Network Security (NetSec)

Summer 2015
Chapter 04: Transport Level Security
Module 01: Overview and a Toy SSL Protocol

Prof. Dr.-Ing. Matthias Hollick

Technische Universität Darmstadt
Secure Mobile Networking Lab - SEEMOO
Department of Computer Science
Center for Advanced Security Research Darmstadt - CASED

Mornewegstr. 32
D-64293 Darmstadt, Germany
Tel.+49 6151 16-70922, Fax. +49 6151 16-70921
http://seemoo.de or http://www.seemoo.tu-darmstadt.de

Prof. Dr.-Ing. Matthias Hollick
matthias.hollick@seemoo.tu-darmstadt.de

Learning Objectives & Outline

Security objectives, mechanisms and limitations on transport layer (or between network layer and application layer)

- Identify the scope of protection as well as the trade-offs involved in securing networks on transport layer
- Understand the fundamental design principles of transport layer security protocols
- Discuss toy and real-world transport layer security protocols

Outline

(1) The scope of protection on transport layer

(2) Some history of transport layer security

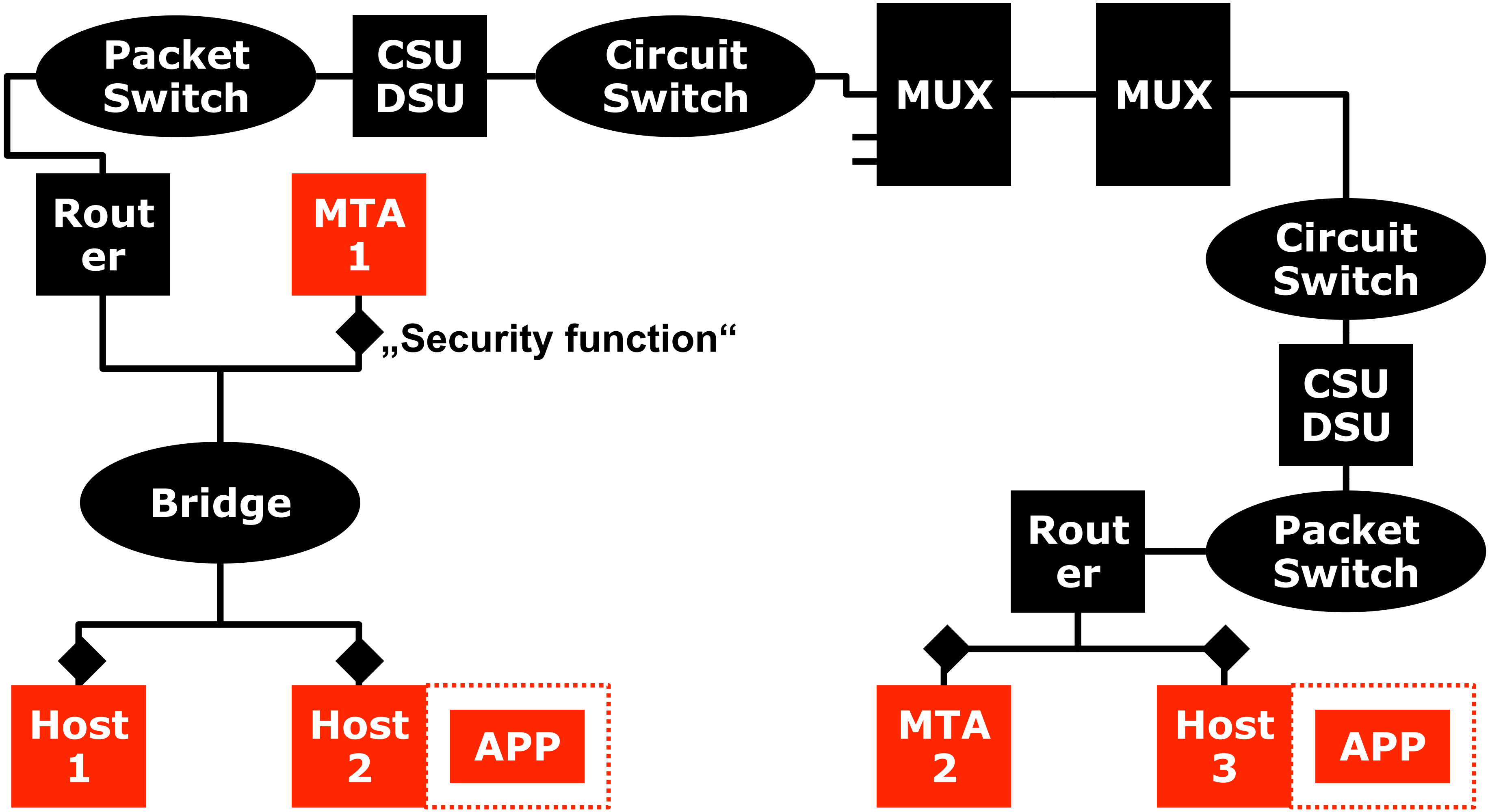
(3) A toy SSL protocol

(4) Recommended readings

Chapter 04, Module 01

Pros and Cons of L4 Security?

L4 Scope of Protection



Transport Layer Security

Characteristics

- No network technology dependence
- Yet significant protocol suite dependence

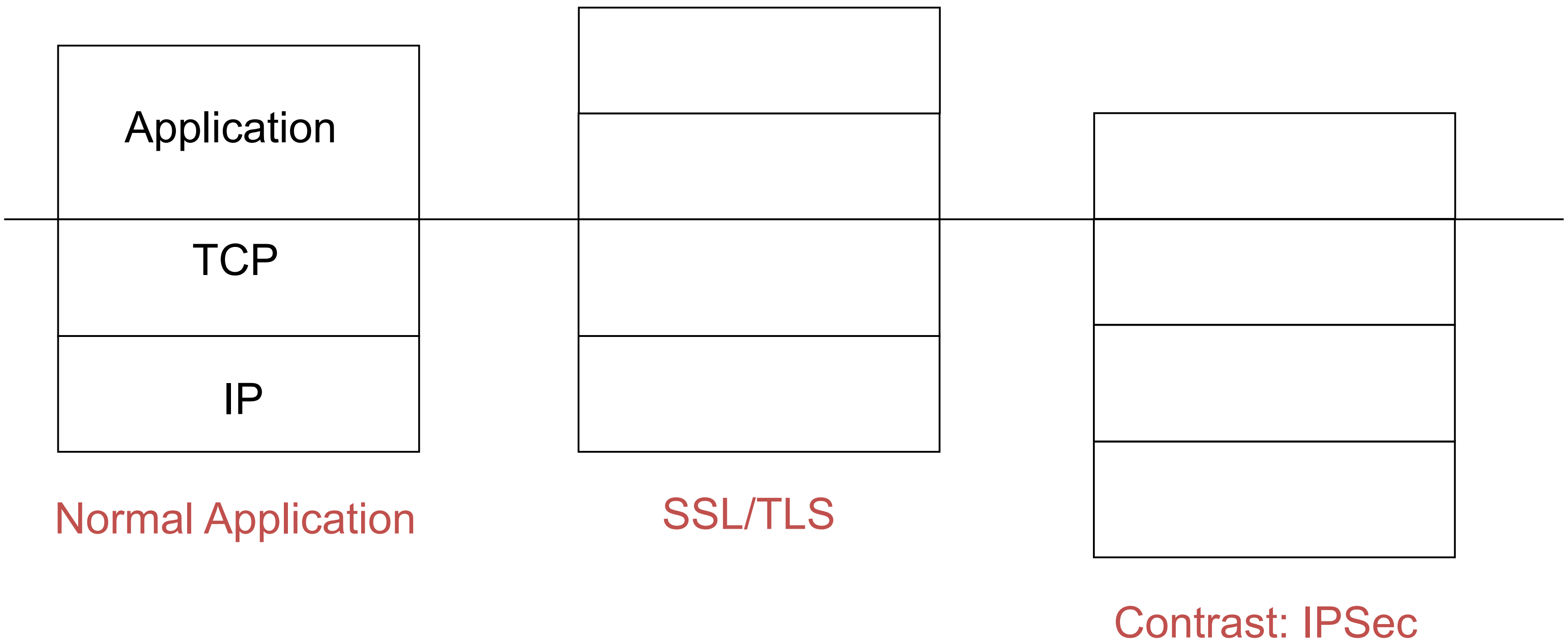
Protection

- Black (data protected)
  - circuits & muxes, circuit & packet switches, LANs & bridges, routers
- Red (data unprotected)
  - MTAs, hosts
- Protection granularity: end host, per-connection

Security services that can be provided on transport layer

- Confidentiality
- Data origin authentication, Peer entity authentication
- Connectionless integrity, Connection-oriented integrity with recovery
- Access control

SSL and TCP/IP



Developer’s view

- SSL provides application programming interface (API)to applications
- C and Java SSL libraries/classes readily available

SSL: Secure Sockets Layer

Widely deployed security protocol
- Supported by almost all browsers and web servers
- "https" protocol means http over SSL
- Tens of billions $ spent per year over SSL

Originally designed by Netscape in 1993-1995

Number of variations:
- SSLv1, SSLv2, SSLv3
- TLS: transport layer security, v1: RFC 2246, v1.1: RFC 4346, v1.2: RFC 5246

Provides
- Confidentiality
- Integrity
- Authentication

Original goals:
- Had Web e-commerce transactions in mind
- Encryption (especially credit-card numbers)
- Web-server authentication
- Optional client authentication
- Minimum hassle in doing business with new merchant

Available to all TCP applications
- Secure socket interface

Standardization Madness

- RFC 3943: “Transport Layer Security (TLS) Protocol Compression Using Lempel-Ziv-Stac (LZS)”.
- RFC 4132: “Addition of Camellia Cipher Suites to Transport Layer Security (TLS)”.
- RFC 4162: “Addition of SEED Cipher Suites to Transport Layer Security (TLS)”.
- RFC 4217: “Securing FTP with TLS”.
- RFC 4279: “Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)”.
- RFC 4347: “Datagram Transport Layer Security”
- RFC 4366: “Transport Layer Security (TLS) Extensions”.
- RFC 4492: “Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)”.
- RFC 4507: “Transport Layer Security (TLS) Session Resumption without Server-Side State”.
- RFC 4680: “TLS Handshake Message for Supplemental Data”.
- RFC 4681: “TLS User Mapping Extension”.
- RFC 4785: “Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS)”.
- RFC 5054: “Using the Secure Remote Password (SRP) Protocol for TLS Authentication”.
- RFC 5746: “Transport Layer Security (TLS) Renegotiation Indication Extension”.

Standardization Madness Contd.

SECURE MOBILE NETWORKING

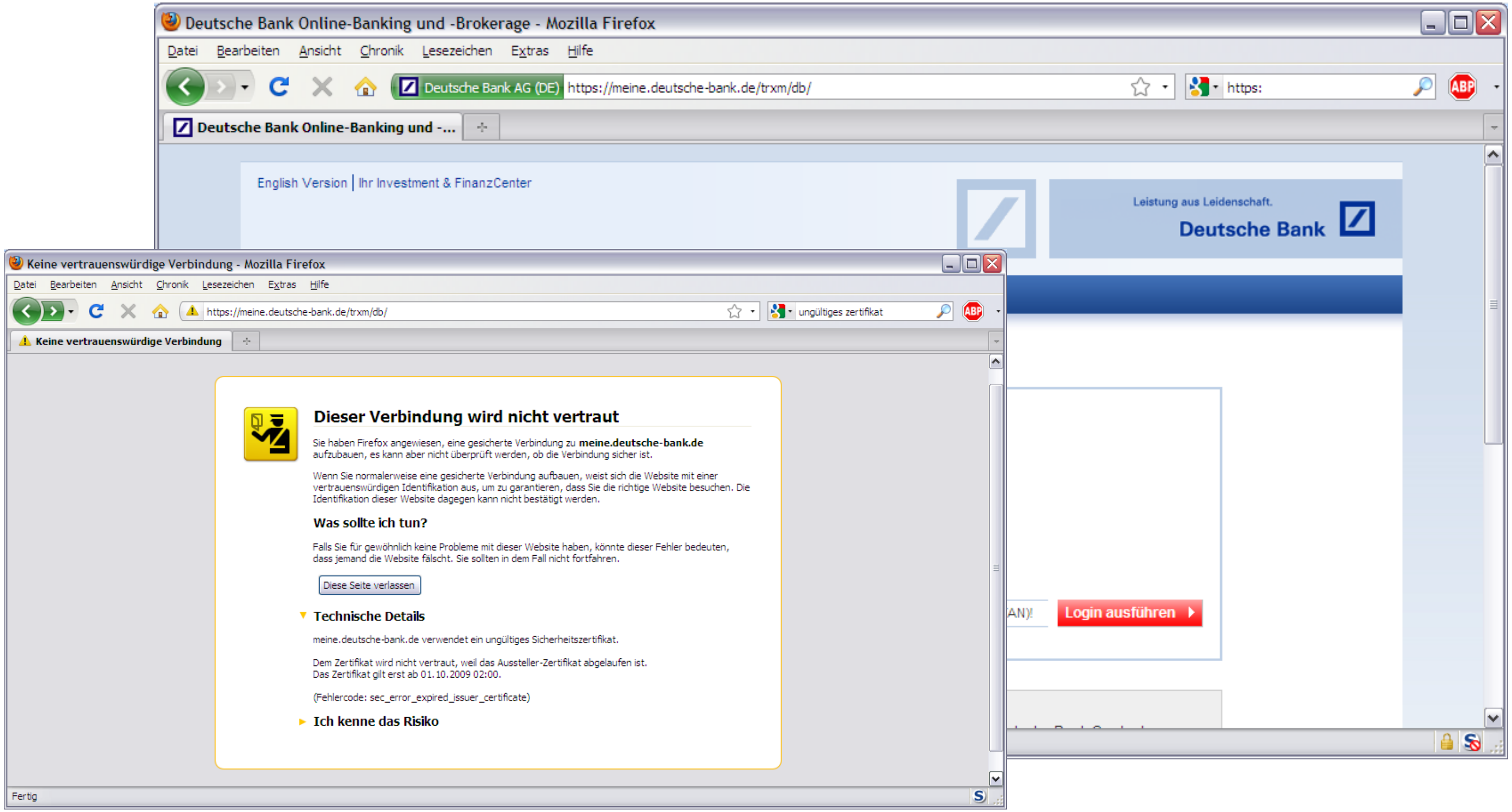
nsport Layer Security (tls)

datatracker.ietf.org/wg/tls/

(draft-ietf-tls-srp)	Protocol for TLS Authentication				
RFC 5081 (draft-ietf-tls-openpgp-keys)	Using OpenPGP Keys for Transport Layer Security (TLS) Authentication	2007-11	RFC 5081 (Experimental) Obsoleted by RFC 6091		Russ Housley
RFC 5246 (draft-ietf-tls-rfc4346-bis)	The Transport Layer Security (TLS) Protocol Version 1.2	2008-08	RFC 5246 (Proposed Standard) Updated by RFC 5746, RFC 5878, RFC 6176 Errata	1	Tim Polk
RFC 5288 (draft-ietf-tls-rsa-aes-gcm)	AES Galois Counter Mode (GCM) Cipher Suites for TLS	2008-08	RFC 5288 (Propo…		Pasi Eronen
RFC 5289 (draft-ietf-tls-ecc-new-mac)	TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter M… (GCM)	200…			Pasi Eronen
RFC 5469 (draft-ietf-tls-des-idea)	DES…		…(Informational)	1	Tim Polk
RFC 5487 (draft-ietf-tls-psk-…	…Counter Mode	2009-03	RFC 5487 (Proposed Standard)		Pasi Eronen
RFC 5489 (draft-ietf-tls-ecdhe…	…PSK Cipher Suites for Transport Layer Security (TLS)	2009-03	RFC 5489 (Informational)		Pasi Eronen
RFC 5705 (draft-ietf-tls-extractor)	Keying Material Exporters for Transport Layer Security (TLS)	2010-03	RFC 5705 (Proposed Standard)	3	Pasi Eronen
RFC 5746 (draft-ietf-tls-renegotiation)	Transport Layer Security (TLS) Renegotiation Indication Extension	2010-02	RFC 5746 (Proposed Standard)		Pasi Eronen
RFC 6066 (draft-ietf-tls-rfc4366-bis)	Transport Layer Security (TLS) Extensions: Extension Definitions	2011-01	RFC 6066 (Proposed Standard)		Sean Turner
RFC 6176 (draft-ietf-tls-ssl2-must-not)	Prohibiting Secure Sockets Layer (SSL) Version 2.0	2011-03	RFC 6176 (Proposed Standard)		Alexey Melnikov
RFC 6347 (draft-ietf-tls-rfc4347-bis)	Datagram Transport Layer Security Version 1.2	2012-01	RFC 6347 (Proposed Standard)	3	Sean Turner
RFC 6520 (draft-ietf-tls-dtls-heartbeat)	Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension	2012-02	RFC 6520 (Proposed Standard)		Sean Turner

check out datatracker.ietf.org

User Awareness

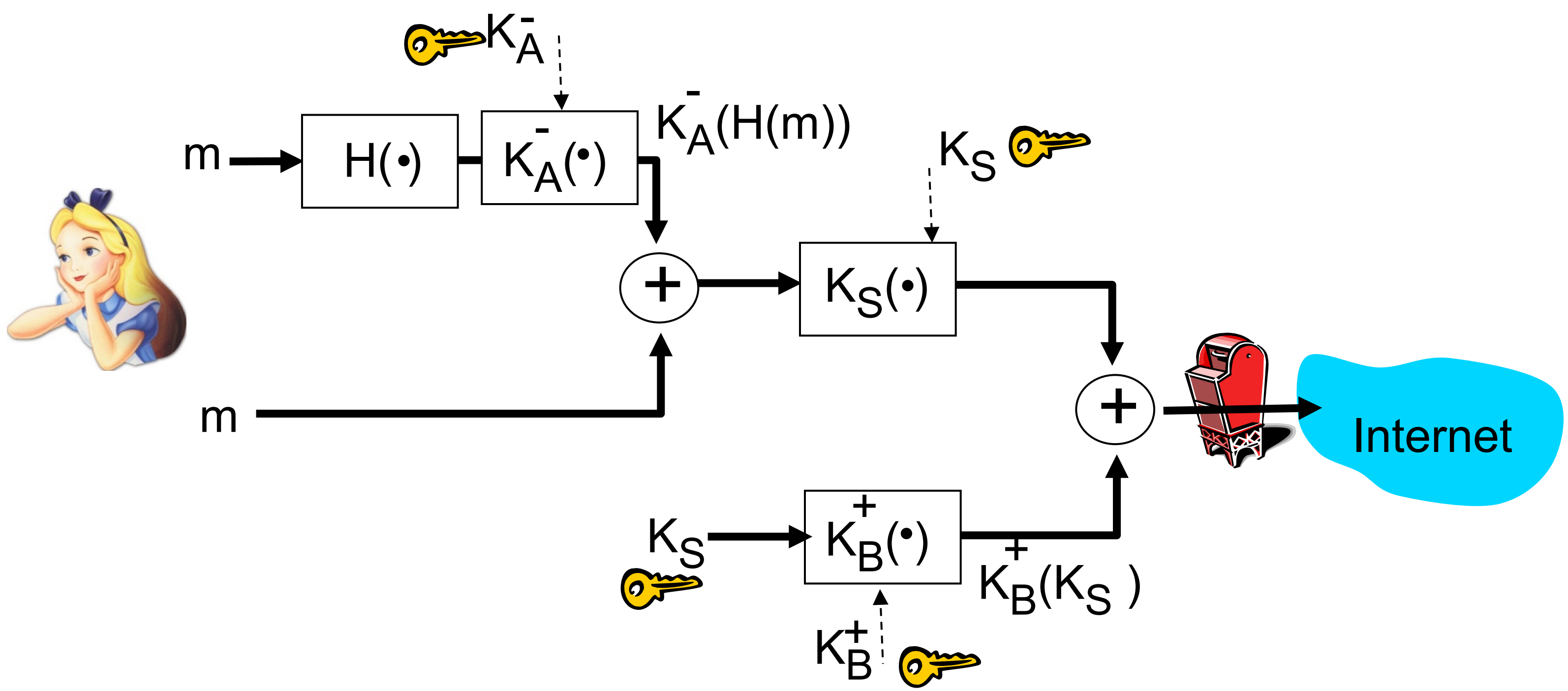


Could we borrow from PGP?

But want to send byte streams & interactive data

Want a set of secret keys for the entire connection

Want certificate exchange part of protocol: handshake phase



Toy SSL: A Simple Secure Channel

Handshake:

- Alice and Bob use their certificates and private keys to authenticate each other and exchange shared secret

Key Derivation:

- Alice and Bob use shared secret to derive set of keys

Data Transfer:

- Data to be transferred is broken up into a series of records

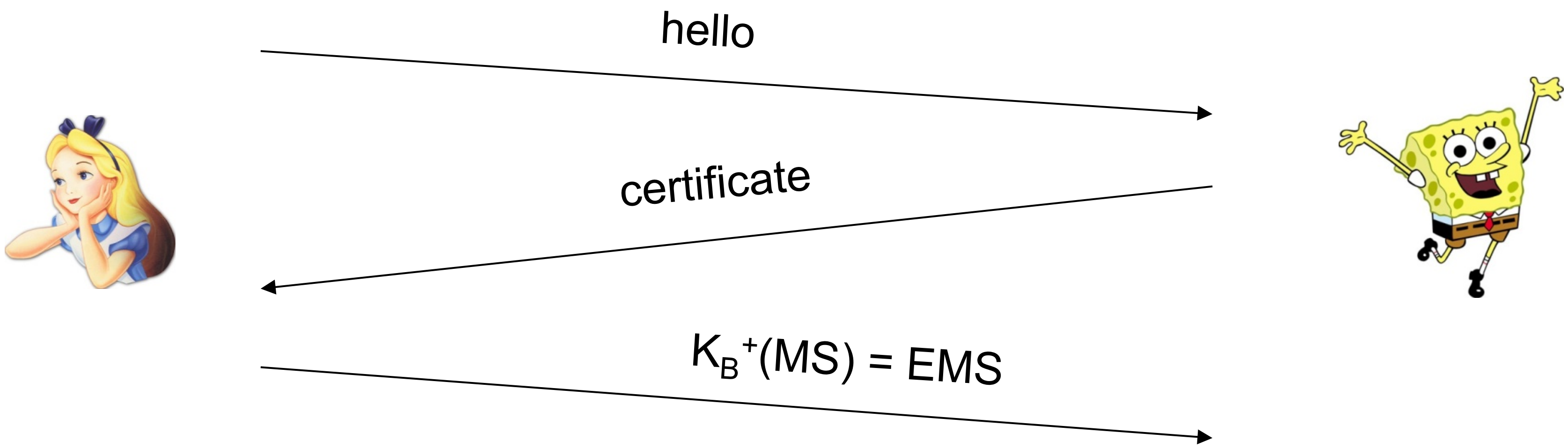
Connection Closure:

- Special messages to securely close connection

Toy SSL: A simple handshake

MS = master secret

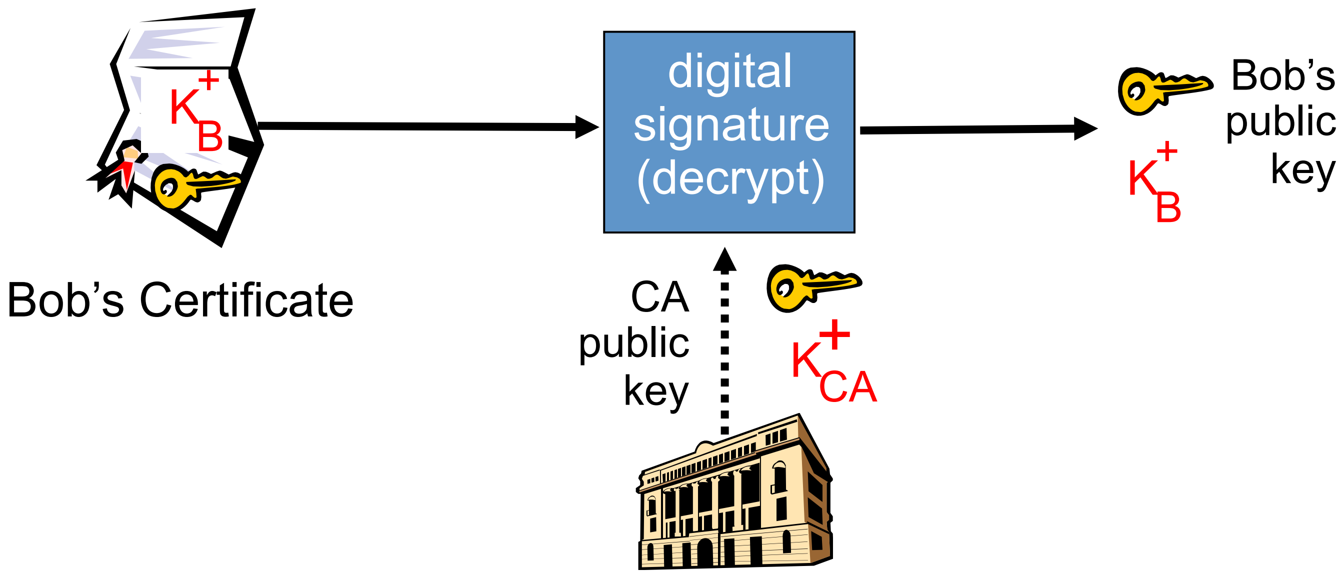
EMS = encrypted master secret



Certification Authorities

When Alice wants Bob’s public key:

- gets Bob’s certificate (from Bob or elsewhere).
- apply CA’s public key to Bob’s certificate, get Bob’s public key



Beyond secure handshake: What do we want?

Toy SSL: Key Derivation

Considered bad to use same key for more than one cryptographic operation

- Use different keys for message authentication code (MAC) and encryption

Four keys:

- K_c = encryption key for data sent from client to server
- M_c = MAC key for data sent from client to server
- K_s = encryption key for data sent from server to client
- M_s = MAC key for data sent from server to client

Keys derived from key derivation function (KDF)

- Takes master secret and (possibly) some additional random data and creates the keys
- Which other properties might be of interest for the keys?

Toy SSL: Data Records

Why not encrypt data in constant stream as we write it to TCP?

- Where would we put the MAC? If at end, no message integrity until all data processed.
- For example, with instant messaging, how can we do integrity check over all bytes sent before displaying?

Instead, break stream in series of records

- Each record carries a MAC
- Receiver can act on each record as it arrives

Issue: in record, receiver needs to distinguish MAC from data

- Want to use variable-length records

Toy SSL: Sequence Numbers

Attacker can capture and replay record or re-order records (and happily increase the unprotected TCP sequence number so its packets get delivered to the TOY SSL protocol)

Solution: put sequence number into MAC:

- MAC = MAC(M_x, sequence||header||data)
- In the SSL record, there is a field for SSL seq. numbers? True or False?

False. Both sides maintain Seq. no independently.

Attacker could still replay all of the records

- Use random nonce

Toy SSL: Control Information

Truncation attack:

- attacker forges TCP connection close segment
- One or both sides thinks there is less data than there actually is.

Solution: record types, with one type for closure

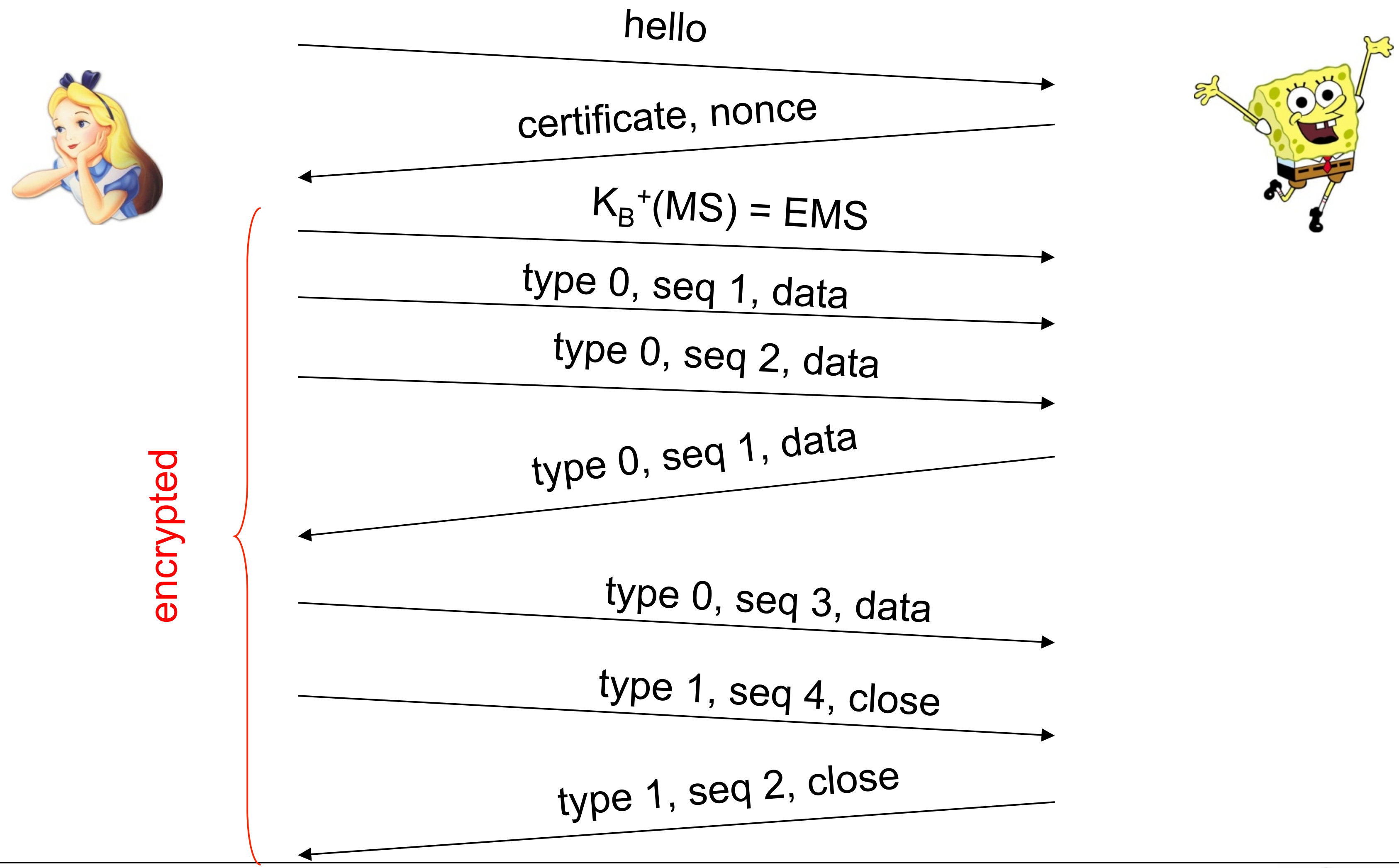
- type 0 for data; type 1 for closure

MAC = MAC(M_x, sequence||type||data)

length	type	data	MAC

What is cryptographically protected?

Toy SSL: Summary



Question

Suppose Seq. number is NOT used.

Question: In an SSL session, Can Trudy ((wo)man-in-the-middle) delete a TCP segment?

Question: What effect will it have?

Question

Suppose Seq. number is USED.

In an SSL session, an attacker inserts a bogus TCP segment into a packet stream with correct TCP checksum and seq. no.

Question: Will TCP at the receiving side accept bogus packet and pass the payload to SSL?

Question: Will SSL at the receiving side accept bogus packet and pass the payload to application level?

Toy SSL isn't complete

Handshake sufficient?

How long are the fields?

What cryptographic protocols supported?

No negotiation

- Allow client and server to support different encryption algorithms
- Allow client and server to choose together specific algorithm before data transfer

And potentially much more

Acks & Recommended Reading

Selected slides of this chapter courtesy of

- K. Ross with changes of myself incorporated
- Few others by S. Kent with changes

Recommended reading

- [KaPeSp2002] Charlie Kaufman, Radia Perlman, Mike Speciner: Network Security – Private Communication in a Public World, 2nd Edition, Prentice Hall, 2002, ISBN: 978-0-13-046019-6
- [Stallings2014] William Stallings, Network Security Essentials, 4th Edition, Prentice Hall, 2014, ISBN: 978-0-136-10805-4
- [Schäfer2003] G. Schäfer. Netzsicherheit - Algorithmische Grundlagen und Protokolle. dpunkt.verlag, 2003.

Copyright Notice

Contact

Prof. Dr.-Ing. Matthias Hollick
Department of Computer Science

SEEMOO
Mornewegstr. 32
64293 Darmstadt/Germany
matthias.hollick@seemoo.tu-darmstadt.de

Phone +49 6151 16-70920
Fax +49 6151 16-70921
www.seemoo.tu-darmstadt.de