

INF105 - 18
Kort intro til sikkerhet

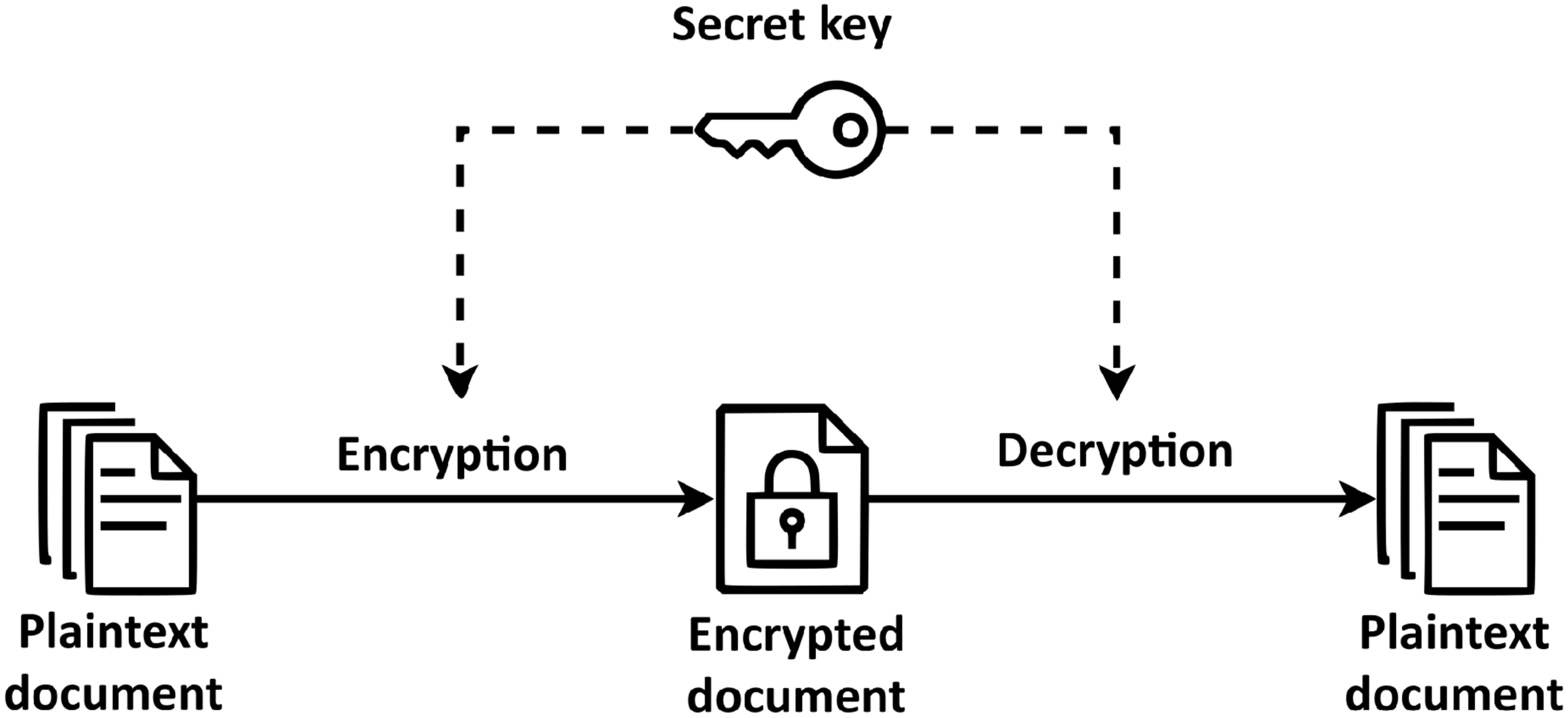
David Grellscheid

HTTP packet trace

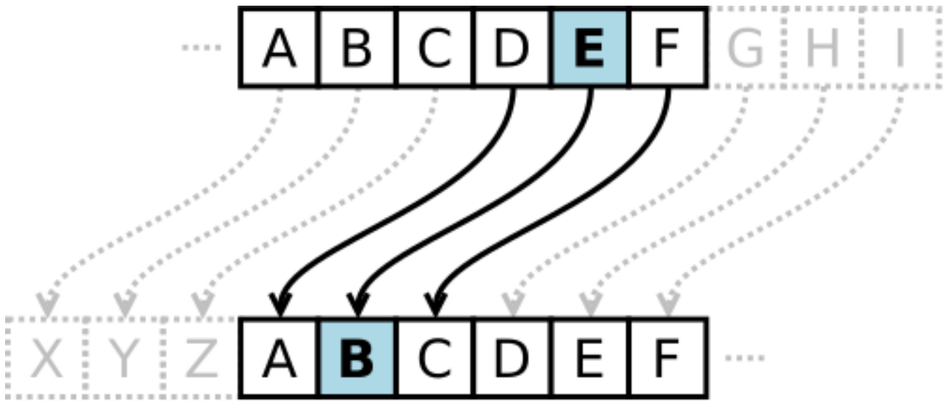
- Alle som har tilgang til nettverksutstyr (switches, routere) kan se innholdet i pakkene
- f.eks HTTP sender innholdet uten videre kryptering
- http basic auth sender username / passwd i klartekst
- demo wireshark

Kryptering

- vi trenger skjult kommunikasjon der alle andre ikke skal ha tilgang
- med ett unntak: mottakeren skal kunne lese
- krever en hjemmelig nøkkel som bare sender og mottaker vet om
- når nøklen avtales på forhånd: **symmetrisk kryptering**



By Michel Bakni - This file was derived from: Simple symmetric encryption.png, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=153764747

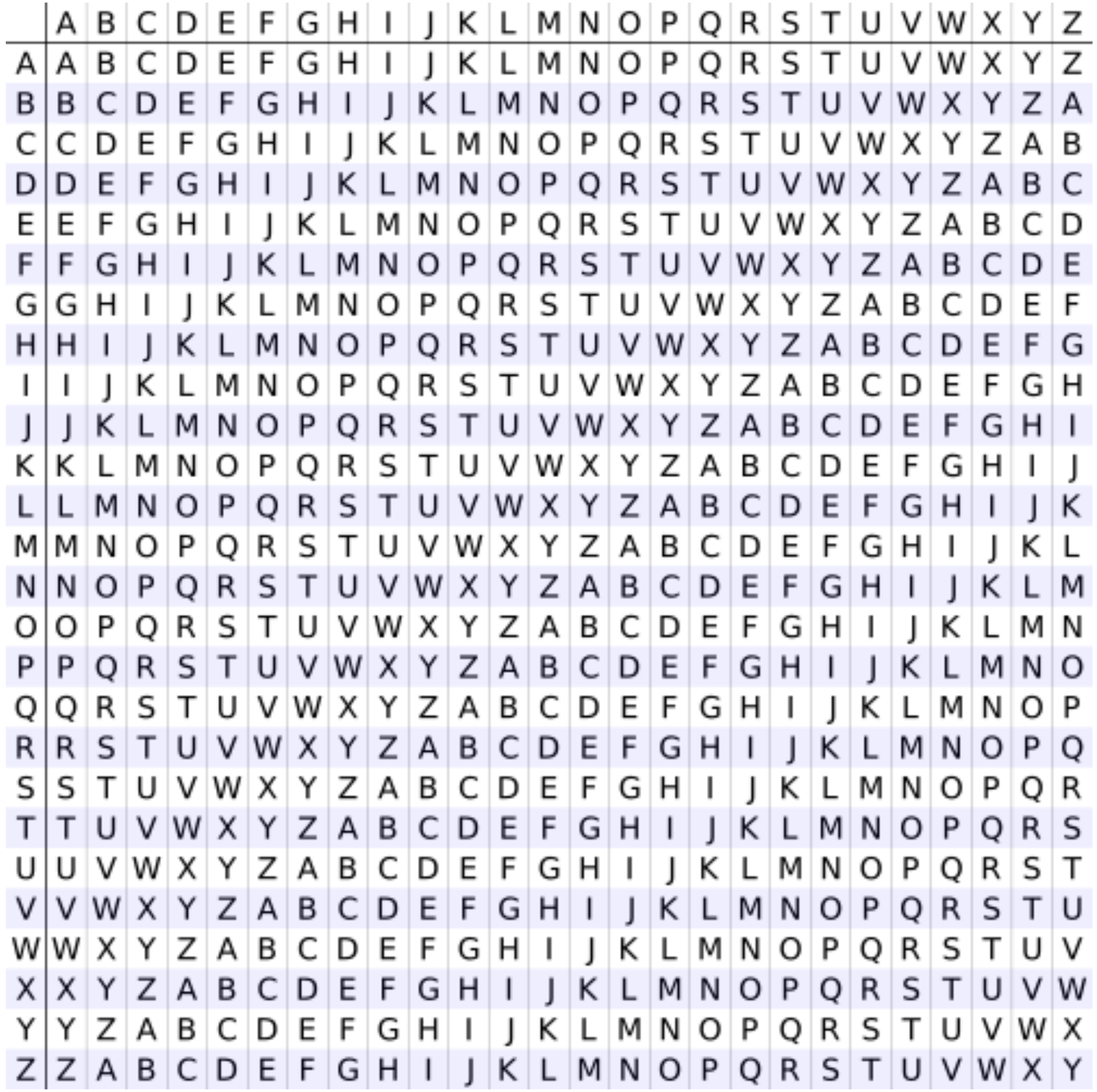


- Caesar cypher: erstatter hvert bokstav med en annen, fast offset
- hva er nøklen?

- Vigenère cypher:

  erstatter hvert bokstav med en annen, nøkkeltekst bestemmer hvilken

- ```
  helloworld
  keykeykeyk <- nøkkel
  ----------
  rijvsuyvin
  ```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
  ```

Kryptografi <-> Kryptanalyse

- Kryptanalyse:

  prøve å finne skjult nøkkel / krypteringsmetoden
  ved bruk av linguistikk / matematikk / statistikk

- Caesar cypher: hvilke bokstaver brukes oftest?

  - Eller **brute force**, sjekk alle 26 mulighetene

- må ha brute-force i bakhodet! Metoden må ha stor nok problem space

Kryptografi <-> Kryptanalyse

- f.eks Enigma machine:
- https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma

- "Keine besonderen Ereignisse" ofte brukt i meldinger
- En bokstav aldri blir kryptert som samme bokstav
- forenklet kryptanalyse enormt

kryptering er ikke
noe å lage selv

krever ekspertise

symmetrisk kryptering

- må utveksle krypteringsnøklene på en sikker måte

Public key exchange

- Først publisert av Diffie / Hellman i 1976
- Krever ***one way functions,***
  matematiske funksjoner der det er vanskelig å gjette input fra gitt output
- Første implementasjon:

  (g^(ab) mod p) vanskelig å få når bare

  (g^a mod p) og (g^b mod p) er kjent



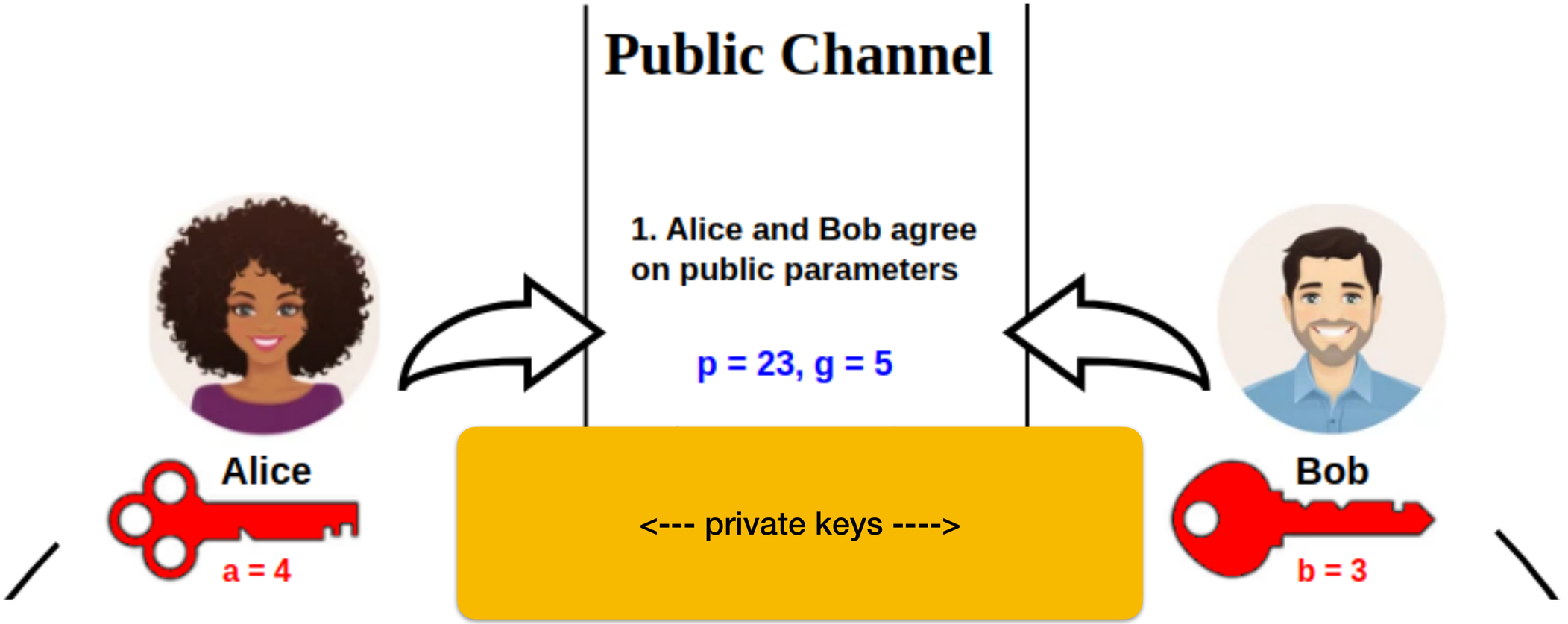
Public Channel

1. Alice and Bob agree on public parameters

$p = 23, g = 5$



p er primtall med > 2048 bit
ikke alle g er like godt, må velge forsiktig


Public Channel
1. Alice and Bob agree on public parameters
p = 23, g = 5
Alice
a = 4
<--- private keys ---->
Bob
b = 3



Alice



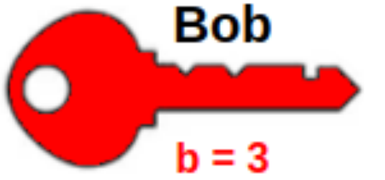
a = 4

1. Alice and Bob agree on public parameters

p = 23, g = 5

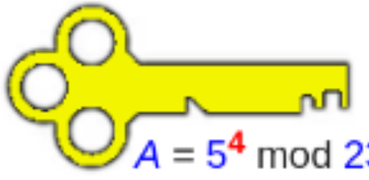


Bob



b = 3

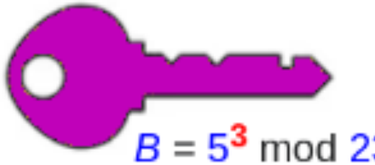
2. Alice combines her secret key (a) with the parameters and sends the resulting public key (A) to Bob



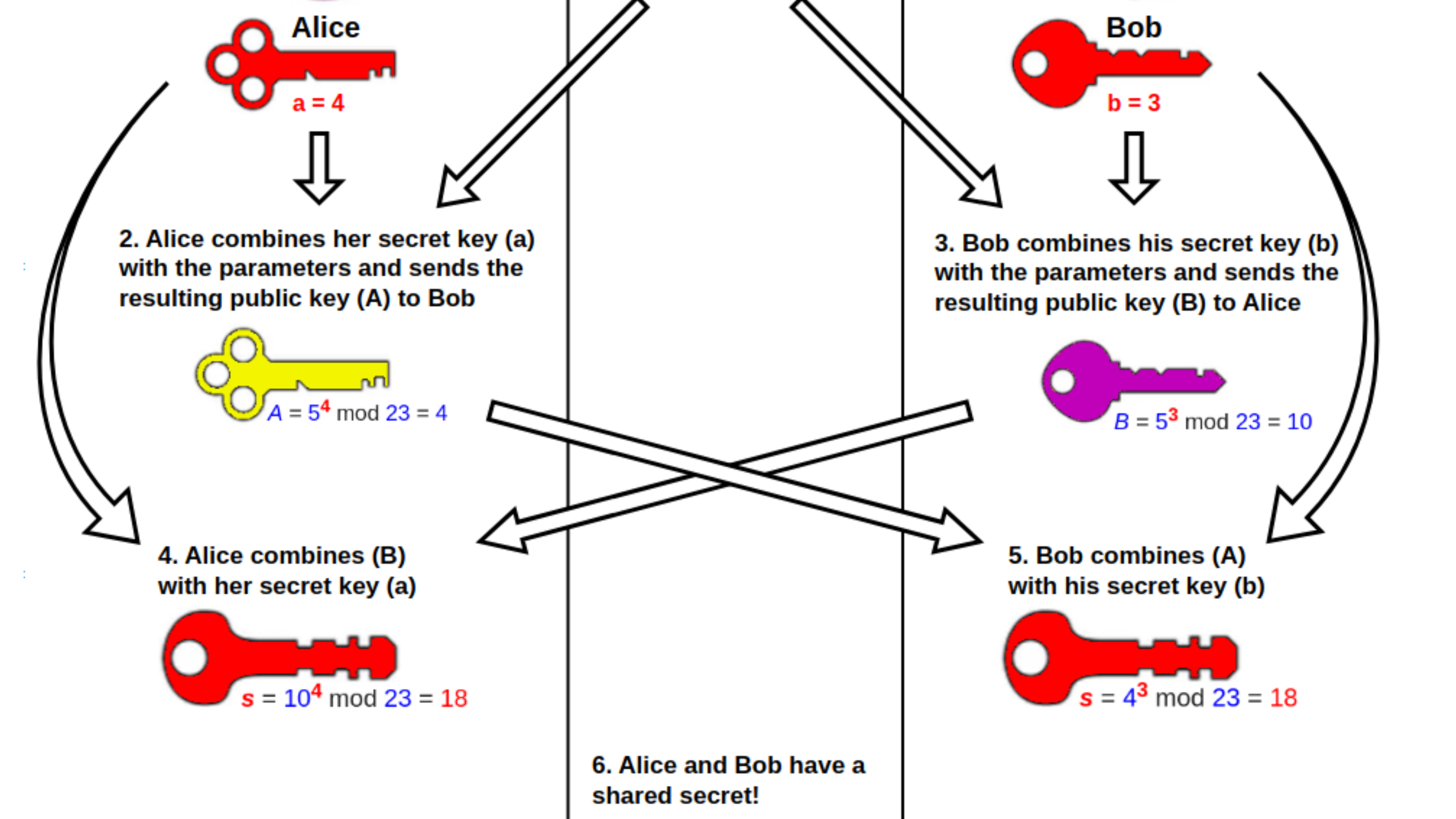
$A = 5^4 \bmod 23 = 4$

<--- public keys ---->

3. Bob combines his secret key (b) with the parameters and sends the resulting public key (B) to Alice

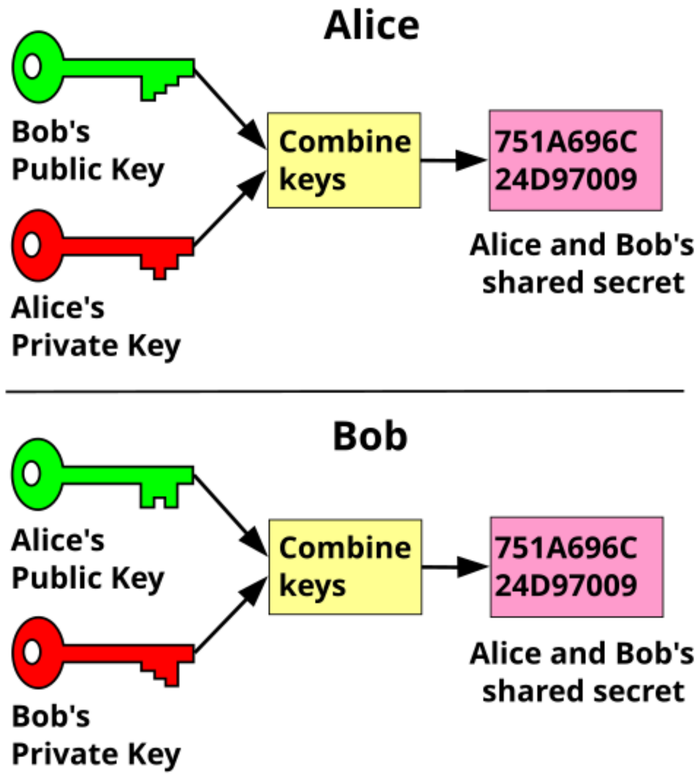


$B = 5^3 \bmod 23 = 10$

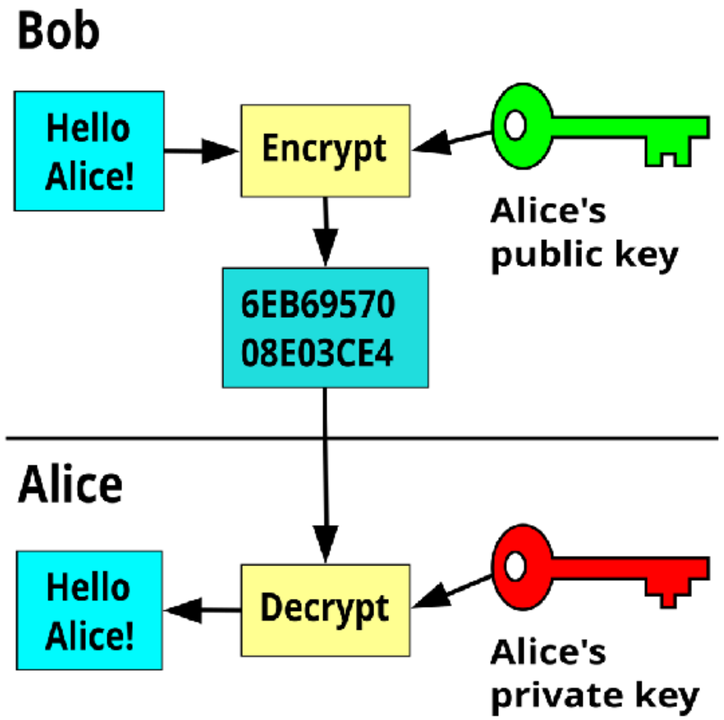

Alice
a = 4
Bob
b = 3
2. Alice combines her secret key (a) with the parameters and sends the resulting public key (A) to Bob
A = 5^4 mod 23 = 4
3. Bob combines his secret key (b) with the parameters and sends the resulting public key (B) to Alice
B = 5^3 mod 23 = 10
4. Alice combines (B) with her secret key (a)
s = 10^4 mod 23 = 18
5. Bob combines (A) with his secret key (b)
s = 4^3 mod 23 = 18
6. Alice and Bob have a shared secret!

- Kan bruke andre ***one way functions:*** f.eks *elliptic curve algorithms*

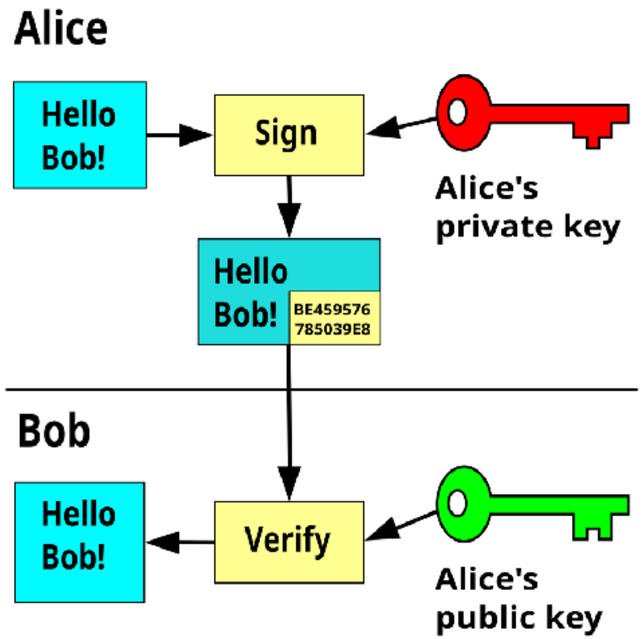
- https://en.wikipedia.org/wiki/Curve25519



Key exchange



Public key cryptography
(f.eks RSA)



Public key
signature verification

Obs!

- en god one-way function med gode parametere er helt avgjørende
- private key må holdes skjult
- Hører public key til den du skal snakke med? *MITM attacks*

SSH (secure shell)

- Brukeren har en secret key
- Public key ligger i remote maskin authorized_keys
- når du prøver å logge på:
  - debug1: Offering public key: /Users/dg/.ssh/id_ed25519
    debug1: Server accepts key: /Users/dg/.ssh/id_ed25519
- server sender en "challenge" kryptert med din pubkey
  om du kan svare riktig på den, så har du privkey
- Det samme omvendt for å autentisere host med known_hosts
- key exchange, og symmetrisk kryptert forbindelse videre

HTTPS

- basert på TLS (transport layer security), erstattet tidligere SSL
- krever pålitelig layer-4 protokoll, som TCP
- datasegmentene er kryptert, bare mottakerens system kan dekryptere

TLS

- kan ses i wireshark
- klient spør om sikker forbindelse, med en liste aksepterte algoritmer
- serveren velger en, og presenterer sitt *certificate* til autentifisering
- om klienten akspeterer sertifikatet,
  key-exchange av en midlertidig session nøkkel, symmetrisk kryptering
- kryptert innhold til layer 4 - segments

Certificates

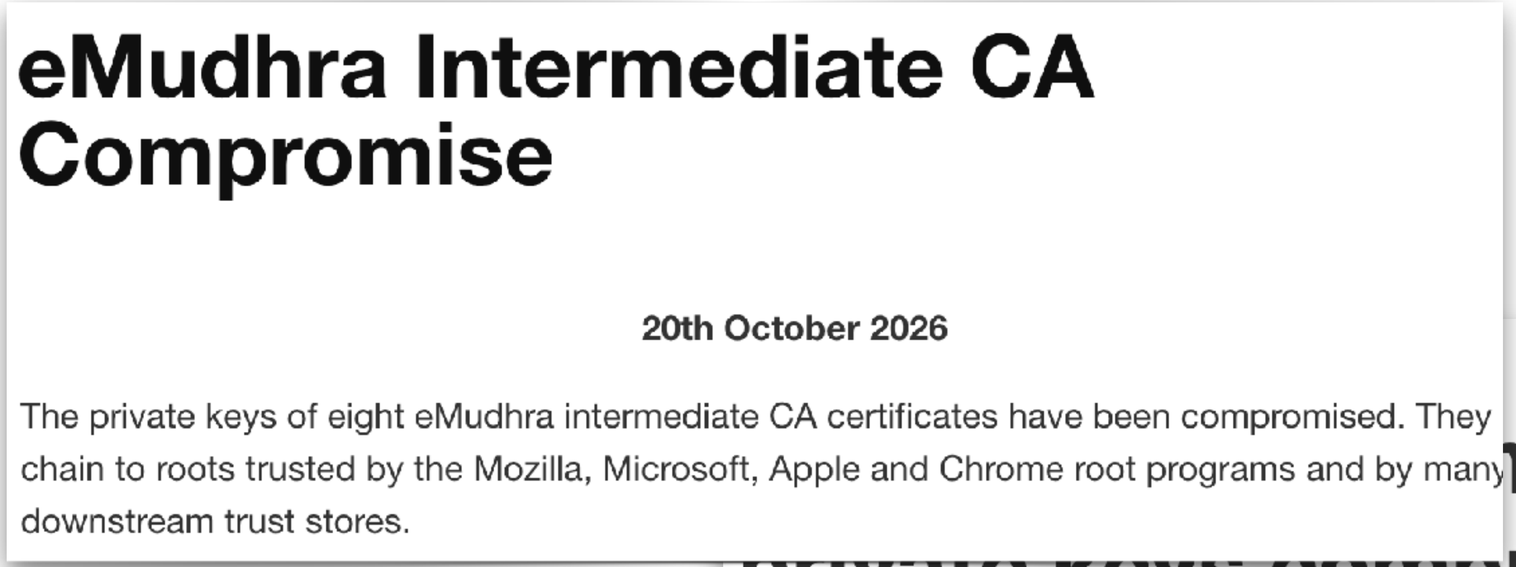
- Server certs kreves for å unngå MITM
- noen må ha bekreftet at sertifikatet jeg får faktisk hører til www.inf105.org
- Certification Authority som LetsEncrypt (certbot) sjekker
  - om du kan legge inn en fil på http://www.inf105.org/.well-known/letsencrypt/
  - eller legge in en DNS TXT record til www.inf105.org med et gitt innhold

Certificate Authorities

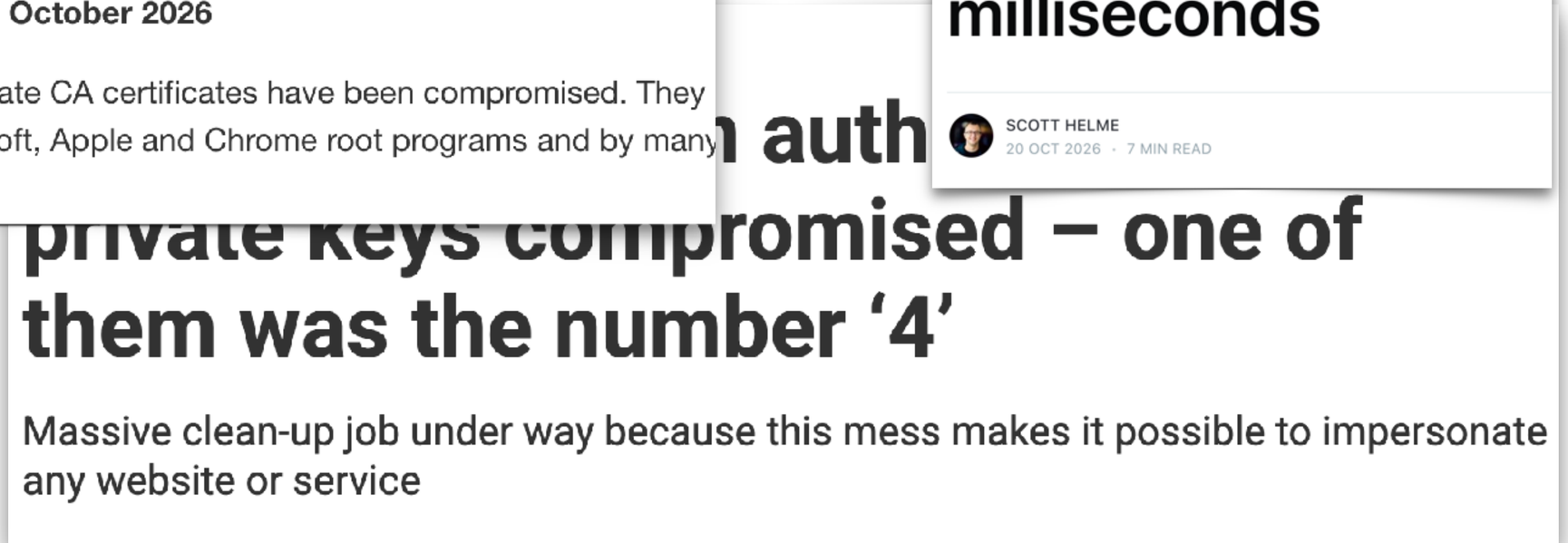
- Det finnes mange CA, root-sertifikatene til dem er innebygt i browsere og OS
- blir stadig oppdatert
- stor MITM risk hvis en untrusted CA ender opp der, kan signere certs til alle domener, kan utgi seg for å være hvem som helst

I går:

- https://www.theregister.com/security/2026/10/20/indian-certification-authority-emudhras-private-keys-compromised-one-of-them-was-the-number-4/5304312
- https://emudhra.we-have-a-problem.com/ https://scotthelme.co.uk/eight-bits-of-trust-emudhras-ca-keys-cracked-in-milliseconds/







I går

- https://scotthelme.co.uk/eight-bits-of-trust-emudhras-ca-keys-cracked-in-milliseconds/

- These are elliptic-curve keys on the P-384 curve. The private key is just a number, d. The public key, the thing that goes in the certificate for the world to see, is d × G, where G is a fixed point on the curve that everyone uses and "×" means adding G to itself d times using the curve's special arithmetic.

- Going forwards is easy, but going backwards, from the public key to d, is considered to be impractical. That's the entire security model, and it only holds if d is enormous. A P-384 private key should be a random 384-bit number, which gives you 2^{384} possibilities (or roughly 10^{115}), so nobody is ever going to search that.

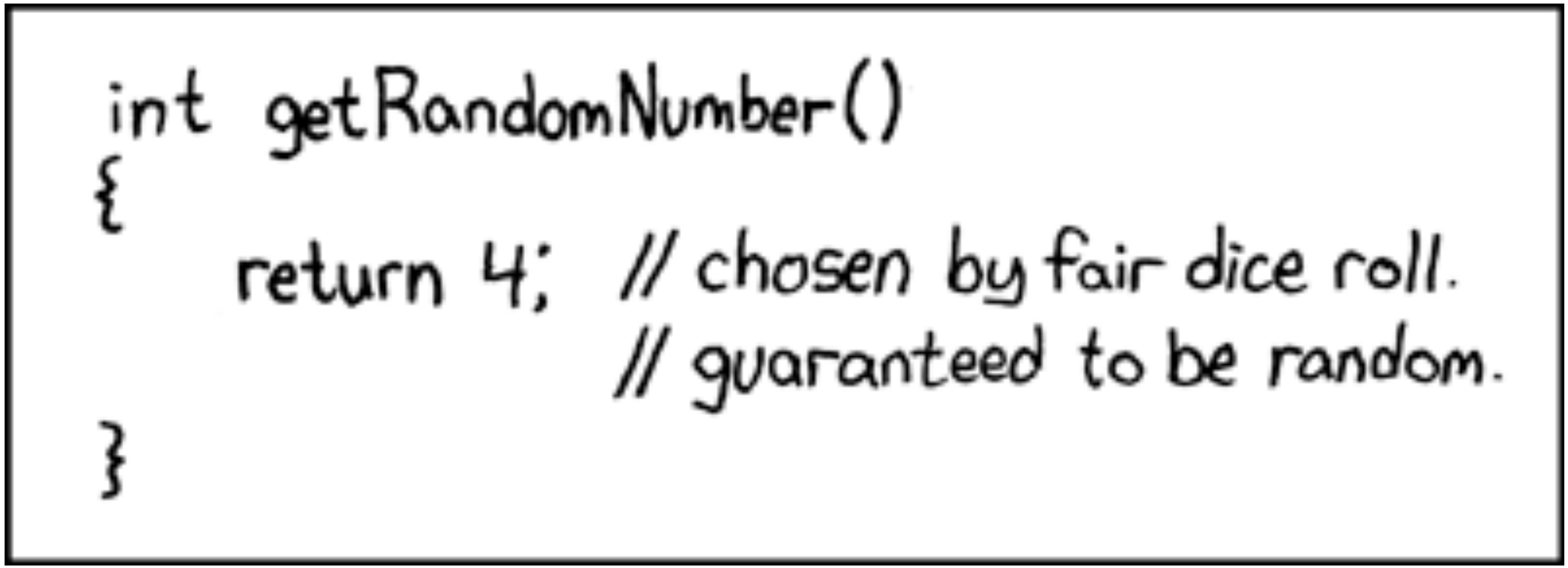
- eMudhra's keys look like this:

  - d = 0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000004

Problemet:

- offentlig private key som kan signere sertifikater til hvilken som helst domene, signert av eMudhra
- eMudhra er en av de trusted root certs
- enheter som kan få oppdateringer fjerner eMudhra -> OK
- enheter uten oppdateringer -> ???

alltid en XKCD som passer...



CC-BY-NC https://xkcd.com/221/

Sikkerhet

- ikke bare kryptering
- må se på hele bildet





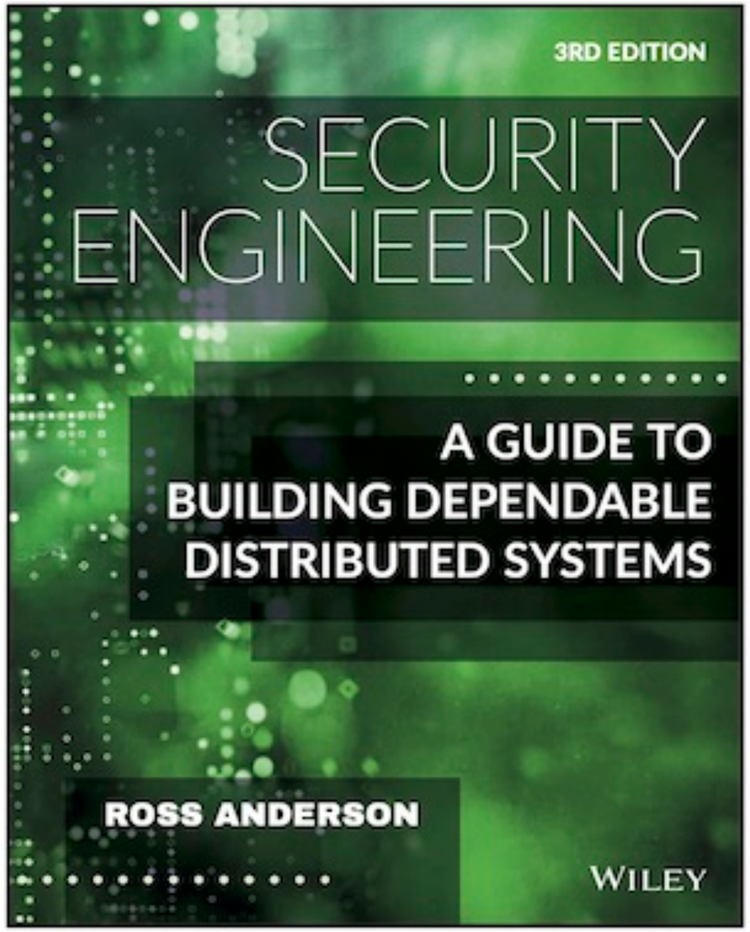
Randall Munroe CC-BY-NC https://xkcd.com/538/

Sikkerhet

- må se på hele bildet:

  sikkerhet mot hvem?
  hva er risiko? hva er kostnad?





https://www.cl.cam.ac.uk/archive/rja14/book.html

Sikkerhet

- side channel attacks:
  shouldersurfing, timing attacks, screen flicker

- DDOS: DNS flood, ping flood, ...
  distributed denial of service
  ikke egentlig et sikkerhetsbrudd, men tjenesten er utilgjengelig

- email spam

- blanding av control and content:
  SQL insertion, de fleste LLM systemer

- Vigilo case

Vigilo 2019

- Vigilo-systemet brukte folkeregisteret for å knytte barn til foreldre
- hentet uansett foreldreansvar
- delte skjult adresse, annen informasjon til foreldre som ikke skal ha kontakt
- folkeregisteret ikke har informasjon om begrensninger i tilgang, må legges inn manuelt av kommunen
- neste oppdatering fra folkereg. overskrev alle manuelle endringene

https://www.bt.no/nyheter/i/RRQo42/foreldre-med-besoeksforbud-ble-lagt-til-i-skoleappen-vigilo

Vigilo 2019

- Kommunaldirektør:
  "Vi har ikke forstått at dette kom til å skje. Vi trodde at våre endringer ville bli stående etter den nye innlastingen fra Folkeregisteret"

- Admin. direktør Vigilo:
  "Vi er leverandør og har et selvstendig ansvar. Samtidig er det Bergen kommune som har databehandleransvaret. [...]"

Incident investigation reports

- comp.risks , https://web.archive.org/web/20130907214101/http://nsc.nasa.gov/SFCS/
- Therac 25: https://en.wikipedia.org/wiki/Therac-25
- Air France 447: https://en.wikipedia.org/wiki/Air_France_Flight_447
- lett å si "human error", men mye ansvar på systemdesign
- software burde ha undersøkelseskomisjoner, se f.eks P.H.Kamp:
  - https://queue.acm.org/doi/10.1145/3475965.3475967
    https://cacm.acm.org/practice/the-software-industry-is-the-problem/
    https://cacm.acm.org/practice/the-software-industry-is-still-the-problem/