IN2120 Informasjonssikkerhet
Høst 2026

Del 2: Kryptografi



Åvald Åslaugson Sommervoll
Universitetet i Oslo

Kort beskjed:

- Eksamen er kun på norsk ikke på engelsk
  - Det ble sagt feil under forrige forelesning
  - https://www.uio.no/studier/emner/matnat/ifi/IN2120/h26/beskjeder/eksamensspraket-er-norsk.html

IN2120 - Informasjonssikkerhet

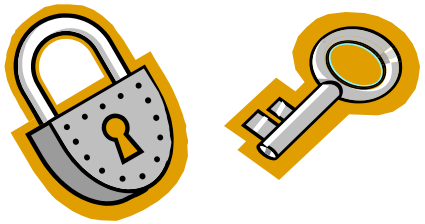
Eksamensspråket er norsk

Hei, det ble dessverre sagt feil på forelesningen i dag. (Foilen er riktig.)
Eksamenssettet er bare tilgjengelig på norsk, ikke engelsk. Beklager feilen.

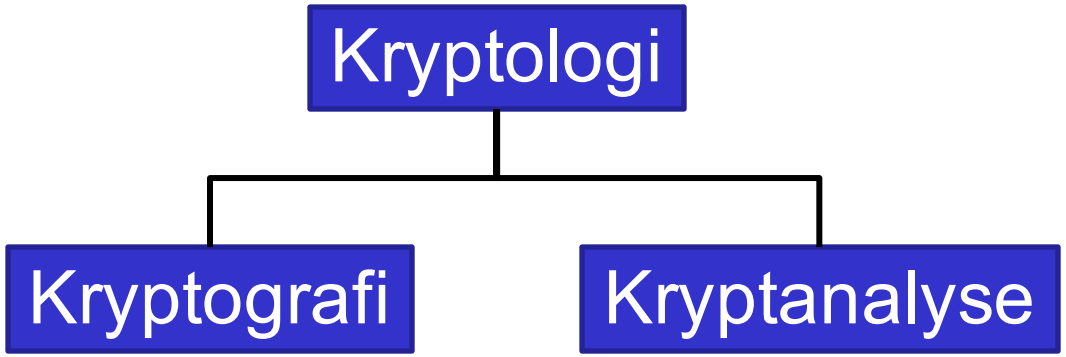
Publisert 17. aug. 2026 19:41 - Sist endret 17. aug. 2026 19:41

Oversikt

- Hva er kryptografi?
  - Hva brukes det til? Hvor mye er kryptografi?
- Symmetrisk kryptografi
  - Kort kryptohistorikk
  - Strømchiffer
  - Blokkchiffer
- Hashfunksjoner
- Asymmetrisk kryptografi
  - Asymmetrisk kryptering generelt
  - Diffie-Hellman nøkkelbytte
  - Digitale signaturer
- Post-Quantum Crypto

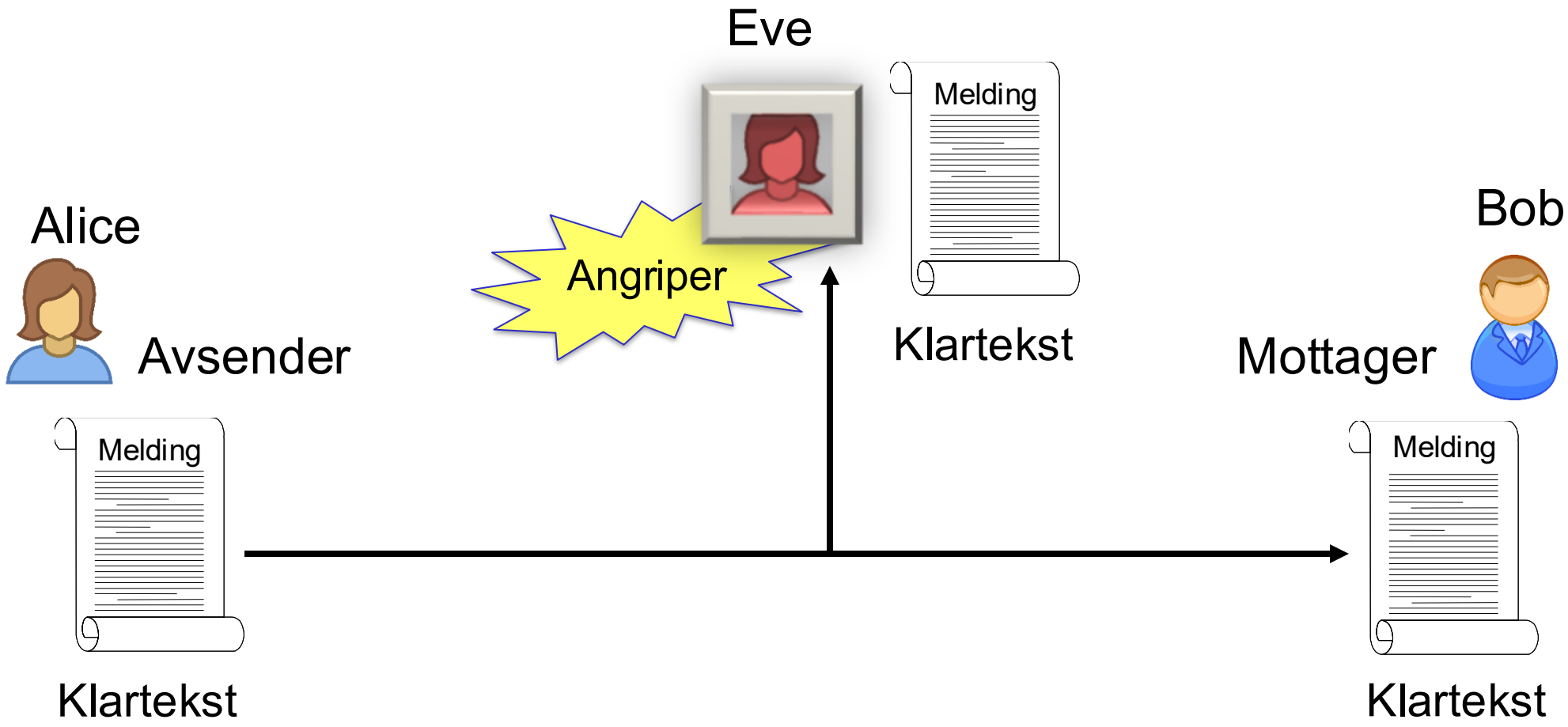


Terminologi

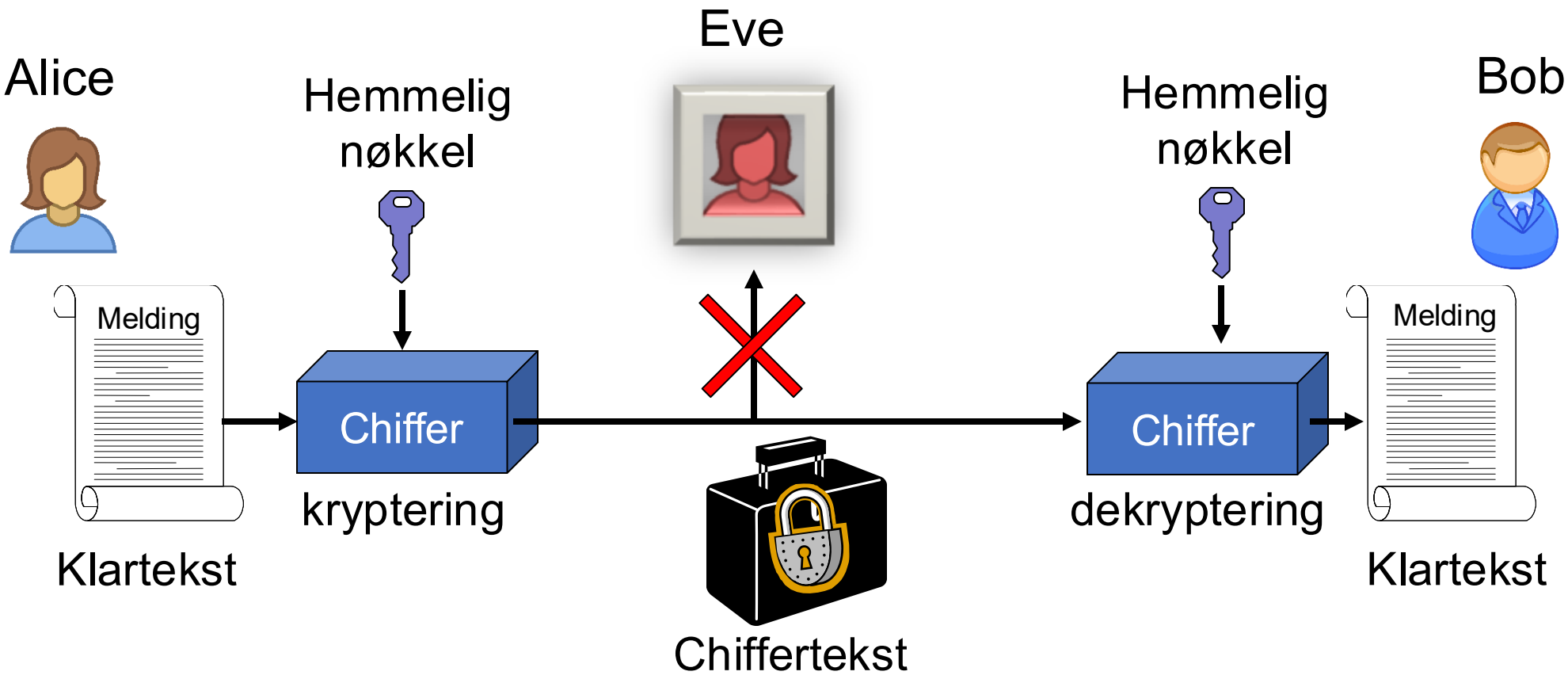


- **Kryptografi** er vitenskapen om hemmelig skrift med det formål å skjule betydningen av en melding.
- **Kryptanalyse** er vitenskapen om å knekke kryptografi.
- **Kryptologi** dekker både kryptografi og kryptanalyse.

Nødvendighet av kryptografi i et blikk



Anvendelse av kryptografi i et blikk



Hva kan kryptografi brukes til?

- Kryptografi støtter følgende sikkerhetsmål:
  - **Konfidensialitet:**
    - Gjør data uleselige for enheter som ikke har de riktige kryptografiske nøklene, selv om de har dataene.
  - **Dataintegritet:**
    - Enheter med riktige kryptografiske nøkler kan bekrefte at data er korrekt og ikke er blitt endret av uautoriserte.
  - **Autentisering:**
    - Entiteter som kommuniserer kan få visshet om at identiteten til den andre brukeren/entiteten eller avsenderen av en melding er det den påstår å være.
  - **Digital Signatur og PKI (Public-Key Infrastructure):**
    - Sterkt bevis på dataautentisitet som kan verifiseres av tredjeparter.
    - Skalerbar (til hele Internett) sikker distribusjon av kryptografiske nøkler.

Konfidensialitet:

- Gjør data uleselige for enheter som ikke har de riktige kryptografiske nøklene, selv om de har dataene.

- Eksempel på en melding som ikke har konfidensialitet:
  eDtte re ne ehmmelig emlding

- Det å bytte om første og andre bokstav gir ikke konfedensialitet
  Dette er en hemmelig melding

Dataintegritet:

- Enheter med riktige kryptografiske nøkler kan bekrefte at data er korrekt og ikke er blitt endret av uautoriserte.

- Eksempel på en melding som ikke har dataintegritet:
- Cpc tfoefs 100 ls ujm Bmjdf
  - (Vi vet at det handler om en pengeoverføring)

  - Kan endres uten å vite hva som star I meldingen til f.eks.

- Cpc tfoefs 1 000 000 ls ujm Bmjdf
  - (Bob sender 1 million kr til Alice)

Autentisering

- Entiteters om kommuniserer kan få en visshet om at identiteten til den andre entiteten er det den påstår å være.



Jeg burde komme backstage jeg er Beyonce

Nei, det er du ikke.
Det ser jeg, autentisering mislykkes.



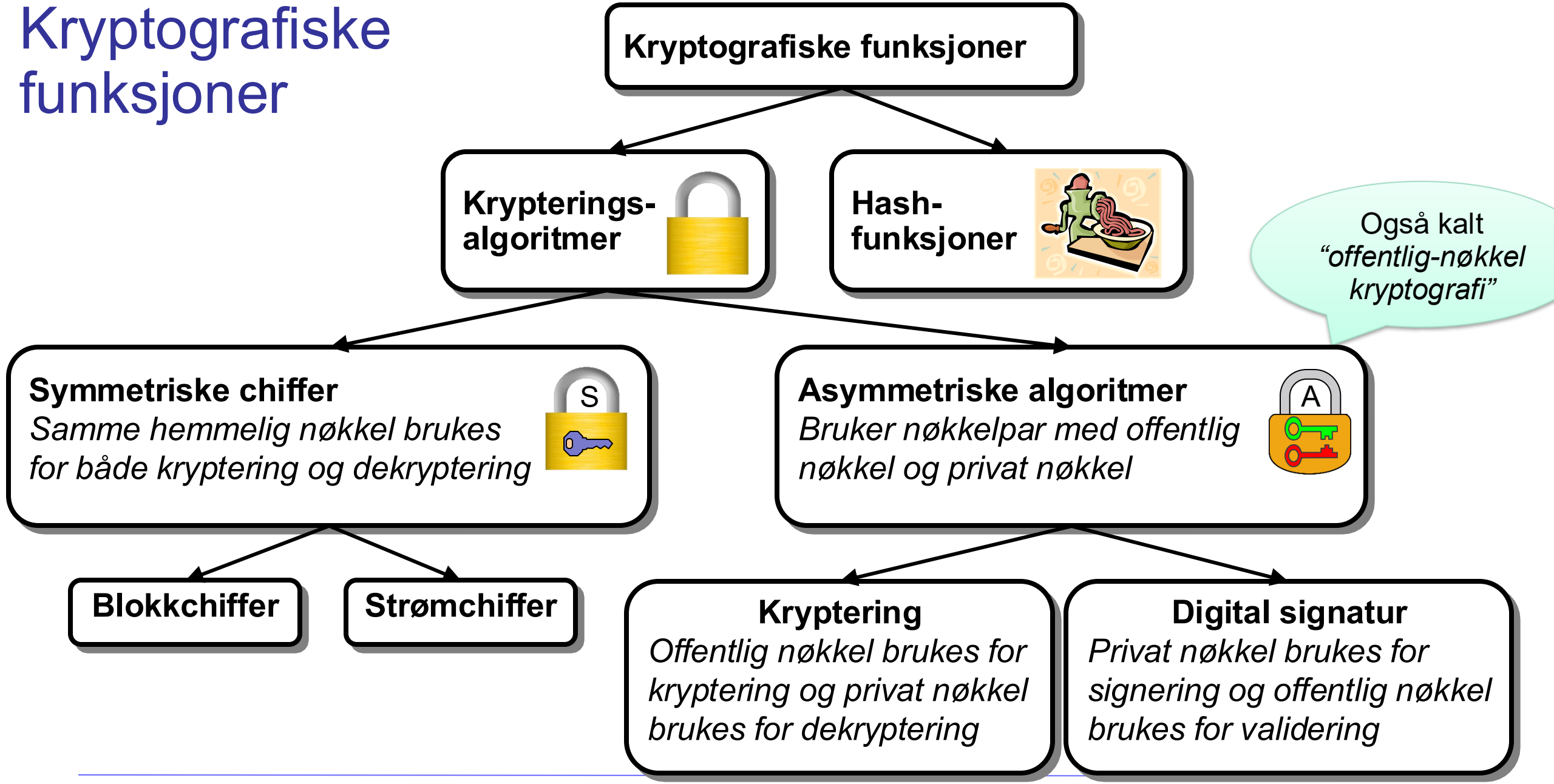


https://en.wikipedia.org/wiki/File:Beyonc%C3%A9_-_Tottenham_Hotspur_Stadium_-_1st_June_2023_(10_of_118)_(52946364598)_(best_crop).jpg

Hva kan kryptografi brukes til?

- Kryptografi støtter følgende sikkerhetsmål:
  - **Konfidensialitet:**
    - Gjør data uleselige for enheter som ikke har de riktige kryptografiske nøklene, selv om de har dataene.
  - **Dataintegritet:**
    - Enheter med riktige kryptografiske nøkler kan bekrefte at data er korrekt og ikke er blitt endret av uautoriserte.
  - **Autentisering:**
    - Entiteter som kommuniserer kan få visshet om at identiteten til den andre brukeren/entiteten eller avsenderen av en melding er det den påstår å være.
  - **Digital Signatur og PKI (Public-Key Infrastructure):**
    - Sterkt bevis på dataautentisitet som kan verifiseres av tredjeparter.
    - Skalerbar (til hele Internett) sikker distribusjon av kryptografiske nøkler.

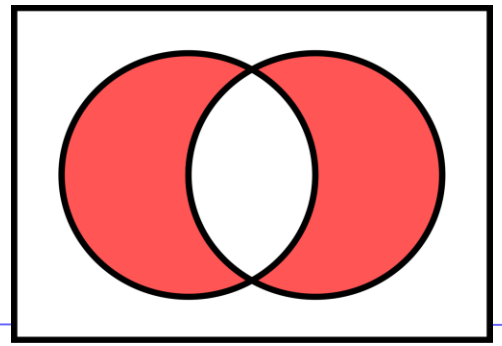
Kryptografiske funksjoner



Terminologi

- Kryptering: klartekst *M* transformeres med krypteringsfunksjon E til chiffertekst *C* styrt av krypteringsnøkkel *k*.
  - Formell skrivemåte: $C = E(M, k)$.
- Dekryptering: chiffertekst *C* transformeres med dekrypteringsfunksjon D til klartekst *M* styrt av krypteringsnøkkel *k*.
  - Formell skivemåte: $M = D(C, k)$.

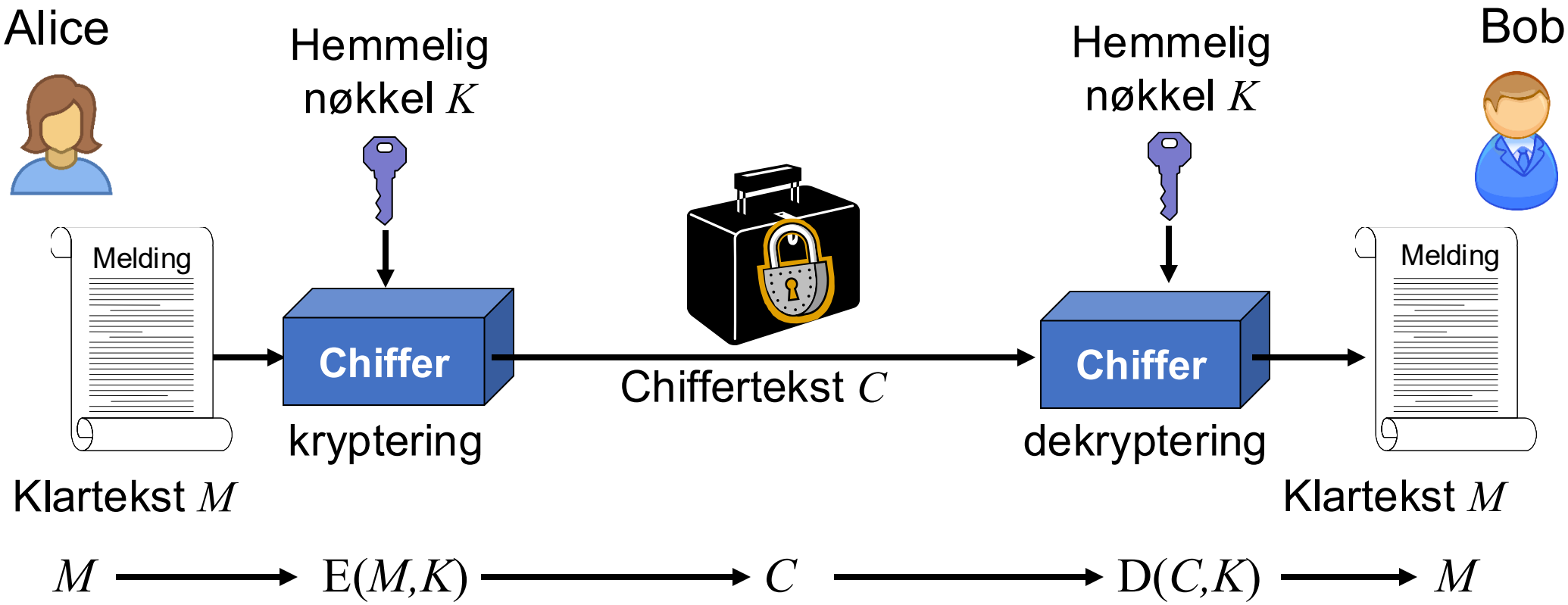
- XOR: En binær operasjon som gir 0 hvis bits-ene er like og 1 hvis bits-ene er forskjellige. Ofte kalt "exclusive or", og representert med symbolet: $\oplus$.



$0 \oplus 0 = 0$ $\quad$ $0 \oplus 1 = 1$

$1 \oplus 1 = 0$ $\quad$ $1 \oplus 0 = 1$

Symmetrisk kryptering (med hemmelig nøkkel)



$$M \longrightarrow \mathrm{E}(M,K) \longrightarrow C \longrightarrow \mathrm{D}(C,K) \longrightarrow M$$

- "Hemmelig nøkkel" betyr at nøkkelen er delt *i hemmelighet* mellom alle entiteter som er autorisert til å kryptere/dekryptere.

Kryptografiens historie

Klassiske chiffer

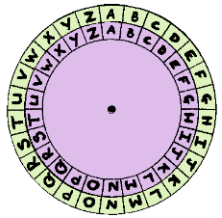
Transposisjon

Scytale



Substitusjon

Cæsar-chiffer



Middelalder chiffer

Poly-alfabetisk substitusjon og transposisjon

Vigenère 1566



Pre-WW2 chiffer

Engangs-nøkkel

Vernam 1917



WW2 chiffer

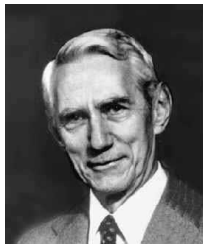
Kompleks mekanikk

Enigma



SP-nettverk Info-teori

Shannon



Pre-2000 chiffer

DES

Feistel



Asymmetrisk krypto

Diffie Hellman



Post-2000 chiffer

AES

Rijmen & Daemen



SHA-3

Post-kvante asymmetrisk krypto

→ BC | AD → 1799 | 1800 → 1939 | 1940 → 1975 | 1976 → 2000 | 2001 →

Styrken av et chiffer



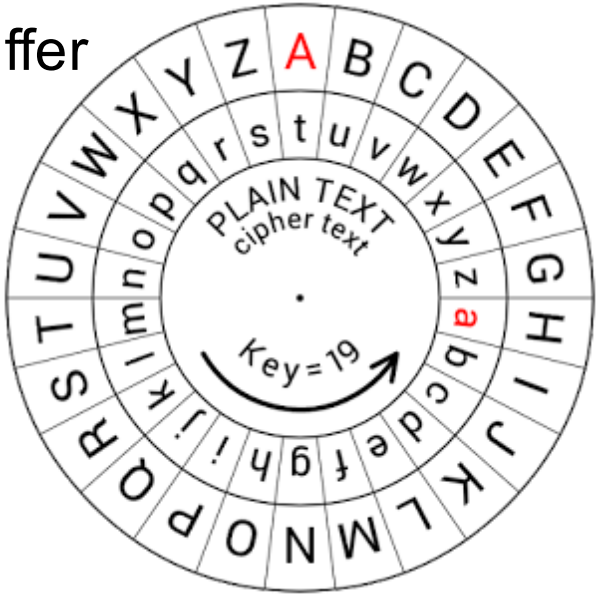
Faktorer som bestemmer chifferstyrke:

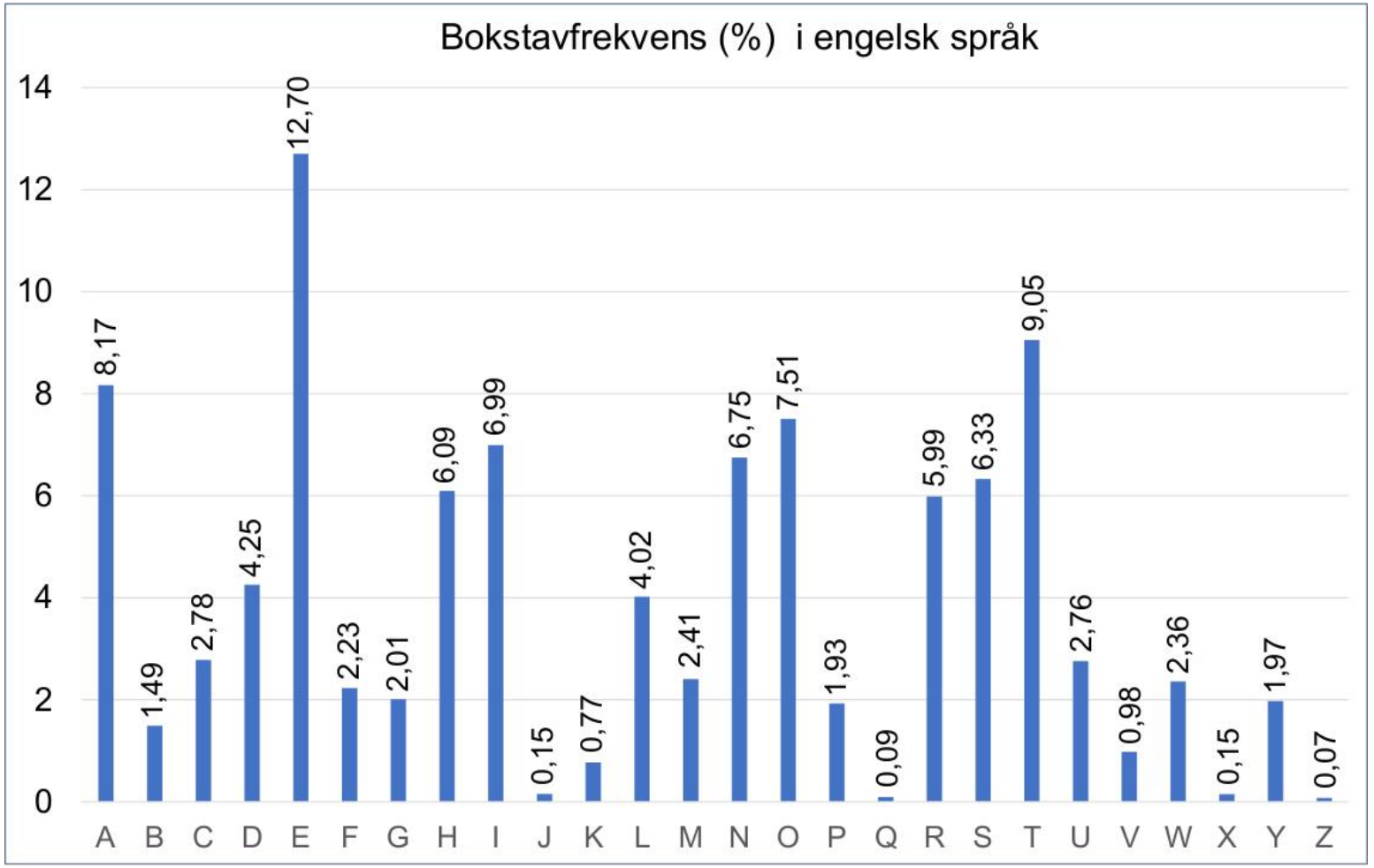
- **Nøkkelstørrelse**.
  - Nødvendig tid for fullstendig søk blant alle nøkler avhenger av størrelse.
  - Typisk størrelse for symmetrisk blokkchiffer er 256 bits.
  - Angriper må gjennomsnittlig prøve $2^{256}/2$ forskjellige nøkler for å finne den riktige, noe som ville ta millioner av år og som derfor ikke er praktisk.
  - Hvis det fins N ulike nøkler vil nøkkelstørrelse være: $\log_2(N)$.
- **Algoritmens styrke**.
  - Å finne nøkkelen ved kryptanalyse kan utnytte statistiske ujevnheter i chifferteksten.
  - For å forhindre kryptanalyse bør bitmønstrene / tegnene i chifferteksten ha en uniform/jevn fordeling, det vil si at alle bitmønstre / tegn skal være like sannsynlige.

Bokstavfrekvenser→ Statistisk kryptanalyse

- Klassiske chiffer, som Cæsar-chifferet, er svake fordi de ikke klarer å skjule statistiske ujevnheter i chifferteksten.

Cæsar-chiffer





Viginère cipher



Brukte flere alfabeter, hvilket alfabet som ble brukt ble bestemt av et kodeord eller setning.

Ble også knukket av frekvensanalyse. Noe bruteforce for å finne lengden på kodeordet.

Mer robust og krever at det er nok tekst. Lange kodeord generelt mer sikre.

	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

Den perfekte chiffermaskinen: One-Time-Pad



- Teleks med OTP på hullbånd, produsert av STK på Økern
- Moderne versjoner kan bruke usb med mye tilfeldige data

Vernam Cipher og addisjon med XOR (1917)



- Kryptering:
  $C = E(M,K) = M \oplus K$
- Dektryptering:
  $M = D(C, K) = C \oplus K$

$0 \oplus 0 = 0$	$0 \oplus 1 = 1$
$1 \oplus 1 = 0$	$1 \oplus 0 = 1$

- Perfekt kryptering:
  - Kalles ofte for perfekt kryptering hvis nøkkelen er helt tilfeldig, like lang som meldingen og bare brukes en gang til kryptering.
  - Dette er siden uansett hva den orginale meldingen er så vil det være 50% sannsynlighet for at et bit endres og 50% sannsynlighet for at det forblir uendret.

Viktig egenskap ved XOR

$$M \oplus U \oplus U = M$$

- Uansett hva den ukjente U eller meldingen M er, så vil melding M være uendret etter å XOR-e med U to ganger

$$0 \oplus 0 \oplus 0 = 0 \qquad 1 \oplus 0 \oplus 0 = 1$$
$$0 \oplus 1 \oplus 1 = 0 \qquad 1 \oplus 1 \oplus 1 = 1$$

Vernam Cipher og addisjon med XOR



- Eksempel kryptering og dekryptering: $M = 1111 \quad K = 1001$
  - Kryptering: $C = 1001 \oplus 1111 = 0110$
  - Dekryptering: $M = 1001 \oplus 0110 = 1111$

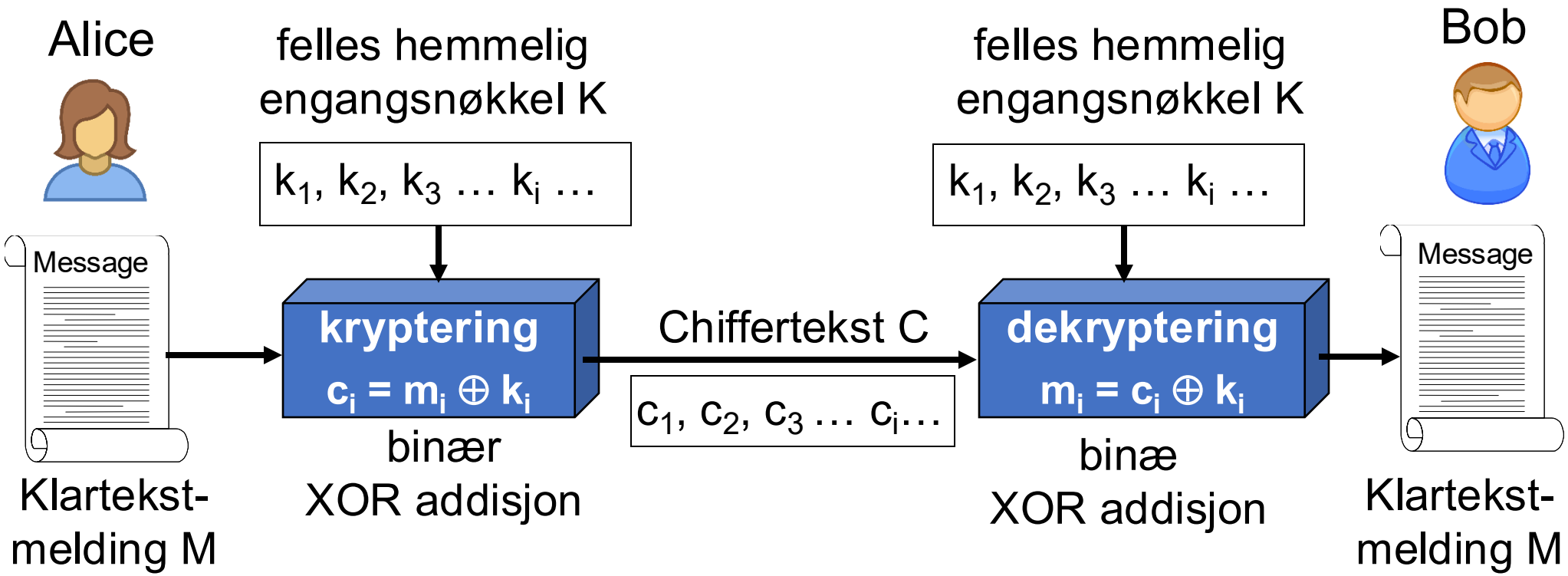
$0 \oplus 0 = 0$	$0 \oplus 1 = 1$
$1 \oplus 1 = 0$	$1 \oplus 0 = 1$

- En fare er at nøkkelen blir gjenbrukt. Da blir krypteringen svekket.
  $M' = 0100 \; K = 1001$
  $C' = 1101$

  $C' \oplus C = M' \oplus K \oplus M \oplus K = M' \oplus M$
  Obs! lekker informasjon om meldingene!

Engangsnøkkel: One-Time Pad (Gilbert Vernam, 1917)





- Bit for bit binær XOR addisjon: $c_i = m_i \oplus k_i$
  $m_i = c_i \oplus k_i = m_i \oplus k_i \oplus k_i = m_i \oplus 0 = m_i$
- OTP gir perfekt sikkerhet ved å anta at OTP-nøkkelen er helt tilfeldig, av samme lengde som meldingen, og brukes bare en gang.

Konkret eksempel

- Enkel melding, bare ja eller nei, ikke 1 og 0

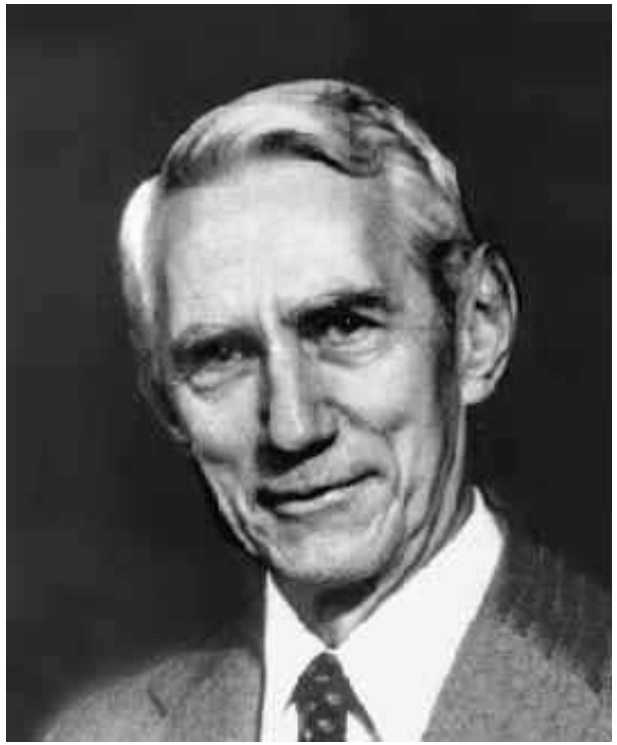
- Nøkkelen er resultatet til å slå mynt:
  - Kron (heads) -> Jeg sender motsatt
  - Mynt (tails) -> Jeg sender det jeg egentlig tenkte

- Hvis Eve observerer **ja,** så er det 50% sannsynlighet for meldingen var ja, men også 50% sannsynlighet for at meldingen var nei
- Tilsvarende hvis Eve observerer **nei.**

Claude Shannon (1916 – 2001)

Informasjonsteoriens far – MIT / Bell Labs

- Informasjonsteori
  - Definerte "binært siffer" (bit) som minste informasjonsenhet
  - Definert informasjonsentropi som mål på mengden informasjon
- Kryptografi
  - Modell for hemmelighetssikre systemer
  - Definert perfekt hemmelighetssikkerhet
  - Prinsipp for kryptering med SP-nettverk (substitusjon og permutasjon) for å viske ut statistiske ujevnheter



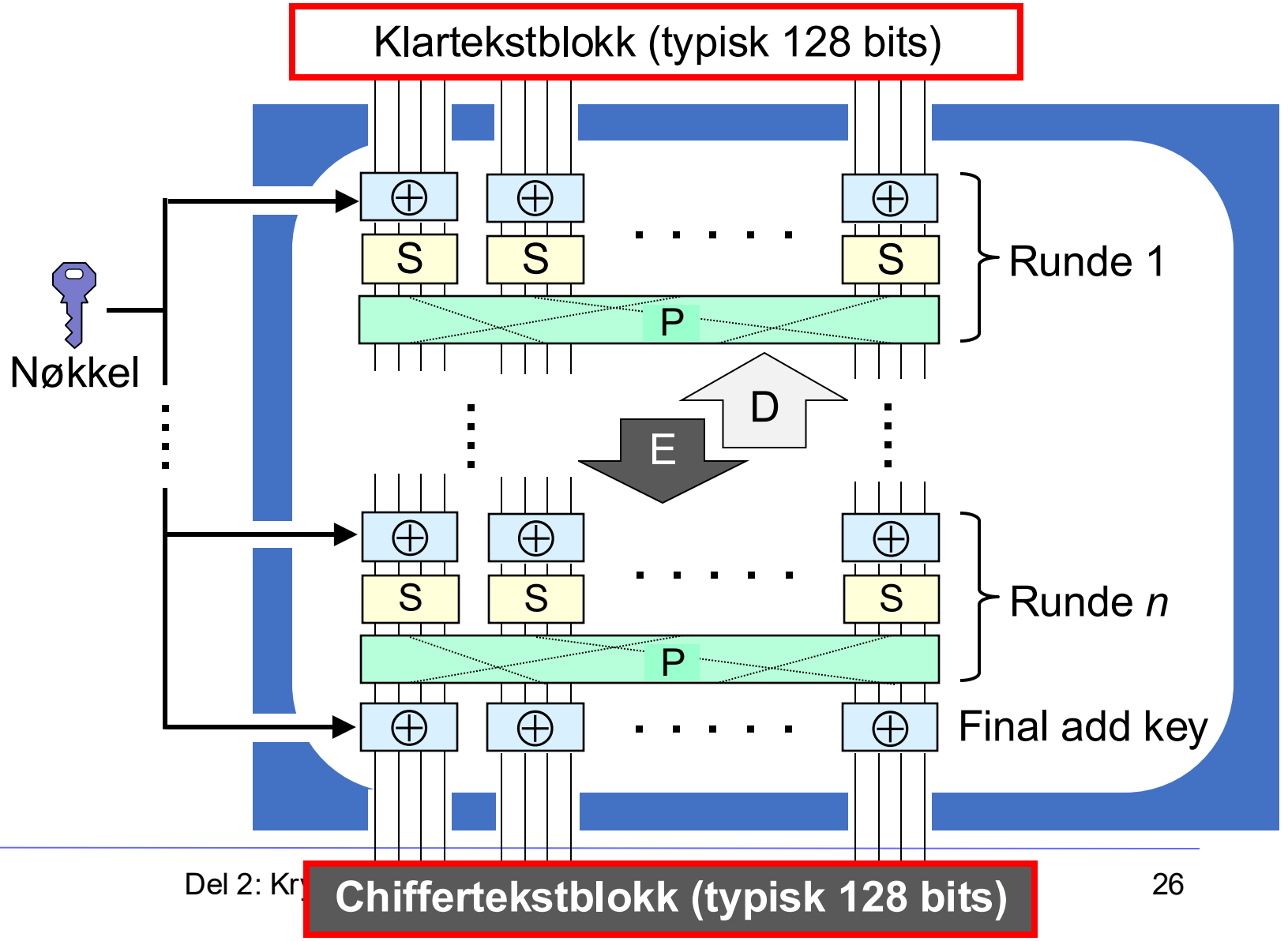
Shannon's blokkchiffer SP-nettverk

Visker ut statistiske ujevnheter

E : Kryptering
D : Dekryptering

⊕: Add key bits
S : Substitution
P : Permutation

- Repeter substitusjon og permutasjon, typisk 10-20 runder.
- Substitusjon
  - Sub-blokker substitueres (erstattes) med andre blokker
  - Gir «confusion» dvs. skjuler sammenheng mellom klartekst og chiffertekst.
- Permutasjon
  - Sub-blokker flyttes rundt omkring i blokken.
  - Gir "diffusion", dvs. å endre en enkelt bit i klartekst gir stor endring i chiffertekst.
- Nøkkelen inngår i S eller P, eller i separat add-key-funksjon.

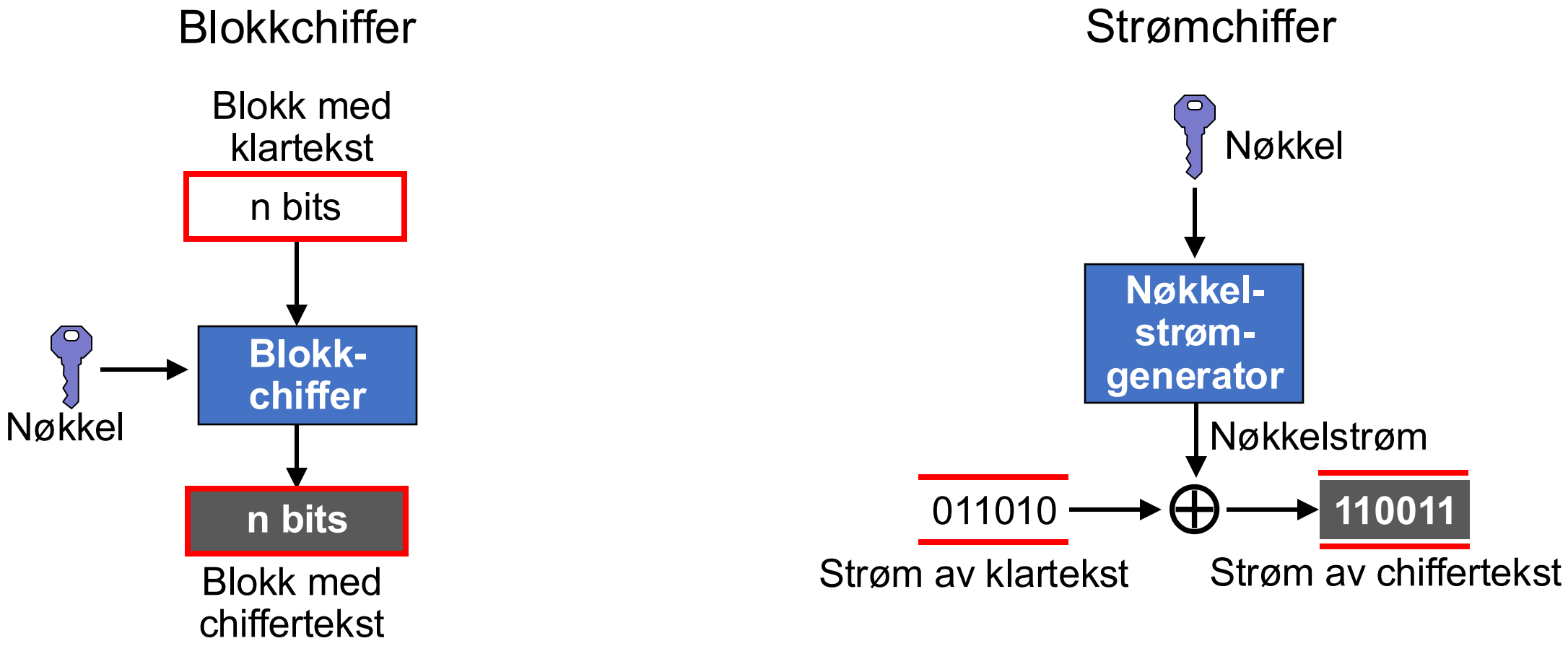


AES - Advanced Encryption Standard

- DES (Data Encryption Standard) fra 1977 hadde en 56-bits nøkkel og en 64-bits blokk. På midten av 1990-tallet kunne DES bli knekket med fullstendig nøkkelsøk.
- I 1997 kunngjorde NIST en åpen konkurranse om å designe et nytt blokkchiffer for å erstatte DES.
- Det beste forslaget kalt "Rijndael" (designet av Vincent Rijmen og Joan Daemen fra Belgia) ble ansett som best, og nominert til å bli AES (Advanced Encryption Standard) i 2001.
- AES har nøkkelstørrelser på 128, 192 eller 256 bit og blokkstørrelse på 128 bit.
- AES følger Shannons prinsipp med SP-nettverk.



Blokkchiffer og strømchiffer



Blokkchiffer: Operasjonsmoduser

- Et blokkchiffer krypterer en blokk på (typisk) 128 bits, som bare er omtrent 16 bokstaver.
- For kryptering av mer enn en blokk benyttes en spesifikk modus.
- Krypteringsmodusene har ulike egenskaper.
- Vanlige moduser er:
  - **C**oun**T**e**R** Mode (CTR)
  - **C**ipher **B**lock **C**haining (CBC)
  - **O**utput **F**eed**B**ack (OFB)
  - **C**ipher **F**eed**B**ack (CFB)



sikker

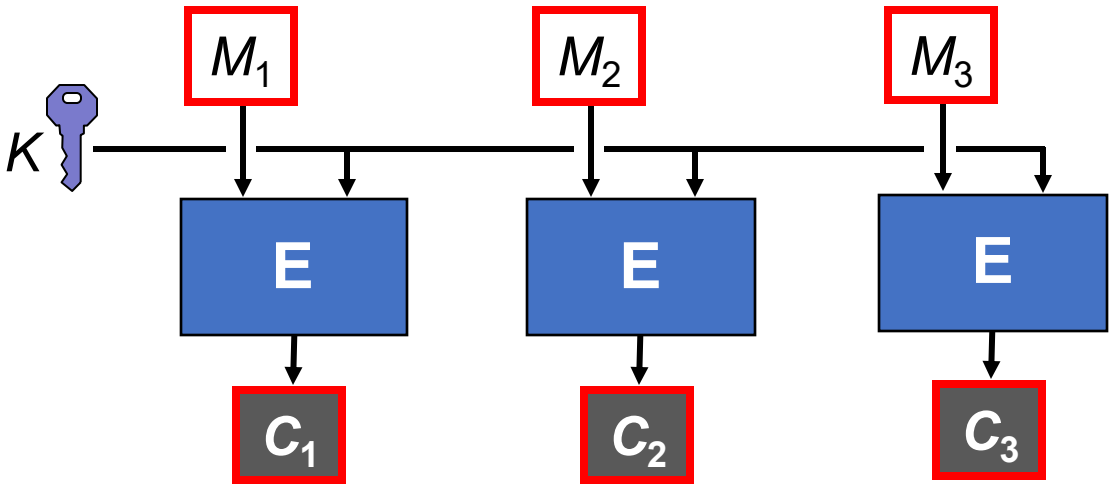
  - **E**lectronic **C**ode **B**ook (ECB)



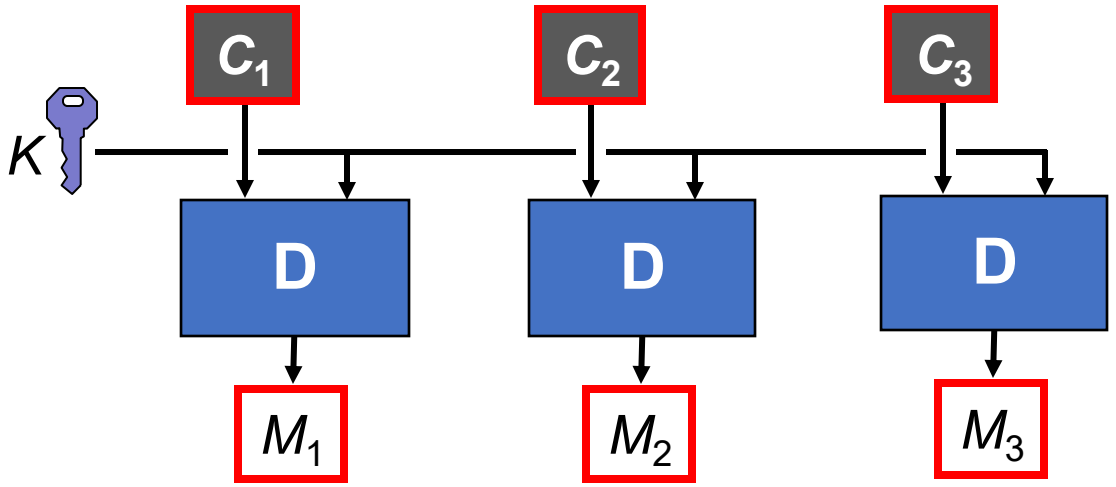
usikker

Electronic Code Book (ECB)

- Enkleste krypteringsmodus
- Klarteksten deles opp i blokker $M_1, M_2, \ldots, M_n$
- Hver blokk krypteres separat.
  - Notasjon kryptering: $C_1 = \mathrm{E}(M_1, K)$
  - Notasjon dekryptering: $M_1 = \mathrm{D}(C_1, K)$
  - Like klartekstblokker gir like chiffertekstblokker, dette er problemet !



Kryptering

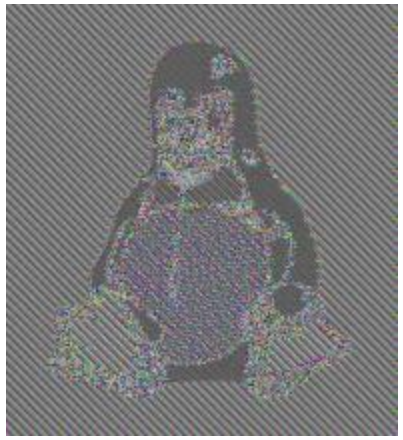


Dekryptering

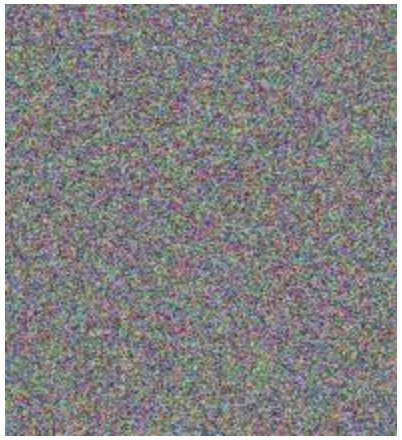
Sårbarhet ved å bruke ECB-modus



Klartekst



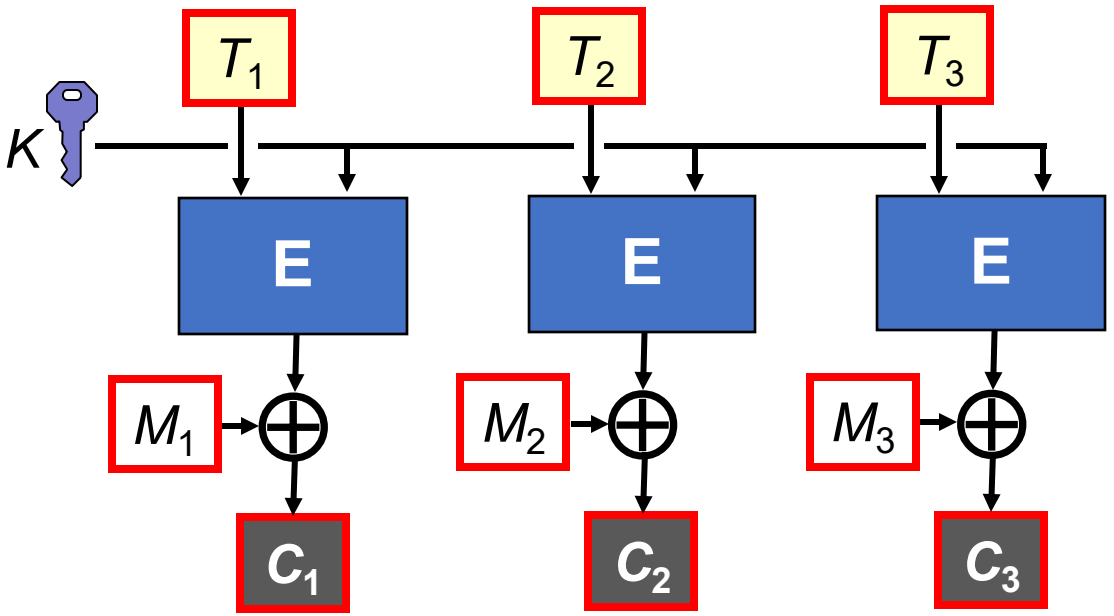
Chiffertekst med ECB-modus



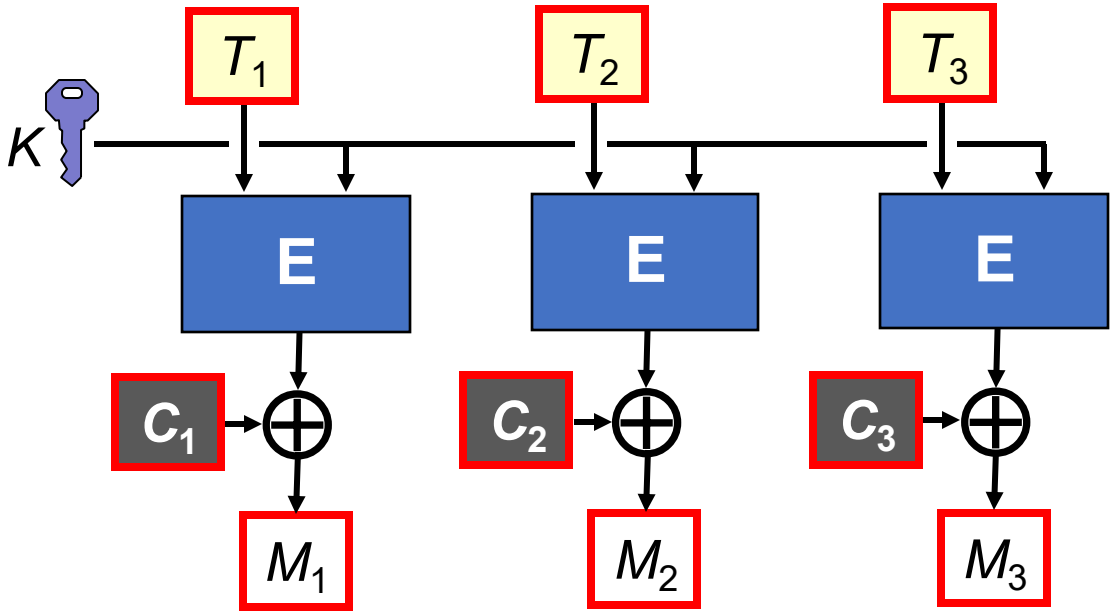
Chiffertekst med sikker modus

Tellermodus - Counter Mode (CTR)

- Klarteksten deles opp i blokker $M_1, M_2, \ldots, M_n$
- En inkrementerende tellerverdi T krypteres
- Hver krypterte telleverdi adderes til klartekstblokken med binær XOR $\oplus$
  - Like klartekstblokker gir forskjellige chiffertekstblokker, dette gir sikkerhet !



Kryptering



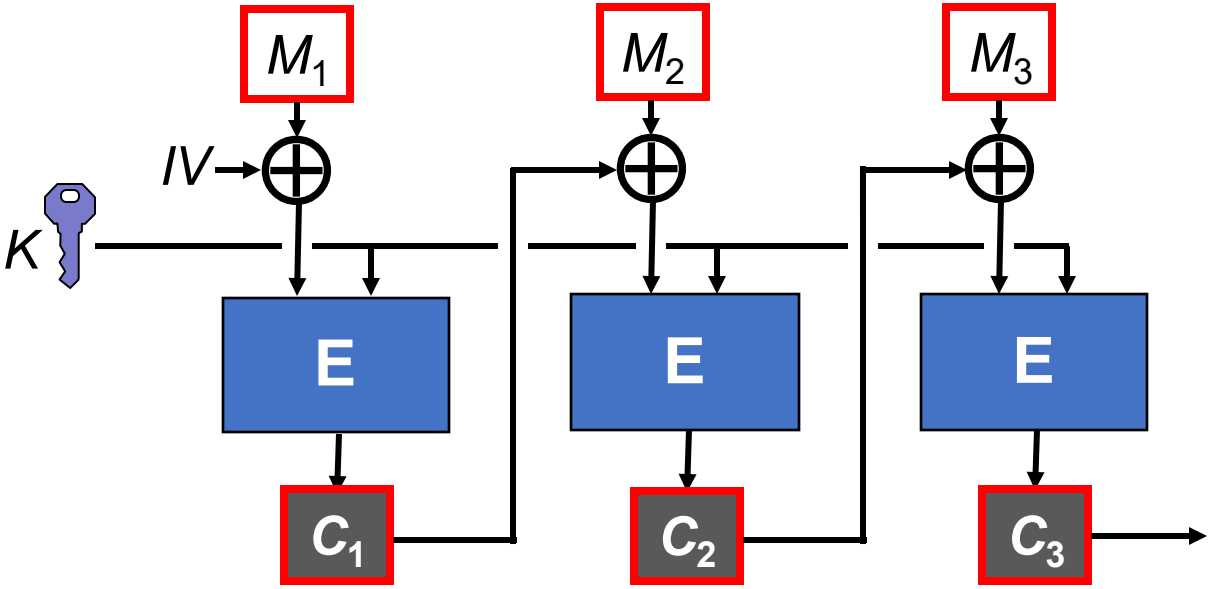
Dekryptering

CTR-kryptering og binæraddisjon med XOR

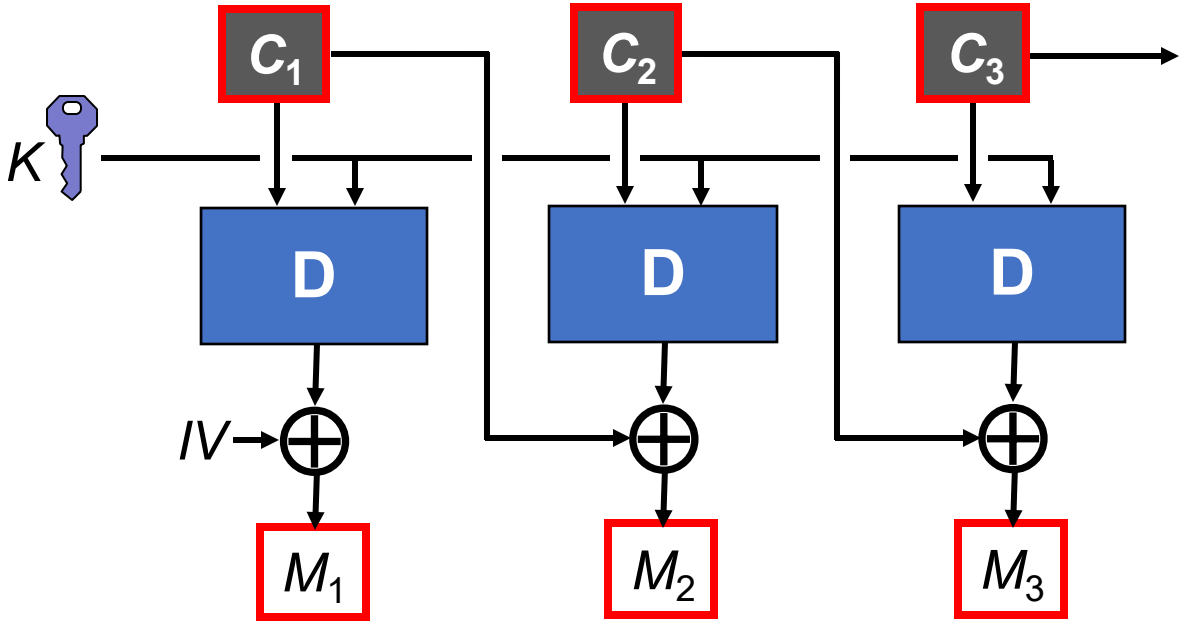
- Klarteksten deles opp i blokker: $M_1, M_2, \ldots, M_n$
- Inkrementerende tellerverdier med samme blokkstørrelse: $T_1, T_2, \ldots, T_n$
- Tellerverdiene krypteres og adderes til klartekstblokkene:
  - Notasjon kryptering: $C_1 = \mathrm{E}(T_1, K) \oplus M_1$
  - Notasjon dekryptering: $M_1 = \mathrm{E}(T_1, K) \oplus C_1 \quad = \mathrm{E}(T_1, K) \oplus \mathrm{E}(T_1, K) \oplus M_1$
  - Krypteringsfunksjonen $\mathbf{E}$ benyttes til både kryptering og dekryptering

Cipher Block Chaining (CBC)

- Klarteksten deles opp i blokker $M_1, M_2, \ldots, M_n$
- Hver chifferblokk adderes til neste klartekstblokken med binær XOR ⊕ før kryptering
  - Starter med en tilfeldig *IV* (initialiseringsvektor) som ikke trenger å være hemmelig
  - Like klartekstblokker krypteres til forskjellige chiffertekstblokker, dette gir god sikkerhet !



Kryptering



Dekryptering

Notasjon for CBC

- Klarteksten deles opp i blokker: $M_1, M_2, \ldots, M_n$
- Hver chifferblokk adderes til neste klartekstblokk med binær XOR før kryptering

  – Notasjon kryptering:

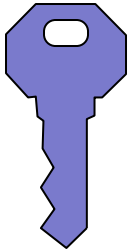
$$C_1 = \mathrm{E}((IV \oplus M_1), K)$$
$$C_2 = \mathrm{E}((C_1 \oplus M_2), K)$$
$$C_n = \mathrm{E}((C_{n\text{-}1} \oplus M_n), K)$$

  – Notasjon dekryptering:

$$M_1 = \mathrm{D}(C_1, K) \oplus IV$$
$$M_2 = \mathrm{D}(C_2, K) \oplus C_1$$
$$M_n = \mathrm{D}(C_n, K) \oplus C_{n\text{-}1}$$

HASHFUNKSJONER OG MELDINGSAUTENTISERING





Hva er en hash?

- En funksjon tar data av en vilkårlig størrelse og returner verdier av en fast størrelse.

- Høres komplisert ut, hoved ideen er å ta noe stort å lage en representasjon som er liten.



Hashfunksjoner

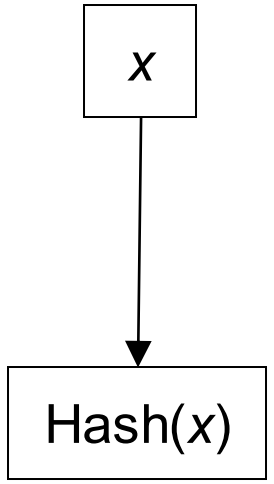
Krav til en hashfunksjon Hash:

1. Lett å beregne: Gitt inputdata *x*, det skal være lett å beregne Hash(*x*).
2. Komprimering: Komprimerer vilkårlig stor *x* til en hash-verdi Hash(*x*) med fast størrelse *n* (typisk 256 bits eller 512 bits).
3. Enveis: Gitt hash-verdi *y*, det skal være praktisk umulig å finne inputdata *x* slik at Hash(*x*) = *y*.
4. Kollisjonsresistens (svak): Gitt inputdata *x* og tilhørende hash-verdi Hash(*x*), det skal være praktisk umulig å finne en annen *x'* slik at Hash(*x*) = Hash(*x'*) (svak kollisjonsresistens).
5. Kollisjonsresistens (sterk): Det skal være praktisk umulig å finne to ulike inputdata *x* og *x'* slik at Hash(*x*) = Hash(*x'*) (sterk kollisjonsresistens).

1.Lett å beregne

- Gitt inputdata *x*,
  det skal være lett å beregne Hash(*x*).

- Lett for en datamaskin, slik at beregningen er rask.

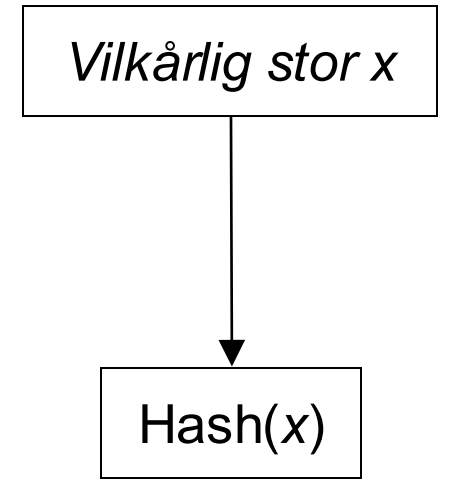


Lett å beregne

2. Komprimering

- Komprimerer vilkårlig stor *x* til en hash-verdi Hash(*x*)med fast størrelse *n* (typisk 256 bits eller 512 bits).

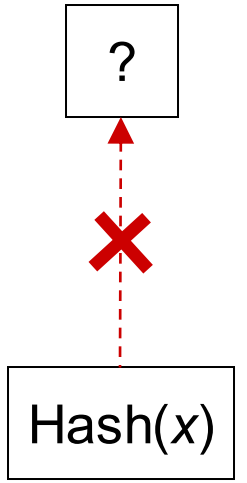
- Uansett hvor stor input det er så skal det bli 256 bits, om det er hele Ringenes Herre sagaen eller en facebook melding.



Komprimering til fast størrelse

3. Enveis

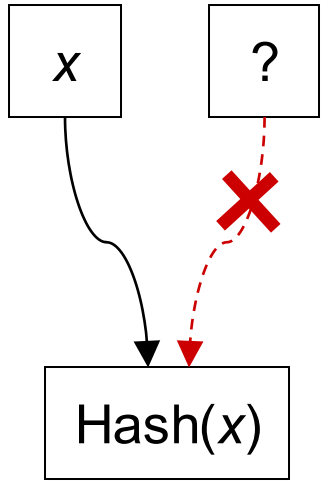
- Gitt hash-verdi *y*, det skal være praktisk umulig å finne inputdata *x* slik at Hash(*x*) = *y*.



Enveis-
funksjon

Kollisjonsresistens (svak)

Gitt inputdata *x* og tilhørende
hashverdi Hash(*x*),
det skal være praktisk umulig å finne en annen *x'*
slik at Hash(*x*) = Hash(*x'*) (svak kollisjonsresistens).

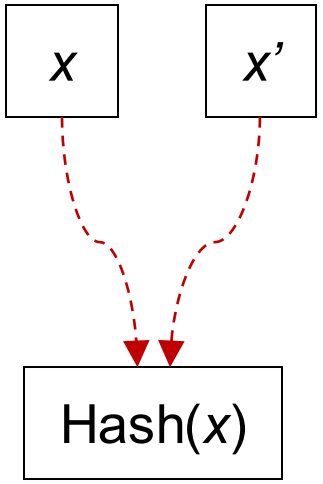


Svak kollisjons-resistens

Bursdags eksempel: Hvor mange her har vært i klasse med noen som har samme bursdag som dem?

Kollisjonsresistens (Sterk)

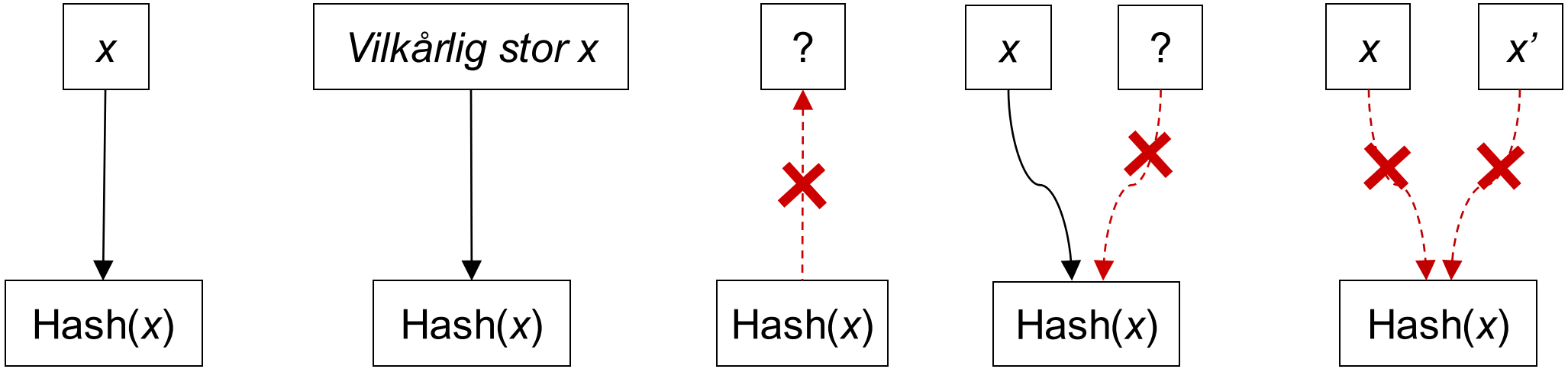
Det skal være praktisk umulig å finne to
ulike inputdata *x* og *x'*
slik at Hash(*x*) = Hash(*x'*) (sterkkollisjonsresistens).



Sterk
Kollisjons-
resistens

Bursdags eksempel: Hvor mange her har vært
i klasse der to elever (trenger ikke være deg selv) har
hatt samme bursdag?

Egenskaper ved hashfunksjoner



Lett å beregne

Komprimering til fast størrelse

Enveis-funksjon

Svak kollisjons-resistens

Sterk Kollisjons-resistens

Velkjente hashfunksjoner

- **MD5** (1991): 128 bits hash-verdi. Lett å finne kollisjoner, på grunn av liten hash-størrelse og dårlig design. Skal ikke lenger brukes.

- **SHA-1** (Secure Hash Algorithm):160 bits hash-verdi. Designet av NSA i 1995. Relativt lett å finne kollisjoner. Skal ikke lenger brukes, men forekommer i fremdeles eldre applikasjoner.

- **SHA-2** designet av NSA i 2001. Kan generere 256, 384 og 512 bits hashverdi. Betraktes som sikker. Erstatning for SHA-1.

- **SHA-3:** designet av Joan Daemen + andre i 2010. Standardisert i 2015. Kan generere: 256, 384, og 512 bits hashverdi. SHA-3 har liten bruk, fordi SHA-2 fremdeles regnes som sikker.

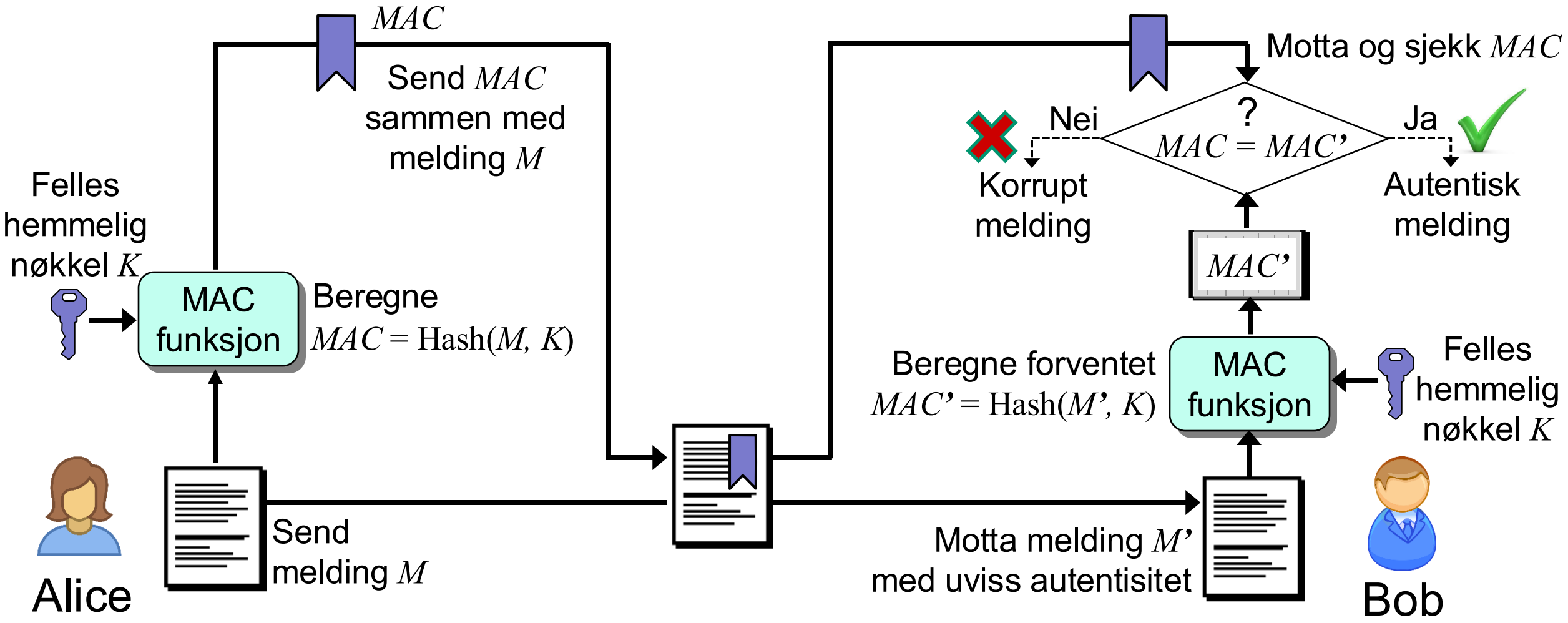
Anvendelser av hashfunksjoner

- Sammenligning av filer
- Beskyttelse av passord
- Integritetssjekk, f.eks. av SW-distribusjoner
- Generering av meldingsautentiseringskoder (MAC)
- Digitale signaturer
- Bitcoin og kryptovaluta
- Generering av pseudotilfeldige tall
- Generering av kryptonøkler

Meldingsautentiseringskode – MAC (Message Authentication Code)

- En melding *M* med en enkel hash-verdi Hash(*M*) kan lett endres av angriper.
- For å hindre angrep er det nødvendig å bruke en autentisert hash-verdi.
- MAC (meldingsautentiseringskode) inkluderer en hemmelig nøkkel *k* for beregning av hashfunksjon, som gir en autentisert hash-verdi *MAC* = Hash(*M, k*) .
- For å validere og autentisere en melding må mottakeren ha den samme hemmelige nøkkel *k* som ble brukt av avsender til å beregne *MAC*.
- En tredjepart som ikke kjenner nøkkelen kan ikke validere *MAC* -verdien.

Meldingsautentisering med MAC

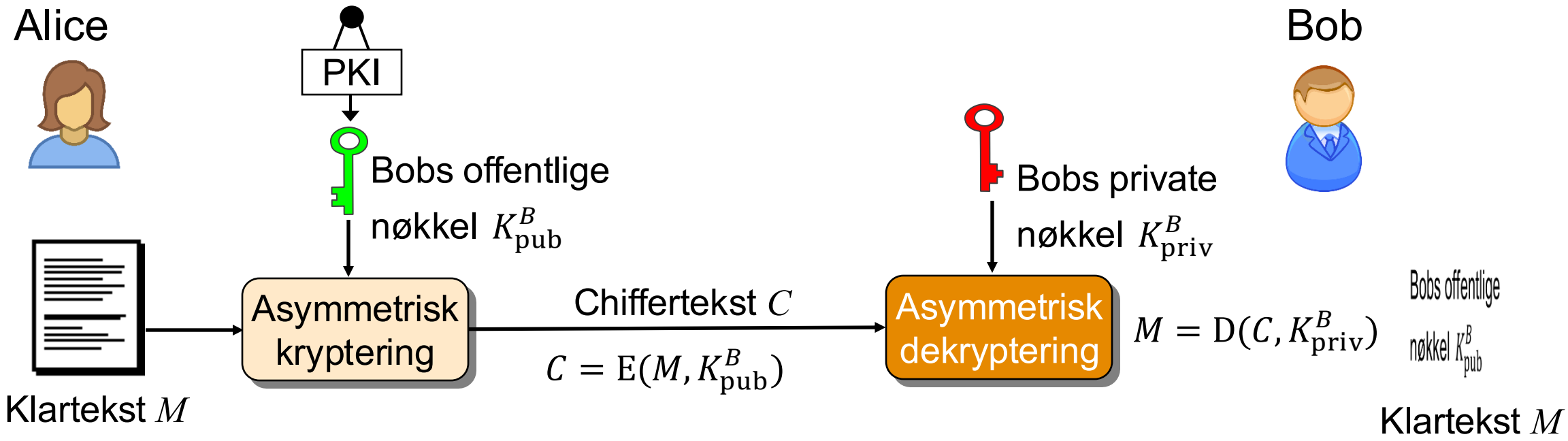


ASYMMETRISK KRYPTOGRAFI

- Asymmetrisk kryptering
- Diffie-Hellman nøkkelutveksling
- Digital signatur



Asymmetrisk kryptering – grunnleggende prinsipp



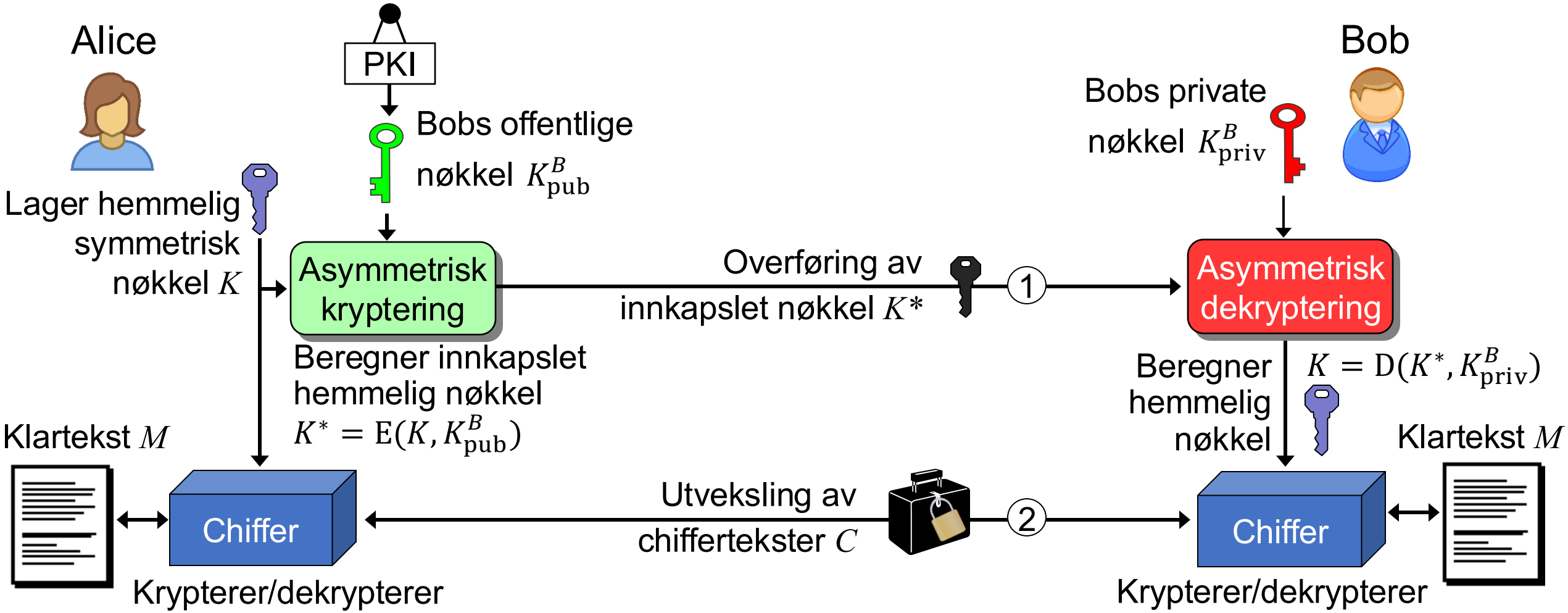
- Asymmetrisk kryptering og dekryptering krever tung beregning, og brukes ikke til direkte kryptering slik som vist over.
- I praksis brukes hybrid kryptering som kombinerer både en asymmetrisk og en symmetrisk algoritme.

Tradisjonelle asymemtriske krypteringsalgoritmer

- RSA: mest kjent asymmetrisk algoritme.
  - RSA = Rivest, Shamir og Adleman (utgitt 1977)
  - Historie: Den britiske kryptografen Clifford Cocks oppfant den samme algoritmen i 1973, men publiserte den ikke fordi det ble hemmeligstemplet.
  - Krever etterhvert store nøkler (typisk 2048 bits) for å opprettholde sikkerhet

- Elliptisk kurvekryptografi
  - Basert på vanskeligheten med å løse EC diskrete logaritme.
  - Nøkler har mindre størrelse (typisk 256 bits) enn RSA.

Hybrid kryptering (gir ikke fremoverhemmelighold)



Hybrid kryptering

- Symmetriske chiffer er mye raskere enn asymmetriske chiffer (fordi symmetriske chiffre benytter relativt enkle matematiske beregninger)
- Asymmetriske chiffer støtter effektiv nøkkeldistribusjon
- Praktisk å benytte en kombinasjon av både symmetriske og asymmetriske chiffer - et hybridsystem:
  - Det asymmetriske chifferet brukes til å distribuere en hemmelig symmetrisk nøkkel.
  - Den hemmelige symmetriske nøkkelen med det symmetriske chifferet brukes for rask kryptering av datasett og meldinger.

- Svakheten ved hybrid kryptering er at det **ikke** gir (perfekt) fremoverhemmelighold (eng. (perfect) forward secrecy).
  - Derimot kan perfekt fremoverhemmelighold oppnås ved å etablere øktnøkkel basert på Diffie-Hellman protokollen (som i TLS 1.3).

Fremoverhemmelighold (Forward secrecy)

- Fremoverhemmelighold sikrer at en angriper ikke kan dekryptere tidligere traffikk, selv om en langtidsnøkkel (som en privatnøkkel) blir komprimitert på et senere tidspunkt.

- Kalles også perfect forward secrecy (PFS), eller på norsk perfekt fremoverhemmelighold.

- For å få til fremoverhemmelighold benyttes ofte Diffie-Hellman nøkkelutveksling.

Diffie-Hellman nøkkelutveksling





Alice velger privat delnøkkel a

Bob velger privat delnøkkel b

Alice beregner offentlig $g^a \pmod p$

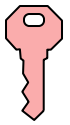
Bob beregner offentlig $g^b \pmod p$

Alice sender til Bob $\longrightarrow$ $g^a \pmod p$

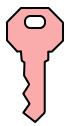
$g^b \pmod p$ $\longleftarrow$ Bob sender til Alice

Alice beregner hemmelig $g^{ab} = (g^b)^a \pmod p$

Bob beregner hemmelig $g^{ab} = (g^a)^b \pmod p$

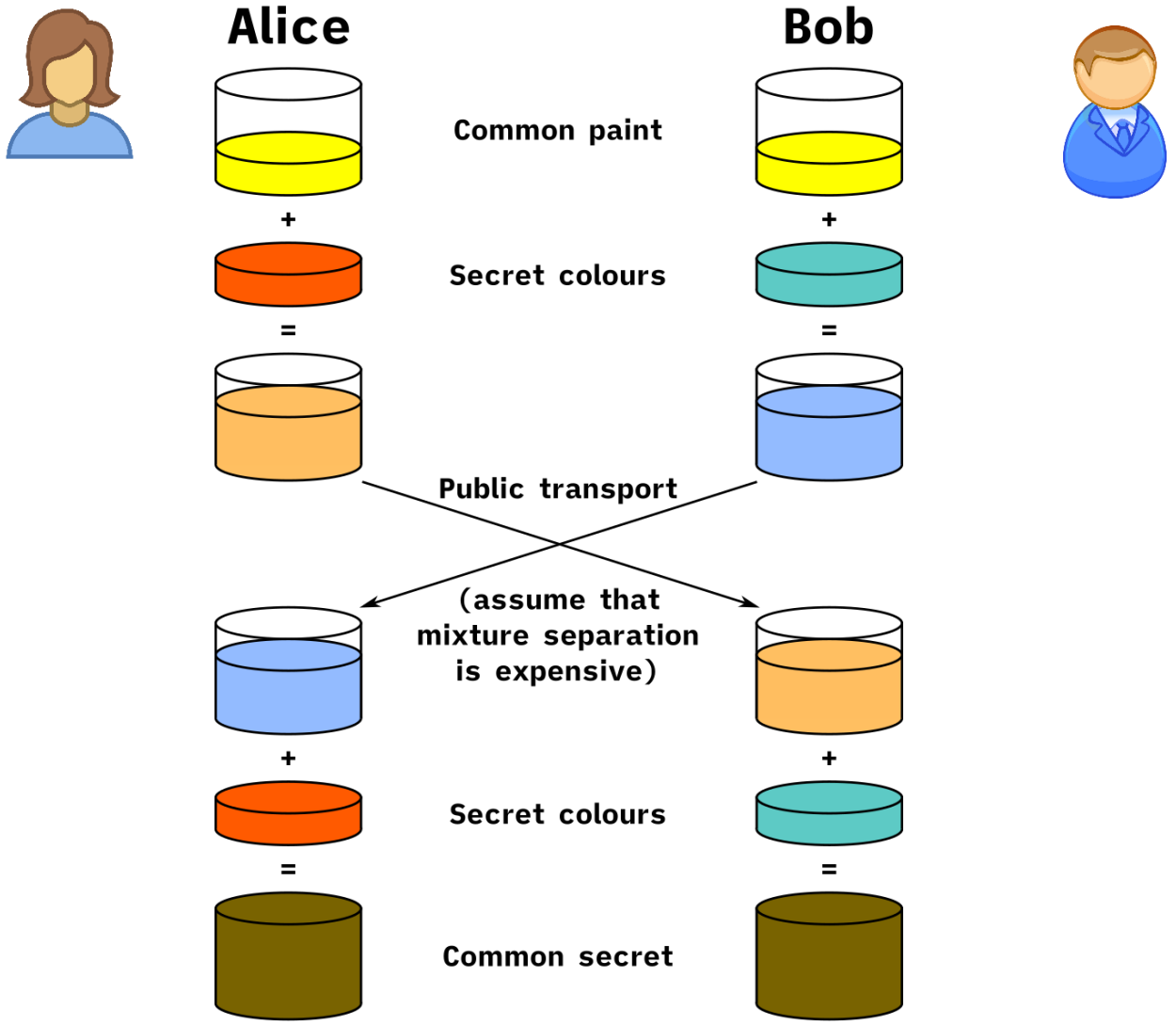


Alice og Bob har utvekslet anonym hemmelig nøkkel g^{ab}



Angripere kan ikke finne de hemmelige delnøklene *a* og *b* fordi beregning av diskret logaritme av store heltall er vanskelig. Dvs. beregning av $a = \log(g^a \pmod p)$ og $b = \log(g^b \pmod p)$ er vanskelig. Dermed kan angripere ikke beregne den hemmelige nøkkelen = $g^{ab} \pmod p$.

Diffie-Hellman nøkkelutveksling



Diffie-Hellman nøkkelutveksling





Alice velger privat delnøkkel a

Bob velger privat delnøkkel b

Alice beregner offentlig $g^a \pmod p$

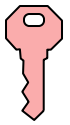
Bob beregner offentlig $g^b \pmod p$

Alice sender til Bob $\longrightarrow$ $g^a \pmod p$

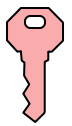
$g^b \pmod p$ $\longleftarrow$ Bob sender til Alice

Alice beregner hemmelig $g^{ab} = (g^b)^a \pmod p$

Bob beregner hemmelig $g^{ab} = (g^a)^b \pmod p$



Alice og Bob har utvekslet anonym hemmelig nøkkel g^{ab}



Angripere kan ikke finne de hemmelige delnøklene *a* og *b* fordi beregning av diskret logaritme av store heltall er vanskelig. Dvs. beregning av $a = \log(g^a \pmod p)$ og $b = \log(g^b \pmod p)$ er vanskelig. Dermed kan angripere ikke beregne den hemmelige nøkkelen = $g^{ab} \pmod p$.

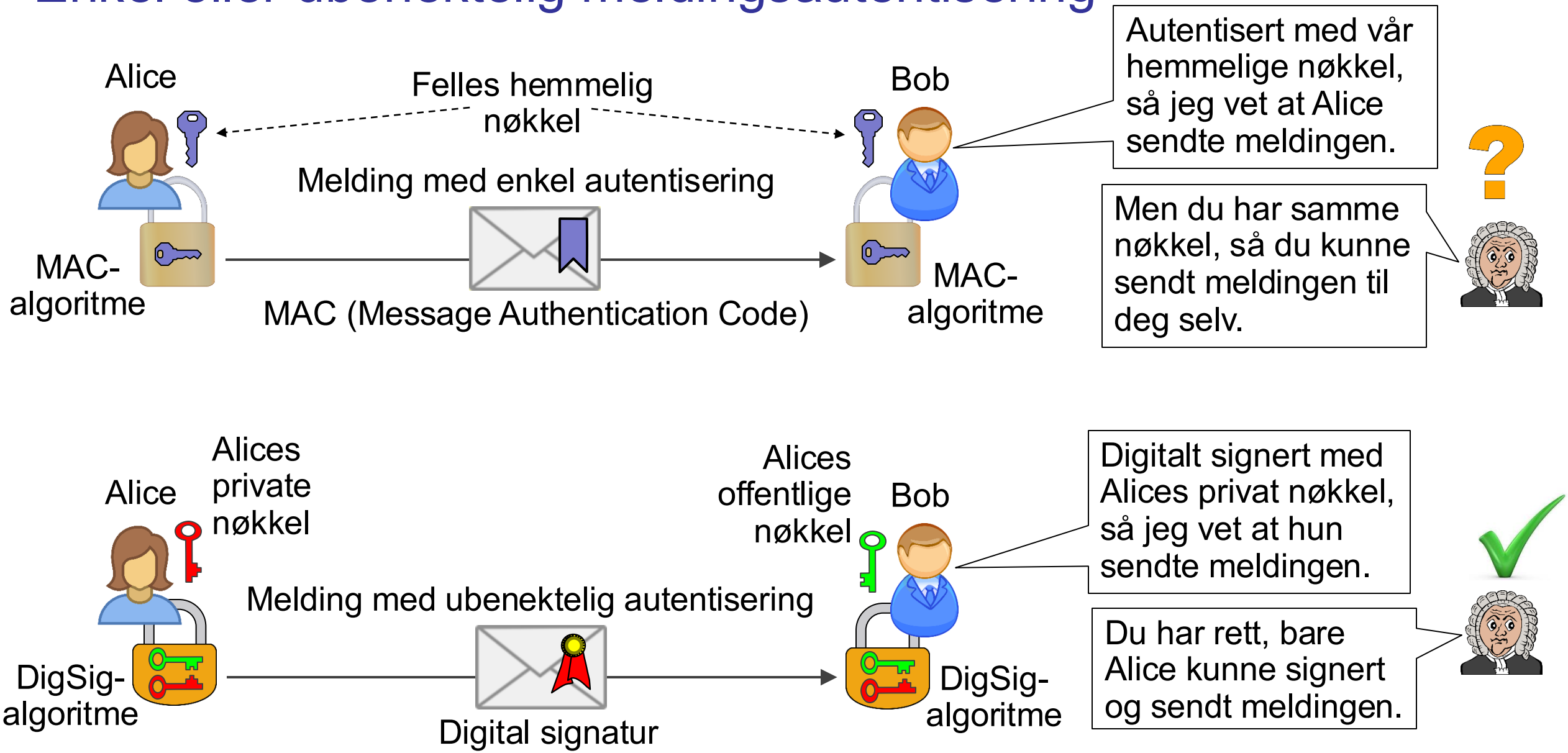
Diffie-Hellman nøkkelutveksling

- Problem:
  - Gir ingen autentisering
  - Alice og Bob kan ikke vite hvem de kommuniserer med
  - Mann-i-midten angrep mulig
- Løsning
  - Kombinasjon med digital signatur gir autentisert nøkkelutveksling
- Applikasjoner:
  - TLS (Transport Layer Security) og https
  - IKE (Internet Key Exchange) og IPSec (IP Security)

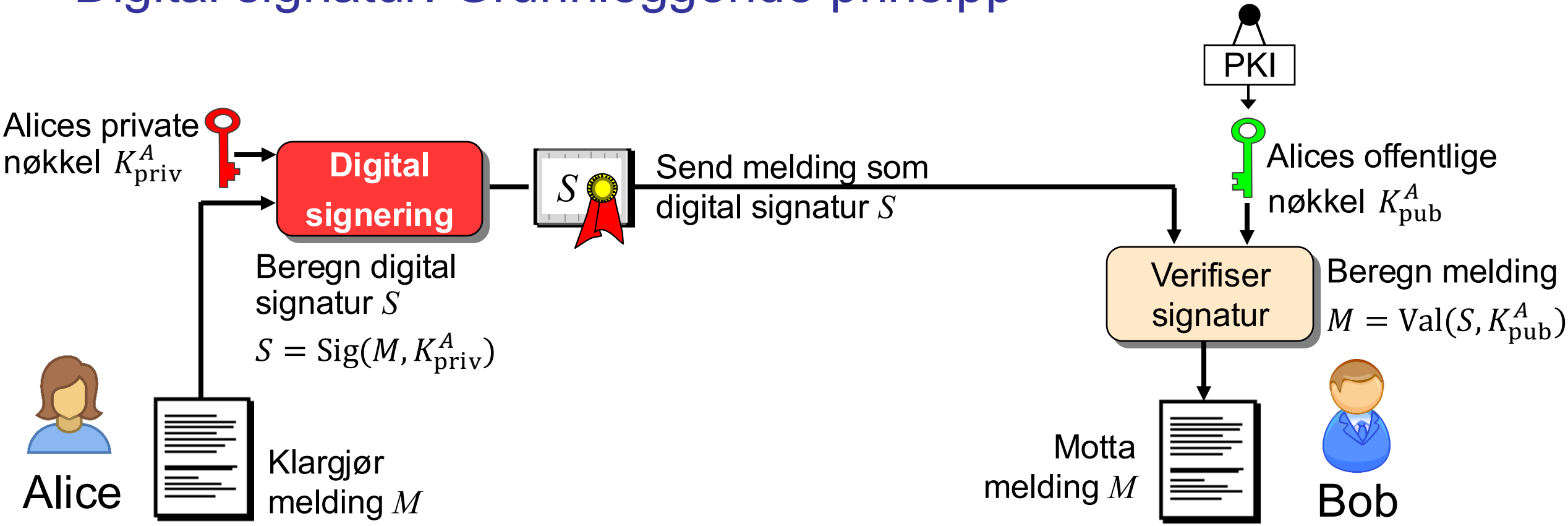
Behov for digital signatur

- En MAC kan ikke brukes som bevis for data-autentisitet som skal verifiseres av en tredjepart
- Digitale signaturer kan valideres av tredjepart
  - Gir ubenektelig (sterk) data-autentisering,
- Funksjoner for digitale signatur:
  - Signering (bruker privat nøkkel)
  - Verifisering (bruker offentlig nøkkel)

Enkel eller ubenektelig meldingsautentisering

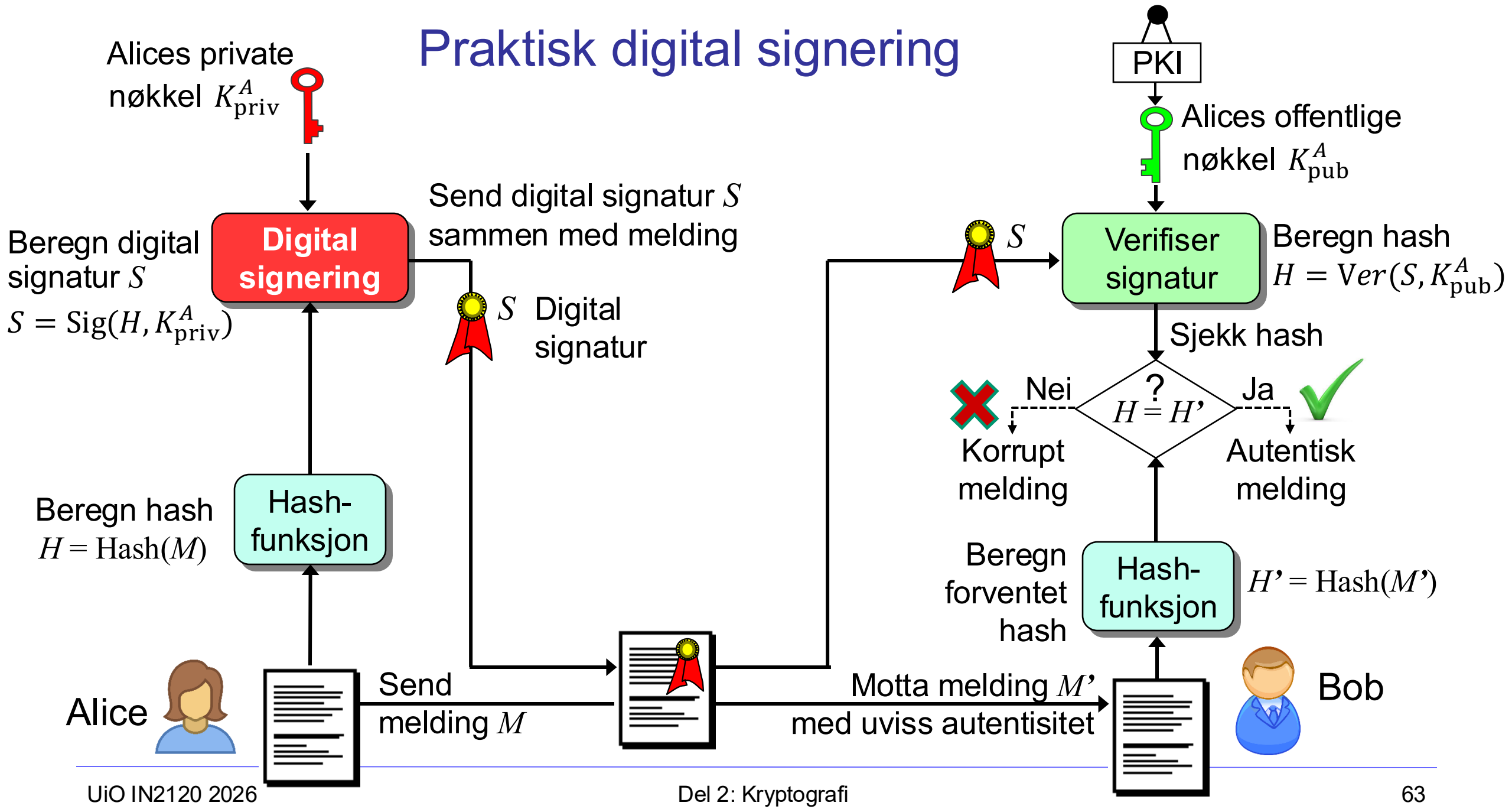


Digital signatur: Grunnleggende prinsipp

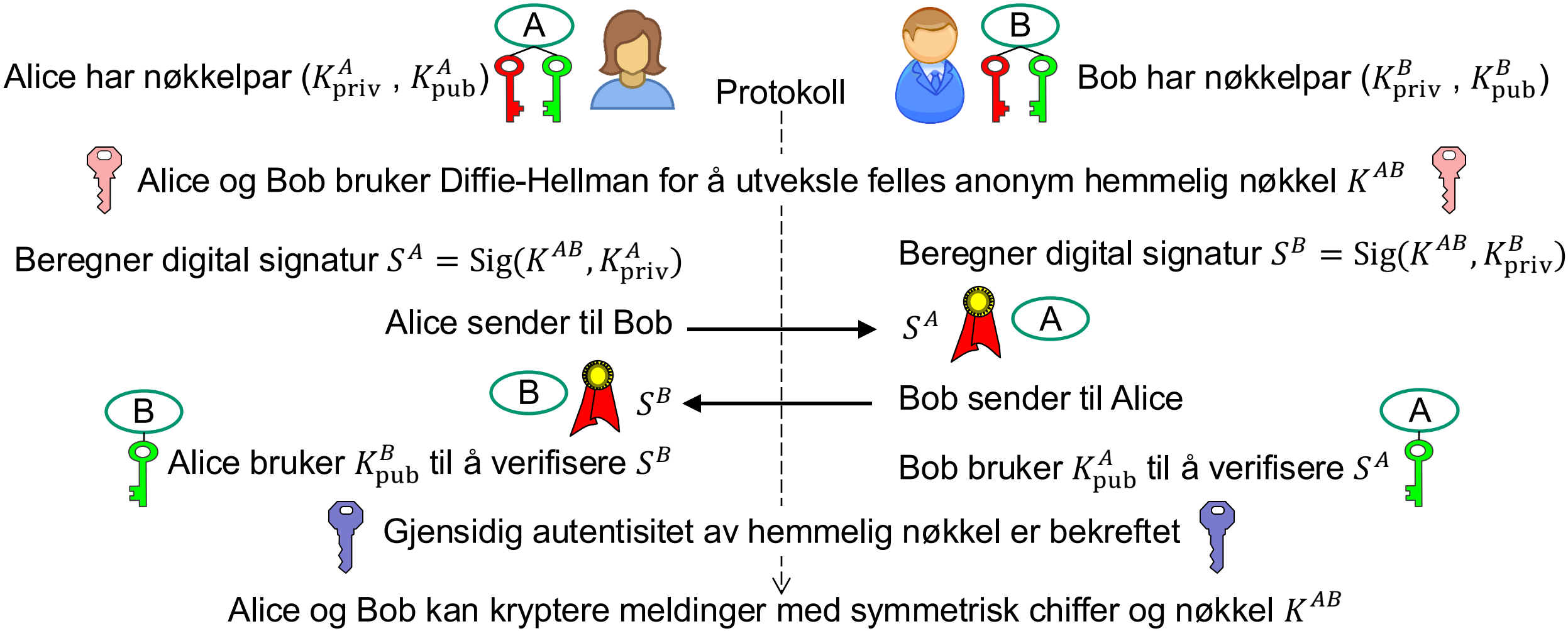


- Digital signering og validering krever tung beregning, og brukes ikke til direkte signering slik som vist over.
- I praksis brukes hybrid signering som kombinerer en hashfunksjon og digital signering.

Praktisk digital signering



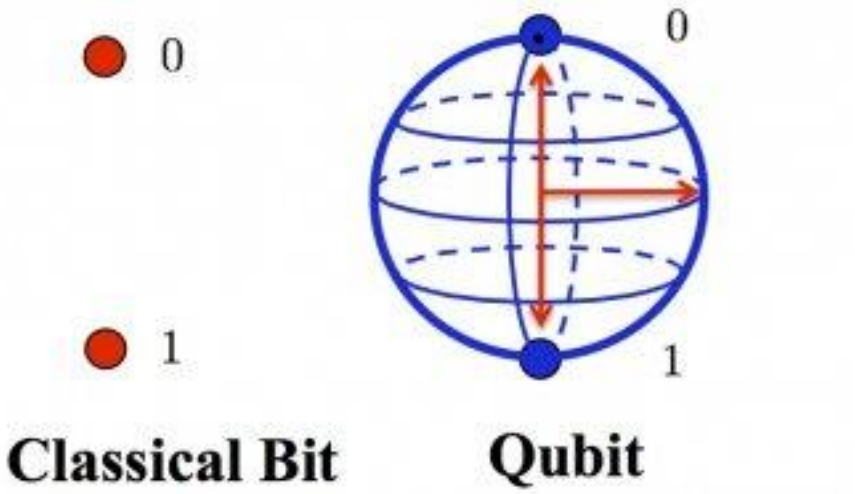
Autentisert nøkkelutveksling for fremoverhemmelighold



POSTKVANTEKRYPTOGRAFI

Kvantecomputere

- Kvanteberegning (Quantum Computing - QC) er basert på kvante «qubits» istedetfor binære bit



Eksperimentell kvantecomputer



- Kvantealgoritmer, dvs. algoritmer for kvantecomputere, kan potensielt knekke vanlige asymmetriske krypto-algoritmer, f.eks. RSA, DSA and Diffie-Hellmann.

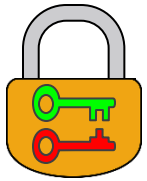
Hvordan kvantecomputere kan påvirke kryptoalgoritmer



Symmetrisk kryptografi
e.g. AES, SHA2, SHA3

Datakonfidensialitet,
-autentisitet og -integritet





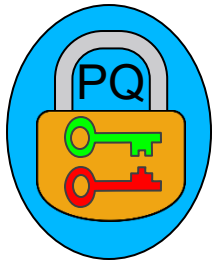
~~Tradisjonell asymmetrisk kryptografi~~
~~f.eks. RSA, DSA, Diffie-Hellman~~

~~Digital signatur, PKI og~~
~~nøkkeldistribusjon~~

Kvantetrusselen

- Symmetrisk krypto blir lite påvirket av kvantekomputere.
- Tradisjonell asymmetrisk krypto kan i teorien knekkes av (superkraftige) kvantecomputere.

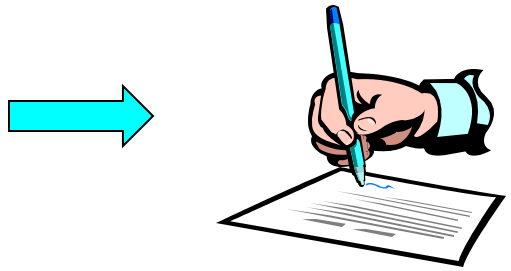
Asymmetriske kryptoalgoritmer som ikke kan knekkes av kvantecomputere



Post-Quantum asymmetrisk kryptering & digital signatur f.eks. med Lattice-based, Multivariate, Hash-based algoritmer

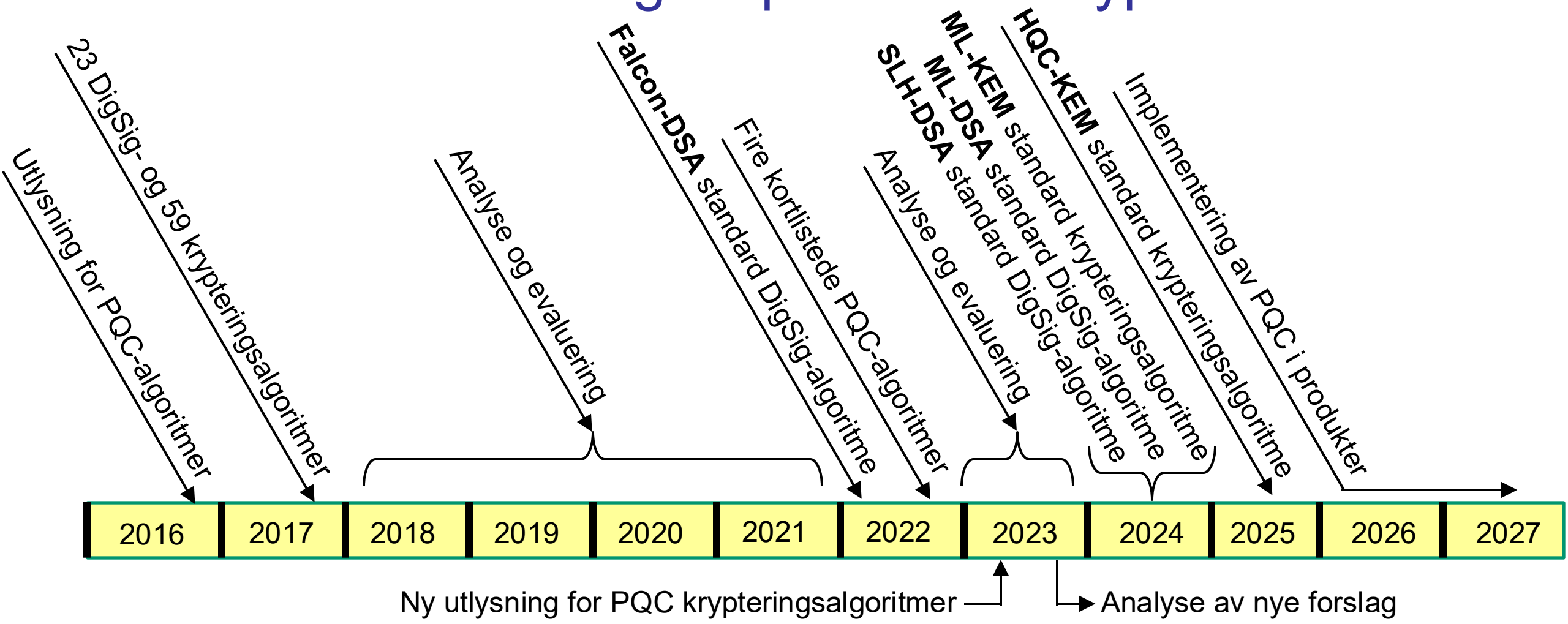


Med postkvantekrypto kan vi beholde DigSig og PKI selv med kvantecomputere som har kapasitet mer enn 1 million qubit.



Digital signatur
PKI/nøkkeldistribusjon

Standardisering av postkvantekrypto



- Begrepet “postkvantekrypto” (Post-Quantum Crypto) betyr kryptografi som ikke kan knekkes av kvantecomputere, også kalt kvanteresistente algoritmer.

Standard PQC-algoritmer, august 2025
FIPS: Federal Information Processing Standard

ML-KEM: FIPS 203

- **M**odule-**L**attice-Based **K**ey-**E**ncapsulation **M**echanism
- Kallenavn: «CRYSTALS-Kyber»

HQC-KEM

- **H**amming **Q**uasi-**C**yclic **K**ey-**E**ncapsulation **M**echanism

Falcon-DSA

- **D**igital **S**ignature **Al**gorithm

ML-DSA: FIPS 204

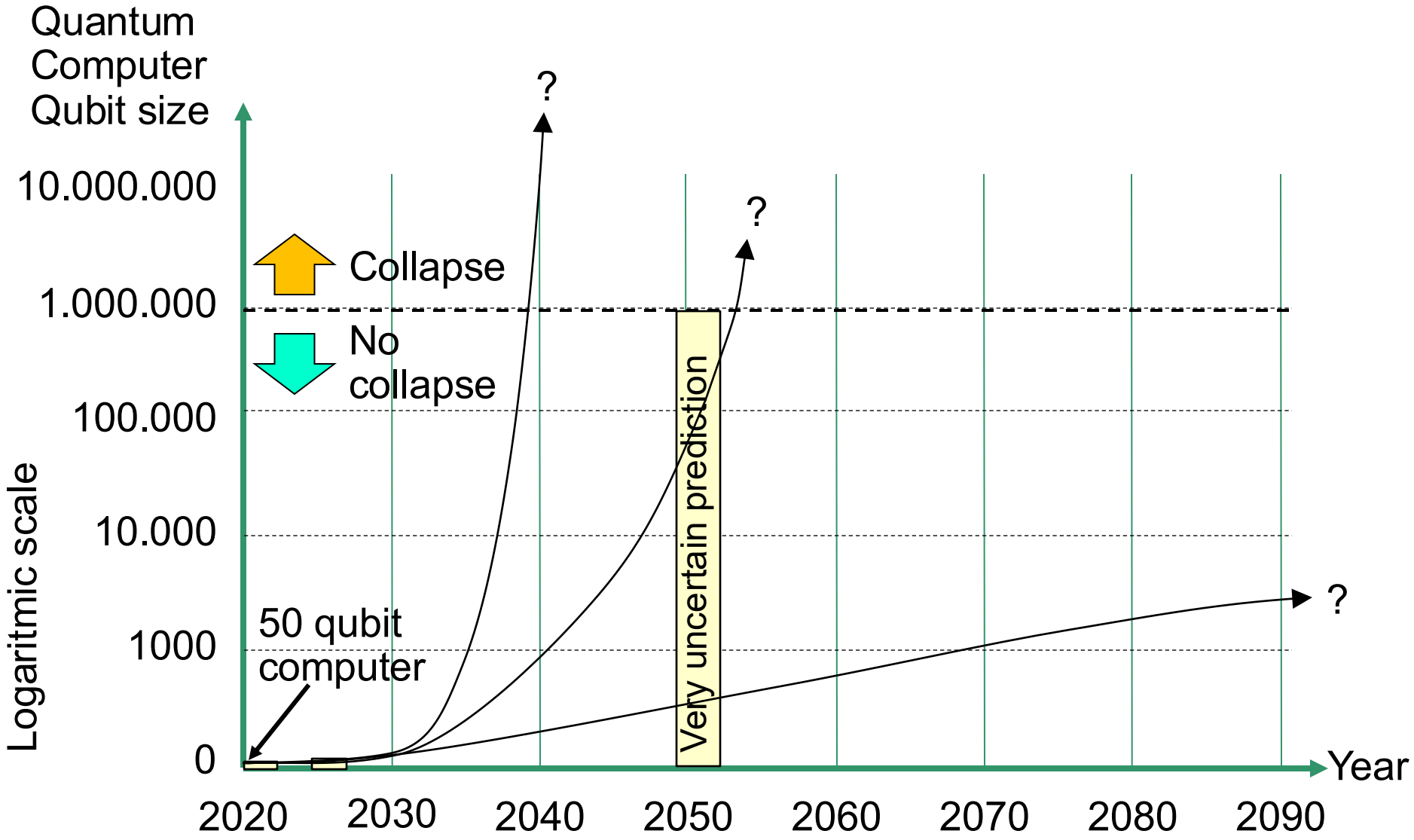
- **M**odule-**L**attice-Based **D**igital **S**ignature **A**lgorithm
- «CRYSTALS-Dilithium»

SLH-DSA: FIPS 205

- **S**tate**L**ess **H**ash-Based **D**igital **S**ignature **Al**gorithm
- Kallenavn: «SPHINCS+»

NIST ønsker å standardisere nye DigSig-algorithmer som ikke er basert på lattice eller hash. Dette for å ha variasjon i det matematiske grunnlaget i tilfellet matematisk gjennombrudd som gjør at alle PQC-algoritmer med samme matematikk, knekkes.

Sammenbrudd av trad. asymmetrisk crypto?



Hvorfor gå over til postkvantekryptografi?

A. Bruk postkvantekrypto fordi kvantecomputere sannsynligvis vil knekke RSA, Diffi-Hellmann og DSA en gang i fremtiden.
B. Bruk postkvantekrypto fordi du ikke vil at organisasjonen din skal havne på forsiden i avisen, beskyldt for å være uansvarlig.


SINET ≡ DN.no Logg inn Kunde

Norge kan snart ikke vokte statshemmeligheter lenger

TEKNOLOGI

TEKST Osman Kibar FØLG MEG
FOTO Åge Peterson, Gorm K. Gaare & Helge Skodvin
01 DESEMBER 2017 - KOLSÅS/OSLO

DN.no, 1. desember 2017

Aftenposten.no, 10. mai 2018

https://www.aftenposten.no

Aftenposten A-magasinet Osloby Sport Meninger



For abonnenter

Norges hemmeligheter skjules bak sterke krypteringer. Om få år vil kodene kanskje kunne knekkes.

SLUTT PÅ PRESENTASJONEN