On the Security of TLS Renegotiation

Florian Giesen (Ruhr University Bochum)
Florian Kohlar (Ruhr University Bochum)
Douglas Stebila (Queensland University of Technology)

RUB

2013/11/05
CCS Berlin

Supported by:

Australian Technology Network–German Academic Exchange Service (ATN-DAAD) Joint Research Cooperation Scheme

Australian Research Council Discovery Project

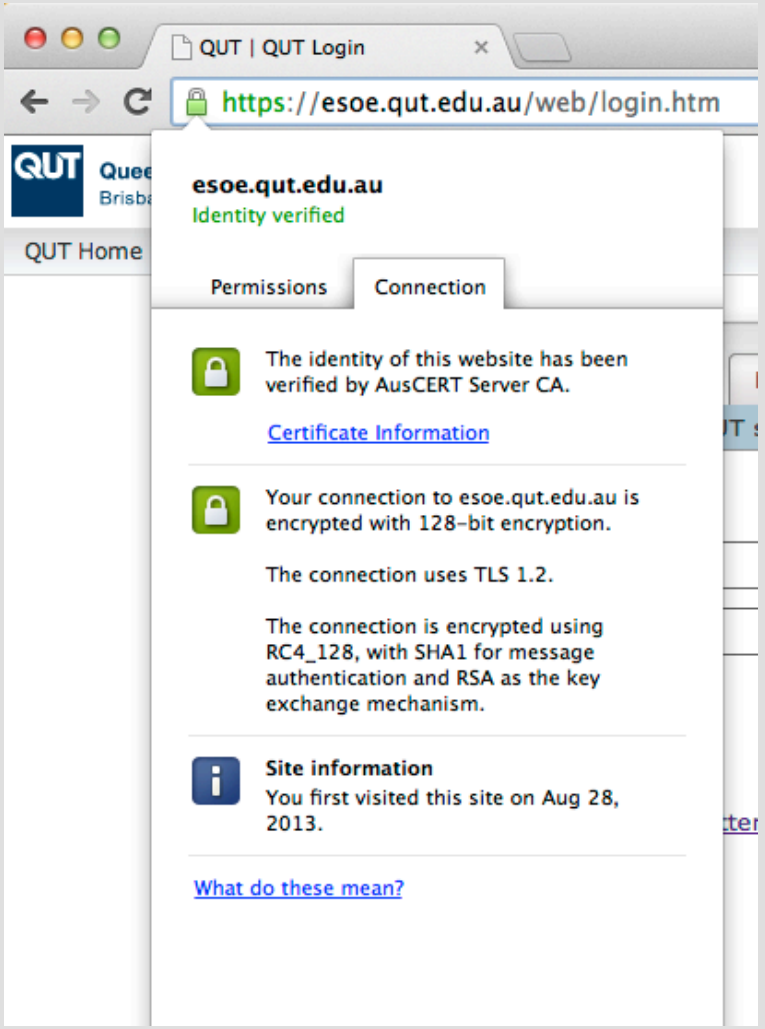
European Network of Excellence in Cryptology II (ECRYPT II)

Agenda

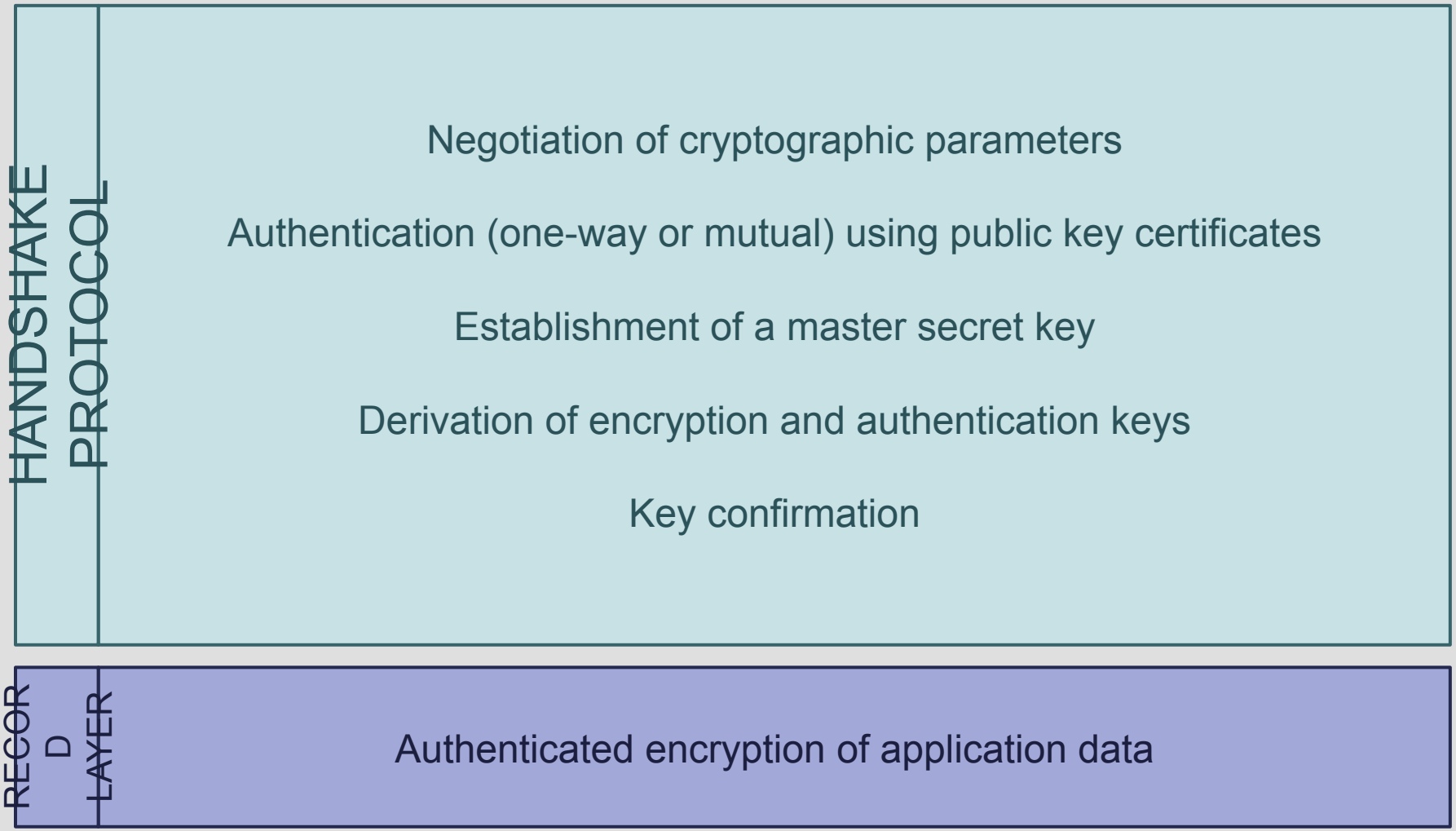
- What is TLS?
- TLS Renegotiation
- Renegotiation security

TLS (Transport Layer Security) protocol
a.k.a. SSL (Secure Sockets Layer)

- Protects http, ftp, smtp, imap, ...
- The **most important cryptographic protocol on the Internet** — used to secure billions of connections every day.



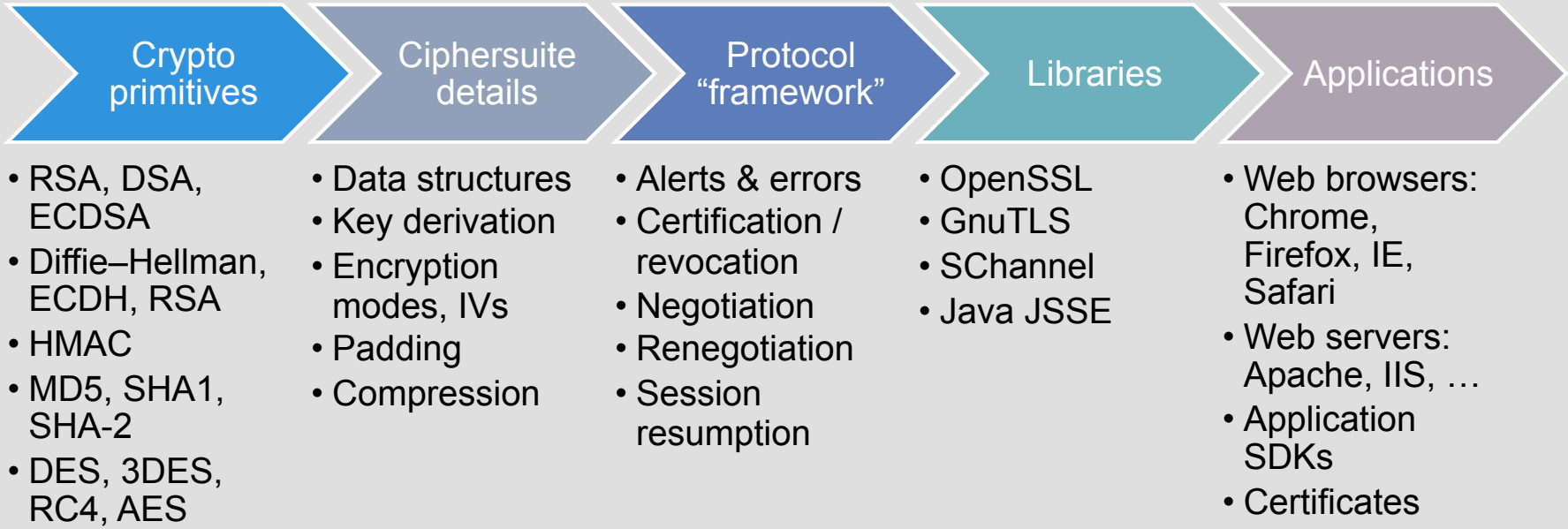
Structure of TLS



TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA TLS_DH_DSS_WITH_DES_CBC_SHA TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_RSA_WITH_DES_CBC_SHA TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA TLS_DHE_DSS_WITH_DES_CBC_SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA TLS_DHE_RSA_WITH_DES_CBC_SHA TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DH_anon_EXPORT_WITH_RC4_40_MD5 TLS_DH_anon_WITH_RC4_128_MD5 TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA TLS_KRB5_WITH_DES_CBC_SHA TLS_KRB5_WITH_3DES_EDE_CBC_SHA TLS_KRB5_WITH_RC4_128_SHA TLS_KRB5_WITH_IDEA_CBC_SHA
TLS_KRB5_WITH_DES_CBC_MD5 TLS_KRB5_WITH_3DES_EDE_CBC_MD5 TLS_KRB5_WITH_RC4_128_MD5 TLS_KRB5_WITH_IDEA_CBC_MD5 TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA
TLS_KRB5_EXPORT_WITH_RC2_CBC_40_SHA TLS_KRB5_EXPORT_WITH_RC4_40_SHA TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5 TLS_KRB5_EXPORT_WITH_RC2_CBC_40_MD5
TLS_KRB5_EXPORT_WITH_RC4_40_MD5 TLS_PSK_WITH_NULL_SHA TLS_DHE_PSK_WITH_NULL_SHA TLS_RSA_PSK_WITH_NULL_SHA TLS_RSA_WITH_AES_128_CBC_SHA
TLS_DH_DSS_WITH_AES_128_CBC_SHA TLS_DH_RSA_WITH_AES_128_CBC_SHA TLS_DHE_DSS_WITH_AES_128_CBC_SHA TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA TLS_DH_DSS_WITH_AES_256_CBC_SHA TLS_DH_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA TLS_DH_anon_WITH_AES_256_CBC_SHA TLS_RSA_WITH_NULL_SHA256 TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_DH_DSS_WITH_AES_128_CBC_SHA256 TLS_DH_RSA_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 TLS_DH_DSS_WITH_AES_256_CBC_SHA256
TLS_DH_RSA_WITH_AES_256_CBC_SHA256 TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 TLS_DH_anon_WITH_AES_128_CBC_SHA256
TLS_DH_anon_WITH_AES_256_CBC_SHA256 TLS_RSA_WITH_CAMELLIA_256_CBC_SHA TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA
TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA TLS_PSK_WITH_RC4_128_SHA
TLS_PSK_WITH_3DES_EDE_CBC_SHA TLS_PSK_WITH_AES_128_CBC_SHA TLS_PSK_WITH_AES_256_CBC_SHA TLS_DHE_PSK_WITH_RC4_128_SHA TLS_DHE_PSK_WITH_3DES_EDE_CBC_SHA
TLS_DHE_PSK_WITH_AES_128_CBC_SHA TLS_DHE_PSK_WITH_AES_256_CBC_SHA TLS_RSA_PSK_WITH_RC4_128_SHA TLS_RSA_PSK_WITH_3DES_EDE_CBC_SHA
TLS_RSA_PSK_WITH_AES_128_CBC_SHA TLS_RSA_PSK_WITH_AES_256_CBC_SHA TLS_RSA_WITH_SEED_CBC_SHA TLS_DH_DSS_WITH_SEED_CBC_SHA TLS_DH_RSA_WITH_SEED_CBC_SHA
TLS_DHE_DSS_WITH_SEED_CBC_SHA TLS_DHE_RSA_WITH_SEED_CBC_SHA TLS_DH_anon_WITH_SEED_CBC_SHA TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 TLS_DH_RSA_WITH_AES_128_GCM_SHA256 TLS_DH_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 TLS_DH_DSS_WITH_AES_128_GCM_SHA256 TLS_DH_DSS_WITH_AES_256_GCM_SHA384
TLS_DH_anon_WITH_AES_128_GCM_SHA256 TLS_DH_anon_WITH_AES_256_GCM_SHA384 TLS_PSK_WITH_AES_128_GCM_SHA256 TLS_PSK_WITH_AES_256_GCM_SHA384
TLS_DHE_PSK_WITH_AES_128_GCM_SHA256 TLS_DHE_PSK_WITH_AES_256_GCM_SHA384 TLS_RSA_PSK_WITH_AES_128_GCM_SHA256 TLS_RSA_PSK_WITH_AES_256_GCM_SHA384
TLS_PSK_WITH_AES_128_CBC_SHA256 TLS_PSK_WITH_AES_256_CBC_SHA384 TLS_PSK_WITH_NULL_SHA256 TLS_PSK_WITH_NULL_SHA384 TLS_DHE_PSK_WITH_AES_128_CBC_SHA256
TLS_DHE_PSK_WITH_AES_256_CBC_SHA384 TLS_DHE_PSK_WITH_NULL_SHA256 TLS_DHE_PSK_WITH_NULL_SHA384 TLS_RSA_PSK_WITH_AES_128_CBC_SHA256
TLS_RSA_PSK_WITH_AES_256_CBC_SHA384 TLS_RSA_PSK_WITH_NULL_SHA256 TLS_RSA_PSK_WITH_NULL_SHA384 TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA256 TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA256 TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA256
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA256 TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256 TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA256
TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA256 TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA256 TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256
TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA256 TLS_EMPTY_RENEGOTIATION_INFO_SCSV TLS_ECDH_ECDSA_WITH_NULL_SHA TLS_ECDH_ECDSA_WITH_RC4_128_SHA
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA TLS_ECDHE_ECDSA_WITH_NULL_SHA
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDH_RSA_WITH_NULL_SHA TLS_ECDH_RSA_WITH_RC4_128_SHA TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_NULL_SHA TLS_ECDHE_RSA_WITH_RC4_128_SHA TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_ECDH_anon_WITH_NULL_SHA TLS_ECDH_anon_WITH_RC4_128_SHA
TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA TLS_ECDH_anon_WITH_AES_128_CBC_SHA TLS_ECDH_anon_WITH_AES_256_CBC_SHA TLS_SRP_SHA_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_RSA_WITH_3DES_EDE_CBC_SHA TLS_SRP_SHA_DSS_WITH_3DES_EDE_CBC_SHA TLS_SRP_SHA_WITH_AES_128_CBC_SHA TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA
TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA TLS_SRP_SHA_WITH_AES_256_CBC_SHA TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_PSK_WITH_RC4_128_SHA TLS_ECDHE_PSK_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256 TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384
TLS_ECDHE_PSK_WITH_NULL_SHA TLS_ECDHE_PSK_WITH_NULL_SHA256 TLS_ECDHE_PSK_WITH_NULL_SHA384 TLS_RSA_WITH_ARIA_128_CBC_SHA256 TLS_RSA_WITH_ARIA_256_CBC_SHA384
TLS_DH_DSS_WITH_ARIA_128_CBC_SHA256 TLS_DH_DSS_WITH_ARIA_256_CBC_SHA384 TLS_DH_RSA_WITH_ARIA_128_CBC_SHA256 TLS_DH_RSA_WITH_ARIA_256_CBC_SHA384
TLS_DHE_DSS_WITH_ARIA_128_CBC_SHA256 TLS_DHE_DSS_WITH_ARIA_256_CBC_SHA384 TLS_DHE_RSA_WITH_ARIA_128_CBC_SHA256 TLS_DHE_RSA_WITH_ARIA_256_CBC_SHA384
TLS_DH_anon_WITH_ARIA_128_CBC_SHA256 TLS_DH_anon_WITH_ARIA_256_CBC_SHA384 TLS_ECDHE_ECDSA_WITH_ARIA_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_ARIA_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_ARIA_128_CBC_SHA256 TLS_ECDH_ECDSA_WITH_ARIA_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_ARIA_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_ARIA_256_CBC_SHA384
TLS_ECDH_RSA_WITH_ARIA_128_CBC_SHA256 TLS_ECDH_RSA_WITH_ARIA_256_CBC_SHA384 TLS_RSA_WITH_ARIA_128_GCM_SHA256 TLS_RSA_WITH_ARIA_256_GCM_SHA384
TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256 TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384 TLS_DH_RSA_WITH_ARIA_128_GCM_SHA256 TLS_DH_RSA_WITH_ARIA_256_GCM_SHA384
TLS_DHE_DSS_WITH_ARIA_128_GCM_SHA256 TLS_DHE_DSS_WITH_ARIA_256_GCM_SHA384 TLS_DH_DSS_WITH_ARIA_128_GCM_SHA256 TLS_DH_DSS_WITH_ARIA_256_GCM_SHA384
TLS_DH_anon_WITH_ARIA_128_GCM_SHA256 TLS_DH_anon_WITH_ARIA_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_ARIA_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_ARIA_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_ARIA_128_GCM_SHA256 TLS_ECDH_ECDSA_WITH_ARIA_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 TLS_ECDH_RSA_WITH_ARIA_128_GCM_SHA256 TLS_ECDH_RSA_WITH_ARIA_256_GCM_SHA384 TLS_PSK_WITH_ARIA_128_CBC_SHA256
TLS_PSK_WITH_ARIA_256_CBC_SHA384 TLS_DHE_PSK_WITH_ARIA_128_CBC_SHA256 TLS_DHE_PSK_WITH_ARIA_256_CBC_SHA384 TLS_RSA_PSK_WITH_ARIA_128_CBC_SHA256
TLS_RSA_PSK_WITH_ARIA_256_CBC_SHA384 TLS_PSK_WITH_ARIA_128_GCM_SHA256 TLS_PSK_WITH_ARIA_256_GCM_SHA384 TLS_DHE_PSK_WITH_ARIA_128_GCM_SHA256
TLS_DHE_PSK_WITH_ARIA_256_GCM_SHA384 TLS_RSA_PSK_WITH_ARIA_128_GCM_SHA256 TLS_RSA_PSK_WITH_ARIA_256_GCM_SHA384 TLS_ECDHE_PSK_WITH_ARIA_128_CBC_SHA256
TLS_ECDHE_PSK_WITH_ARIA_256_CBC_SHA384 TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_CAMELLIA_128_CBC_SHA256 TLS_ECDH_ECDSA_WITH_CAMELLIA_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384 TLS_ECDH_RSA_WITH_CAMELLIA_128_CBC_SHA256 TLS_ECDH_RSA_WITH_CAMELLIA_256_CBC_SHA384
TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256 TLS_RSA_WITH_CAMELLIA_256_GCM_SHA384 TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256 TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_DH_RSA_WITH_CAMELLIA_128_GCM_SHA256 TLS_DH_RSA_WITH_CAMELLIA_256_GCM_SHA384 TLS_DHE_DSS_WITH_CAMELLIA_128_GCM_SHA256
TLS_DHE_DSS_WITH_CAMELLIA_256_GCM_SHA384 TLS_DH_DSS_WITH_CAMELLIA_128_GCM_SHA256 TLS_DH_DSS_WITH_CAMELLIA_256_GCM_SHA384
TLS_DH_anon_WITH_CAMELLIA_128_GCM_SHA256 TLS_DH_anon_WITH_CAMELLIA_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384 TLS_ECDH_ECDSA_WITH_CAMELLIA_128_GCM_SHA256 TLS_ECDH_ECDSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384 TLS_ECDH_RSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_ECDH_RSA_WITH_CAMELLIA_256_GCM_SHA384 TLS_PSK_WITH_CAMELLIA_128_GCM_SHA256 TLS_PSK_WITH_CAMELLIA_256_GCM_SHA384 TLS_DHE_PSK_WITH_CAMELLIA_128_GCM_SHA256
TLS_DHE_PSK_WITH_CAMELLIA_256_GCM_SHA384 TLS_RSA_PSK_WITH_CAMELLIA_128_GCM_SHA256 TLS_RSA_PSK_WITH_CAMELLIA_256_GCM_SHA384
TLS_PSK_WITH_CAMELLIA_128_CBC_SHA256 TLS_PSK_WITH_CAMELLIA_256_CBC_SHA384 TLS_DHE_PSK_WITH_CAMELLIA_128_CBC_SHA256 TLS_DHE_PSK_WITH_CAMELLIA_256_CBC_SHA384
TLS_RSA_PSK_WITH_CAMELLIA_128_CBC_SHA256 TLS_RSA_PSK_WITH_CAMELLIA_256_CBC_SHA384 TLS_ECDHE_PSK_WITH_CAMELLIA_128_CBC_SHA256
TLS_ECDHE_PSK_WITH_CAMELLIA_256_CBC_SHA384 TLS_RSA_WITH_AES_128_CCM TLS_RSA_WITH_AES_256_CCM TLS_DHE_RSA_WITH_AES_128_CCM TLS_DHE_RSA_WITH_AES_256_CCM
TLS_RSA_WITH_AES_128_CCM_8 TLS_RSA_WITH_AES_256_CCM_8 TLS_DHE_RSA_WITH_AES_128_CCM_8 TLS_DHE_RSA_WITH_AES_256_CCM_8 TLS_PSK_WITH_AES_128_CCM

List of all 314 TLS 1.2 ciphersuites

TLS from primitive to application



Is TLS secure?

1996

SSL v3.0 standardized

2001

Some variant of one ciphersuite of the TLS record layer is a secure encryption scheme

[Kra01]

2002

Truncated TLS handshake using RSA key transport is a secure authenticated key exchange protocol

[JK02]

2008

Truncated TLS handshake using RSA key transport or signed Diffie–Hellman is a secure authenticated key exchange protocol

[MSW08]

"some variant"… "truncated TLS"…
limited ciphersuites

Is TLS secure?

1996 SSL v3.0 standardized

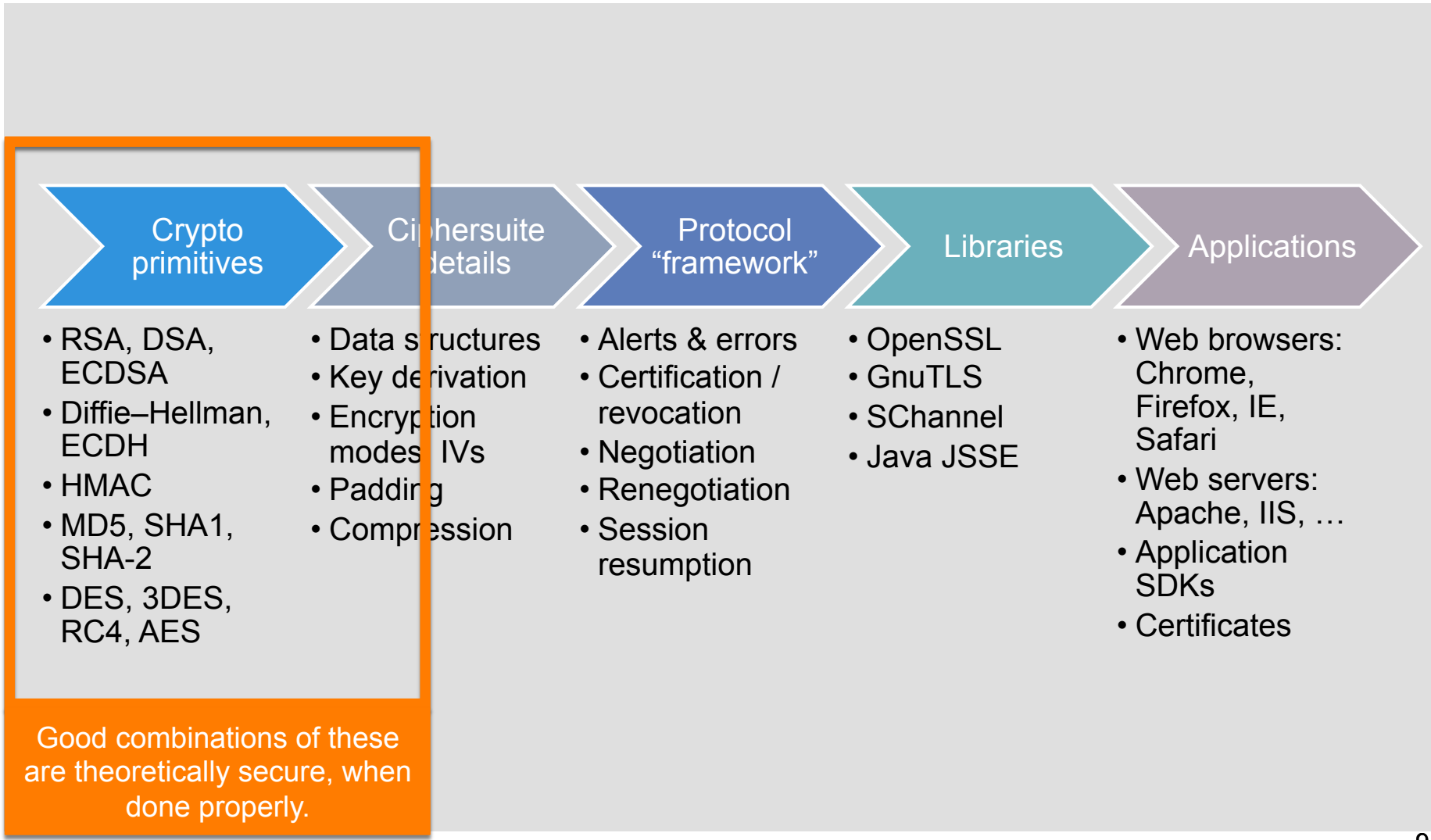
2011 Some modes of TLS record layer are secure authenticated encryption schemes [PRS11]

2012 Unaltered full signed Diffie–Hellman ciphersuite provides a secure channel [JKSS12]

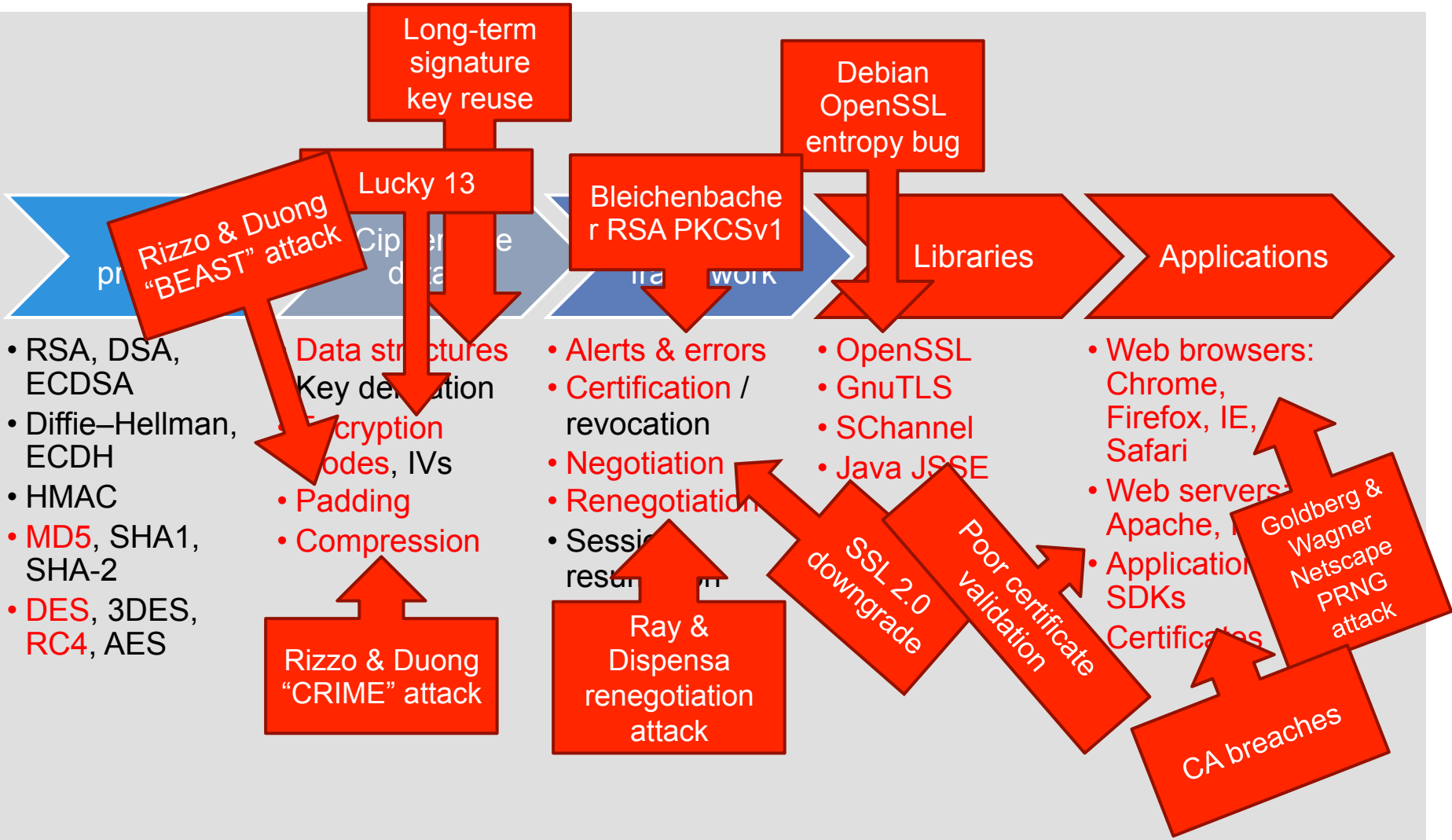
2013 Most unaltered full TLS ciphersuites provide a secure channel [KSS13, KPW13]

"unaltered"… "full"… "most ciphersuites"

Results on the provable security of TLS



Real-world attacks on TLS



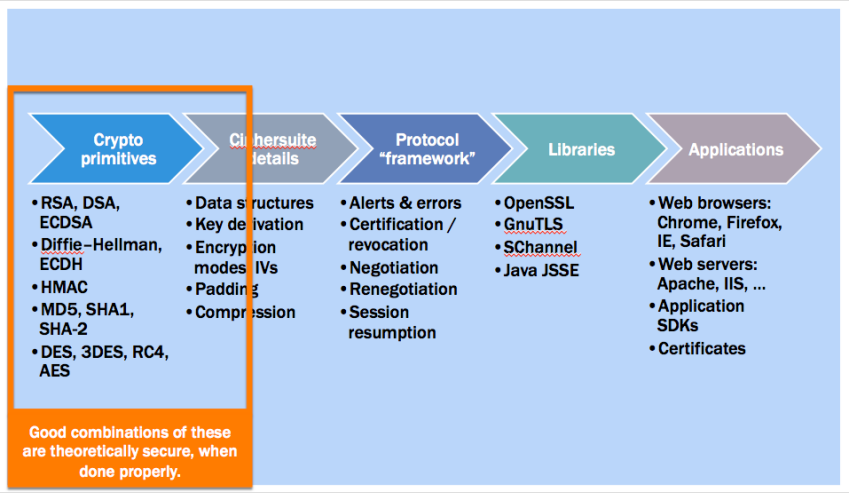
TLS is secure, but NOT

- TLS_DHE is a secure ACCE Protocol [JKSS12]
- Most TLS ciphersuits are secure ACCE Protocols [KSS13, KPW13]

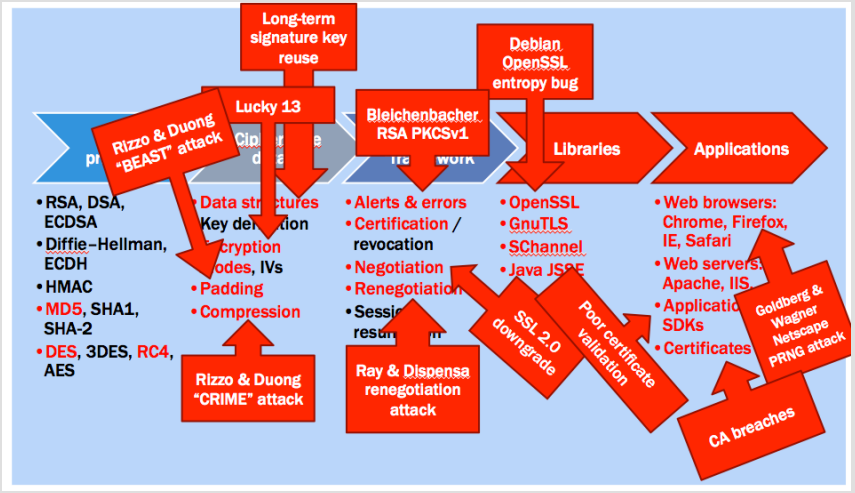
- BEAST attack [DR11]
- Renegotiation attack on TLS [RD09]

The gap between theory and practice

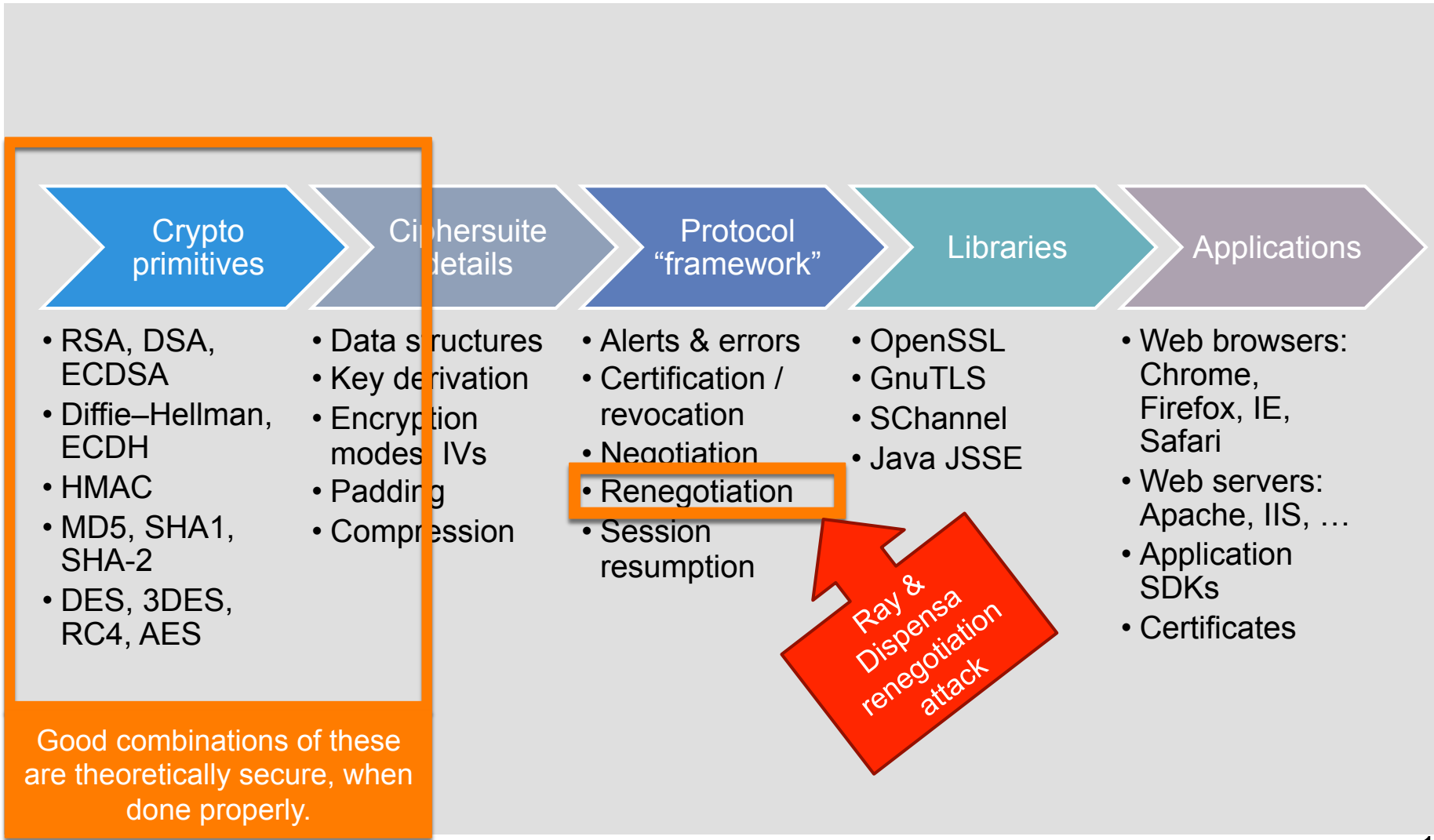
Provable security results



Real-world attacks



Our focus on TLS



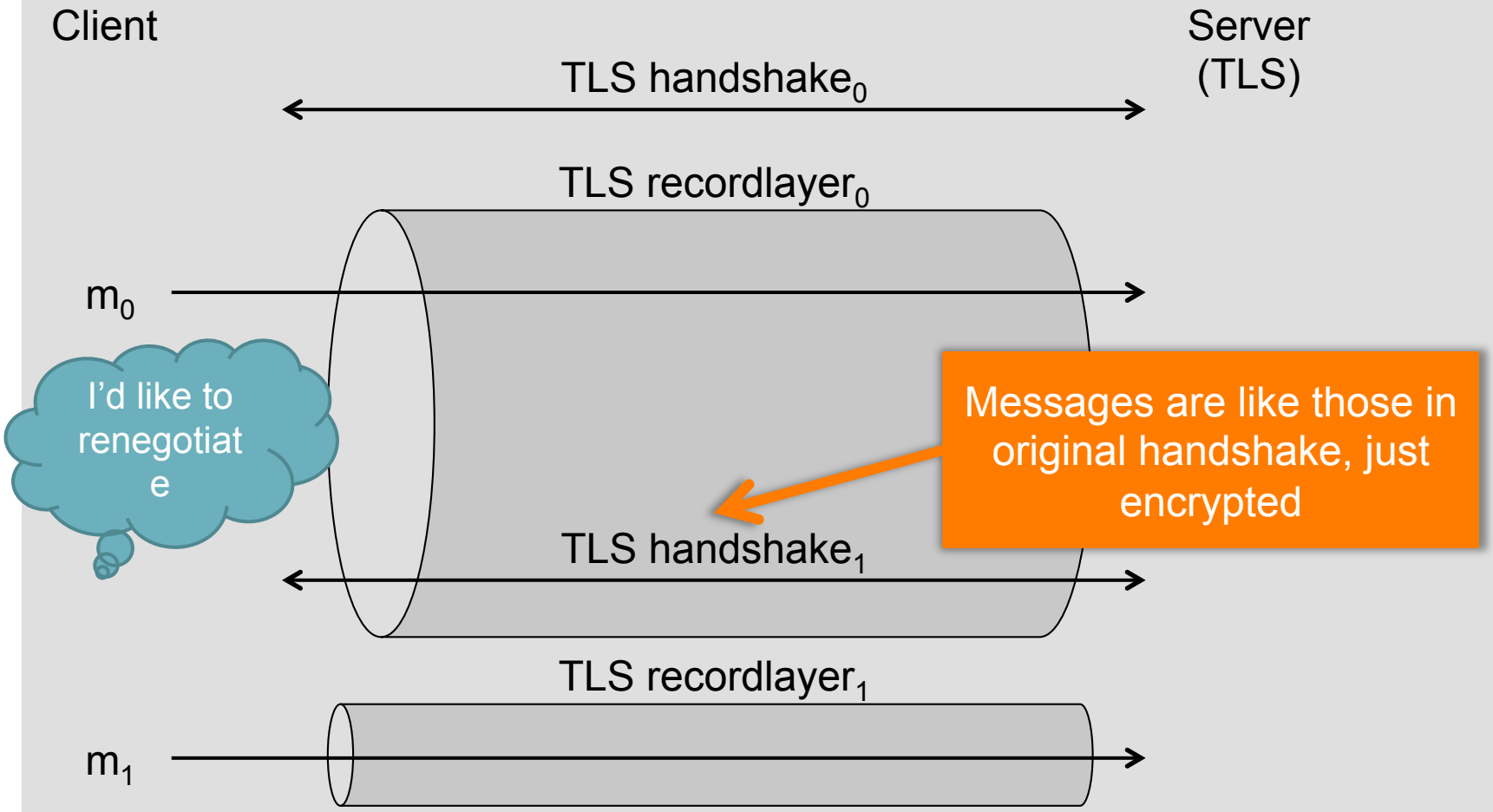
Renegotiation

Allows to create a new TLS channel that continues from the existing one.

Once you've established a TLS channel, why would you ever want to renegotiate it?

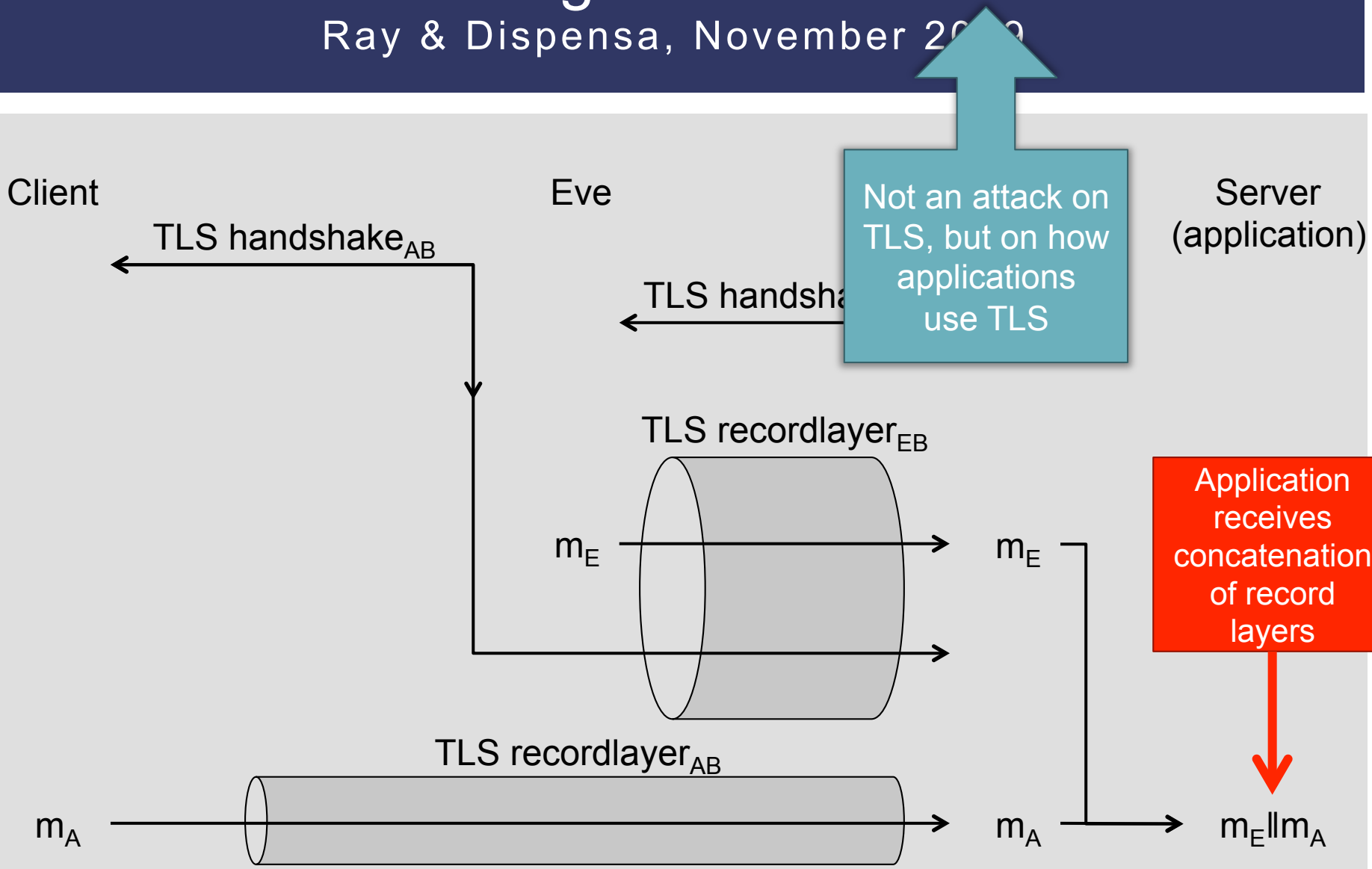
- Change cryptographic parameters
- Refresh encryption keys
- Change authentication credentials
- Identity hiding for client
  1. Establish a one-way authenticated TLS session
  2. Renegotiate using mutual authentication.
     Since handshake messages are sent in the encrypted TLS channel, client's identity is kept private.

Renegotiation in TLS
(pre-November 2009)



TLS Renegotiation "Attack"

Ray & Dispensa, November 20[illegible]



Eve orders a pizza but Alice pays

- Eve sends
  - m_E = "GET /orderPizza?deliverTo=123-Fake-St↩X-Ignore-This: "
- Client sends
  - m_A = "GET /orderPizza?deliverTo=456-Real-St↩Cookie: ↩ Account=1A2B"
- Server's web server receives
  - $m_E \| m_A$ = "GET /orderPizza?deliverTo=123-Fake-St↩
    X-Ignore-This: GET /orderPizza?deliverTo=456-Real-St↩
    Cookie: Account=1A2B"
  - X-Ignore-This: is an invalid header, so the rest of that line gets ignored.
  - Eve's GET request is processed with the cookie supplied by the client.

Renegotiation security

- Q: What property should a secure renegotiable protocol intuitively have?
- A: Whenever two parties successfully renegotiate, they are assured they have the exact same view of everything that happened previously.

Technical approach

1. Extend authenticated and confidential channel establishment (ACCE) security model to include renegotiable, multi-phase protocols.

2. Define security notion for renegotiable protocols.
   - secure renegotiable ACCE (full)
   - weakly secure renegotiable ACCE

3. Show that TLS without fixes does not satisfy security definition.

4. Show that TLS with fixes does satisfy (weak) security definition.
   - Generic reduction: If a TLS ciphersuite satisfies a certain property, then, when combined with fixes, it is a weakly secure renegotiable ACCE.

5. Propose fix to achieve (full) security.

ACCE security

Authenticated and Confidential Channel Establishment

Adversary's goals:

1. Violate authentication:
   - make Alice accept where her intended partner Bob is uncorrupted but has no matching conversation
2. Violate ciphertext integrity or confidentiality:
   - distinguish which of two (chosen) plaintexts were encrypted

Provable security of TLS:

- TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA is a secure ACCE protocol
  - (JKSS; CRYPTO 2012)
- Most TLS ciphersuites are ACCE secure
  - (KSS; ePrint 2013, KPR; CRYPTO 2013)

Secure renegotiable ACCE

- Secure renegotiable ACCE (full):
  - Authentication:
    - when a party successfully renegotiate a new phase, its partner has a phase with a matching handshake and record layer transcript.
- Weakly secure renegotiable ACCE:
  - Authentication:
    - when a party successfully renegotiate a new phase, its partner has a phase with a matching handshake and record layer transcript, ***provided no previous phase's session key was revealed***

Generic renegotiation security of TLS

- Prove generically that any ACCE-secure TLS ciphersuit is also renegotiation-secure
- Can‘t do this because of Ray-Dispensa attack
- Next best thing: prove that any ACCE-secure TLS ciphersuite with the fixes is also renegotiation-secure
- Problem: hard to generically insert the fixes into existing ACCE protocols

Main technical theorem

- Suppose a TLS ciphersuite is ACCE-secure, even if
  - The Finished message is revealed
  - Arbitrary „tags“ are included in Client/ServerHello messages
- Then that ciphersuite is weakly renegotiable secure if the RFC5746 fixes are used.

Weakly secure renegotiable ACCE

Definition

- Weakly secure renegotiable ACCE:
  - Authentication:
    - when a party successfully renegotiate a new phase, its partner has a phase with a matching handshake and record layer transcript, *provided no previous phase's session key was revealed*

TLS

- TLS without fixes is not a weakly secure renegotiable ACCE.
- TLS with RFC 5746 fixes is a weakly secure renegotiable ACCE.

Secure renegotiable ACCE

Definition

- Secure renegotiable ACCE (full):
  - Authentication:
    - when a party successfully renegotiate a new phase, its partner has a phase with a matching handshake and record layer transcript~~, *provided no previous phase's session key was revealed*~~.

TLS

- TLS with or without RFC 5746 fixes is not a secure renegotiable ACCE.
- Can be fixed by including hash of previous record layer messages in RIE.

Conclusion

Theory

- First paper investigated renegotiation for cryptographic protocols.
  - Different levels of renegotiation security
- Security of a protocol in isolation doesn't imply security with renegotiation.

Practice

- The standardized fixes are provably secure.
- Safe to use fixed renegotiation in TLS.

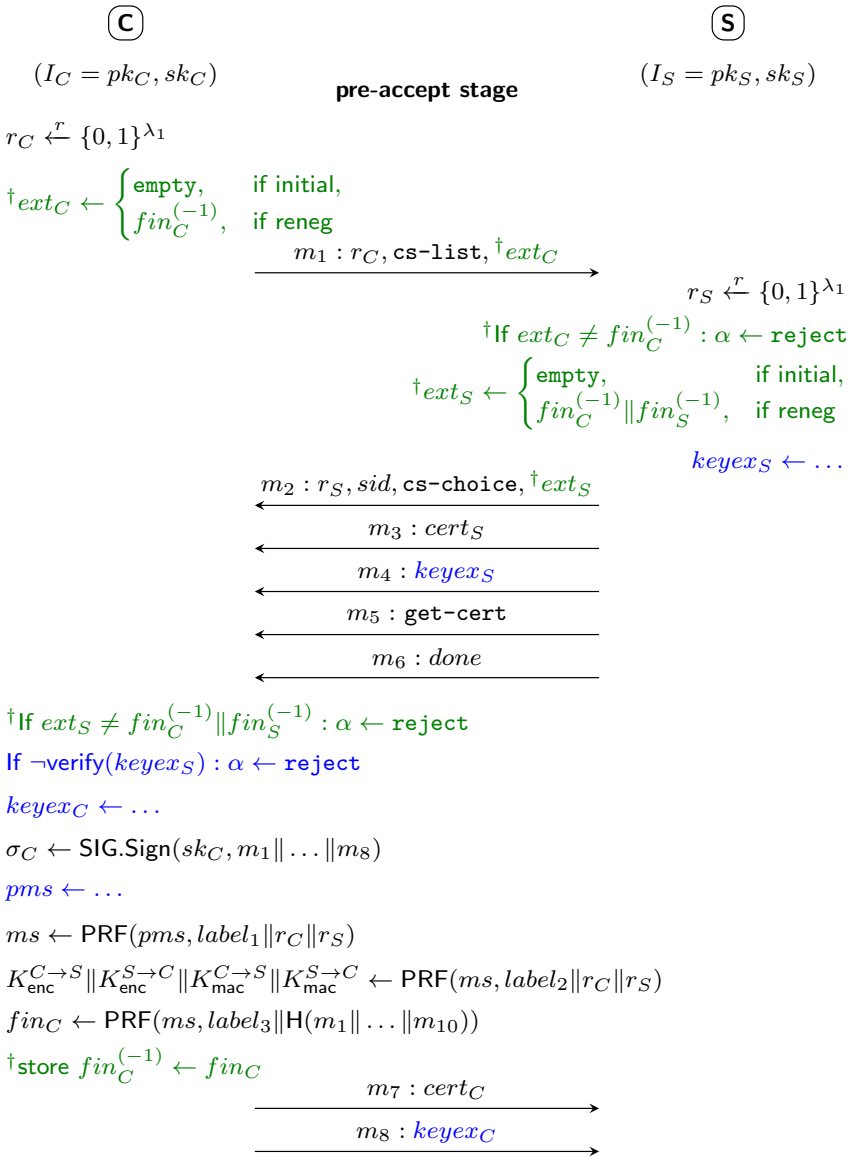
Thank you very much for your attention

Now

Later

Questions?

- Florian Giesen: Florian.Giesen@rub.de
- Florian Kohlar: Florian.Kohlar@rub.de
- Douglas Stebila: stebila@qut.edu.au


C
S
$(I_C = pk_C, sk_C)$
pre-accept stage
$(I_S = pk_S, sk_S)$
$r_C \xleftarrow{r} \{0,1\}^{\lambda_1}$
${}^{\dagger}ext_C \leftarrow \begin{cases} \texttt{empty}, & \text{if initial,} \\ fin_C^{(-1)}, & \text{if reneg} \end{cases}$
$m_1 : r_C, \texttt{cs-list}, {}^{\dagger}ext_C$
$r_S \xleftarrow{r} \{0,1\}^{\lambda_1}$
${}^{\dagger}\text{If } ext_C \neq fin_C^{(-1)} : \alpha \leftarrow \texttt{reject}$
${}^{\dagger}ext_S \leftarrow \begin{cases} \texttt{empty}, & \text{if initial,} \\ fin_C^{(-1)} \| fin_S^{(-1)}, & \text{if reneg} \end{cases}$
$keyex_S \leftarrow \ldots$
$m_2 : r_S, sid, \texttt{cs-choice}, {}^{\dagger}ext_S$
$m_3 : cert_S$
$m_4 : keyex_S$
$m_5 : \texttt{get-cert}$
$m_6 : done$
${}^{\dagger}\text{If } ext_S \neq fin_C^{(-1)} \| fin_S^{(-1)} : \alpha \leftarrow \texttt{reject}$
$\text{If } \neg\text{verify}(keyex_S) : \alpha \leftarrow \texttt{reject}$
$keyex_C \leftarrow \ldots$
$\sigma_C \leftarrow \text{SIG.Sign}(sk_C, m_1 \| \ldots \| m_8)$
$pms \leftarrow \ldots$
$ms \leftarrow \text{PRF}(pms, label_1 \| r_C \| r_S)$
$K_{\text{enc}}^{C \to S} \| K_{\text{enc}}^{S \to C} \| K_{\text{mac}}^{C \to S} \| K_{\text{mac}}^{S \to C} \leftarrow \text{PRF}(ms, label_2 \| r_C \| r_S)$
$fin_C \leftarrow \text{PRF}(ms, label_3 \| \text{H}(m_1 \| \ldots \| m_{10}))$
${}^{\dagger}\text{store } fin_C^{(-1)} \leftarrow fin_C$
$m_7 : cert_C$
$m_8 : keyex_C$

RFC 5746