SAC Summer School 2016

Implementation and analysis of cryptographic protocols

Part 4: Provable security of TLS

Dr. Douglas Stebila

McMaster University

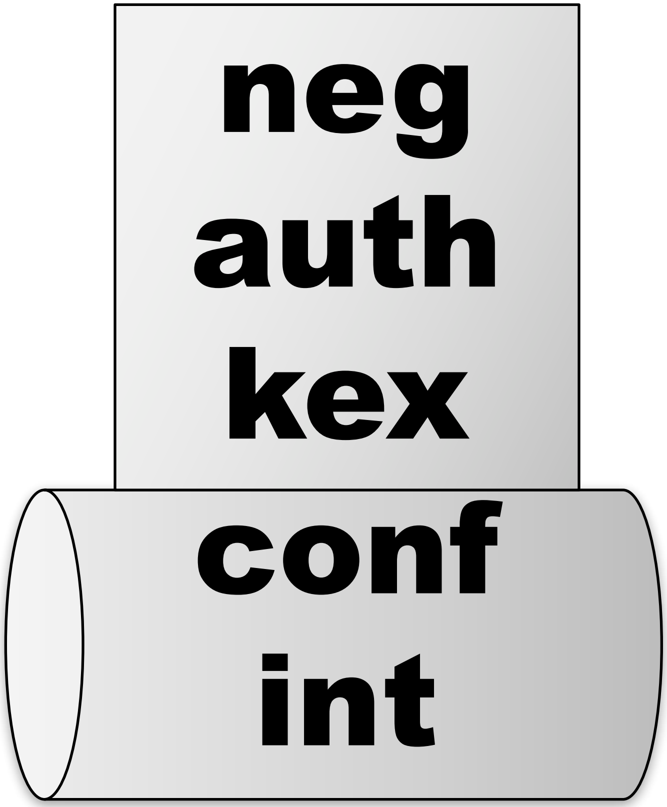
https://www.douglas.stebila.ca/teaching/sac-2016

Provable security

- Define a cryptographic scheme as a set of algorithms.
- Define security as an interactive game between a challenger and an adversary.
- Specify your scheme.
- Prove a theorem that any adve win the security game can be u some hard problem (“reduction”).

Same type of reduction as e.g. proving NP-completeness of travelling salesman problem

Security goals of TLS



From an application perspective, TLS provides:

- (negotiation of parameters)
- entity authentication
- (key exchange)
- confidentiality and integrity of messages

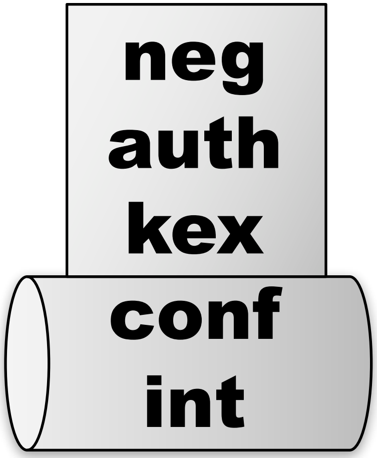
Is TLS secure?

Idea

Prove the TLS handshake is a secure authenticated key exchange protocol

- BR or CK or eCK model: adversary can't distinguish real session key from random session key

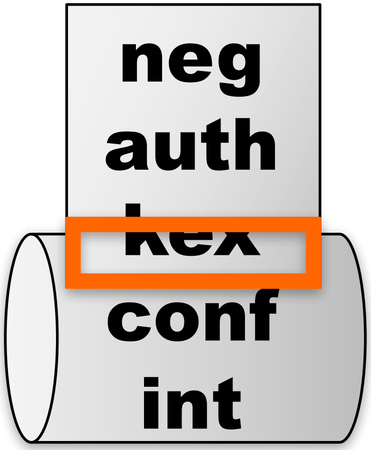
Prove the TLS record layer is a secure authenticated encryption scheme



Problem

TLS handshake sends messages encrypted under the session key

- => overlap between handshake and record layer
- Adversary can distinguish real session key from random



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 AKE [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 is a secure channel

[JKSS12]

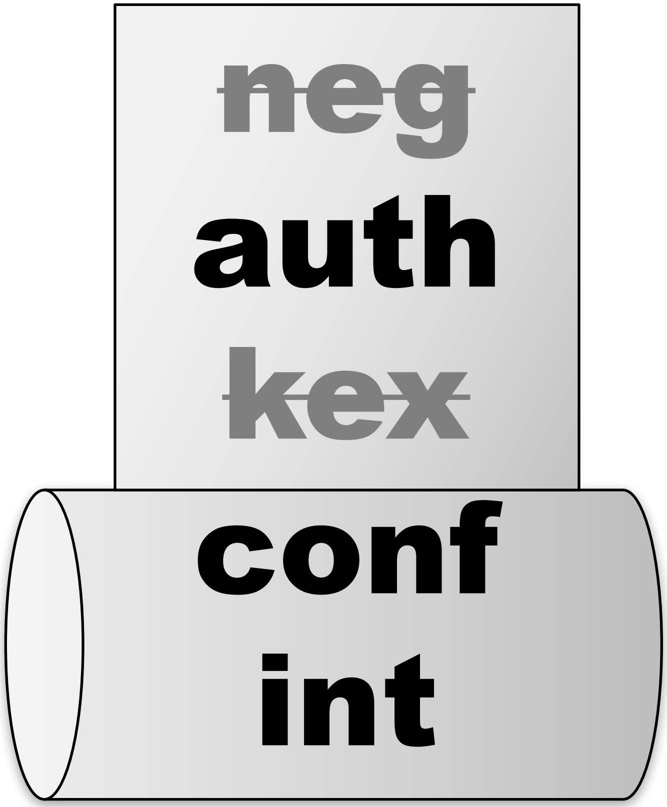
2013

Most unaltered full TLS ciphersuites are a secure channel

[KSS13, KPW13, BFKPS13]

“unaltered”… “full”… “most ciphersuites”

Security goals of TLS



Authenticated and Confidential Channel Establishment (ACCE) security definition [JKSS12] captures:

- entity authentication
- confidentiality and integrity of messages

More results on TLS 1.2

ACCE family

- Renegotiation countermeasure
- Negotiation / downgrade resilience

Constructive cryptography

Formal verification of implementation

- miTLS

SAC Summer School 2016

Implementation and analysis of cryptographic protocols

Part 5: TLS 1.3

Dr. Douglas Stebila

McMaster University

https://www.douglas.stebila.ca/teaching/sac-2016

TLSv1.3: The Next Generation

- Currently under development at the IETF

- Primary goals:
  - remove ciphersuites without forward secrecy
  - remove obsolete / deprecated algorithms
  - provide low-latency mode with fewer round trips
  - encrypt more of the handshake to improve privacy

Zero round trip mode (0-RTT)

- Goal:
  - allow client to send application data on first C-S handshake flow
  - allow server to respond with application data on first S-C handshake flow

- Compared with 3 round trips for TLS 1.2 full handshake and 2 round trips for TLS 1.2 session resumption

Academic involvement in TLS 1.3

- TLS working group actively encouraged academic analyses of TLS 1.3
- TLS 1.3 Ready Or Not (TRON) Workshop
  - January 2016
  - May 2016

Academic results on TLS 1.3

- OPTLS protocol
  - Candidate design for 0-RTT mode
- Provable security of TLS 1.3 handshake candidates
  - draft-05 and draft-10, ECDHE and PSK
- Automated verification of TLS 1.3 modes using Tamarin prover
  - Identified some flaws that have been fixed
- Verified TLS 1.3 implementations
- TLS 1.3 and QUIC weaknesses against PKCS #1 v1.5 encryption
- Provable security analysis of post-handshake authentication

TLS 1.3 timeline

- Working group last call later in 2016?
- ~2? months for additional academic analysis
- Standardization in 2017?
- First implementations in 2017 or 2018
- First attacks…?
    - 0-RTT could be risky:
        - No forward secrecy
        - No solid replay protection
            - How do applications decide which requests are okay without replay protection?