IT-Security (ITS) B1

DIKU, E2020

Lecture plan

Mandag d. 28. september
- kl. 10-12 Cryptography

Fredag d. 2. oktober
- kl. 09-10 Internet security protocols (bemærk ekstra time fra kl. 9 allerede)
- kl. 10-12 Intrusion detection

Mandag d. 5. oktober
- kl. 9-11 Forensics (bemærk flyttet fra kl. 10-12 til kl. 09-11)

Crypto primitives, recap:

Symmetric encryption

Cryprographic hash functions

Message authentication codes

Asymmetric encryption

Digital signatures

Key management

Many keys to protect

Master key

Session key

Signature key

Data encryption key

Key encryption key

…

Protect during entire lifecycle

Generation

Exchange

Storage/backup

Use

Expiration

Revocation

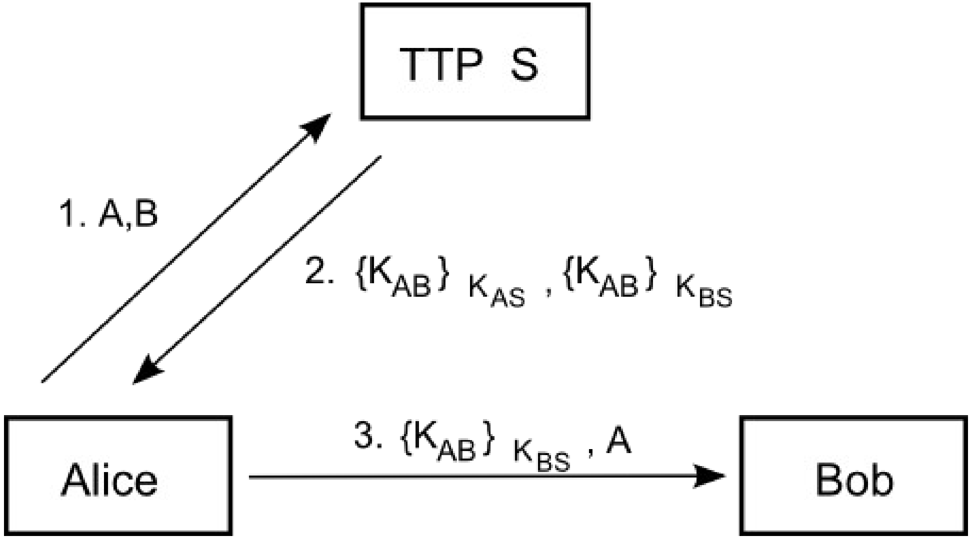
Destruction

Key exchange options include

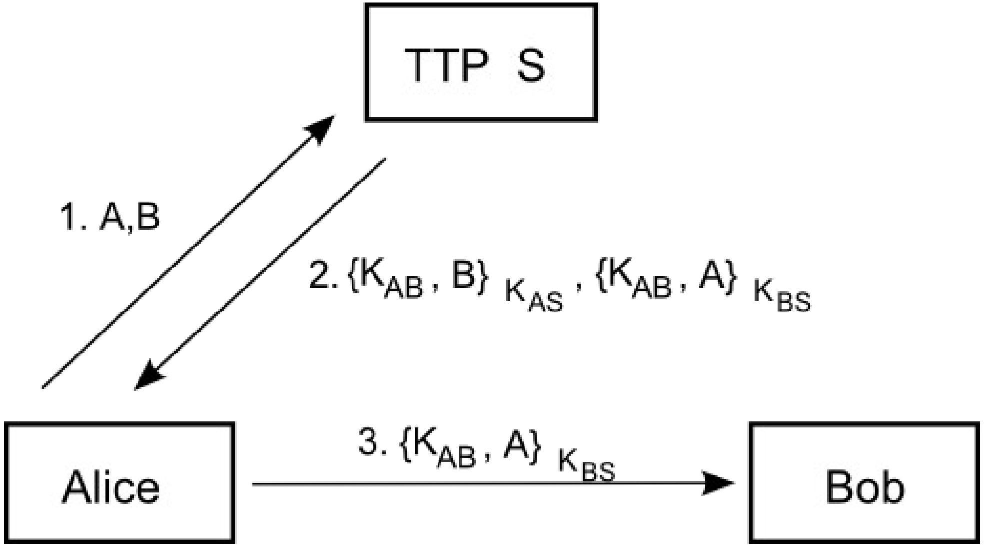
- Pre-distribution
  - Generated and distributed “ahead of time” e.g. physically
- Distribution
  - Generated by a trusted third party (TTP) and sent to all parties
- Agreement
  - Generated by all parties working together
- Asymmetric
  - Is e really yours?

Key distribution

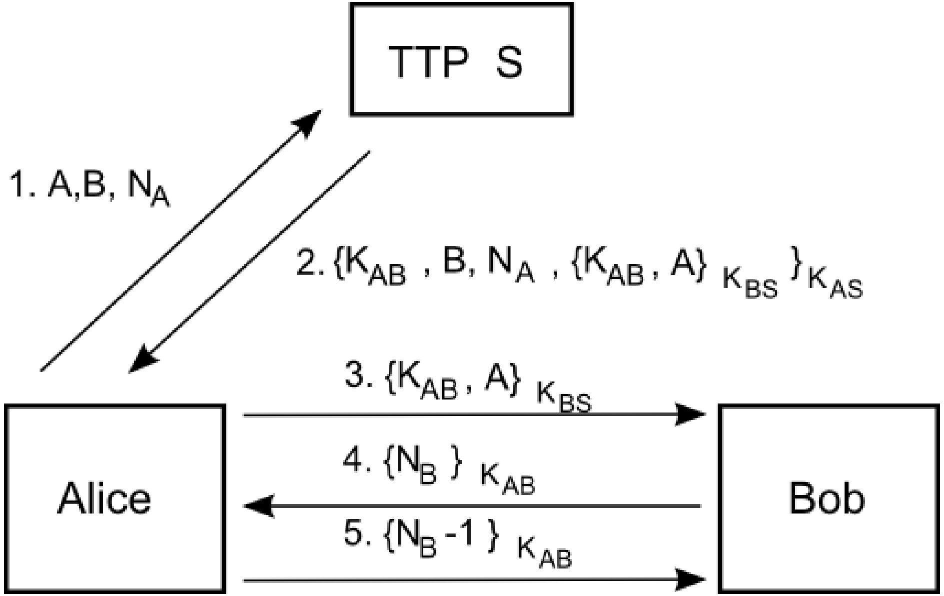
Key distribution



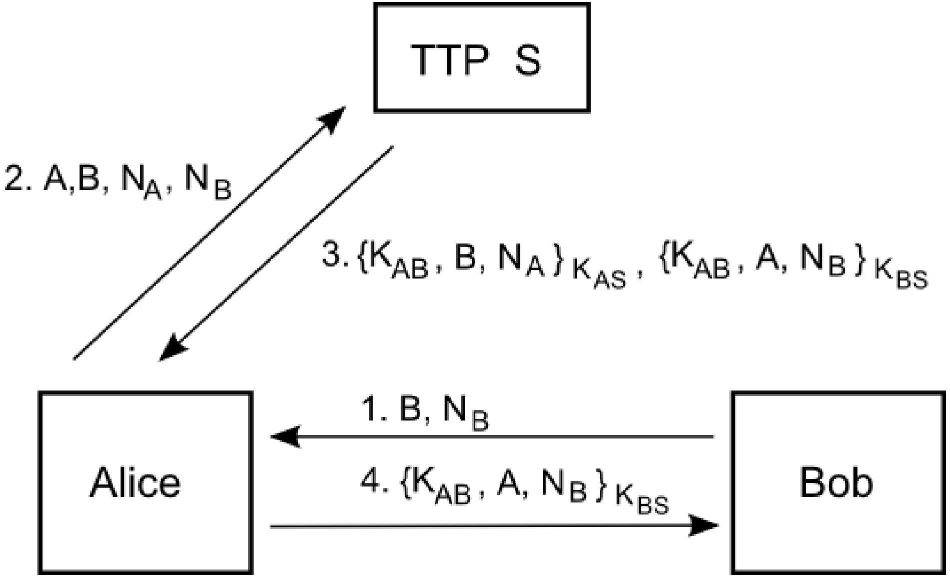
Key distribution



Key distribution



Key distribution



Key agreement

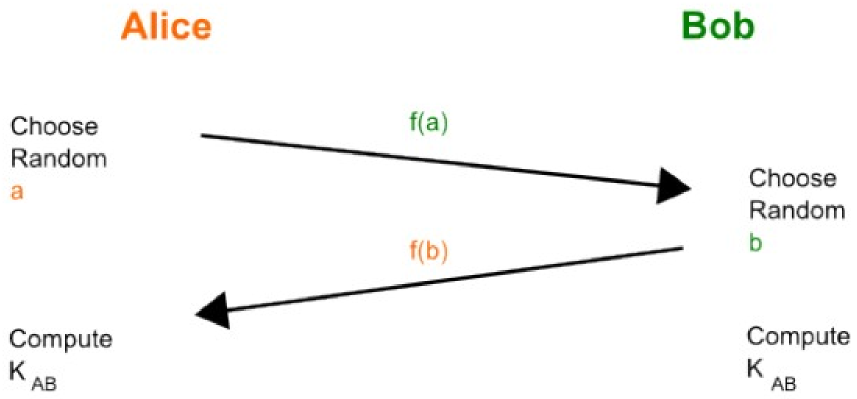
Basic idea

Choose a function f such that

$$f(a,f(b)) = f(b,f(a))$$

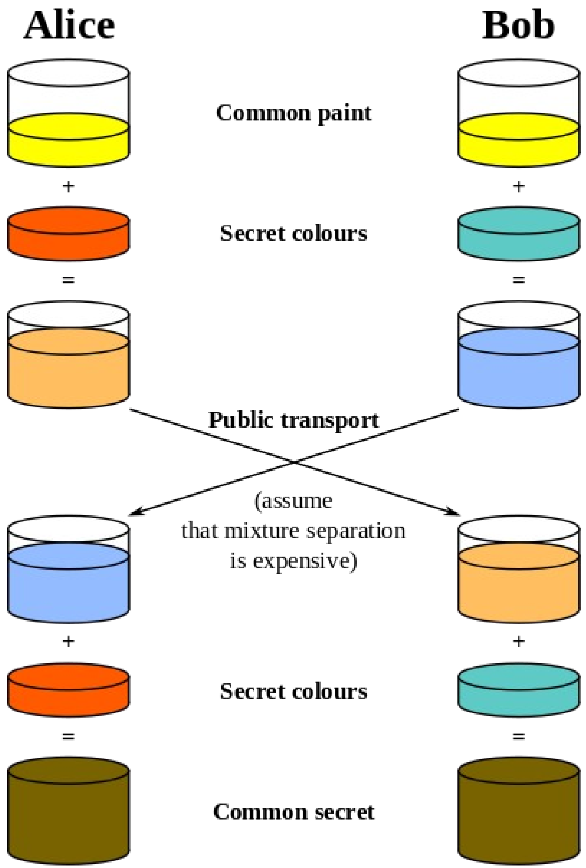
And

$$f^{-1}(x) \text{ is hard}$$



Paint would work

If you wanted to exchange secret paints



Solution by Diffie-Hellman, 1976

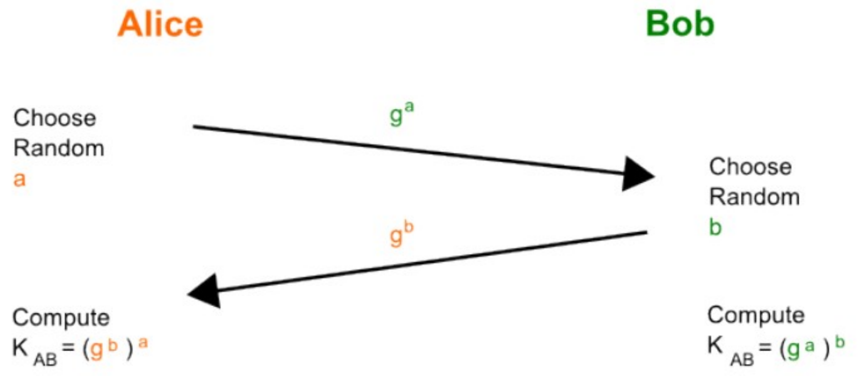
$f(x) = g^x \bmod p$

Given g^a, find x so $g^x = g^a$

Discrete logarithm problem

Given g^a and g^b, find g^{ab}

Computational Diffie-Hellman assumption



Is e really yours?

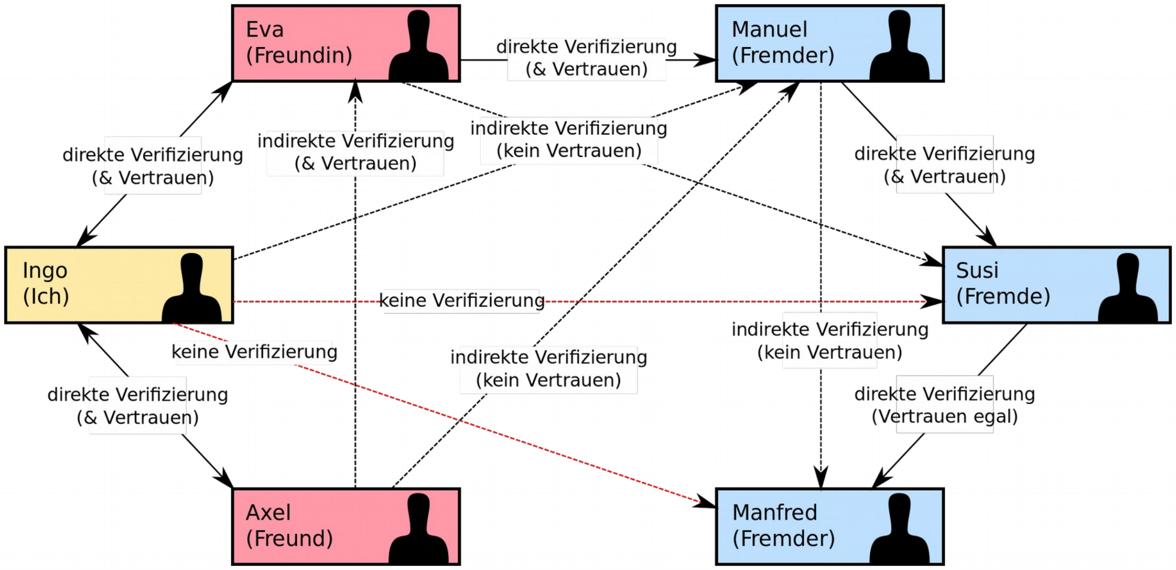
Public-key infrastructure (PKI)

A system for the creation, storage, and distribution of **digital certificates** which are used to verify that a particular public key belongs to a certain entity

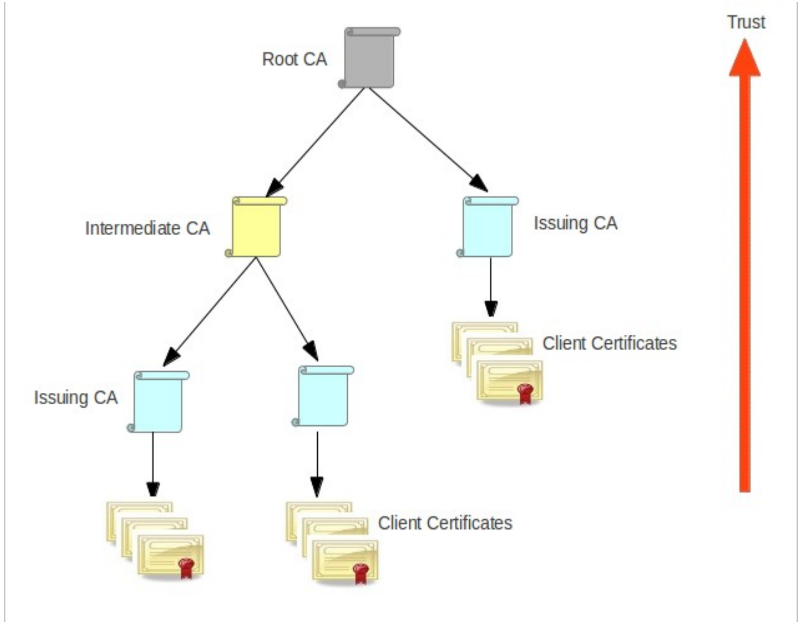
X.509 format for certificates include:

Serial number	– unique identification of certificate
Valid-From/To	– lifespan of the certificate
Subject	– the entity/person/machine/etc. identified
Public key	– the entity's public key
Signature	– the actual signature of the issuer

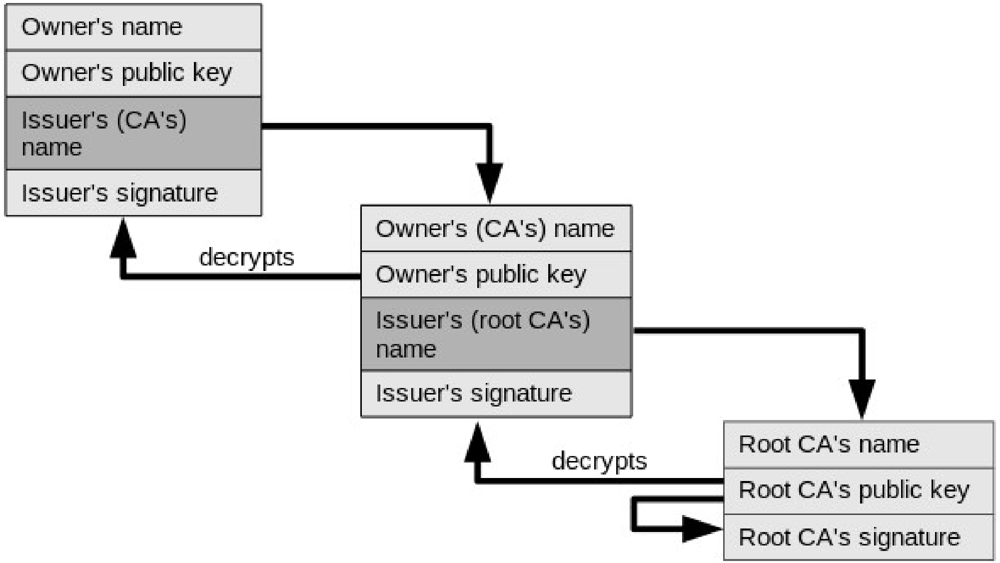
Types of PKI: Web of trust



Types of PKI: CA model

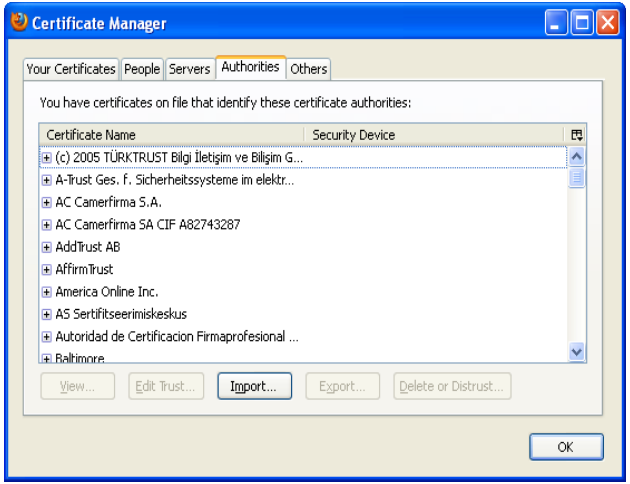


Chain of trust



Trust in browsers

Browsers come pre-configured with a set of root CAs. Do you trust all these CAs (to authenticate properly, to avoid/inform of breaches)?



Revocation of certificates

Certificate revocation list (CRL):

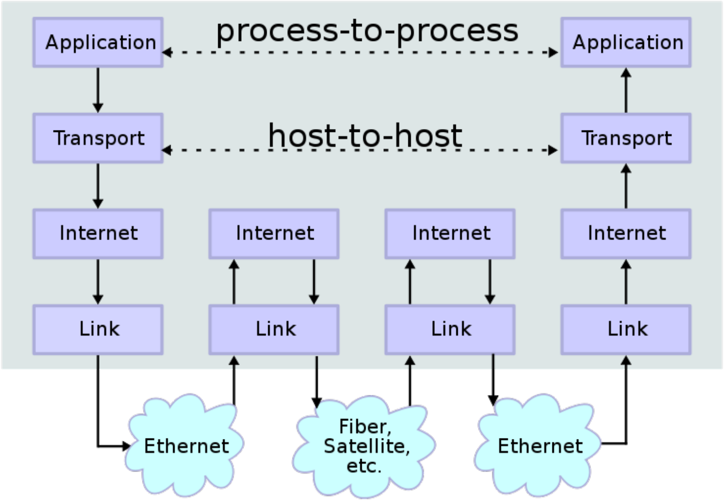
- A list of (serial numbers for) certificates that have been revoked, and therefore, entities presenting those (revoked) certificates should no longer be trusted

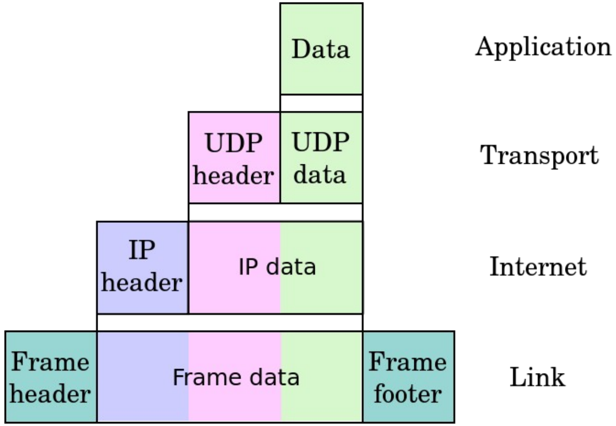
Online Certificate Status Protocol (OCSP):

- Protocol used for obtaining the revocation status of an X.509 digital certificate

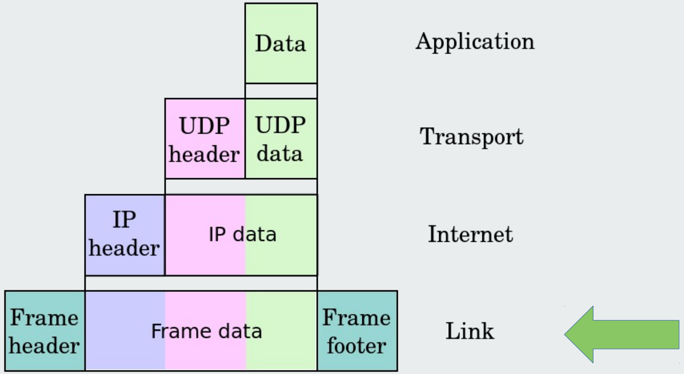
Crypto protocols

Where to encrypt?





The link layer



Wifi

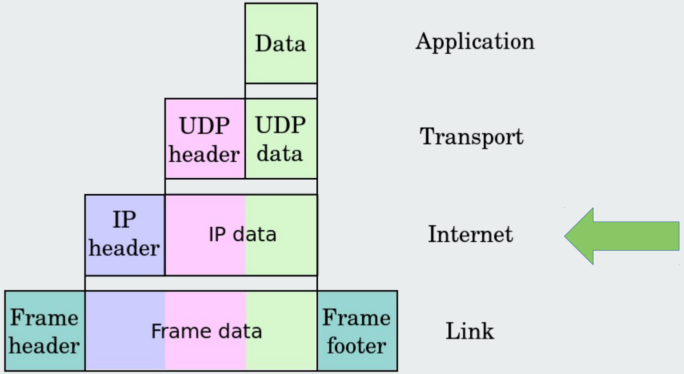
Wifi threats

Eavesdropping (!)

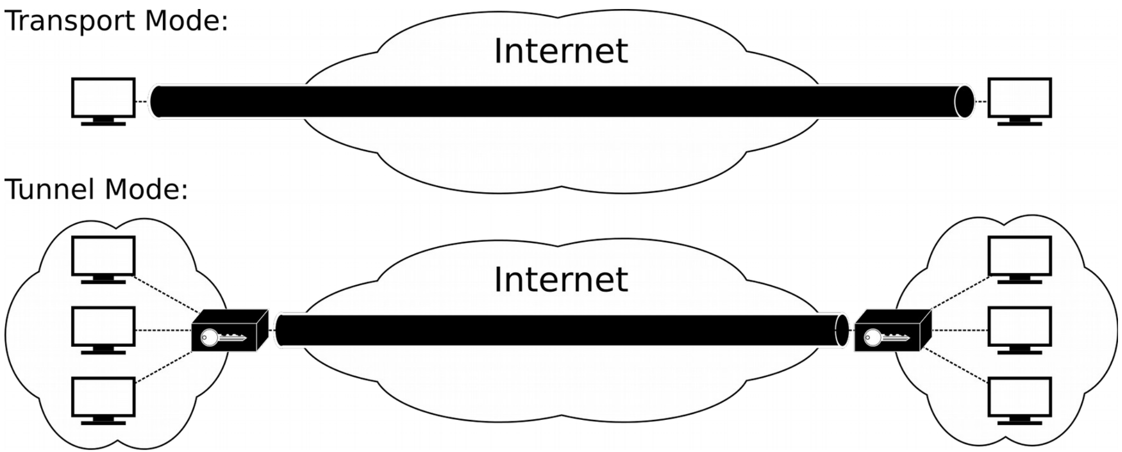
Accidential association

Malicious association

The Internet layer



IPSec – Transport or tunnel mode



IPSec – AH, ESP, SA

Authentication Header (AH)

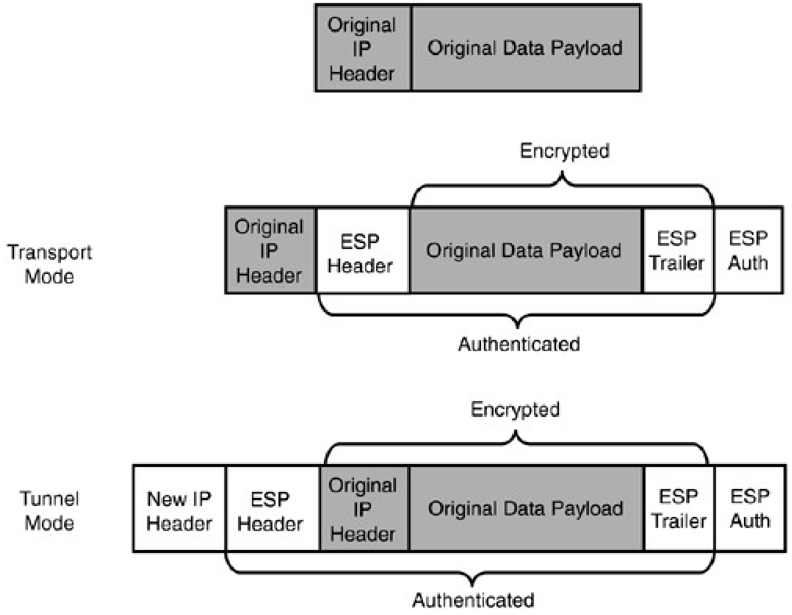
Integrity and authentication

Encapsulating Security Payload (ESP)

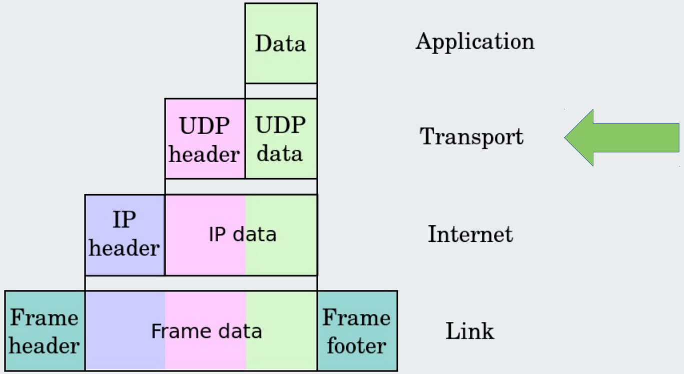
Confidentiality

Security Association

Details on ciphers, keys, lifetime, etc.
One directional



The transport layer



SSL/TLS

Transport Layer Security (TLS), and its now-deprecated predecessor, Secure Sockets Layer (SSL), are cryptographic protocols designed to:

"Provide a secure channel between two communicating peers" [RFC 8446 TLS 1.3]

SSL and TLS protocols

Protocol	Published	Status
SSL 1.0	Unpublished	Unpublished
SSL 2.0	1995	Deprecated in 2011 (RFC 6176)
SSL 3.0	1996	Deprecated in 2015 (RFC 7568)
TLS 1.0	1999	Deprecation planned in 2020[11]
TLS 1.1	2006	Deprecation planned in 2020[11]
TLS 1.2	2008	
TLS 1.3	2018	

Security goals of TLS

Specifically, the secure channel should provide the following properties:

Authentication: The server side of the channel is always authenticated; the client side is optionally authenticated.

Confidentiality: Data sent over the channel after establishment is only visible to the endpoints.

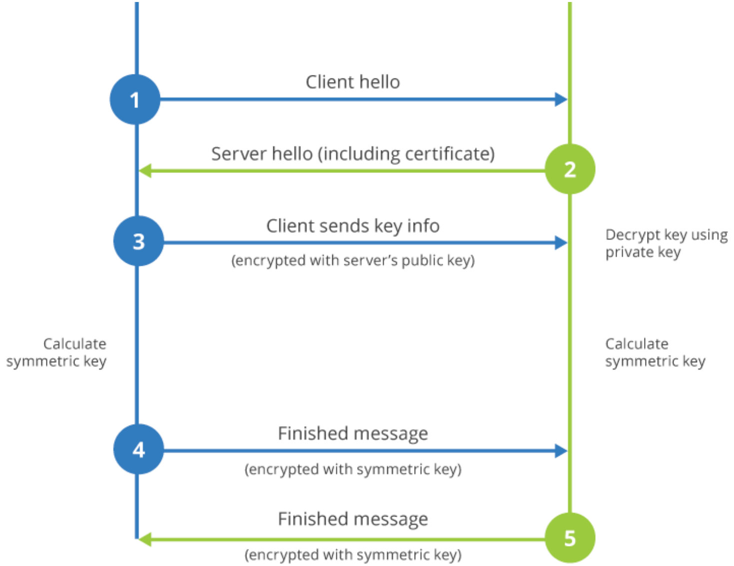
Integrity: Data sent over the channel after establishment cannot be modified by attackers without detection.

Two primary components

A handshake protocol that authenticates the communicating parties, negotiates cryptographic modes and parameters, and establishes shared keying material

A record protocol that uses the parameters established by the handshake protocol to protect traffic between the communicating peers

Handshake protocol (TLS 1.2)

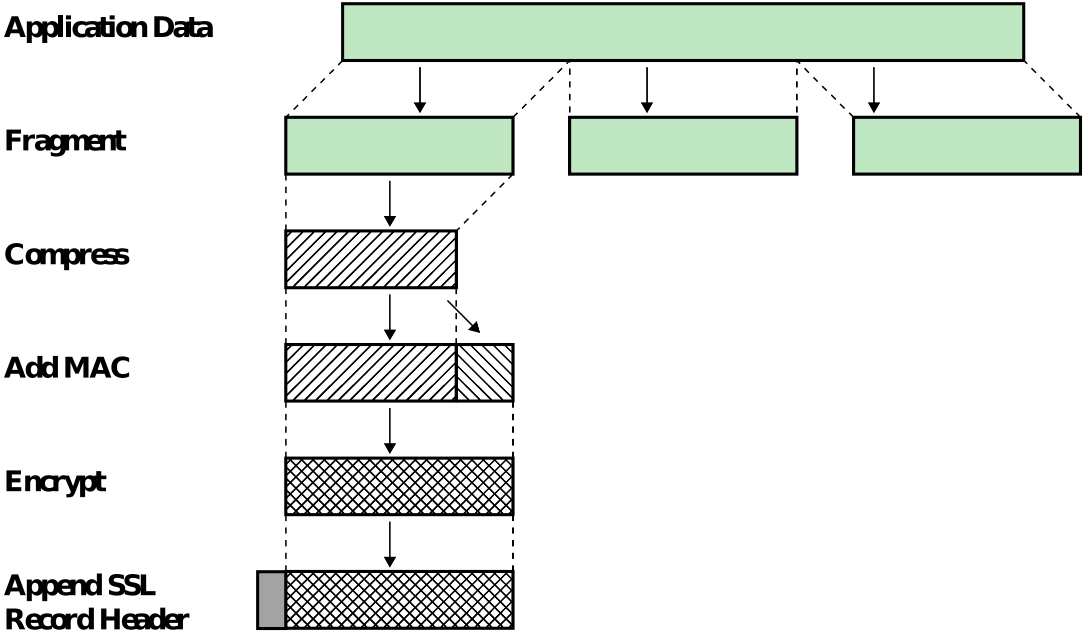


Example ciphersuite (TLS 1.2)

TLS_RSA_WITH_AES_128_CBC_SHA256

TLS	→ TLS 1.2
RSA	→ RSA key exchange
WITH	→ merely filler
AES_128	→ 128-bit AES encryption
CBC	→ Cipher block chaining
SHA256	→ HMAC-SHA256 digest

Record protocol (TLS 1.2)



RSA key exchange (TLS 1.2)

Server has a certificate with a RSA public key

Client creates random “pre-master secret” and encrypts it with the server's public key

What happens if an attacker learns the private key?

Forward secrecy

Forward secrecy, aka perfect forward secrecy, protects past sessions against future compromises by generating *a unique session key for every session* a user initiates, e.g.:

1. Alice and Bob each generate a pair of long-term, asymmetric RSA keys
2. Alice and Bob use Diffie–Hellman to securely agree on a one-time session key (They use the keys from step 1 only to authenticate one another during this process)
3. Alice sends Bob a message, encrypting it with a symmetric cipher using the session key negotiated in step 2
4. Bob decrypts Alice's message using the key negotiated in step 2

The process repeats for each new message sent, starting from step 2

DHE_RSA instead of RSA

TLS_**DHE_RSA**_WITH_AES_128_CBC_SHA256

Server has a certificate with an RSA public key but it's only used for signature

When a client connects, the server generates a new DH key pair and sends the public key to the client; the server signs that message with its permanent RSA private key

The client will respond with a a newly-generated DH public key.

This way, DH yields a fresh **pre-master secret**

Pre-master secret → master secret

master_secret =

HMAC-SHA256 (pre_master_secret, "master secret", ClientHello.random + ServerHello.random)

Master secret → keys used for Records

See RFC5246 TLS 1.2

Ah, I feel safe now

Attacks on SSL/TLS

Renegotiation attack: Marsh Ray (2009)

Timing attacks: Lucky Thirteen

Padding attack: POODLE

Downgrade attack: DROWN

Compression attack: CRIME, BREACH

Faulty CBC in TLS 1.0: BEAST

Implementation attack: HEARTBLEED, GOTO FAIL

What to do?

Use TLS 1.3 (RFC8446)

Removed - static RSA handshake; now only DHE

Removed - insecure modes and ciphers; now only authenticated encryption

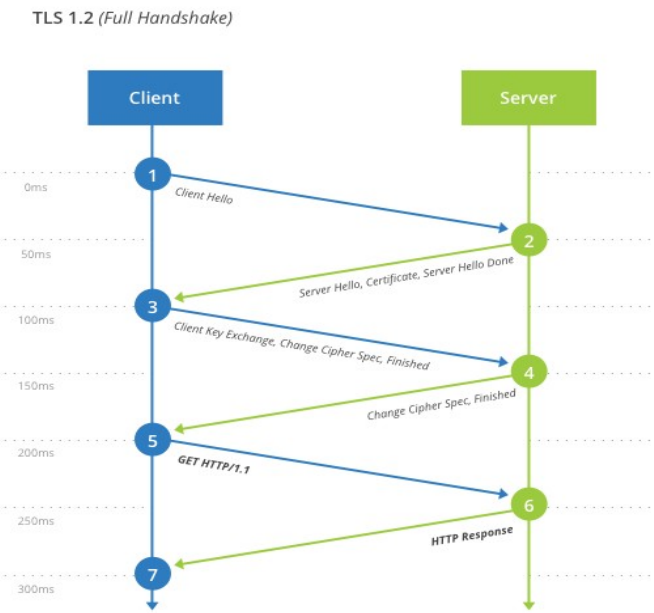
Removed – compression and renegotiation

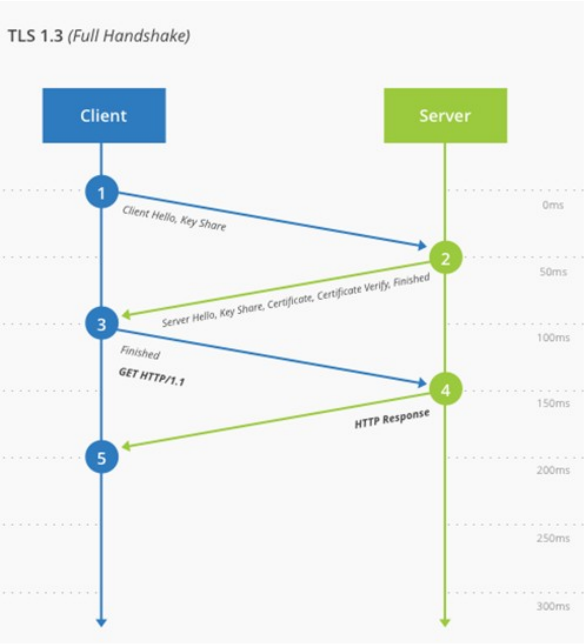
Added – full handshake signature

Added – downgrade protection

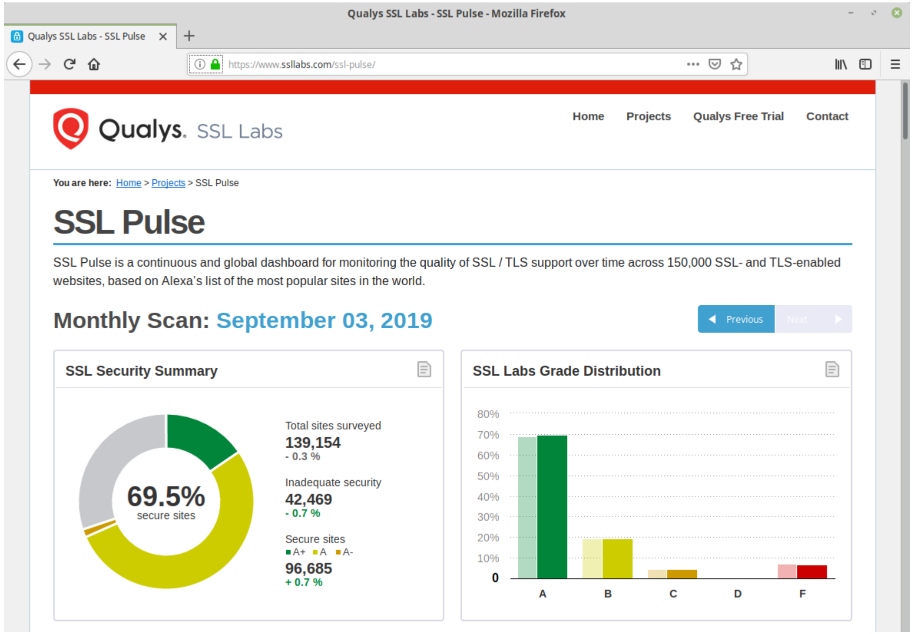
Added – better performance

TLS 1.2 vs TLS 1.3

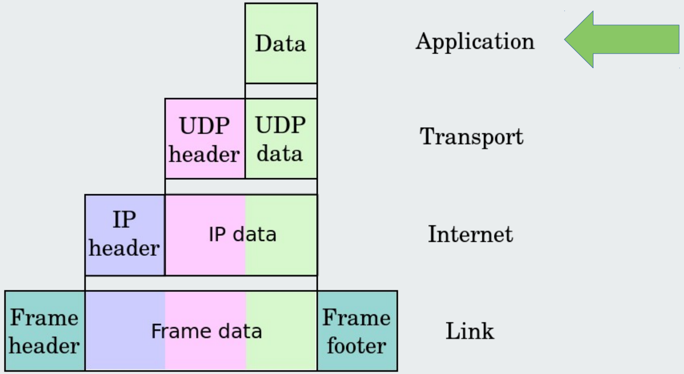




State of SSL/TLS usage



The application layer



Browsing

HTTP over SSL/TLS (HTTPS)

The security of HTTPS lies mainly in the underlying SSL/TLS, but other security mechanism also apply, e.g.:

- HTTP Public Key Pinning (HPKP)
- The lock icon

HTTP Public Key Pinning (HPKP)

HPKP is a security mechanism that can be enabled at the web server

A web server with HPKP will send an HTTP header with each response that looks like this:

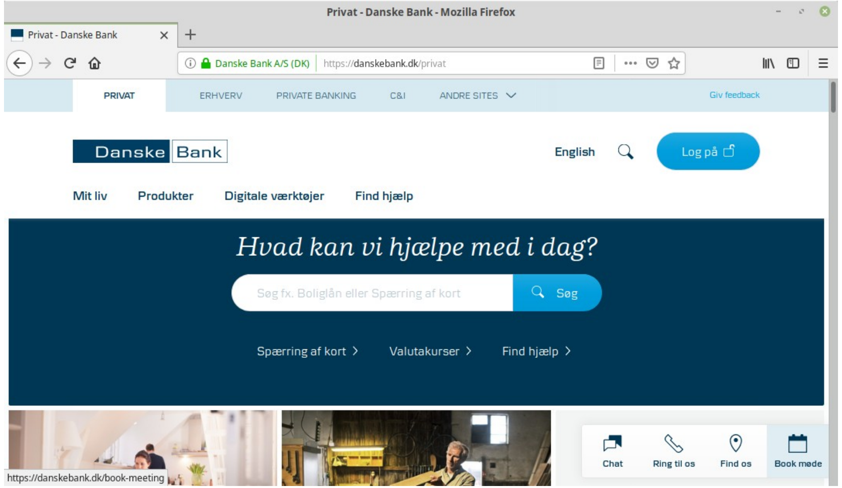
> Public-Key-Pins:
> pin-sha256="8RoC2kEF47SCVwX8Er+UBJ44pDfDZY6Ku5mm9bSXT3o=";
> pin-sha256="78j8kS82YGC1jbX4Qeavl9ps+ZCzb132wCvAY7AxTMw=";
> max-age=31536000;

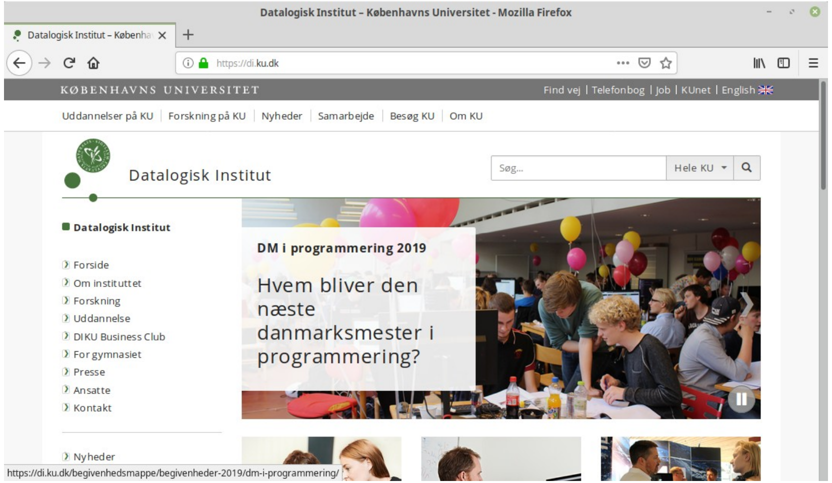
This header tells browsers to refuse any certificate not signed with one of the two public keys identified by their SHA256 sum for “max-age” seconds after visiting the site

What happens if the web site loose control over the public keys?

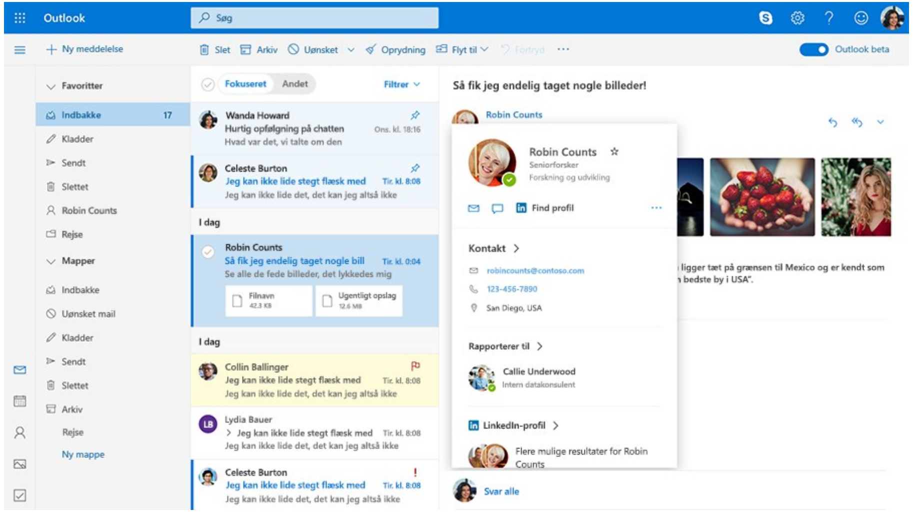
The lock icon

- Secure
- Info or Not secure
- Not secure or Dangerous





E-mail


Outlook
Søg
Ny meddelelse
Slet
Arkiv
Uønsket
Oprydning
Flyt til
Outlook beta
Favoritter
Indbakke 17
Kladder
Sendt
Slettet
Robin Counts
Rejse
Mapper
Indbakke
Uønsket mail
Kladder
Sendt
Slettet
Arkiv
Rejse
Ny mappe
Fokuseret
Andet
Filtrer
Wanda Howard
Hurtig opfølgning på chatten
Ons. kl. 18:16
Hvad var det, vi talte om den
Celeste Burton
Jeg kan ikke lide stegt flæsk med
Tir. kl. 8:08
Jeg kan ikke lide det, det kan jeg altså ikke
I dag
Robin Counts
Så fik jeg endelig taget nogle bill
Tir. kl. 0:04
Se alle de fede billeder, det lykkedes mig
Filnavn
Ugentligt opslag
I dag
Collin Ballinger
Jeg kan ikke lide stegt flæsk med
Tir. kl. 8:08
Jeg kan ikke lide det, det kan jeg altså ikke
Lydia Bauer
Jeg kan ikke lide stegt flæsk med
Tir. kl. 8:08
Jeg kan ikke lide det, det kan jeg altså ikke
Celeste Burton
Jeg kan ikke lide stegt flæsk med
Tir. kl. 8:08
Jeg kan ikke lide det, det kan jeg altså ikke
Så fik jeg endelig taget nogle billeder!
Robin Counts
Robin Counts
Seniorforsker
Forskning og udvikling
Find profil
Kontakt
robincounts@contoso.com
123-456-7890
San Diego, USA
Rapporterer til
Callie Underwood
Intern datakonsulent
LinkedIn-profil
Flere mulige resultater for Robin Counts
ligger tæt på grænsen til Mexico og er kendt som
bedste by i USA".
Svar alle

E-mail security

S/MIME (Secure/Multipurpose Internet Mail Extensions), Pretty Good Privacy (PGP)

Encrypt and authenticate individual emails using certifcates

DomainKeys Identified Mail (DKIM)

Authenticates that the email originates from the owner of the domain

Each email is signed by public key announced in DNS

Sender Policy Framework (SPF)

Authenticates that a mail claiming to come from a specific domain is submitted by an IP address authorized by that domain's administrators announced in DNS

Wrap-up

Lecture plan

Mandag d. 28. september
- kl. 10-12 Cryptography

Fredag d. 2. oktober
- kl. 09-10 Internet security protocols (bemærk ekstra time fra kl. 9 allerede)
- kl. 10-12 Intrusion detection

Mandag d. 5. oktober
- kl. 9-11 Forensics (bemærk flyttet fra kl. 10-12 til kl. 09-11)