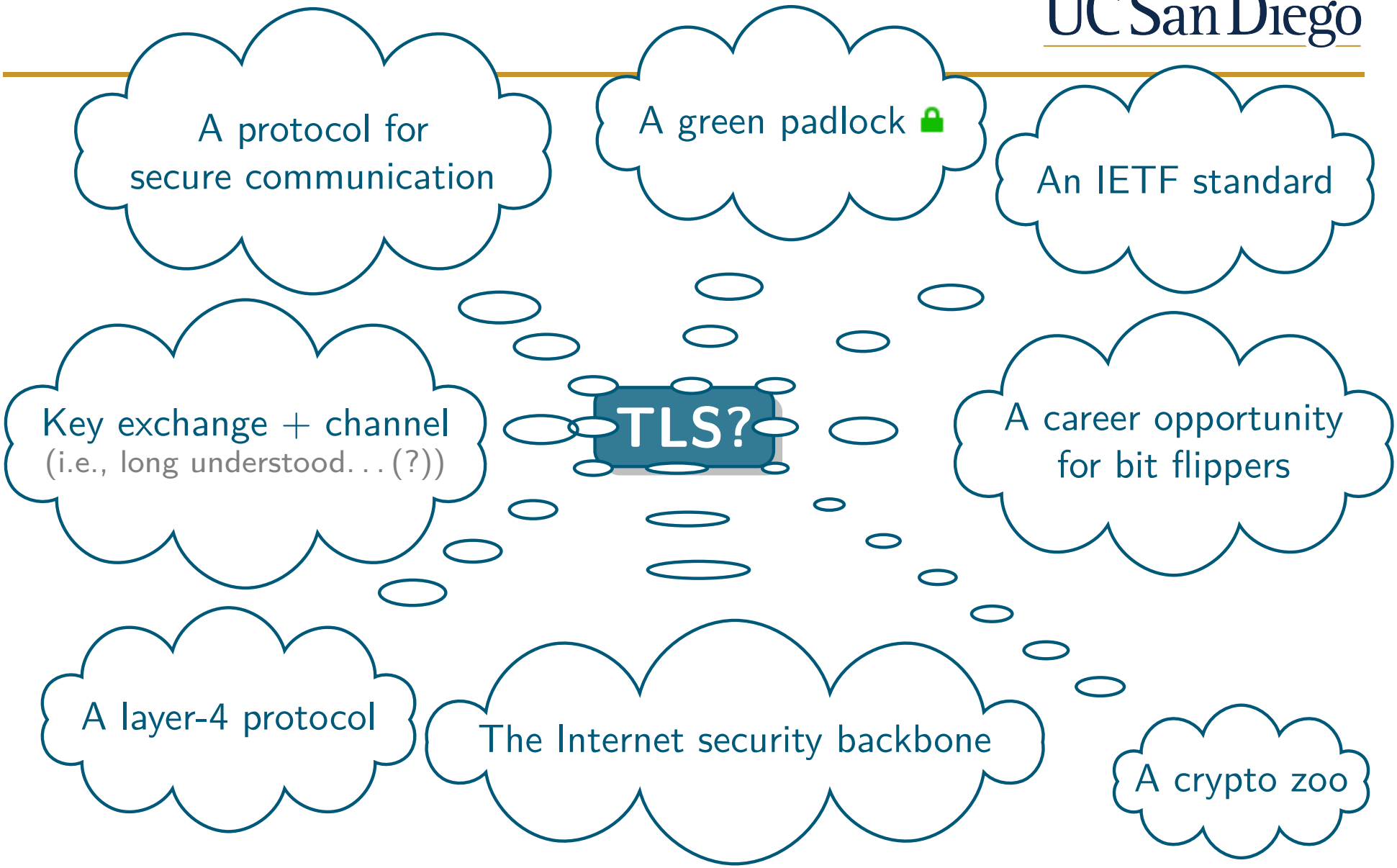
@Implementation: Buffers and Heartbleed

UC San Diego

- core issue: **buffer over-read** in OpenSSL
- **Heartbeat** extension (RFC 6520)
  - client sends "ping back those 4 bytes: `00 01 02 03`"
  - server responds "`00 01 02 03`"
- **Heartbleed** attack [Hea]
  - client sends "ping back those **16 Kbytes**: `00 01 02 03`"
  - server responds "`00 01 02 03 ...<memory dump>`"
  - possibly including sensitive data like server private key etc.
- high severity & public attention — and a catchy logo

So What Is TLS?

UC San Diego

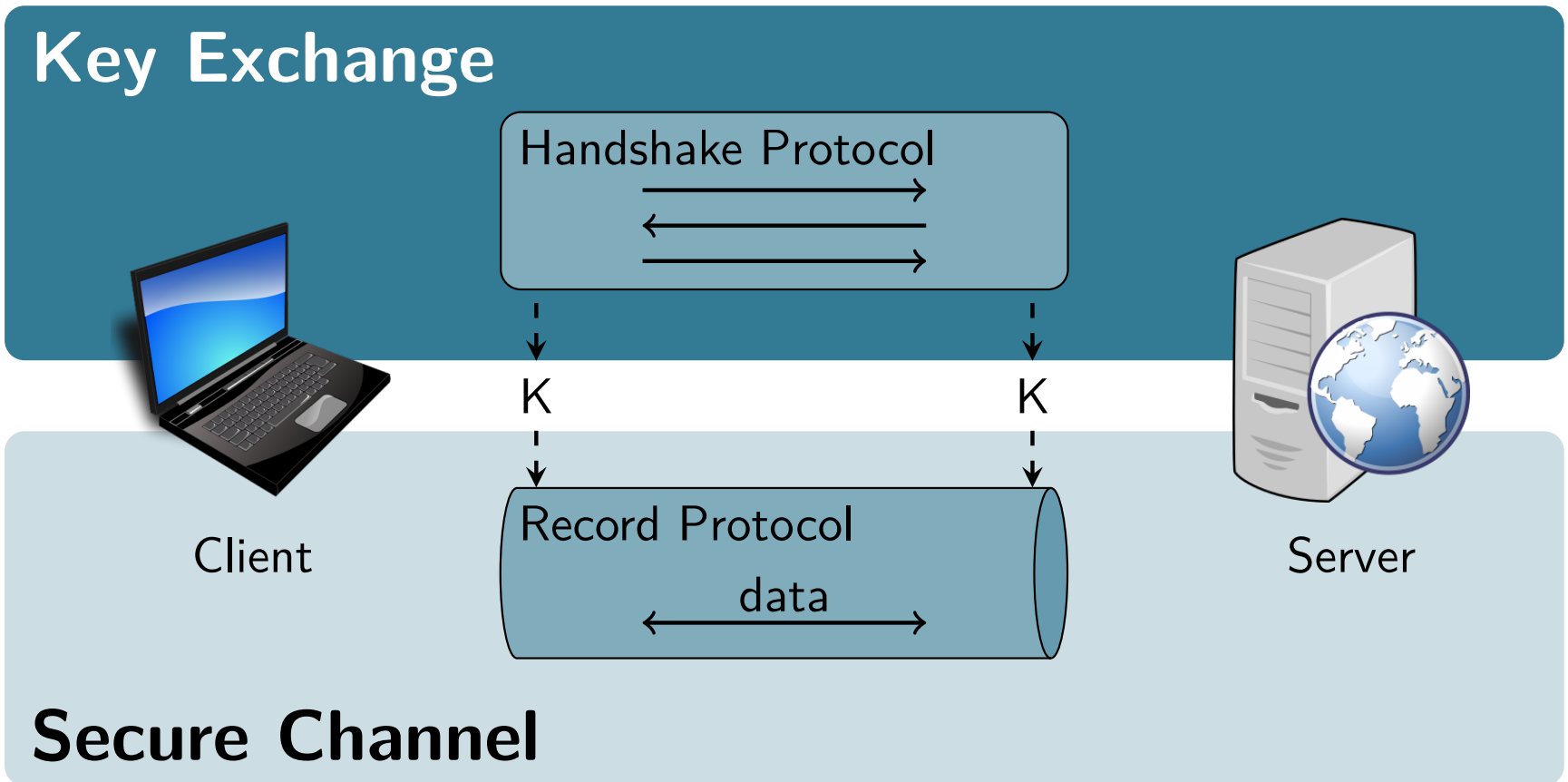


Cryptographic Background

The TLS Protocol Components

(Again, overly simplified)

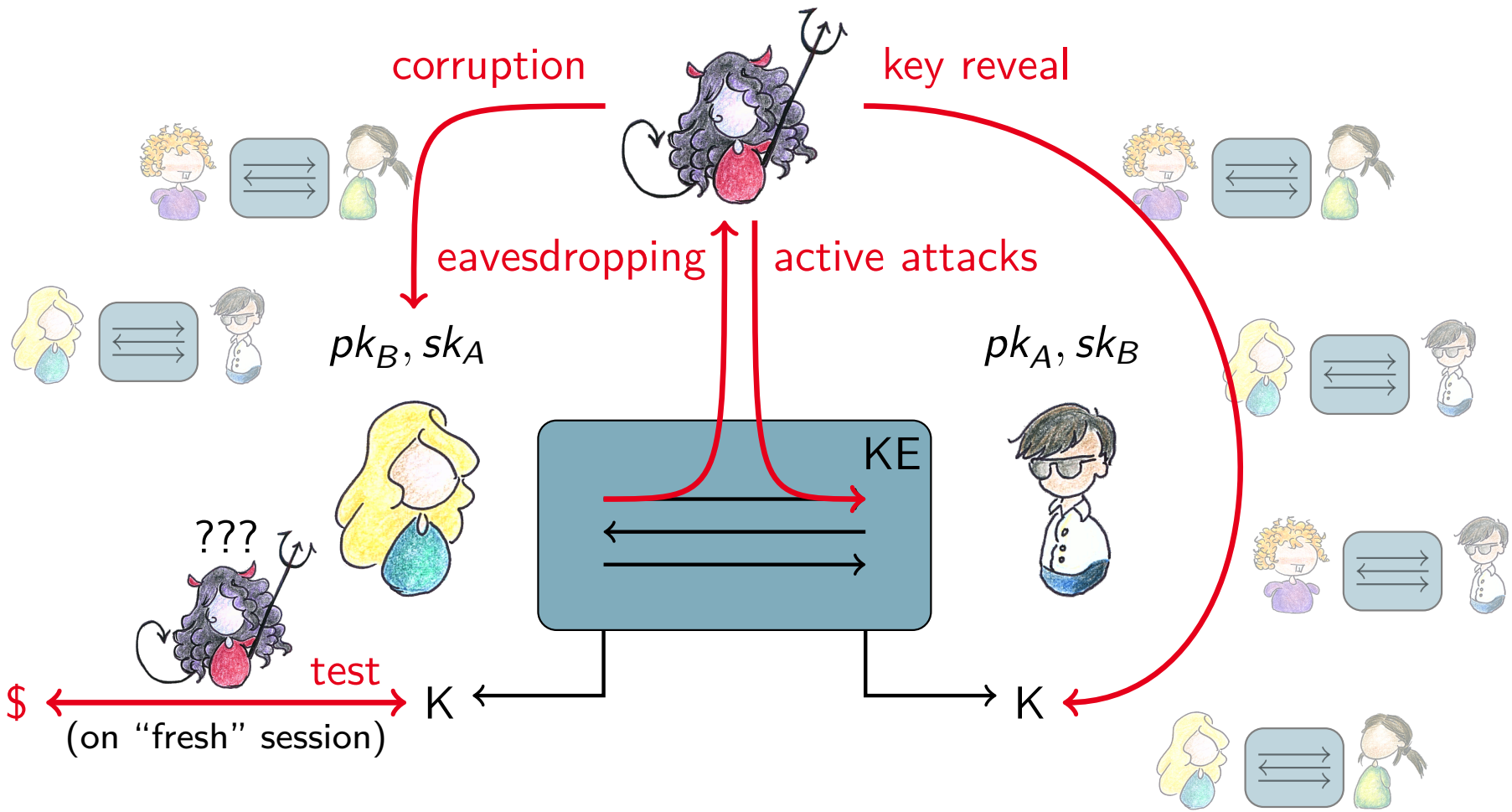
UC San Diego



Key Exchange Security

Bellare, Rogaway 1993 [BR94]

UC San Diego

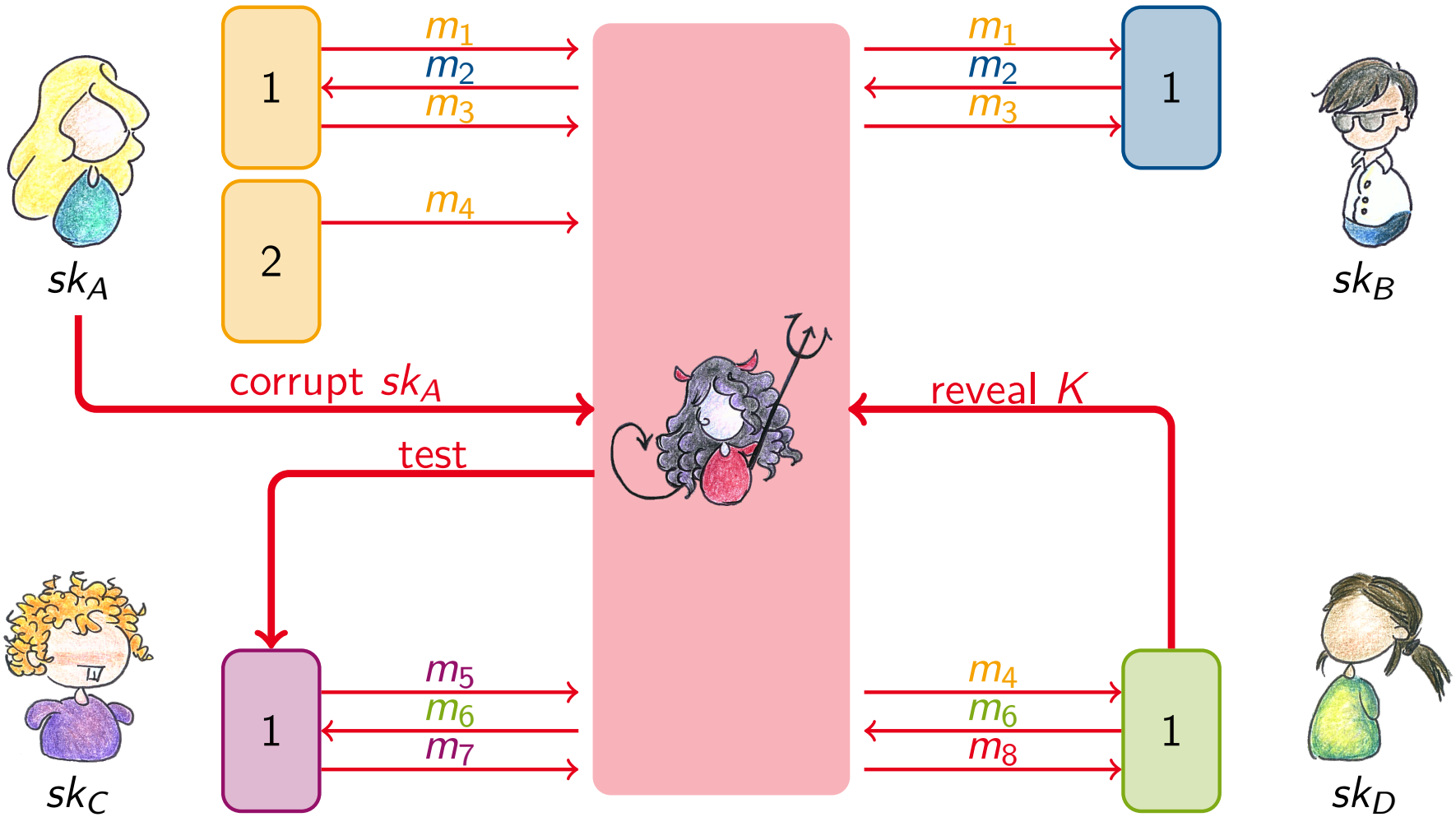


Key Exchange Security

The communication/security model

UC San Diego

$$\mathsf{KE}(id, pid, sk_{id}, pk_{pid}, transcript, \dots) \mapsto (msg, status, K, \dots)$$



Key Exchange Security

What we want

UC San Diego

Authentication

"Adversary can just relay message / act as a wire"

- relaying is the only way adversary can make sessions accept
- session accepts $\Longrightarrow \exists$ other session with matching transcript

Key Secrecy

"uncompromised session keys look random to adversary"

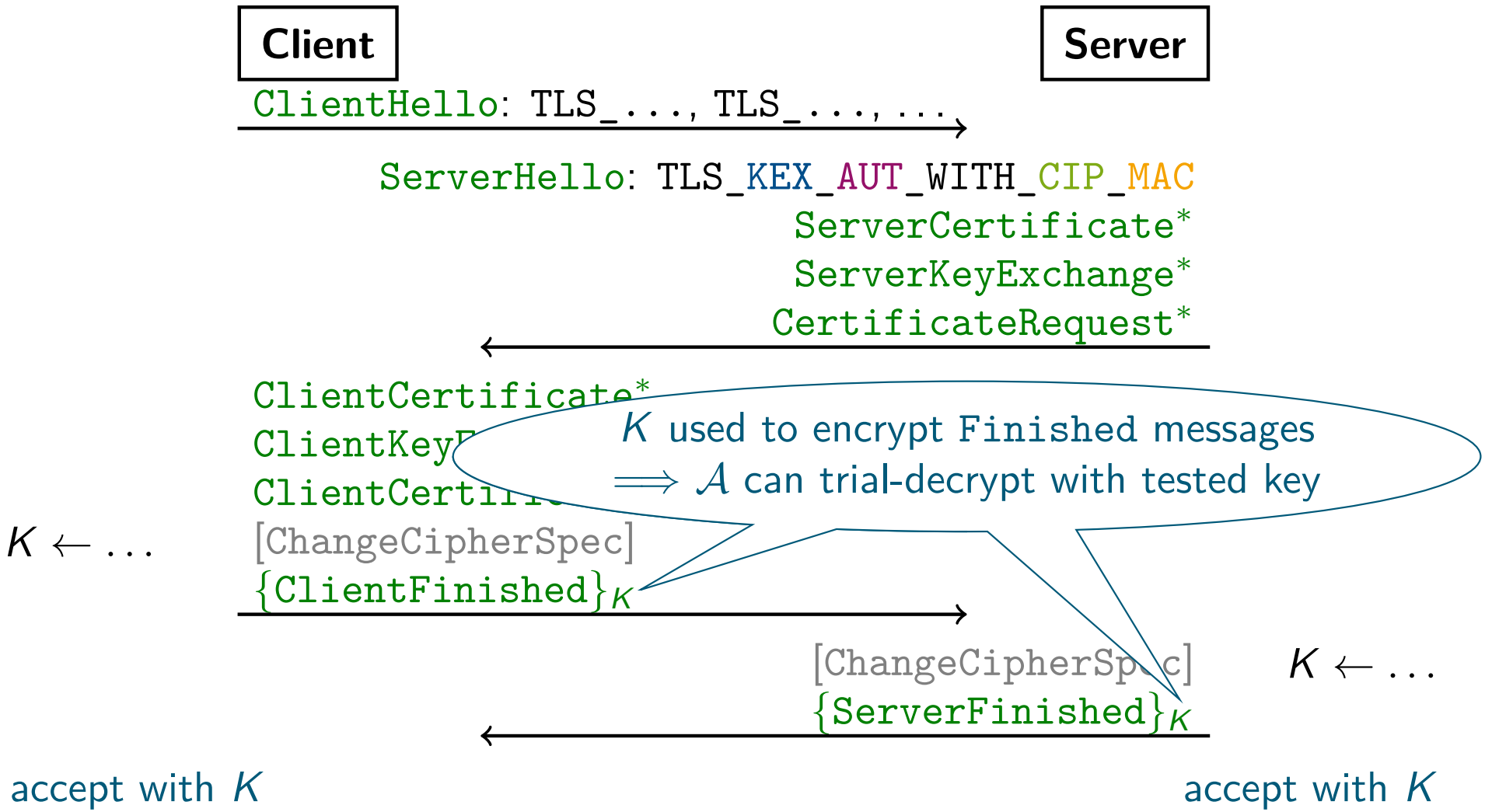
- Test query with hidden bit b
- outputs real K if $b = 0$, else random key $\leftarrow\!\!\$ \{0,1\}^{|K|}$
- adversary is only allowed to issue Test on "fresh"/uncompromised sessions
- $\Pr[\mathcal{A} \Rightarrow b] - \frac{1}{2} \approx \text{negl}.$

The TLS (1.2) Handshake

Why is it not BR-secure?

Joseph
Feb 7

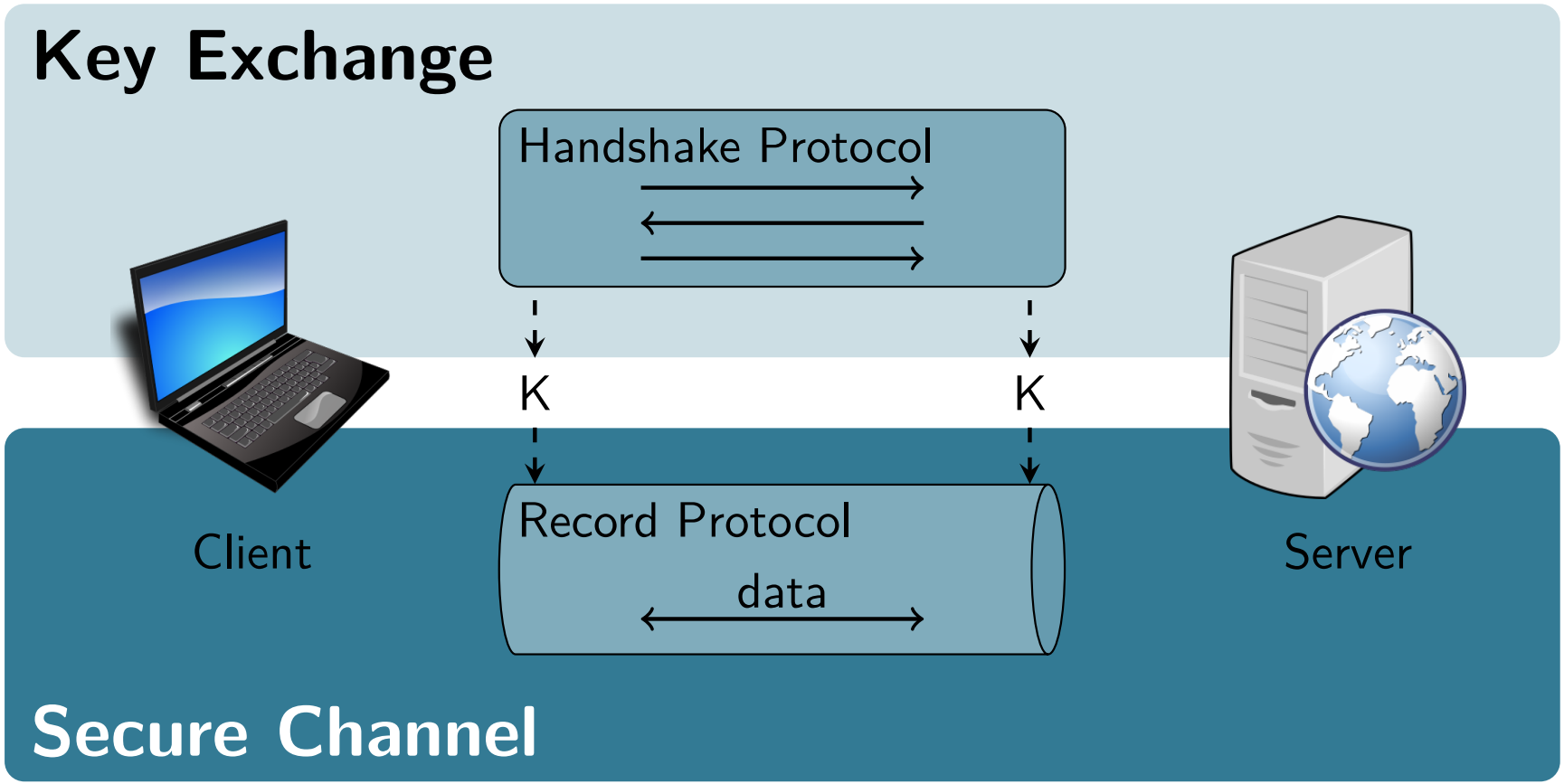
UC San Diego



The TLS Protocol Components

(Again, overly simplified)

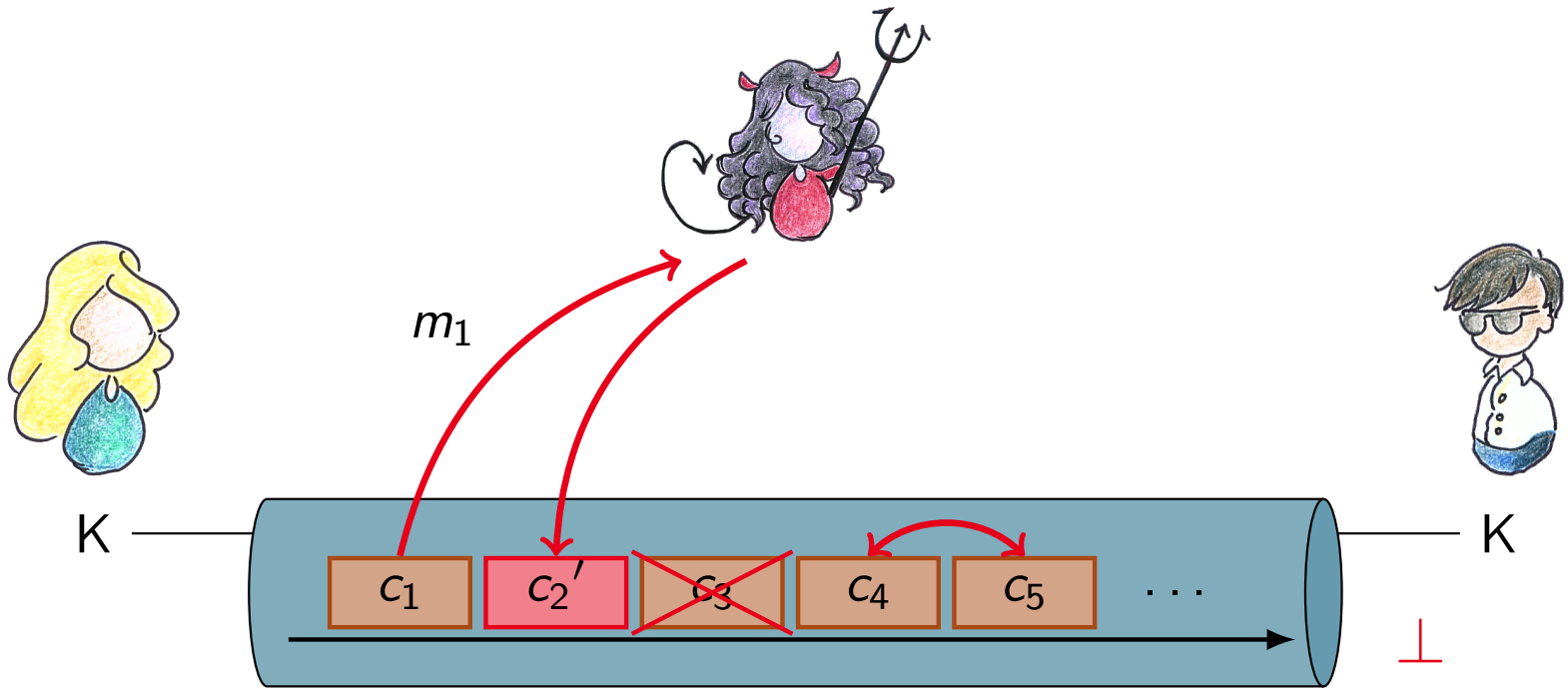
UC San Diego



Channel Security

Bellare, Kohno, Namprempre 2002 [BKN02]

UC San Diego



IND-sfCPA (passive confidentiality)

IND-sfCCA (active confidentiality)

INT-sfPTXT (plaintext integrity)

INT-sfCTXT (ciphertext integrity)

Channel Security

Security notions [BKN02]

UC San Diego

IND-sfCCA Security

$\mathrm{Expt}^{\text{IND-sfCCA}}_{\mathcal{E},\mathcal{A}}(1^\lambda)$:

1. $K \leftarrow\!\!\$\ \mathrm{KGen}(1^\lambda)$, $b \leftarrow\!\!\$\ \{0,1\}$
2. $i, j \leftarrow 0$, sync $\leftarrow 1$
3. $b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda)^{\mathcal{O}_{\mathrm{LoR}}(K,\cdot,\cdot),\mathcal{O}_{\mathrm{Dec}}(K,\cdot)}$
4. return $b = b'$

$\mathcal{O}_{\mathrm{LoR}}(K, m_0, m_1)$: // $|m_0| = |m_1|$

1. $i \leftarrow i + 1$
2. return $c_i \leftarrow \mathrm{Enc}(K, m_b)$

$\mathcal{O}_{\mathrm{Dec}}(K, c)$:

1. $j \leftarrow j + 1$
2. if $j > i$ or $c \neq c_j$: sync $\leftarrow 0$
3. if sync $= 0$: return $m \leftarrow \mathrm{Dec}(K, c)$

INT-sfCTXT Security

$\mathrm{Expt}^{\text{INT-sfCTXT}}_{\mathcal{E},\mathcal{A}}(1^\lambda)$:

1. $K \leftarrow\!\!\$\ \mathrm{KGen}(1^\lambda)$
2. $i, j \leftarrow 0$, sync $\leftarrow 1$, win $\leftarrow 0$
3. $\mathcal{A}(1^\lambda)^{\mathcal{O}_{\mathrm{Enc}}(K,\cdot),\mathcal{O}_{\mathrm{Dec}}(K,\cdot)}$
4. return win

$\mathcal{O}_{\mathrm{Enc}}(K, m)$:

1. $i \leftarrow i + 1$
2. return $c_i \leftarrow \mathrm{Enc}(K, m)$

$\mathcal{O}_{\mathrm{Dec}}(K, c)$:

1. $j \leftarrow j + 1$, $m \leftarrow \mathrm{Dec}(K, c)$
2. if $j > i$ or $c \neq c_j$: sync $\leftarrow 0$
3. if sync $= 0$ and $m \neq \perp$: win $\leftarrow 1$

Channel Security

Still work in progress. . .

UC San Diego

- Many more aspects to consider for secure channels in practice:
  - length-hiding / padding [PRS11]
  - fragmentation of ciphertexts [BDPS12]
  - stream-based data [FGMP15]
  - bidirectionality [MP17]
  - multiple keys [GM17]
  - multiplexing [PS18]
  - . . .

- despite being intuitively simple, the crypto community still hasn't really conclusively ventilated the question: What is a secure channel protocol?

Thank You!

TLS ≤1.2

Jan 17	**TLS intro** [TLS12] **& crypto backround** [BR94,BKN02]	Felix
Jan 24	**Lucky 13** [AP13]	Nicholas
Jan 31	no seminar	-
Feb 7	**The ACCE model** [JKSS12,KPW13]	Joseph
Feb 14	**Logjam** [ABD+15]	Mark

TLS 1.3

Feb 21	**TLS 1.3** [TLS13] **& some security models** [FG17,GM17]	Felix
Feb 28	**Multiplexing channels** [PS18]	Vivek
Mar 7	**Symbolic Tamarin analysis** [CHH+17]	Baiyu
Mar 14	**Downgrade resilience** [BBF+16]	Ruth

References I

UC San Diego

[ABD+15] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Z. Béguelin, and P. Zimmermann. "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice". In: *ACM CCS 15*. Ed. by I. Ray, N. Li, and C. Kruegel: ACM Press, Oct. 2015, pp. 5–17.

[AP13] N. J. AlFardan and K. G. Paterson. "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols". In: *2013 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2013, pp. 526–540.

[BKN02] M. Bellare, T. Kohno, and C. Namprempre. "Authenticated Encryption in SSH: Provably Fixing The SSH Binary Packet Protocol". In: *ACM CCS 02*. Ed. by V. Atluri. ACM Press, Nov. 2002, pp. 1–11.

[BN00] M. Bellare and C. Namprempre. "Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm". In: *ASIACRYPT 2000*. Ed. by T. Okamoto. Vol. 1976. LNCS. Springer, Heidelberg, Dec. 2000, pp. 531–545.

[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.

[BBF+16] K. Bhargavan, C. Brzuska, C. Fournet, M. Green, M. Kohlweiss, and S. Z. Béguelin. "Downgrade Resilience in Key-Exchange Protocols". In: *2016 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2016, pp. 506–525.

[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.

References II

UC San Diego

[CHH+17] C. Cremers, M. Horvat, J. Hoyland, S. Scott, and T. van der Merwe. "A Comprehensive Symbolic Analysis of TLS 1.3". In: *ACM CCS 17*. Ed. by B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu. ACM Press, 2017, pp. 1773–1788.

[TLS12] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Aug. 2008.

[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.

[Hea] *Heartbleed bug*. `http://heartbleed.com/`. 2014.

[JKSS12] T. Jager, F. Kohlar, S. Schäge, and J. Schwenk. "On the Security of TLS-DHE in the Standard Model". In: *CRYPTO 2012*. Ed. by R. Safavi-Naini and R. Canetti. Vol. 7417. LNCS. Springer, Heidelberg, Aug. 2012, pp. 273–293.

[KPW13] H. Krawczyk, K. G. Paterson, and H. Wee. "On the Security of the TLS Protocol: A Systematic Analysis". In: *CRYPTO 2013, Part I*. Ed. by R. Canetti and J. A. Garay. Vol. 8042. LNCS. Springer, Heidelberg, Aug. 2013, pp. 429–448.

References III

UC San Diego

[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 18*. Ed. by D. Lie, M. Mannan, M. Backes, and X. Wang. ACM Press, Oct. 2018, pp. 1415–1428.

[TLS13] E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Aug. 2018.

[Vau02] S. Vaudenay. "Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS..." In: *EUROCRYPT 2002*. Ed. by L. R. Knudsen. Vol. 2332. LNCS. Springer, Heidelberg, 2002, pp. 534–546.