Kryptering og steganografi

EJHJUBM SFQSFTFOUBTKPO FS FU LVMU GBH

Jeg avlytter viktig informasjon, sa smarte Tor.

Læreboka kapittel 19

12. november 2008

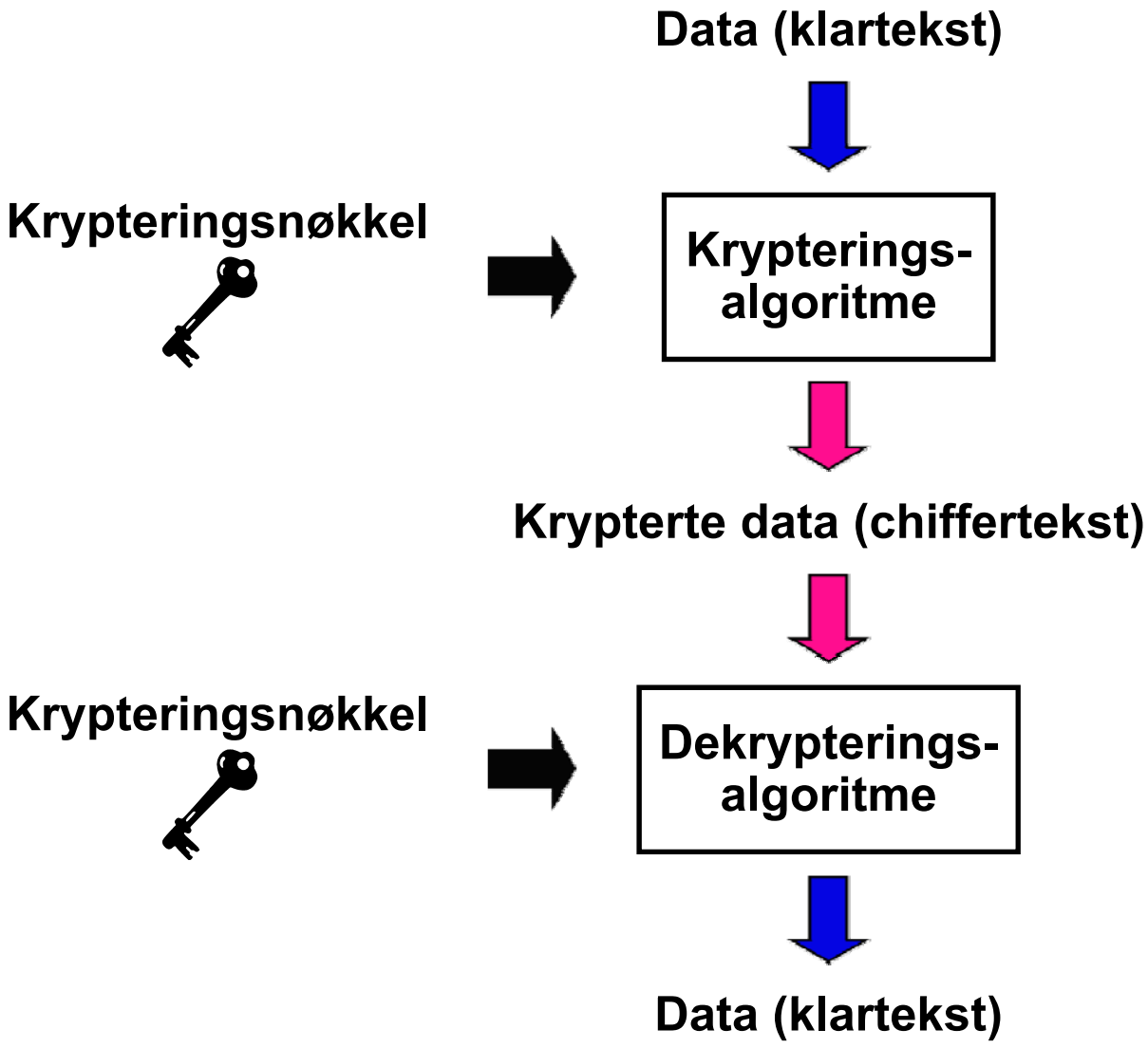
Læringsmål – kryptering og steganografi

- Forstå ulike krypteringsprinsipper.
- Kunne sentrale begreper.
- Kjenne til en del sentrale teknikker.
- Kjenne til steganografi og vannmerking.

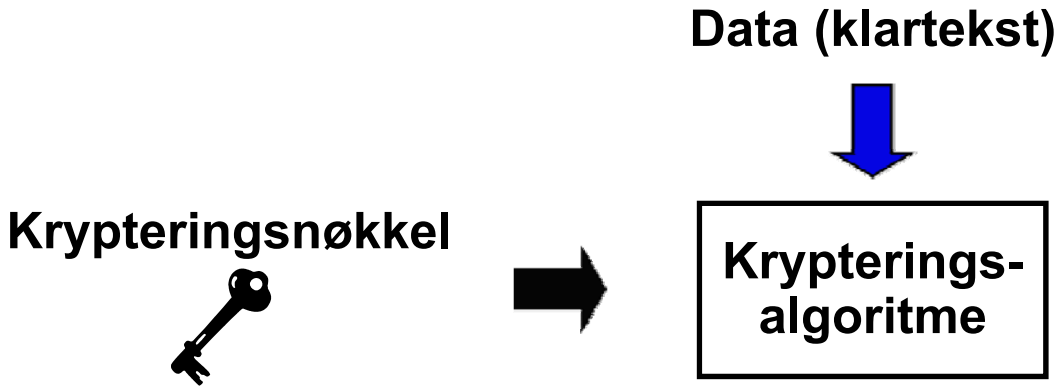
Kryptologi

- Kryptografi
- Kryptering
- Kryptoanalyse

Kryptering av data



Kerckhoffs' assumption



A. Kerckhoff, 1883:

"The security of a cipher must not depend on anything that cannot be easily changed"

"The opponent is not to be underestimated. In particular, the opponent knows the encryption and decryption algorithms. So the strength of a cipher system depends on keeping the key information secret, not the algorithm"

Et enkelt eksempel: Cæsars kode

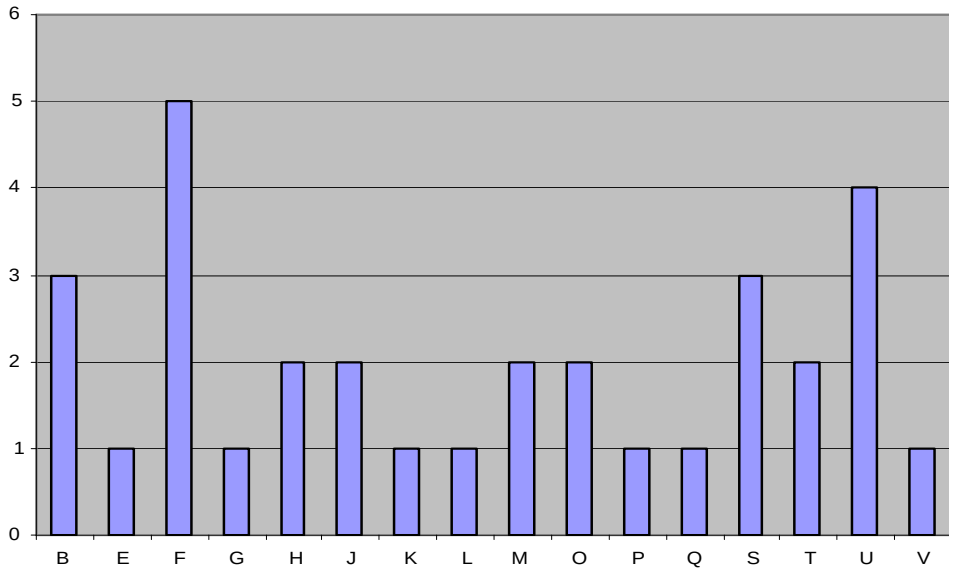
- **Krypteringsalgoritme:**
  Bytt ut hver bokstav med en bokstav tre plasser lenger ut i alfabetet. (Ved slutten av alfabetet, "wrap around".)

  Eksempel: `DIGITAL REPRESENTASJON ER ET KULT FAG`
  `GLJLWDO UHSUHVHQWDVMRQ HU HW NXOW IDJ`

- **Dekrypteringsalgoritme:**
  Bytt ut hver bokstav med bokstaven tre plasser tidligere i alfabetet.

Generalisering av Cæsars kode

- Bytt ut hver bokstav med en bokstav k plasser lenger ut i alfabetet.
- Verdien av k er nøkkelen.
- Hvordan bryte koden når nøkkelen er ukjent?
  - ”Brute force”:
    Prøv alle N muligheter for k, der N er antall tegn i alfabetet, og se hvilken forskyvning som gir et forståelig resultat. (”Exhaustive search of the key space”.)
  - Frekvensanalyse:
    Lag et histogram for tegnene i den krypterte meldingen.
    (Hyppigste bokstaver i norsk:
    e r n t s i a d g l o)



Cæsars kode forts.

- Cæsars kode er et eksempel på
  - substitusjonsprinsippet med syklisk permutasjon.
  - monoalfabetisk substitusjon.
  - symmetrisk kryptering.
  - stream-kryptering.

Krypteringsprinsipper

- Krypteringsnøkkel
  - Symmetrisk kryptering
  - Asymmetrisk kryptering
- Håndtering av data
  - Stream-kryptering
  - Blokk-kryptering
- Algoritmer
  - Substitusjonsprinsippet
  - Transformasjonsprinsippet

Symmetrisk kryptering

- Kryptering av klartekst P med nøkkel k gir chiffertekst C:

  $E_k(P) \rightarrow C$

- Dekryptering av chiffertekst C med *samme* nøkkel k gir klartekst P:

  $D_k(C) \rightarrow P$

Vigenère kryptering

- Ligner Cæsars kryptering, men bruker en *frase* som nøkkel.
- Symmetrisk stream-kryptering med *polyalfabetisk* substitusjon.
- Eksempel:

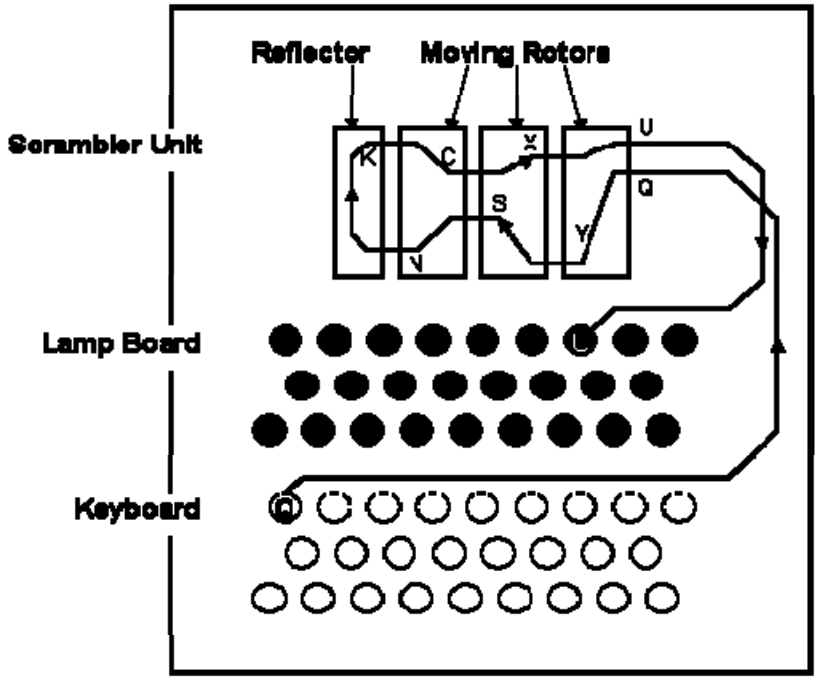
  nøkkel `HEI`

  `REPRESENTASJON`
  `HEIHEIHEIHEIHE`
  `YIXYIALRBHWRVR`

	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

Enigma - Rotormaskin

- Symmetrisk stream-kryptering som bygger på substitusjonsprinsippet.
- Nøkkelen er startposisjonen for rotorene.
- Tilsvarer en nøkkellengde på 380 biter (med “plugboard”).



http://www.math.miami.edu/~harald/enigma/enigma.gif

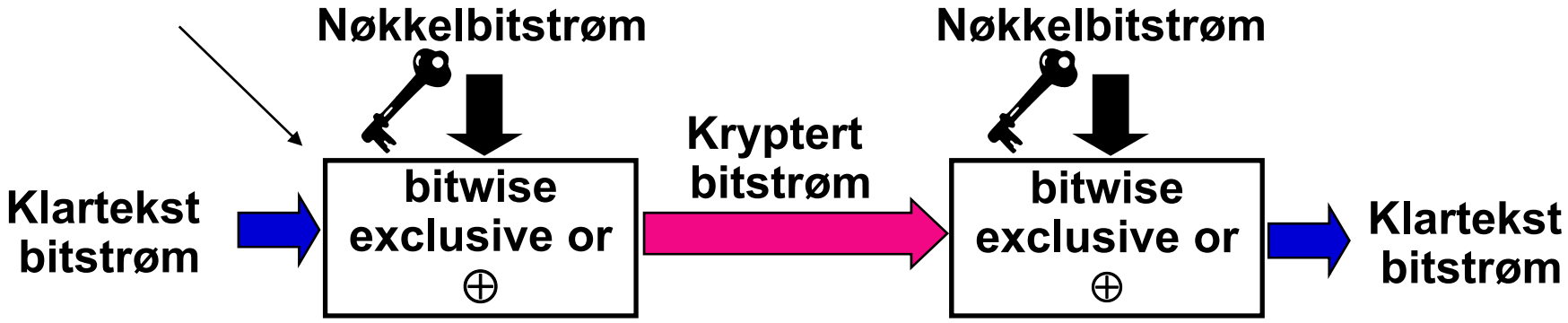
Hva medfører datateknologien?

- Mulighet for mer raffinerte krypterings- og dekrypteringsalgoritmer
- Vi krypterer biter og bytes, ikke tegn
  – krypteringsalgoritmene ser altså ikke forskjell på tekst, bilder og lyd
- Kraftigere verktøy for kryptering…
- …men også kraftigere verktøy for kryptoanalyse
- Et enkelt regnestykke:
  - Nøkkellengde 128 biter, antall mulige nøkler er 2^{128}
  - La oss anta at det tar 0,001 sekund å sjekke ut en nøkkel…
  - …da blir tidsforbruket i gjennomsnitt

$$0{,}5 * 0{,}001 * 2^{128}\ s = 1{,}7 * 10^{35}\ s = 5{,}4 * 10^{27}\ år$$

Vernam-kryptering

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



Eksempel:
'to og to ' i UTF-8: 74 6F 20 6F 67 20 74 6F 20
= 01110100 01101111 00100000 01101111 01100111 00100000 01110100 01101111 00100000

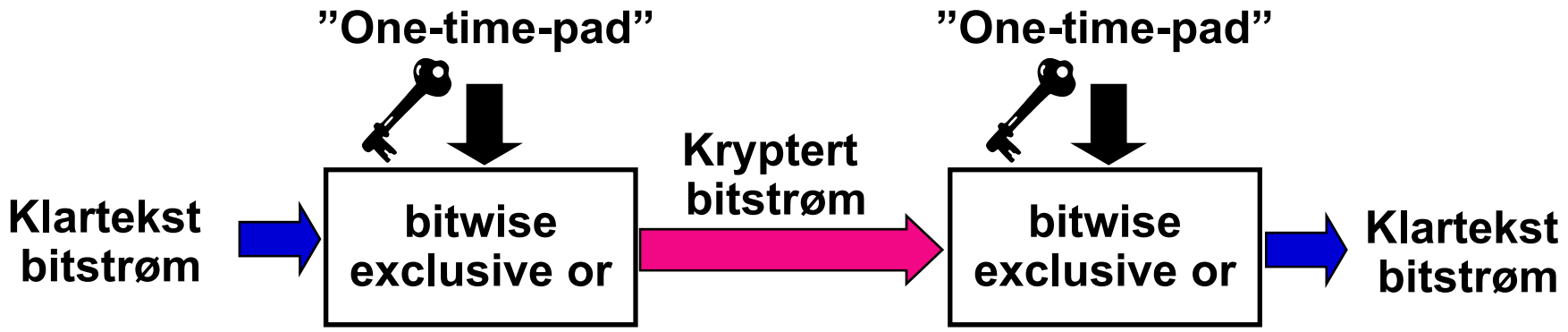
Tilfeldig valgt nøkkel: 01110011 01100101 01111000

klartekst	01110100	01101111	00100000	01101111	01100111	00100000	01110100	01101111	00100000
nøkkel	01110011	01100101	01111000	01110011	01100101	01111000	01110011	01100101	01111000
chiffertekst	00000111	00001010	01011000	00011100	00000010	01011000	00000111	00001010	01011000
nøkkel	01110011	01100101	01111000	01110011	01100101	01111000	01110011	01100101	01111000
klartekst	01110100	01101111	00100000	01101111	01100111	00100000	01110100	01101111	00100000

”One-time-pad”

- **Vernam-kryptering med en tilfeldig nøkkel minst så lang som meldingen, *og som brukes bare en gang***
- **Bevist av Shannon å være 100 % sikker**
  Intuitivt: Enhver tekst med et gitt antall tegn kan genereres med en passende nøkkel

Når ”one-time-pad” er 100 % sikker, hva er da problemet?



Pseudotilfeldige tall

- Mange moderne krypteringsteknikker bygger på sekvenser av pseudotilfeldige tall
- Pseudotilfeldige tall ser ut som tilfeldige tall, men hvert tall i sekvensen er beregnet på grunnlag av det forrige.
  Det første tallet er beregnet på grunnlag av en startverdi (”seed”).
- Nøkkelen består av ”seed” og eventuelle konstanter
  (se neste lysark).
- Med denne nøkkelen og tilgang til algoritmen for beregning av de pseudotilfeldige tallene kan tallsekvensen regenereres av mottakeren av meldingen!
- Men: Robusthet mot knekking er ikke bevist.

Algoritmer for pseudotilfeldige tall

- Eksempel på algoritme (ikke særlig velegnet for kryptering)

  Linear Congruential Pseudo-number Generator

  $$x_{n+1} = (C*x_n + D) \% M$$

  der `C`, `D` og `M` er konstanter valgt slik at:
  - `C` og `D` er relativ prim (ingen felles faktorer)
  - `C-1` er delbar med alle primfaktorer i `M`
  - hvis `M` er delelig med 4, er også `C-1` det.
- Da gir algoritmen alle tall fra 0 til M-1 i "tilfeldig" rekkefølge før sekvensen gjentas.
- Eksempel: C=5, D=3, M=16, seed=5 gir
  12, 15, 14, 9, 0, 3, 2, 13, 4, 7, 6, 1, 8, 11, 10
- For å få en noenlunde sikker nøkkel må M være stor! (> 128 biter)

Krypteringsprinsipper

- Krypteringsnøkkel
  - Symmetrisk kryptering
  - Asymmetrisk kryptering
- Håndtering av data
  - Stream-kryptering
  - Blokk-kryptering
- Algoritmer
  - Substitusjonsprinsippet
  - Transformasjonsprinsippet

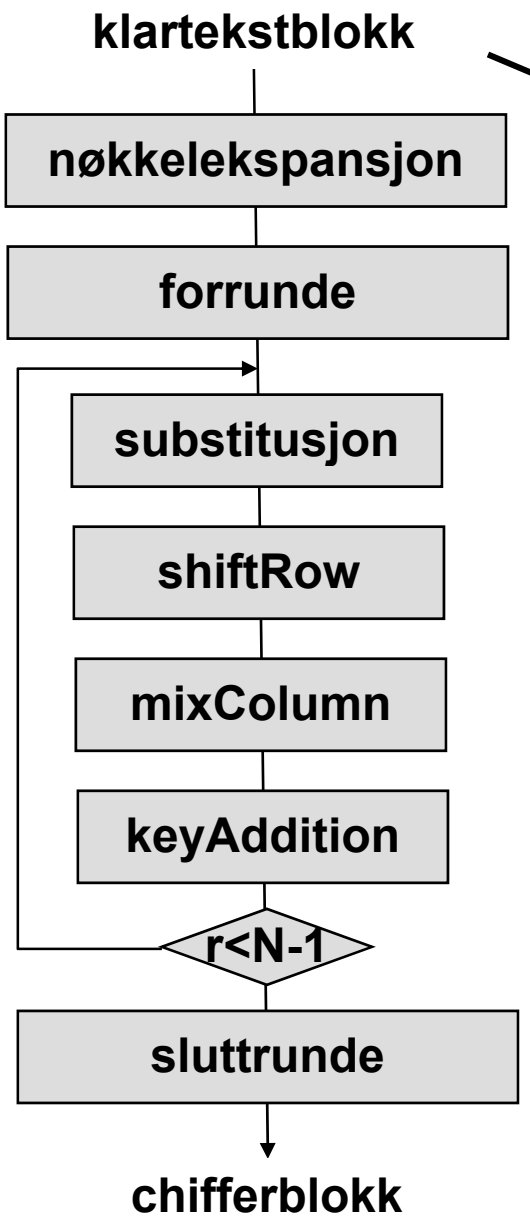
Blokk-kryptering

- Blokk-kryptering opererer på en blokk av biter
  – typisk fra 64 til 384 – ad gangen
- Blokk-kryptering åpner for permutasjon av bitene i blokken
- To hovedprinsipper
  - ”Confusion”
  - ”Diffusion”

Advanced Encryption Standard – AES

- Vinner av en konkurranse utlyst av
  NIST (National Institute of Standards and Technology) i USA
  – vinneren kåret i oktober 2000
- Konstruert av to belgiere, Joan Daemon og Vincent Rijmen, under navnet *Rijndael*
- Blokk- og nøkkellengde 128, 192 eller 256 biter
- Ingen patenter, kan benyttes av alle
- Motstandsdyktig mot alle kjente kryptoanalysemetoder
- Enkel, rask, lett å implementere i maskin- og/eller programvare
- Avløser DES – Data Encryption Standard

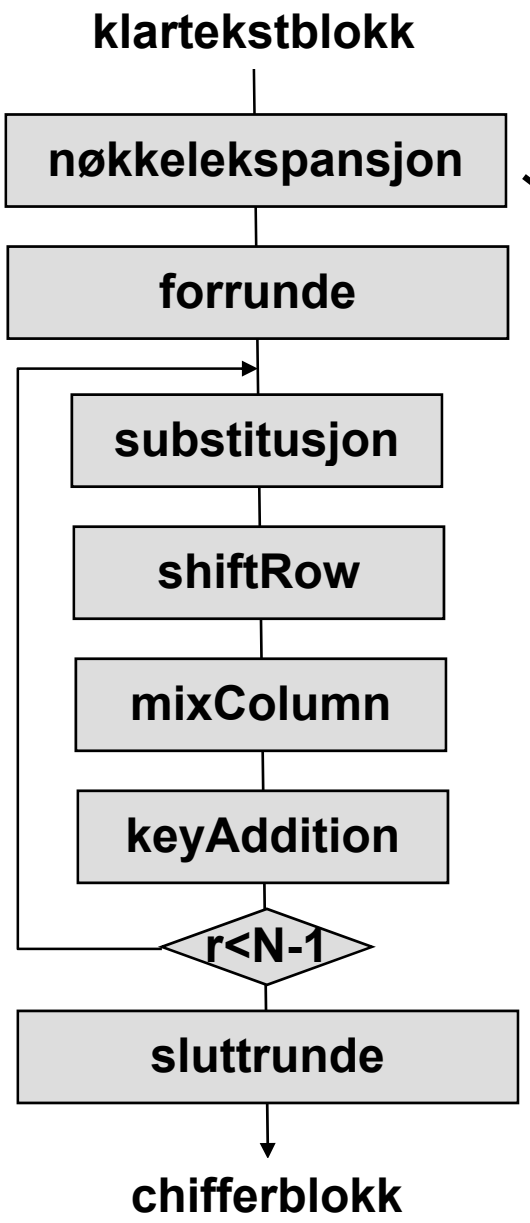
AES – virkemåte (1)

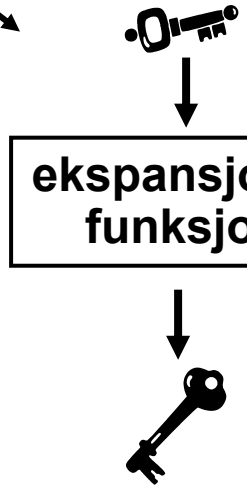


$t_{0,0}$	$t_{0,1}$	$t_{0,2}$	$t_{0,3}$
$t_{1,0}$	$t_{1,1}$	$t_{1,2}$	$t_{1,3}$
$t_{2,0}$	$t_{2,1}$	$t_{2,2}$	$t_{2,3}$
$t_{3,0}$	$t_{3,1}$	$t_{3,2}$	$t_{3,3}$

- Klartekstblokken oppfattes som en 4 * 4 tabell (matrise) med bytes (AES-standarden)
- Generelt kan større blokker og matriser kan benyttes
- Antall runder (10, 12 eller 14) avhenger av blokkstørrelsen

AES – virkemåte (2)

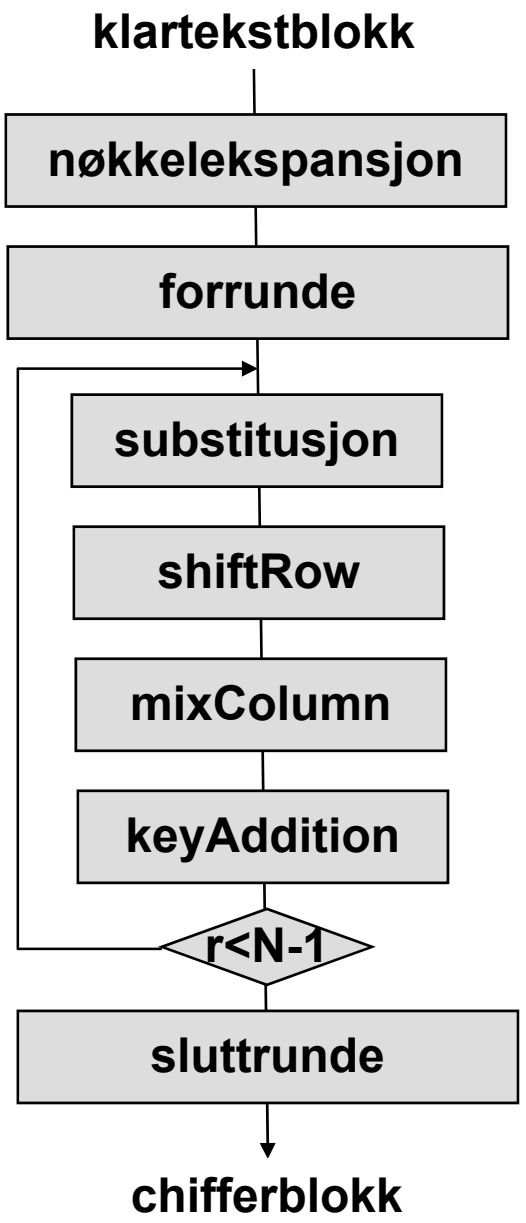


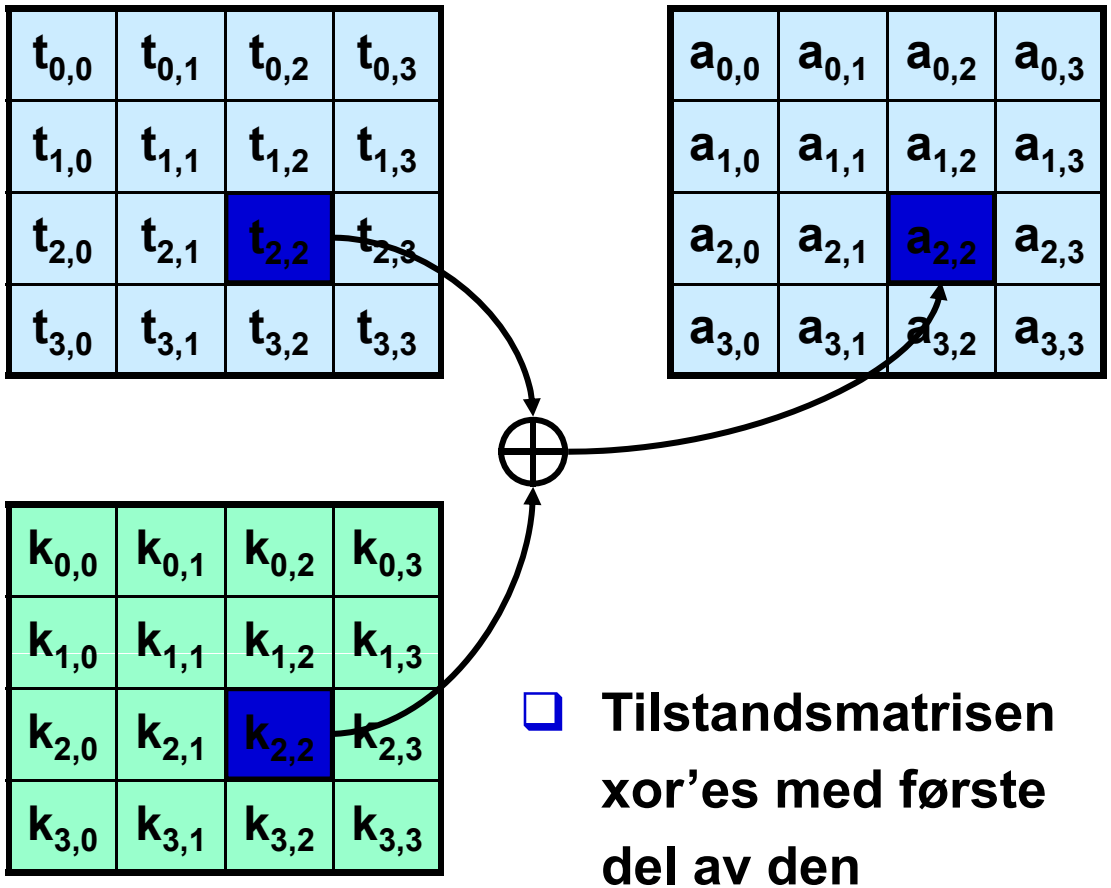


(Rijandel's key schedule)

pseudo-tilfeldig bitmønster
av tilstrekkelig lengde

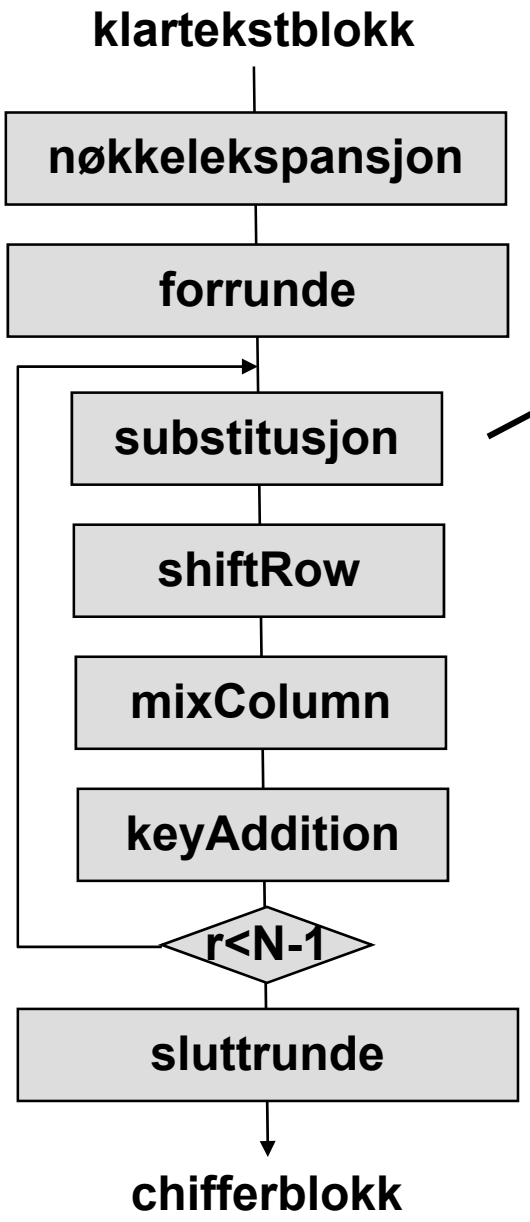
AES – virkemåte (3)

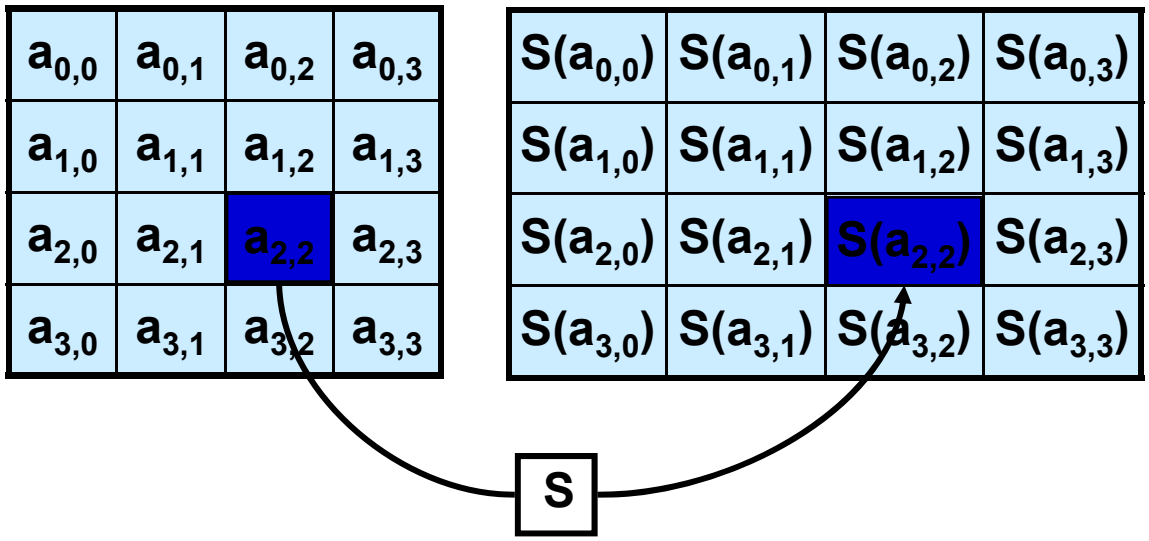




- Tilstandsmatrisen xor'es med første del av den ekspanderte nøkkelen.

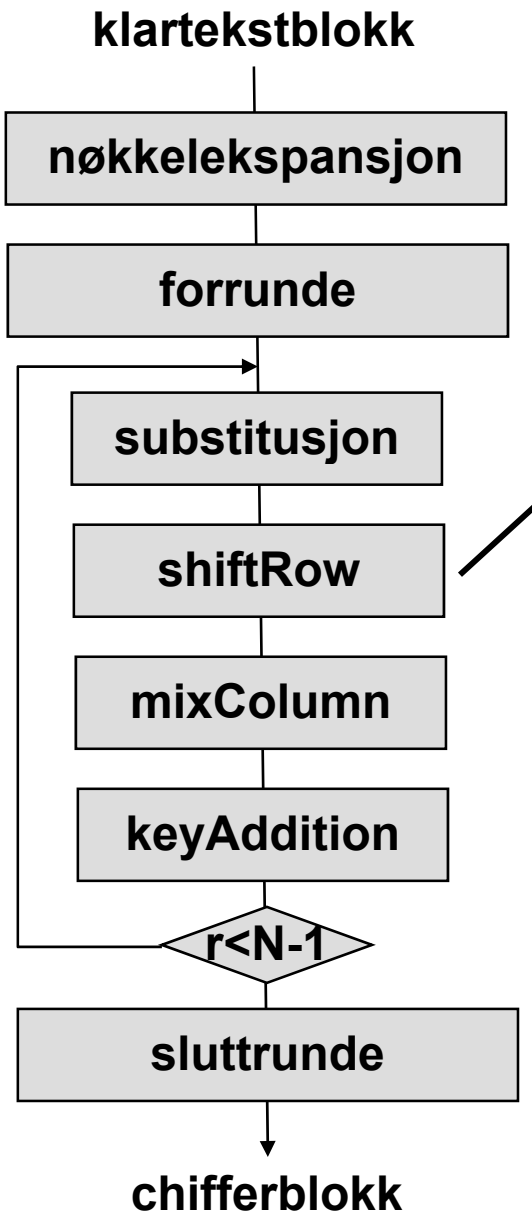
AES – virkemåte (4)

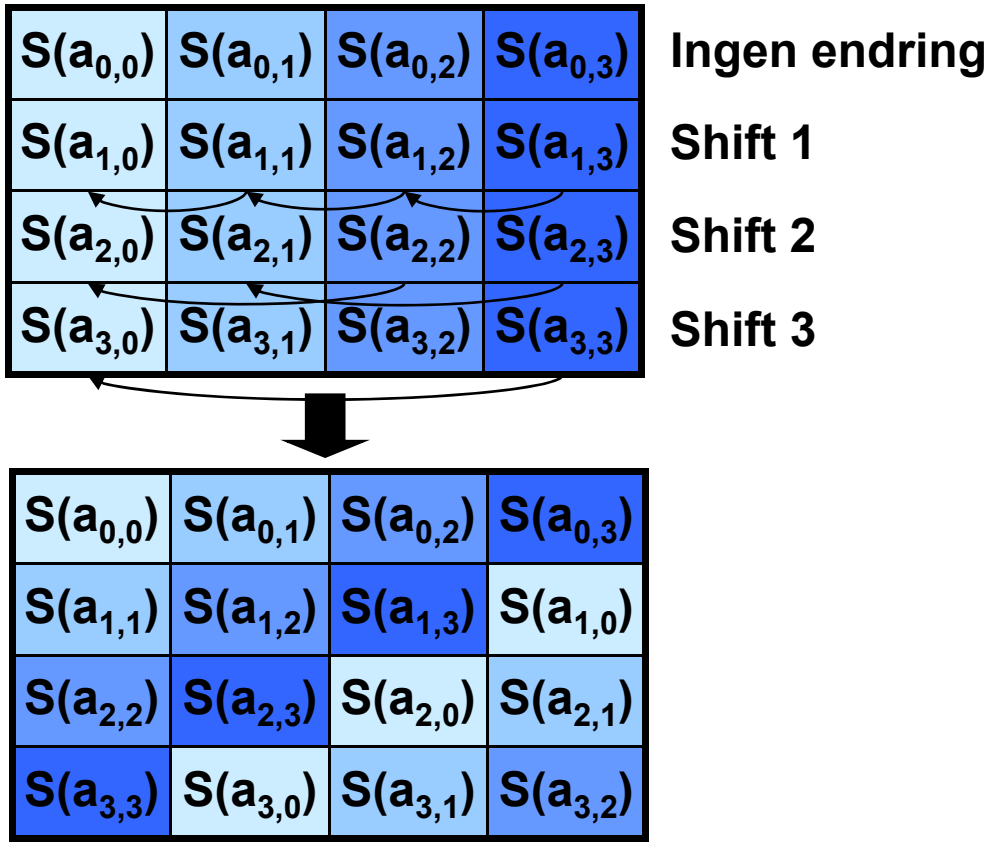




- Hver enkelt byte erstattes med en annen byte som finnes ved å slå opp i en tabell S med 2^8 elementer.
- Tabellen er konstruert slik at vi får best mulig ”confusion”.

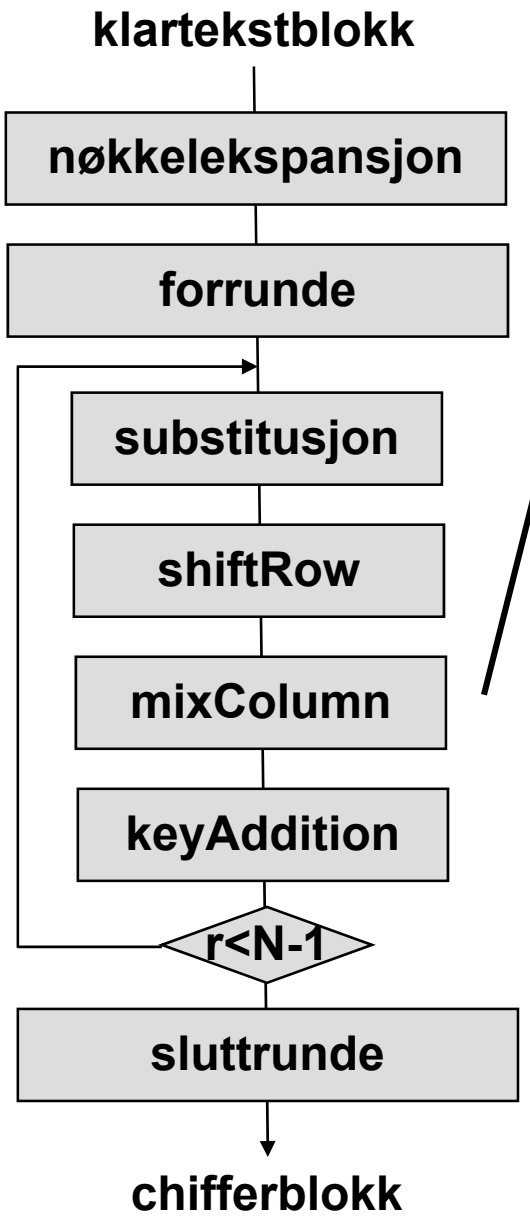
AES – virkemåte (5)





- Linjene i tilstandsmatrisen roteres.
- Dette er første ledd i ”diffusion”.

AES – virkemåte (6)



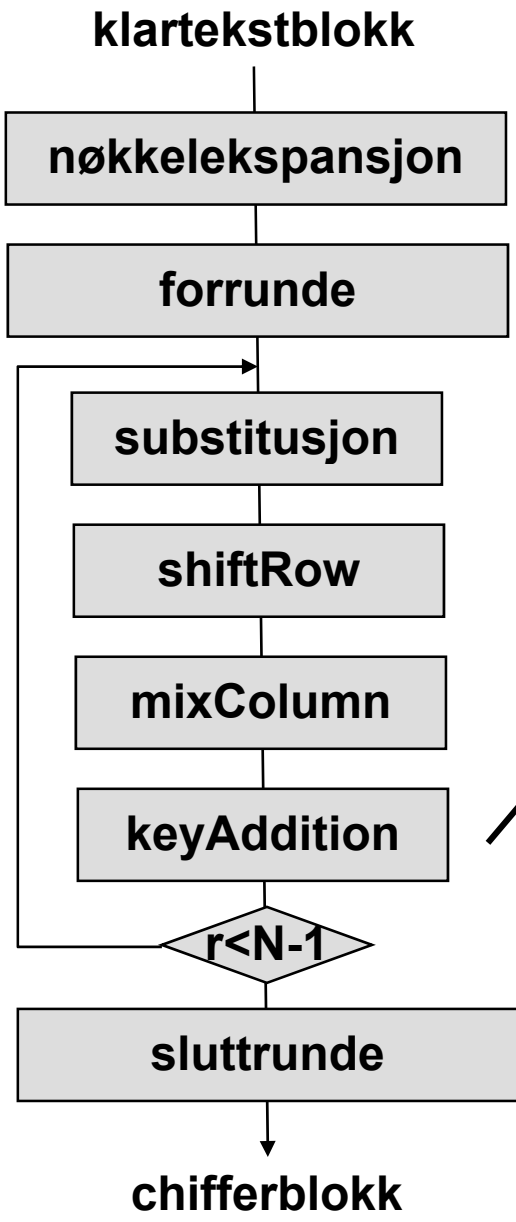
$S(a_{0,0})$	$S(a_{0,1})$	$S(a_{0,2})$	$S(a_{0,3})$
$S(a_{1,1})$	$S(a_{1,2})$	$S(a_{1,3})$	$S(a_{1,0})$
$S(a_{2,2})$	$S(a_{2,3})$	$S(a_{2,0})$	$S(a_{2,1})$
$S(a_{3,3})$	$S(a_{3,0})$	$S(a_{3,1})$	$S(a_{3,2})$

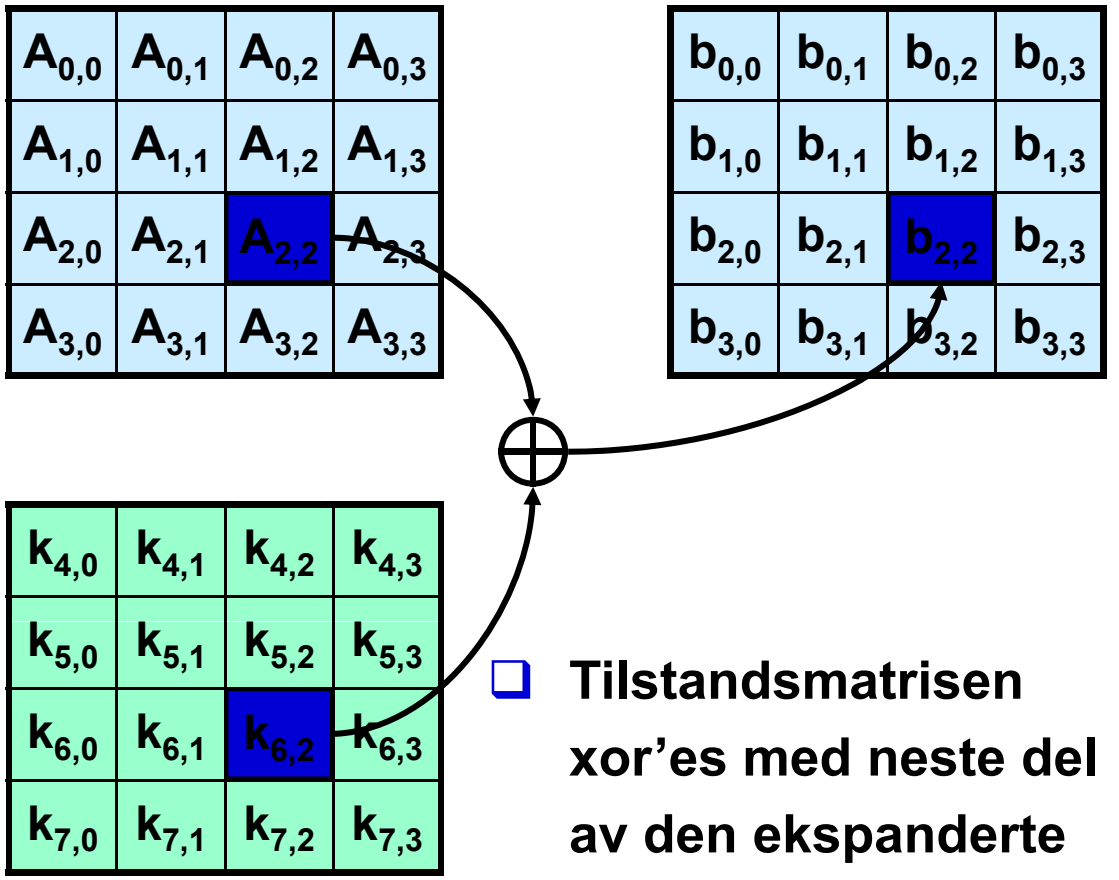
$\otimes$ c(x)

$A_{0,0}$	$A_{0,1}$	$A_{0,2}$	$A_{0,3}$
$A_{1,0}$	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$
$A_{2,0}$	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$
$A_{3,0}$	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$

- Kolonnene matrisemultipliseres med en fast vektor c(x).
- Det innebærer at innholdet i en S(a)-byte påvirker innholdet i 4 A-bytes.
- Dette er andre ledd i ”diffusion”.

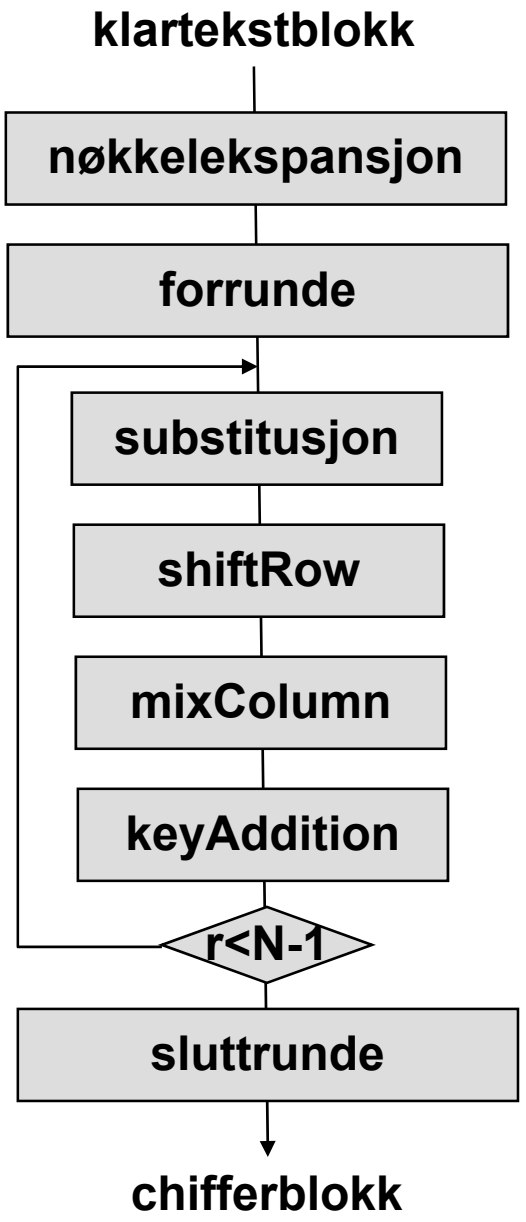
AES – virkemåte (7)





- Tilstandsmatrisen xor'es med neste del av den ekspanderte nøkkelen.

AES – virkemåte (8)



- Sluttrunden er som alle de andre, bortsett fra at skrittet mixColumn er utelatt

Bruksmåter (”modes of operation”)

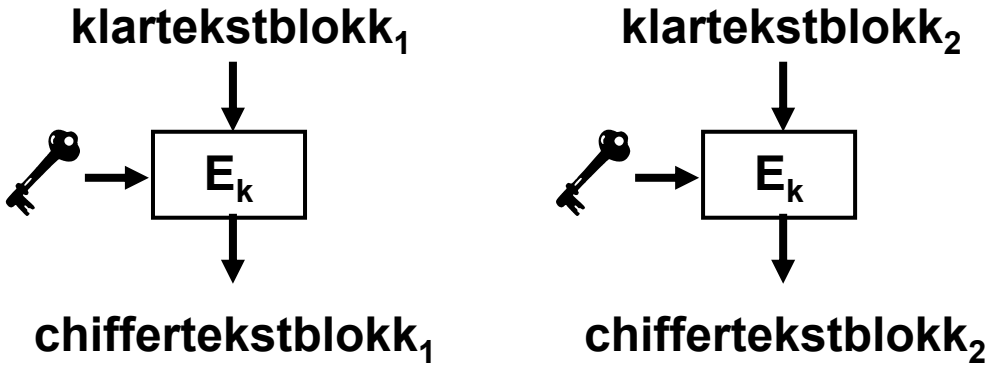
- For å kryptere bitstrømmer lengre enn blokklengden, bakes blokk-krypteringen inn i ulike bruksmåter (”modes of operation”):
  - Electronic Code Book – ECB
  - Cipher feedback – CFB
  - Cipher Block Chaining – CBC
  - Output feedback – OFB
  - Counter mode – CTR

Derfor snakker vi for eksempel om AES-CBC

- Noen av disse utfører ”diffusion” utenfor blokklengden.
- Ulike egenskaper med hensyn på sikkerhet og effektivitet.

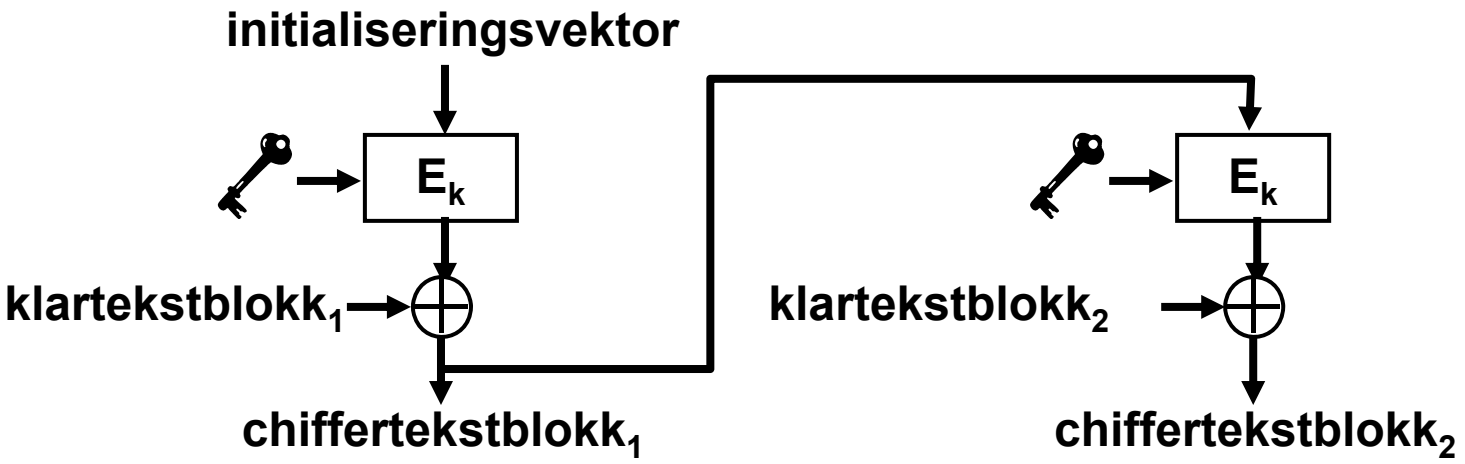
Electronic Code Book - ECB

- Hver blokk krypteres for seg.
- Ulempe: Like klartekstblokker gir like chiffertekstblokker (dårlig ”diffusion”).



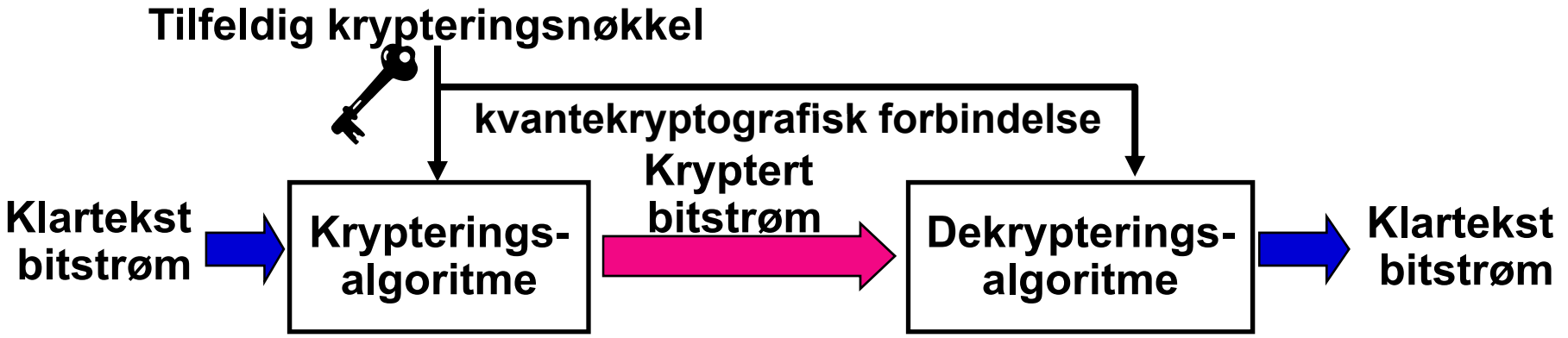
Cipher Feedback - CFB

- Foregående chiffertekstblokk krypteres med nøkkelen – første gang krypteres initialiseringsvektoren.
- Blokken xor'es med resultatet.



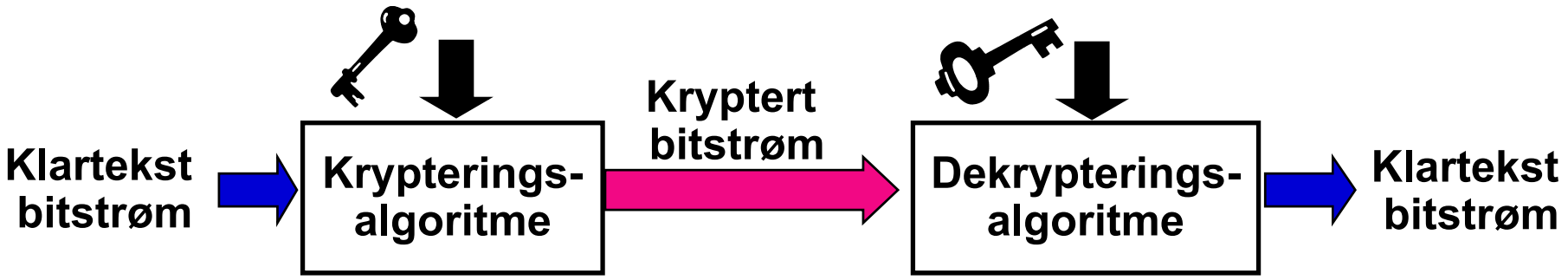
Kvantekryptografi

- Symmetrisk kryptering har et stort problem:
  Oversending av nøkler på en sikker måte
- Kvantekryptografi bygger på at polariseringsretningen for fotoner i lys kan avleses bare én gang (BB84-protokollen)
  - Derfor er det umulig å lytte ubemerket på linjen
- Fungerer bra over opptil 50 km optisk linje
- Foreløpig få kommersielle anvendelser – men mye forskning
- se f.eks. http://www.fysikknett.no/optikk/anvendelser6.php?menuid=3

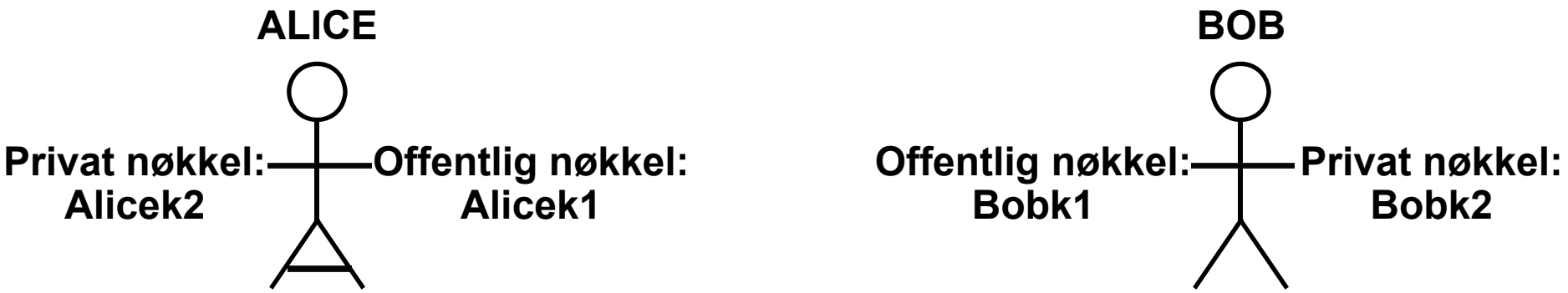


Asymmetrisk kryptering

- Asymmetrisk kryptering arbeider med et *nøkkelpar* k1 og k2 som er matematisk relatert til hverandre – den ene nøkkelen brukes for kryptering, den andre for dekryptering
- Kryptering av klartekst P med nøkkel k1 gir chiffertekst C: $E_{k1}(P) \rightarrow C$
- Dekryptering av chiffertekst C med nøkkel k2 gir klartekst P: $D_{k2}(C) \rightarrow P$



Praktisk bruk av asymmetrisk kryptering



Asymmetrisk krypteringsteknikk

- Vi ønsker at samme algoritme E brukes for både kryptering og dekryptering:
  dvs. $E_{k2}(E_{k1}(P)) \rightarrow P$
- Mest kjente algoritme: RSA (Rivest, Shamir & Adleman)
- RSA er basert på at det er lett å multiplisere to primtall, men vanskelig å faktorisere dem (bare empirisk vist!)

Multiplikasjon av to primtall er et eksempel på en matematisk enveisfunksjon:
Det er enkelt å beregne y = f(x),
og samtidig meget vanskelig å beregne $x = f^{-1}(y)$

Det er en matematisk utfordring å finne ut hvor "enveis" en gitt funksjon virkelig er!

Hvordan distribuere offentlige nøkler?

- En løsning: Digitale sertifikater (delegering av tillit)
  - Top Key Authority med kjent offentlig nøkkel (Verisign, Thawte, BT etc.)
  - Top Key Authority har som oppgave å sertifisere et underliggende nivå med Key Authorities x1, x2, …
  - . . . og så videre nedover i hierarkiet
  - En Key Authority sender deg et sertifikat i form av en melding kryptert med Key Authority private nøkkel
  - Sertifikatet bekrefter at bruker NN har offentlig nøkkel NNoff
- En mindre byråkratisk løsning: ”Web of trust”

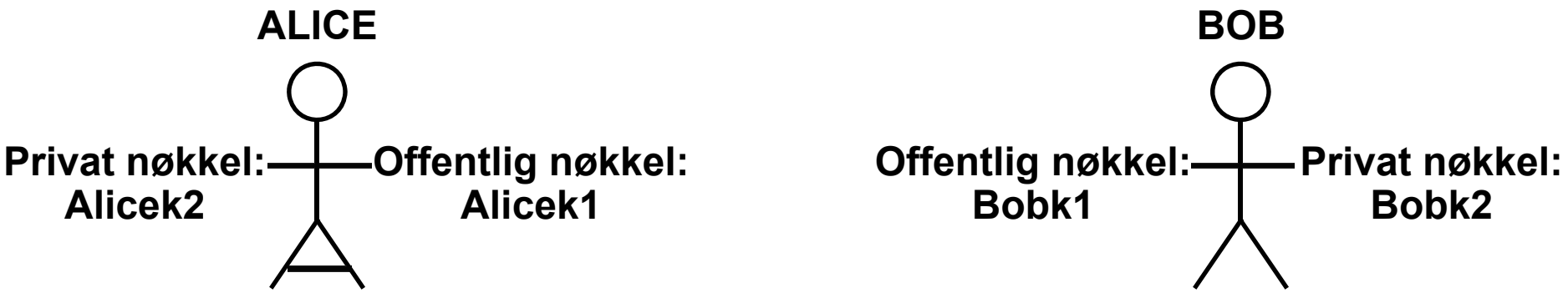
Hvordan oppdage endring av meldinger

- Kryptering forhindrer at utenforstående kan lese dataene, men forhindrer ikke at utenforstående (inklusive støy på linjen) ubemerket endrer dataene
- Mottiltak er å la en algoritme h beregne et såkalt digitalt segl eller "message digest" s ut fra meldingen P.
  - Avsender beregner s = h(P) og overfører s sammen med P
  - Mottaker beregner s = h(P) en gang til og sammenlikner med den overførte s

Beregning av digitalt segl

- En “hash-funksjon” h kan beregne en tilsynelatende mystisk bit-sekvens (dvs. et tall) fra enhver melding P.
- h(P) har alltid samme verdi for samme P.
- Ønskede egenskaper for en god hash-funksjon:
  - Alle deler av P inngår i beregningen av h(P).
  - Uansett innholdet av de aktuelle P gir h(P) en jevn fordeling over hele resultatområdet, dvs. alle verdier h(P) er like sannsynlige.
  - Beregningen er rask.

Metode for sikker meldingsformidling



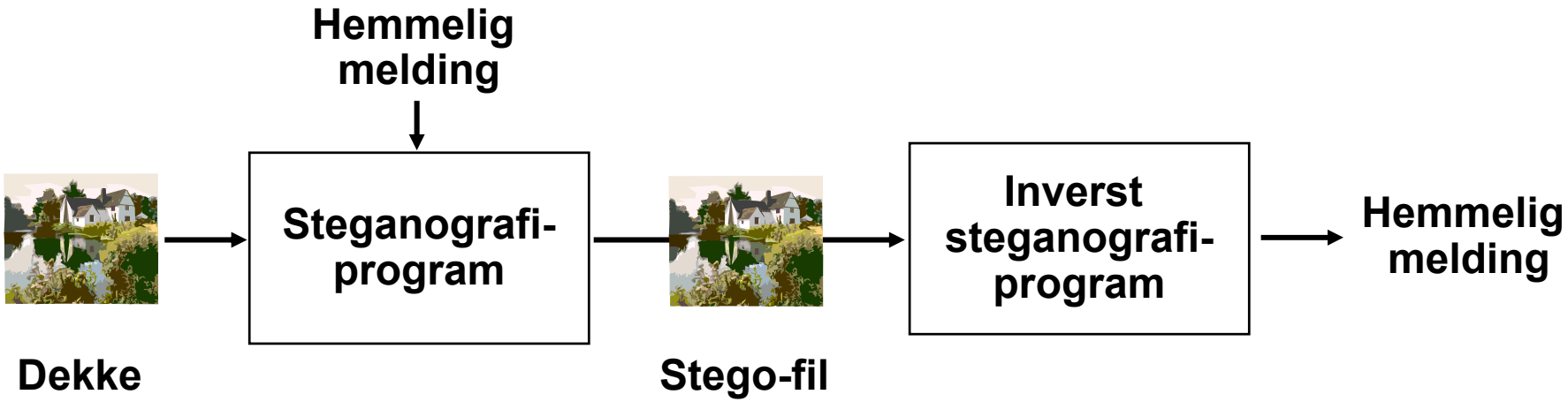
Steganografi

Mitt ønske til mor er glemt,

kan lillebror se etter katten selv?

Dekket

- Dekket er som oftest et bilde eller en lyd (mange biter!)
- Den beste måten å holde noe hemmelig er å holde det hemmelig at det er noe å holde hemmelig
  → dekket må ikke endres påfallende gjennom at noe er skjult



Et eksempel på steganografi

- Steganografi med ” Digital Invisible Ink Toolkit ”

se http://diit.sourceforge.net/index.html

Kart over modifiserte piksler

Originalbilde
7 kb jpeg

Dekke
40 kB jpeg

Stegobilde
332 kB png

Gjenvunnet bilde
7 kB jpeg

Vannmerking

- Utvide en tekst-/bilde-/lydfil med ekstra data som ikke kan la seg fjerne uten å ødelegge originalen.
- Deler av vannmerket kan være observerbart i teksten/bildet/lyden, men vanligvis er det ikke observerbart ved direkte observasjon.
- Brukes for tilleggsopplysninger, som rettighetshaver, copyrightmarkering og eksemplarnummerering.
- Et vannmerke skal ikke kunne fjernes uten å ødelegge tekst/bilde/lyd.
- Vannmerking ligner teknisk sett på steganografi, hovedforskjellen er intensjonen.

Hvordan gjemme data

- ”Trikset” er å gjemme det hemmelige bitmønsteret på bestemte steder i bitmønsteret for dekket
- Eksempel:
  - Et bilde (dekke) er representert som en sekvens av RGB-verdier, hver på 24 bit.
  - Vi skal gjemme tegnet ”A”, Unicode 41 (hex) = 0100 0001
  - Vi sørger for at den siste biten i for eksempel hver 64de RGB-verdi har verdiene 0100 0001
  - Av og til fører dette til at RGB-verdien blir endret, av og til ikke
  - En eventuell endring i intensitet i blått hist og her vil antagelig ikke være synlig

Variasjoner

- Endre i R- eller G-verdi istedenfor
- Endre i de minst signifikante bitplanene i et bilde
- Endre hvor det allerede er store endringer (bildederivasjon!)
- Endre i andre verdiområder (for eksempel frekvensdomenet)
- Manipulere fargetabellen for bildeformater med fargetabell

Stegoanalyse

- Å konstatere at et dekke inneholder et skjult budskap
  – og eventuelt gjenvinne budskapet.
- Vanskelig å gjøre ved direkte observasjon.
- Analyseprogrammer er mye mer effektive
  – er det noe uvanlig i teksten/bildefilen/lydfilen?
- Tilgang til dekket letter stegoanalysen betraktelig
  – bruk ikke kjente digitale bilder som ligger på Internett som dekke!

Oppsummering

- Data kan gjøres uleselige ved hjelp av kryptering (konfidensialitet).
- Urettmessig endring av data kan oppdages med digitale segl (integritet).
- Avsender kan verifiseres ved hjelp av asymmetrisk kryptering (autentisitet/ikke-benektbarhet).
- Kobling mellom individer og nøkler gjøres med sertifikater.
- Ved kryptering brukes oftest en kombinasjon av symmetriske og asymmetriske krypteringsteknikker.
- Vurderinger av sikkerheten mot ”knekking” av krypteringer er kun basert på antagelser og empiri, intet er bevist.
- Steganografi brukes for å skjule en melding i et dekke.
- Vannmerker brukes for å gi tilleggsopplysninger som ikke kan fjernes uten å ødelegge dekket.

Planen videre

- Vi er nå ferdige med de ordinære forelesningene.
- De to neste ukene blir det repetisjon.
- HUSK: Kapittel 20 – Digital representasjon av identitet er pensum selv om det ikke er forelest!