Aalto University
School of Electrical
Engineering

Basic Principles in Networking

Securing Email and TCP

Stephan Sigg

Department of Communications and Networking
Aalto University, School of Electrical Engineering
stephan.sigg@aalto.fi

Version 1.0

Aalto University
School of Electrical
Engineering

Motivation (5 min)

SHA-1 collision (Defcon 2017 conference)

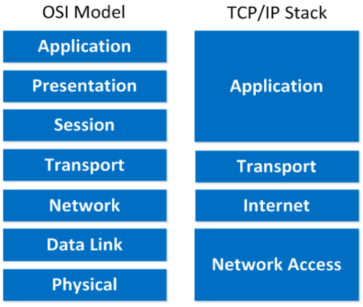
Aalto University
School of Electrical
Engineering

Part I (20 min)

Securing Email

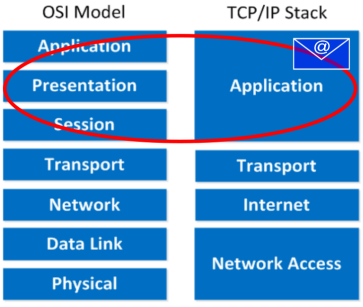
Secure Email

Desired properties



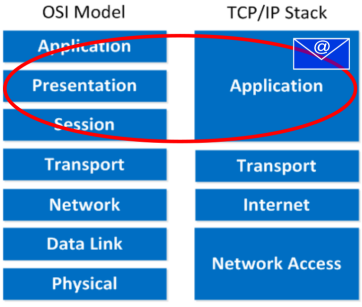
Secure Email

Desired properties



Secure Email

Desired properties



Secure Email

Desired properties

Desired for secure email:

Secure Email

Desired properties

Desired for secure email:

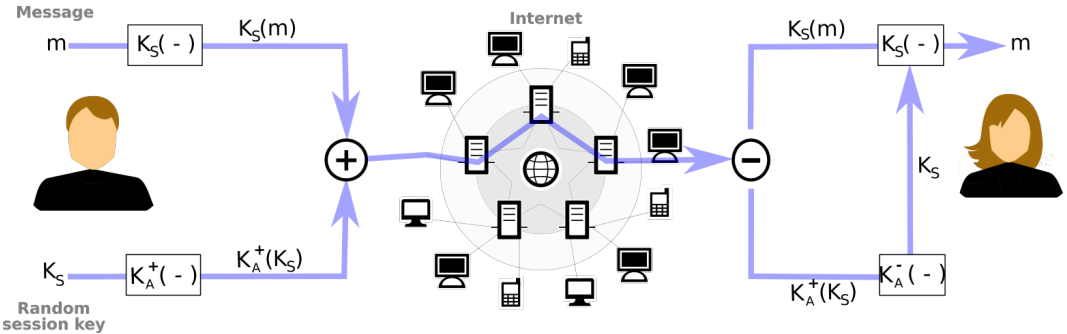
1. Confidentiality
2. Sender authentication
3. Message Integrity
4. Receiver authentication

Secure Email

Confidentiality

Secure Email

Confidentiality

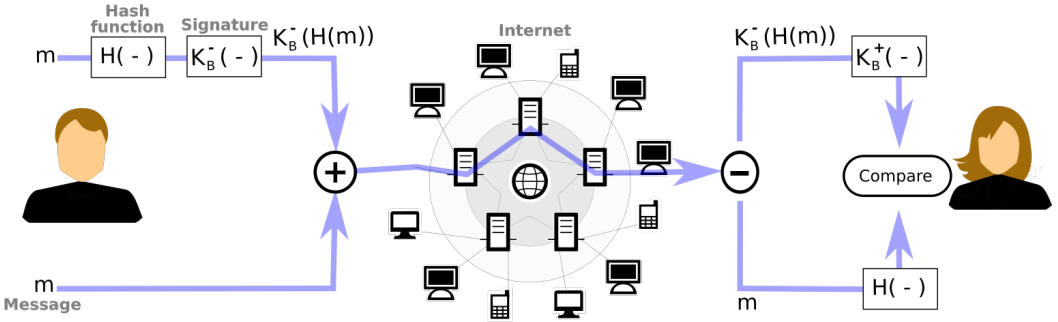


Secure Email

Sender Authentication and Message Integrity

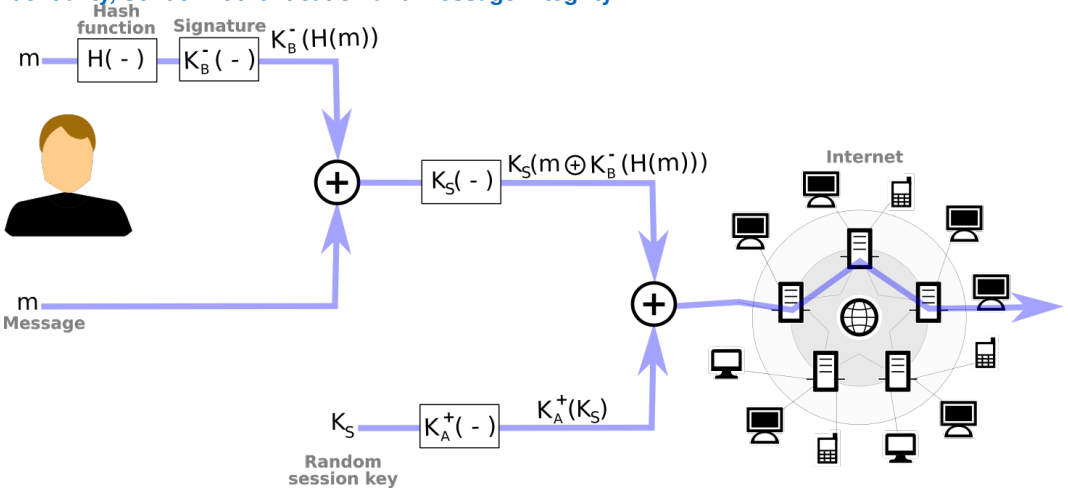
Secure Email

Sender Authentication and Message Integrity



Secure Email

Confidentiality, Sender Authentication and Message Integrity

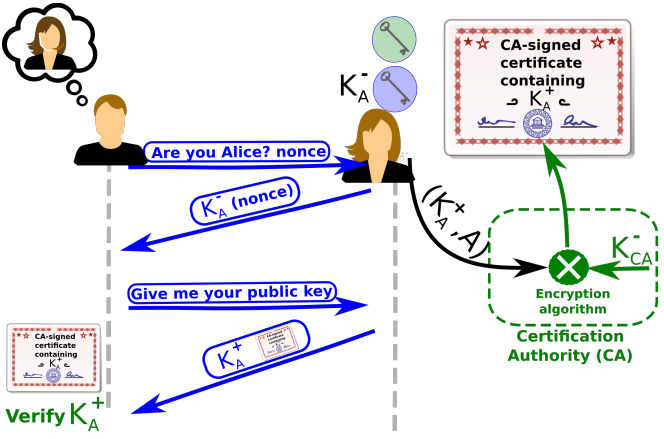


Secure Email

Receiver authentication

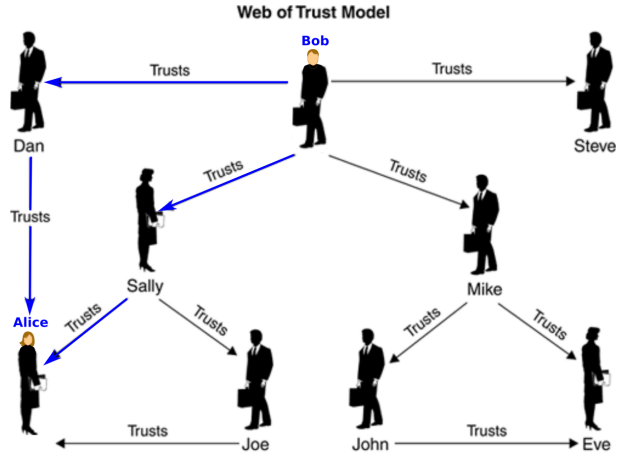
Secure Email

Receiver authentication



Secure Email

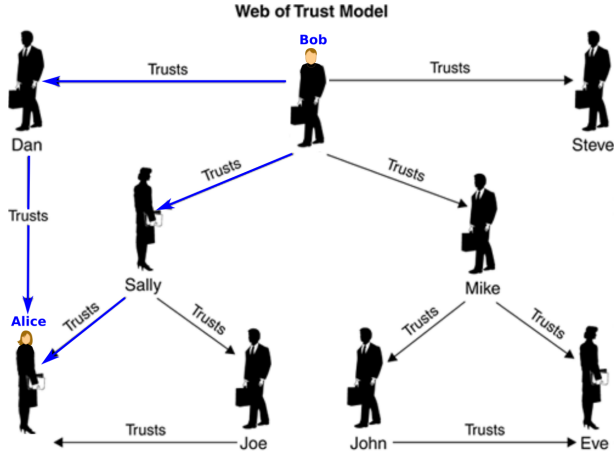
Web of Trust



Secure Email

Web of Trust

PGP

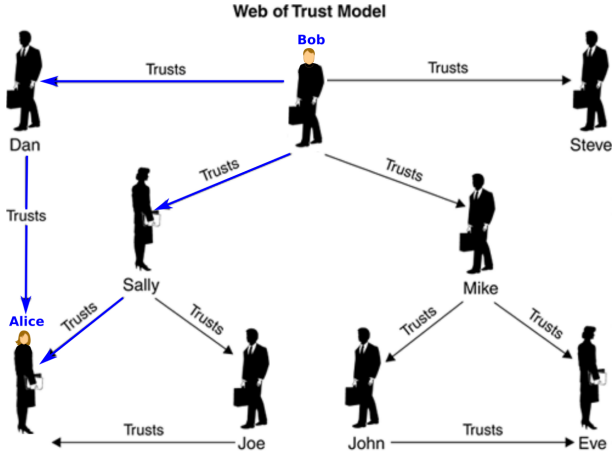


Secure Email

Web of Trust

PGP

- Implements the above scheme
- Replaces trusted third party with Web of Trust

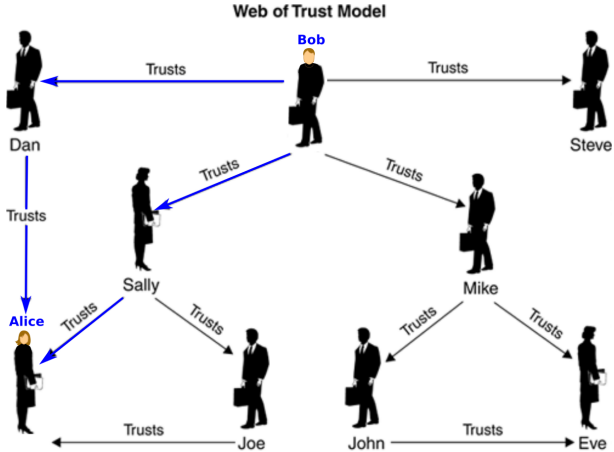


Secure Email

Web of Trust

PGP

- Implements the above scheme
- Replaces trusted third party with Web of Trust
- Idea:
  - Each party accumulates public keys of actors she/he trusts (published).
  - This way, trust-chains evolve
  - Decentralized



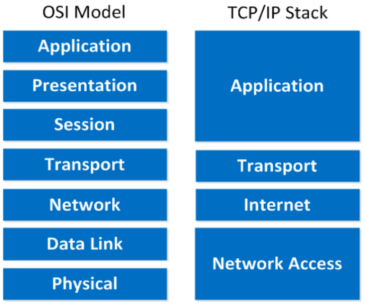
Aalto University
School of Electrical
Engineering

Part II (20 min)

Securing TCP connections

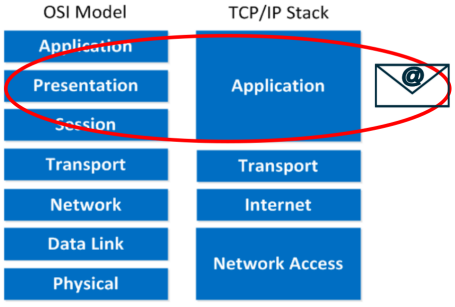
Securing TCP connections

SSL – introduction



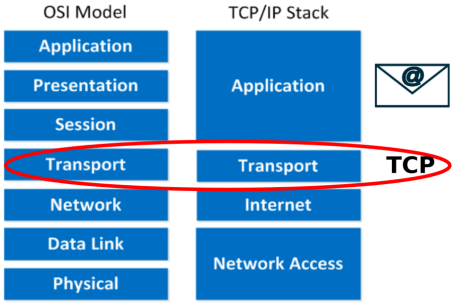
Securing TCP connections

SSL – introduction



Securing TCP connections

SSL – introduction

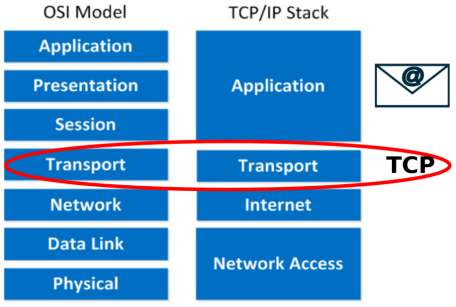


Securing TCP connections

SSL – introduction

Security Services for TCP

- confidentiality
- data integrity
- end-point authentication



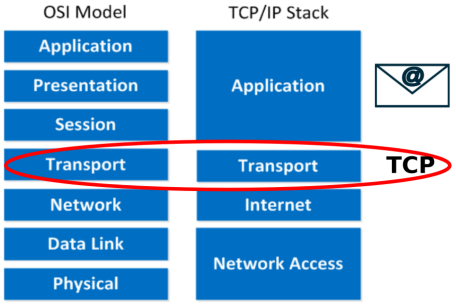
Securing TCP connections

SSL – introduction

Security Services for TCP

- confidentiality
- data integrity
- end-point authentication

In short: Secure Sockets Layer (SSL)



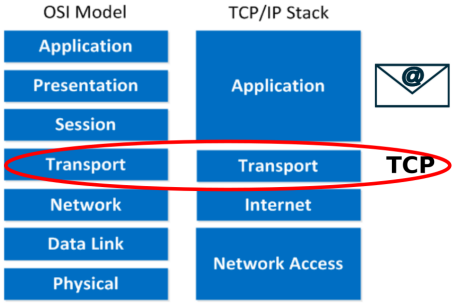
Securing TCP connections

SSL – introduction

Security Services for TCP

- confidentiality
- data integrity
- end-point authentication

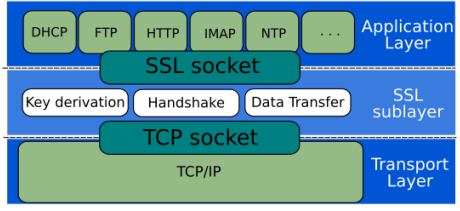
In short: Secure Sockets Layer (SSL)



A modified version of SSL v3 (Transport Layer Security (TLS)) has been standardized by the IETF (RFC 2246)

Securing TCP connections

SSL – historical remarks

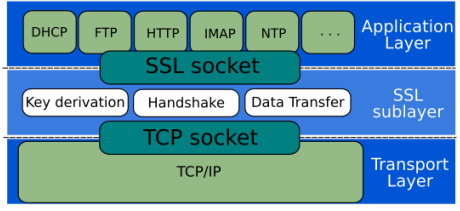


[a] Woo et al. SNP: an interface for secure network programming, USENIX 1994.

Securing TCP connections

SSL – historical remarks

1994 Ideas around securing TCP discussed in academia[a]



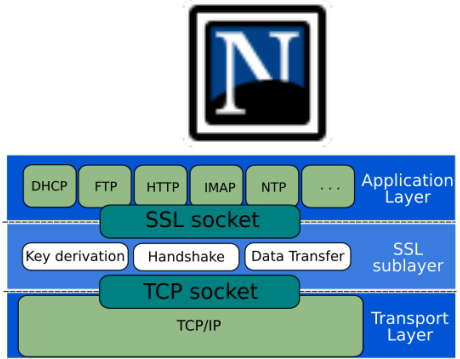
[a] Woo et al. SNP: an interface for secure network programming, USENIX 1994.

Securing TCP connections

SSL – historical remarks

1994 Ideas around securing TCP discussed in academia[a]

1994 SSL designed by Netscape



[a] Woo et al. SNP: an interface for secure network programming, USENIX 1994.

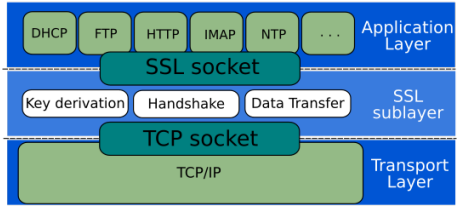
Securing TCP connections

SSL – historical remarks

1994 Ideas around securing TCP discussed in academia[a]

1994 SSL designed by Netscape

Support: all popular web browsers and web servers and all Internet comerce sites.



[a] Woo et al. SNP: an interface for secure network programming, USENIX 1994.

Securing TCP connections

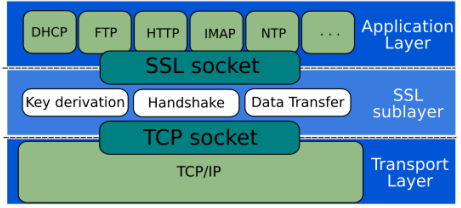
SSL – historical remarks

1994 Ideas around securing TCP discussed in academia[a]

1994 SSL designed by Netscape

Support: all popular web browsers and web servers and all Internet comerce sites.

https SSL is used by the browser when the URL begins with https



[a] Woo et al. SNP: an interface for secure network programming, USENIX 1994.

Securing TCP connections

SSL – historical remarks

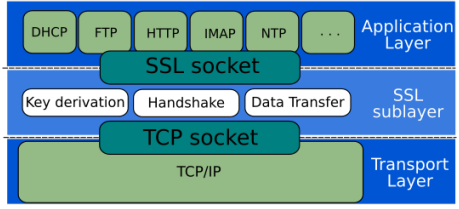
1994 Ideas around securing TCP discussed in academia[a]

1994 SSL designed by Netscape

Support: all popular web browsers and web servers and all Internet comerce sites.

https SSL is used by the browser when the URL begins with https

API API with sockets. Similar to TCP's API

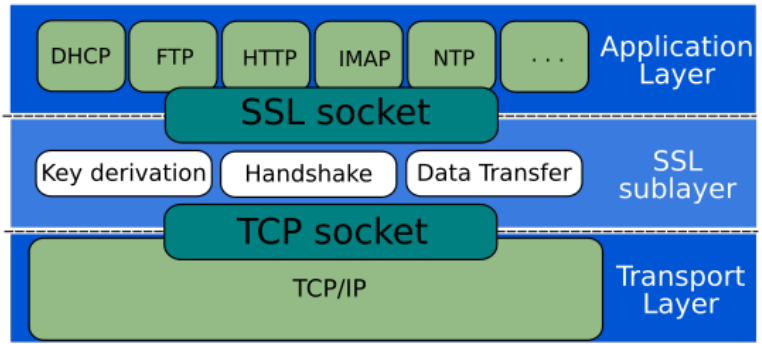


[a] Woo et al. SNP: an interface for secure network programming, USENIX 1994.

Securing TCP connections

SSL – three phases

- Handshake
- Key derivation
- Data transfer

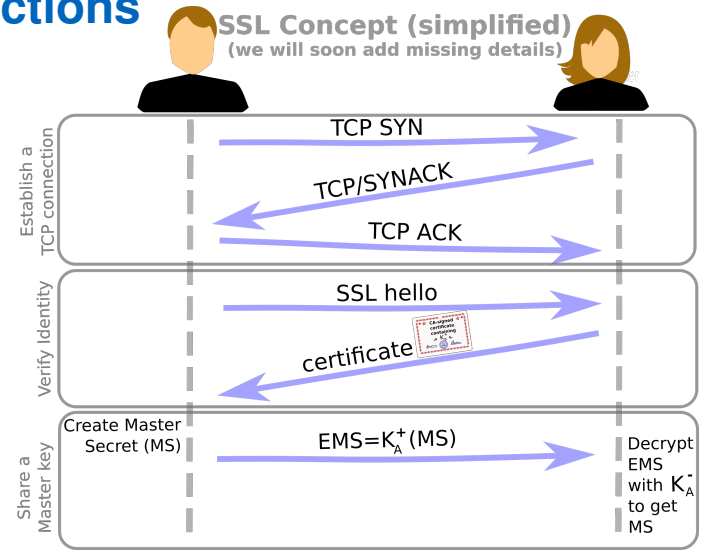


Securing TCP connections

SSL – Handshake

SSL handshake consists of

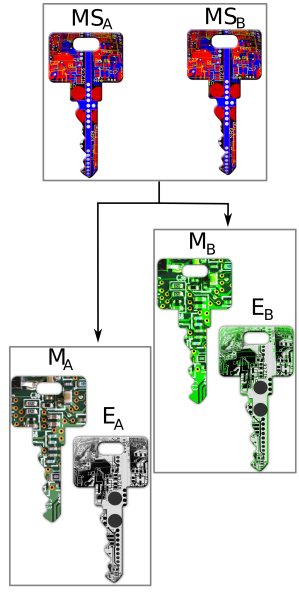
- TCP connection establishment
- Verification of the identity of the communication partner
- Generating a master key



Securing TCP connections

SSL – Key derivation

To increase security, four keys are generated from the MS



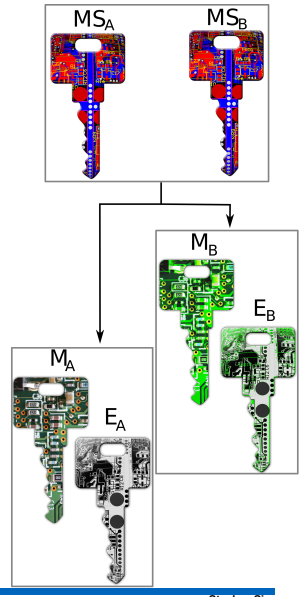
Securing TCP connections

SSL – Key derivation

To increase security, four keys are generated from the MS

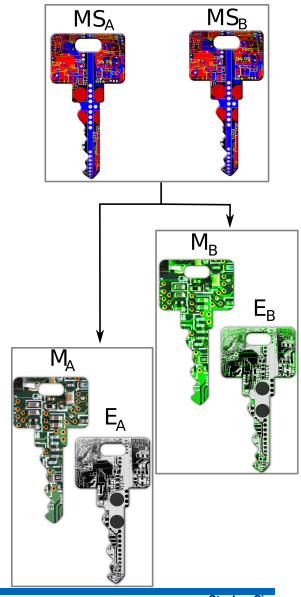
E_B session encryption key for data sent from Bob to Alice

E_A session encryption key for data sent from Alice to Bob



Securing TCP connections

SSL – Key derivation

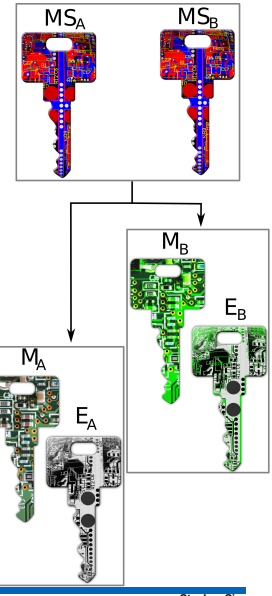


To increase security, four keys are generated from the MS

- E_B session encryption key for data sent from Bob to Alice
- M_B session MAC key for data sent from Bob to Alice
- E_A session encryption key for data sent from Alice to Bob
- M_A session MAC key for data sent from Alice to Bob

Securing TCP connections

SSL – Key derivation



To increase security, four keys are generated from the MS

- E_B session encryption key for data sent from Bob to Alice
- M_B session MAC key for data sent from Bob to Alice
- E_A session encryption key for data sent from Alice to Bob
- M_A session MAC key for data sent from Alice to Bob

E_A, E_B

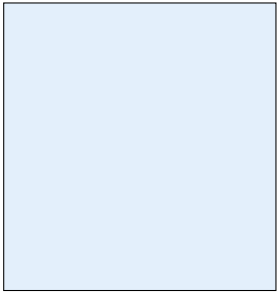
Encryption keys used to encrypt messages,

M_A, M_B

MAC keys to verify integrity of data

Securing TCP

SSL – Data transfer



Securing TCP

SSL – Data transfer

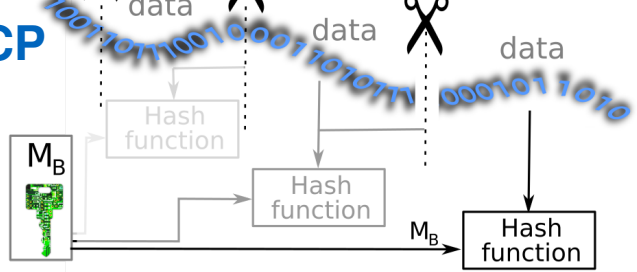
SSL breaks data streams into slices

Securing TCP

SSL – Data transfer

SSL breaks data streams into slices

Create a MAC for integrity

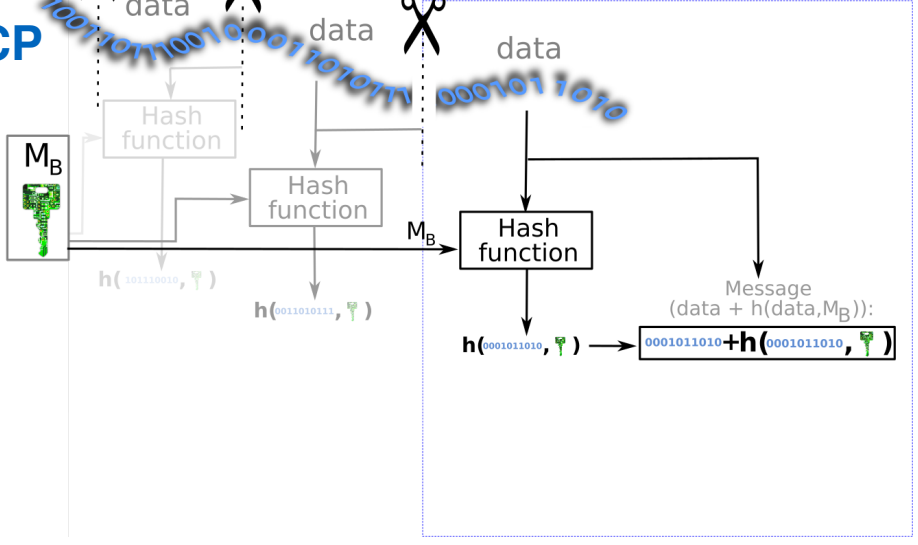


Securing TCP

SSL – Data transfer

SSL breaks data streams into slices

Create a MAC for integrity



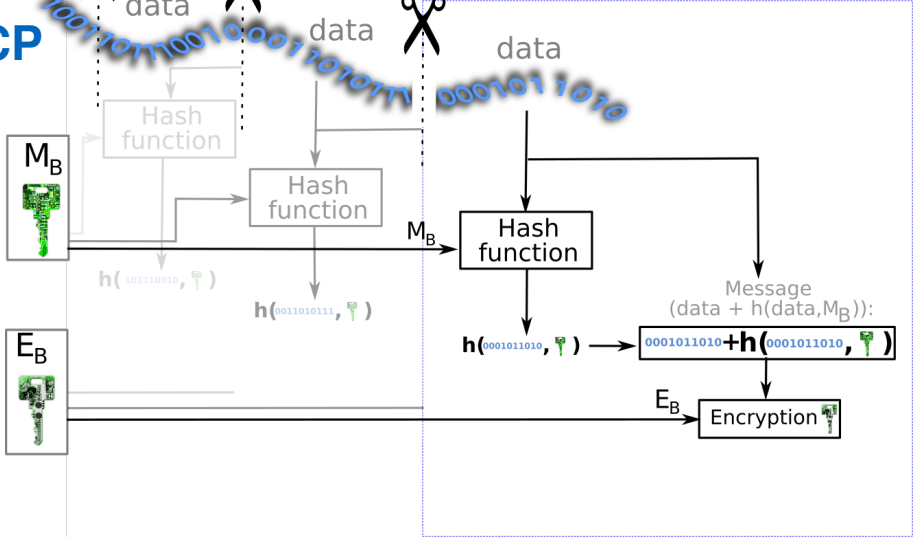
Securing TCP

SSL – Data transfer

SSL breaks data streams into slices

Create a MAC for integrity

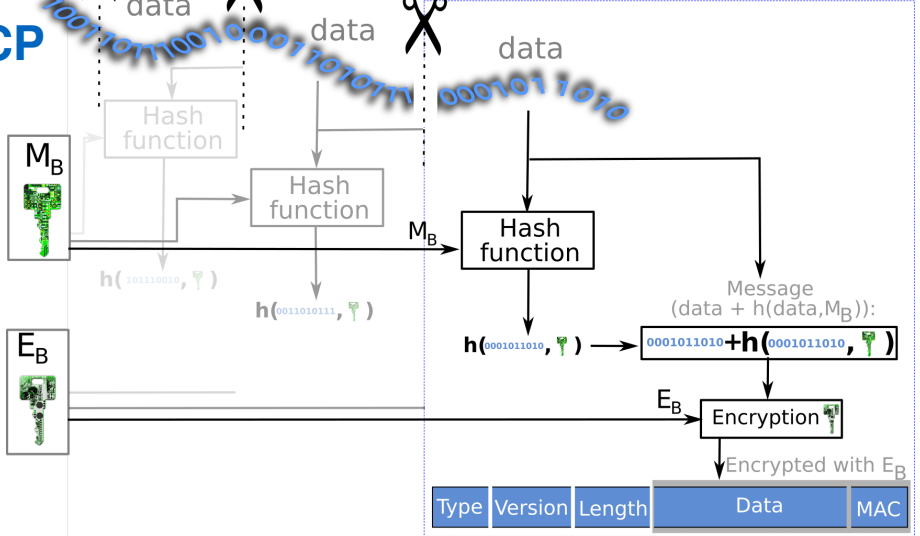
Encrypt the data and the MAC



Securing TCP

SSL – Data transfer

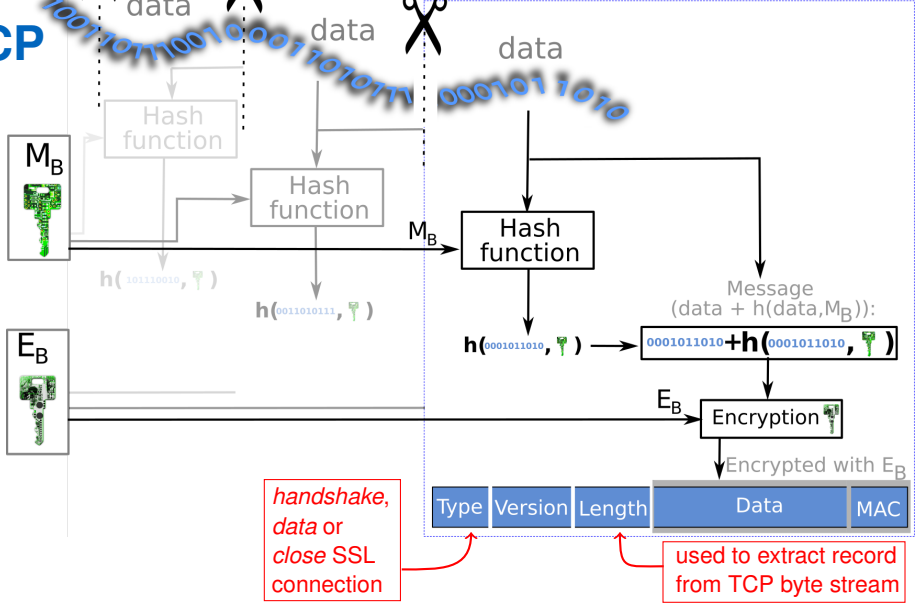
- SSL breaks data streams into slices
- Create a MAC for integrity
- Encrypt the data and the MAC
- Pass to TCP



Securing TCP

SSL – Data transfer

- SSL breaks data streams into slices
- Create a MAC for integrity
- Encrypt the data and the MAC
- Pass to TCP



Securing TCP

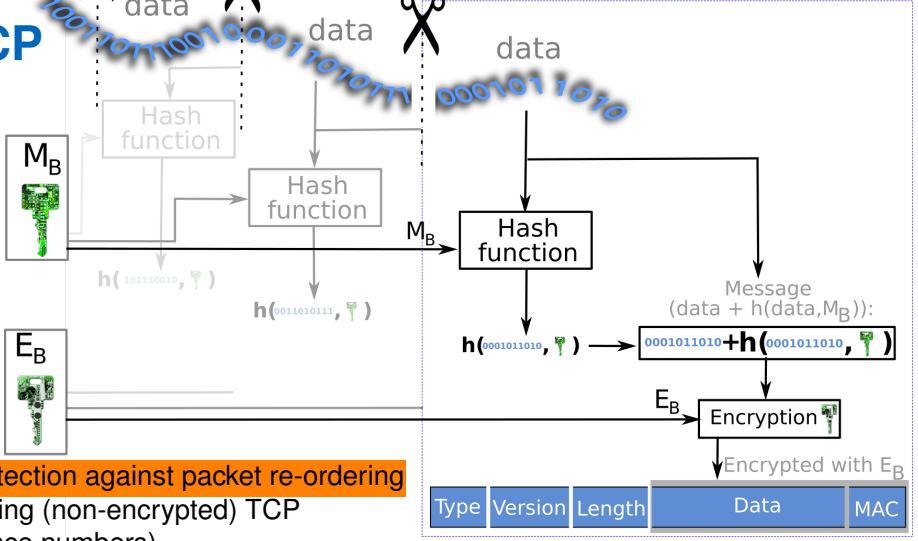
SSL – Data transfer

SSL breaks data streams into slices

Create a MAC for integrity

Encrypt the data and the MAC

Pass to TCP



Integrity issue: No protection against packet re-ordering (changing (non-encrypted) TCP sequence numbers)

Securing TCP

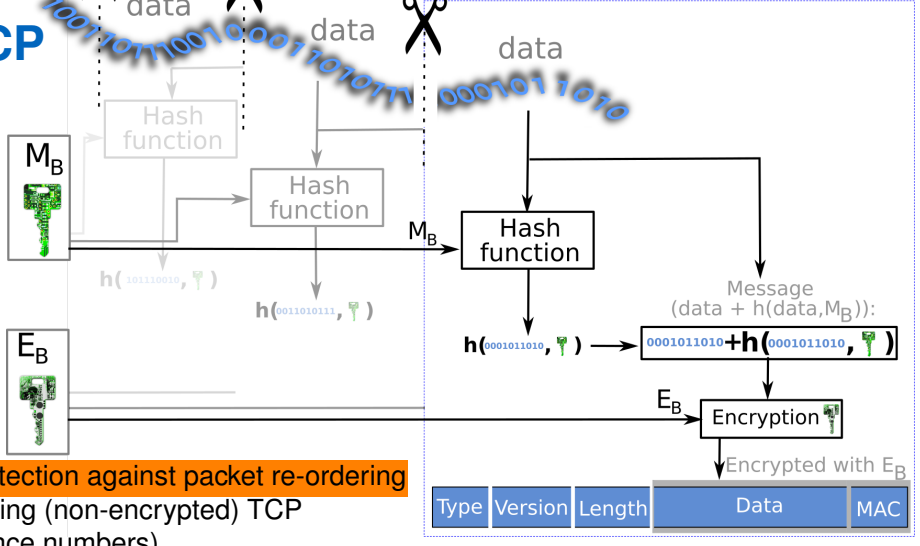
SSL – Data transfer

SSL breaks data streams into slices

Create a MAC for integrity

Encrypt the data and the MAC

Pass to TCP



Integrity issue: No protection against packet re-ordering (changing (non-encrypted) TCP sequence numbers)

Solution: Sequence numbers (included/added in the MAC calculation)

Securing TCP

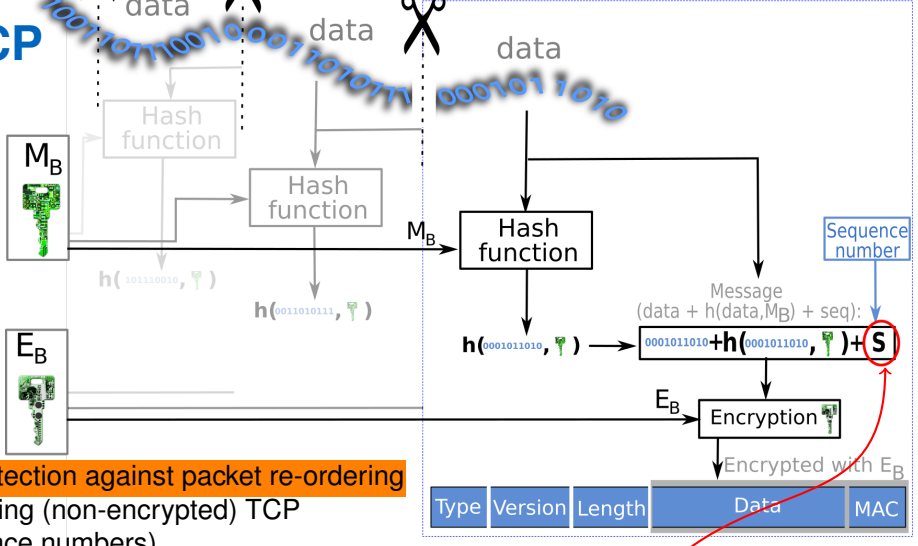
SSL – Data transfer

SSL breaks data streams into slices

Create a MAC for integrity

Encrypt the data and the MAC

Pass to TCP

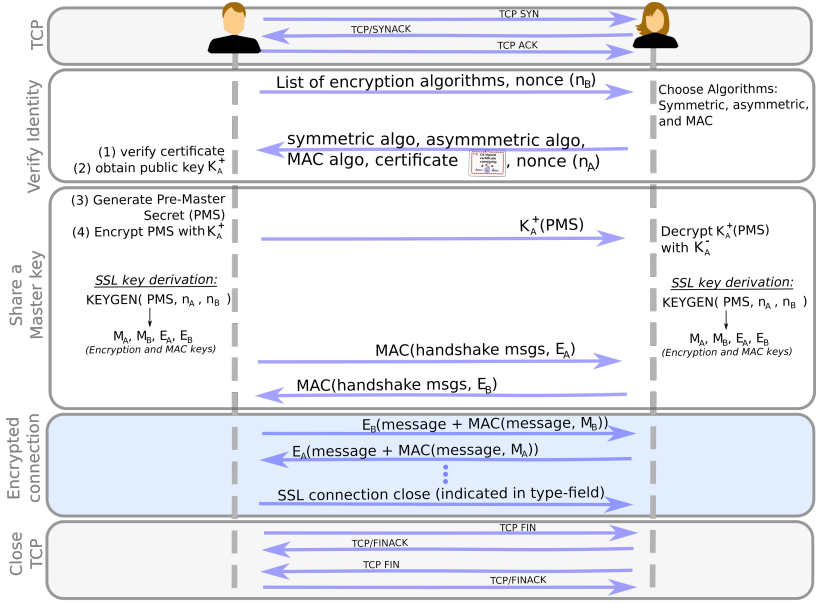


Integrity issue: No protection against packet re-ordering (changing (non-encrypted) TCP sequence numbers)

Solution: Sequence numbers (included/added in the MAC calculation)

Securing TCP

SSL – Complete handshake

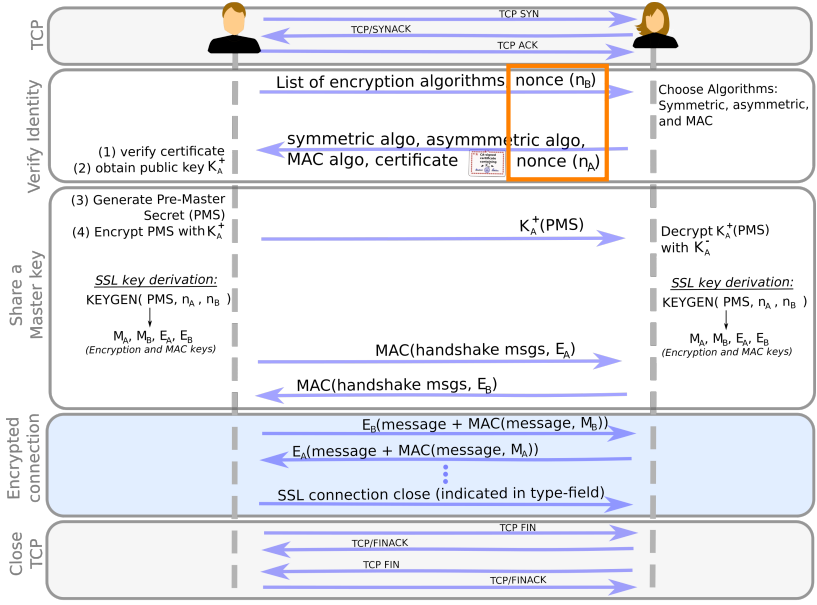


Securing TCP

SSL – Complete handshake

Handshake design choices

Why using n_A and n_B?



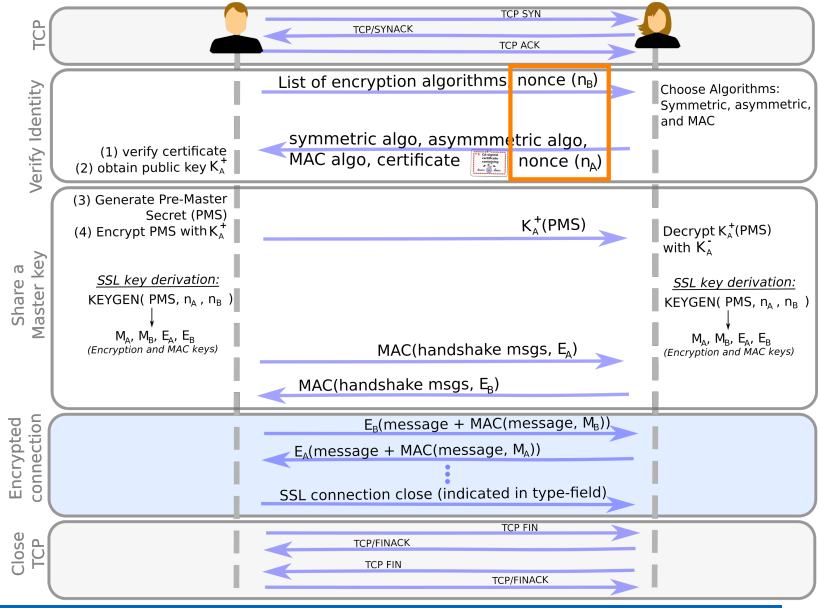
Securing TCP

SSL – Complete handshake

Handshake design choices

Why using n_A and n_B?

Nonces prevent connection replay attack



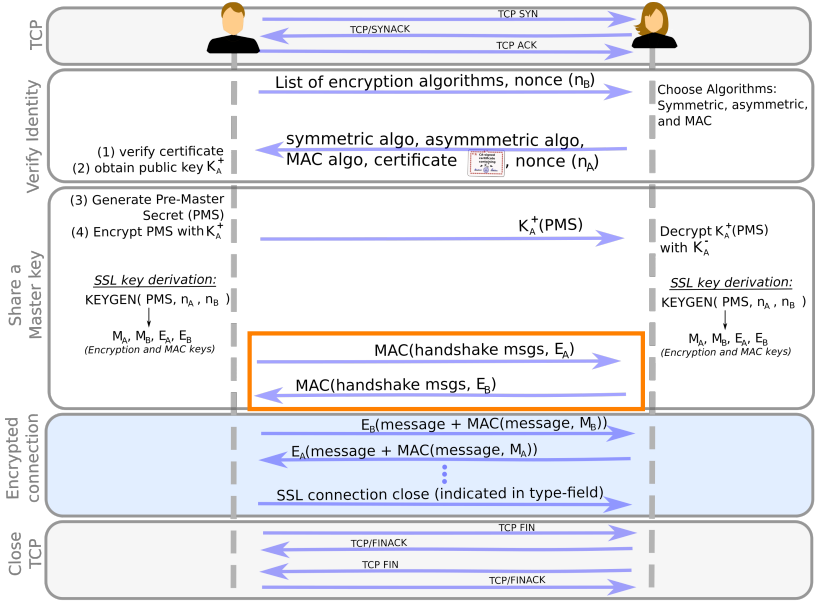
Securing TCP

SSL – Complete handshake

Handshake design choices

Why using n_A and n_B?

Why compute MAC of all handshake messages?



Securing TCP

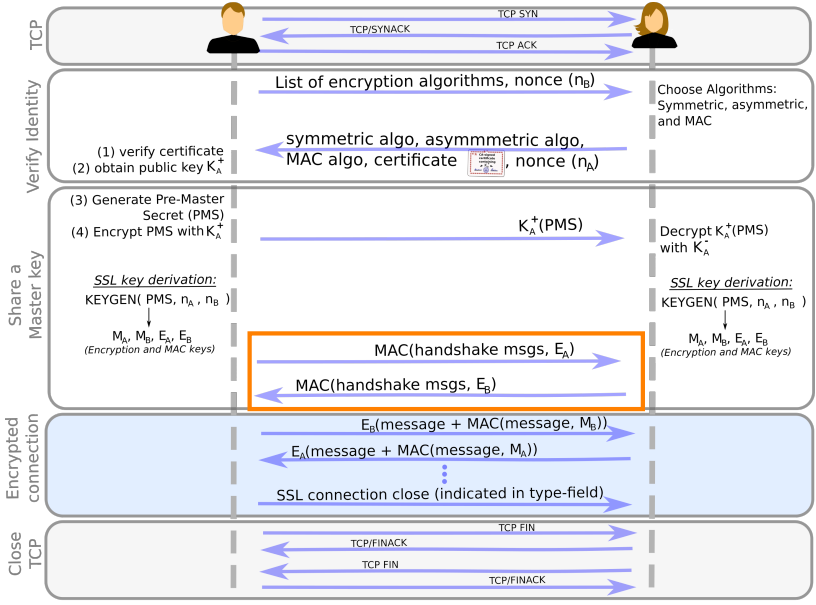
SSL – Complete handshake

Handshake design choices

Why using n_A and n_B?

Why compute MAC of all handshake messages?

Protect handshake from tampering (e.g. removing strong encryption algorithms)

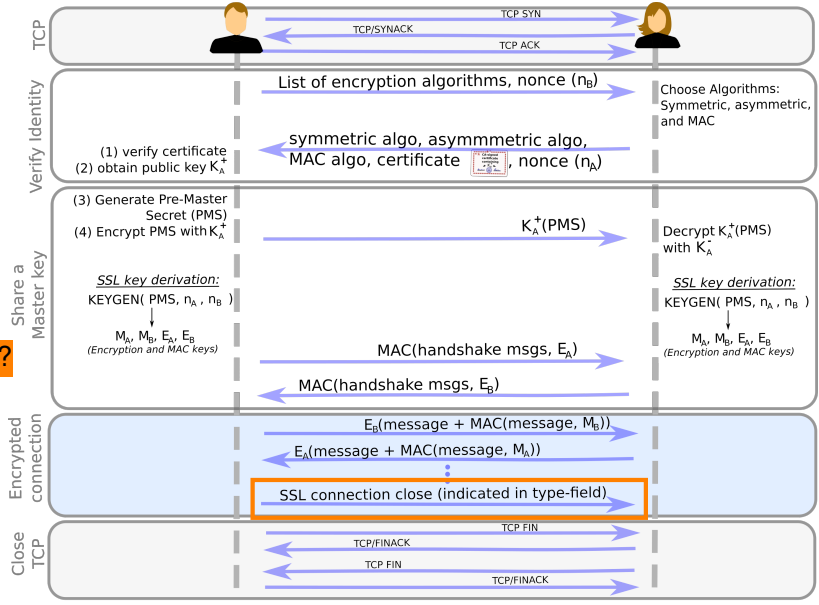


Securing TCP

SSL – Complete handshake

Handshake design choices

- Why using n_A and n_B?
- Why compute MAC of all handshake messages?
- Why SSL connection close?
- Is TCP FIN not sufficient?



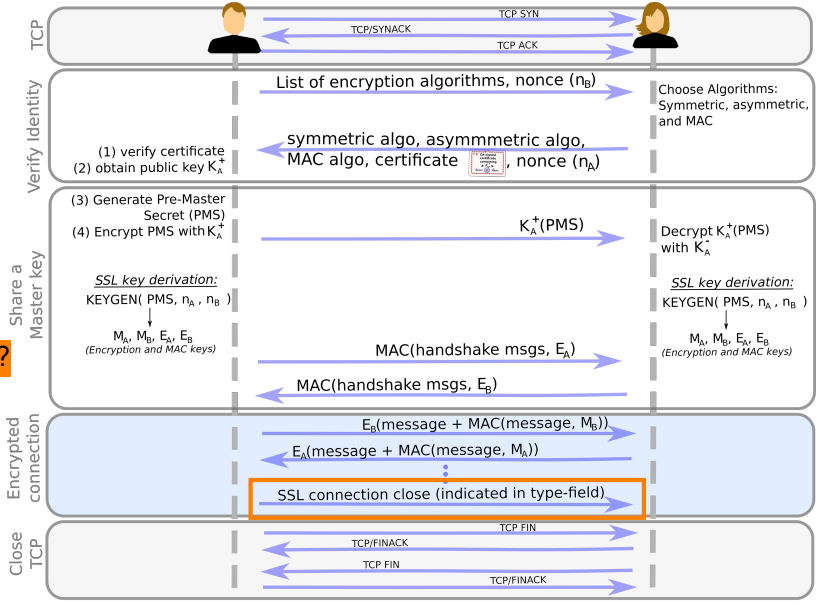
Securing TCP

SSL – Complete handshake

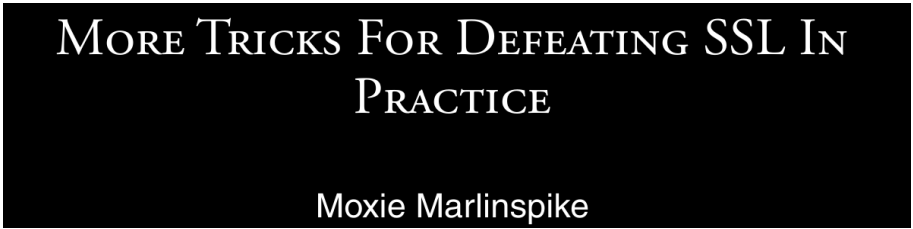
Handshake design choices

- Why using n_A and n_B?
- Why compute MAC of all handshake messages?
- Why SSL connection close?
- Is TCP FIN not sufficient?

Prevent trunctuation attack (adversary closing a connection before it is actually finished)



Further reading



http://2015.hack.lu/archive/2009/moxie-marlinspike-some_tricks_for_defeating_ssl_in_practice.pdf

Aalto University
School of Electrical
Engineering

Video: Trust and privacy in the internet (5 min)

Ron Rivest, Shafi Goldwasser, Whitfield Diffie, Tal Rabin, Paul Kocher

RSA 2019, Cryptographers panel

Questions?

Stephan Sigg
stephan.sigg@aalto.fi

Esa Vikberg
esa.vikberg@aalto.fi

Leo Lazar
leo.lazar@aalto.fi

Literature

- J.F. Kurose,K.W. Ross: Computer Networking: A Top-Down approach (7th edition), Pearson, 2016.
- J.F. Kurose,K.W. Ross: Computer Networking: A Top-Down approach (6th edition), Addison-Wesley, 2012.