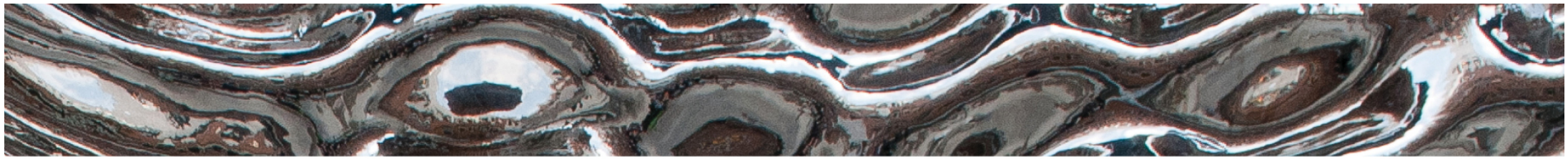


UiO : Institutt for informatikk
Det matematisk-naturvitenskapelige fakultet

IN1020 - Introduksjon til datateknologi

Kryptering til hverdag og fest

21.09.2018

Håkon Kvale Stensland & Andreas Petlund

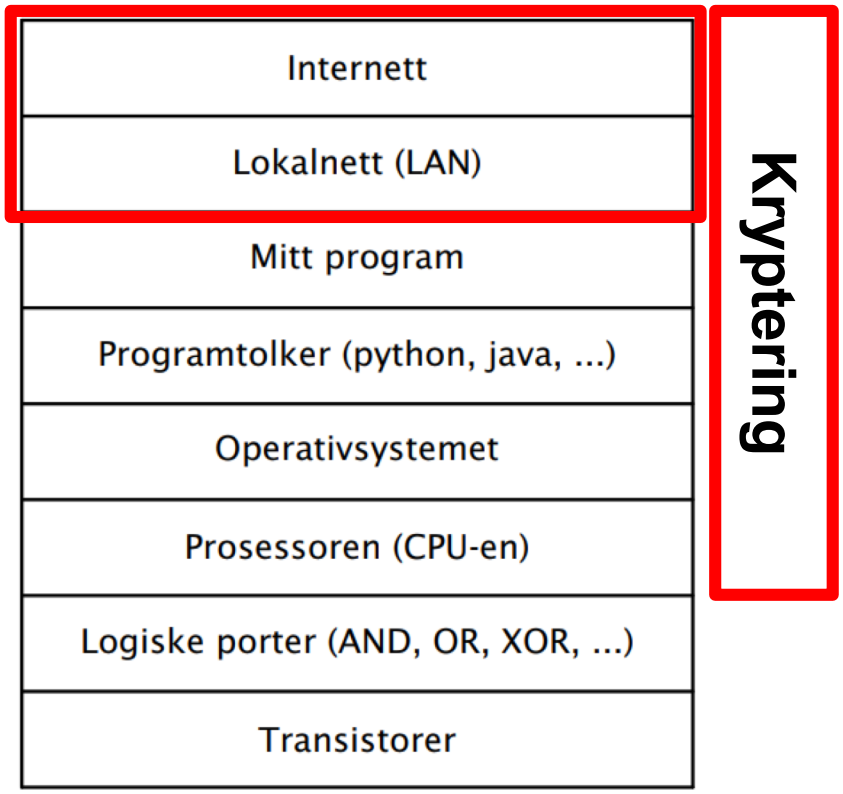
simula

UiO : Institutt for informatikk
Det matematisk-naturvitenskapelige fakultet

Introduksjon til kryptografi

- Hvorfor trenger vi kryptografi?
- Hva er kryptografi?
- Forskjellige typer kryptografiske funksjoner.
- Hvordan ”knekke” en kryptering.
- Hemmelig nøkkel kryptering.
- Offentlig nøkkel kryptering.
- Hash-algoritmer.

Hvor i "stakken" er vi nå?



Kryptografi

κρυπτο γραφη

Kunsten å skrive hemmelig.

Kunsten å "scramble" informasjon til noe ugjenkjennelig, og
Samtidig muliggjøre at rett person med en hemmelig metode kan lese dette

Fundamental Tenet of Cryptography

If lots of smart people have failed to solve a problem,
then it probably won't be solved (soon).

Viktige definisjoner

- Beskjeder:
  - Plaintext / Klar tekst
  - Ciphertext / Krypter tekst
- Ingredienser:
  - Algoritmer
  - Nøkler
- Aktører:
  - *Kryptograf*: Finner opp smarte algoritmer
  - *Kryptoanalytikker*: “Knekker” smarte algoritmer

Kryptografi

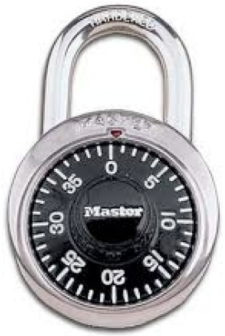
- Algoritmer
  - Nøkler = Hemmelige verdier
- Helt greit at algoritmene er offentlige
  - Ikke greit å bruke svake algoritmer.
  - Offentlighet bra for å luke ut svakheter.
  - Algoritme uten nøkkel hjelper ikke til å dekode informasjon.

Offentligjøre algoritmen?

- Offentlig eller hemmellig algoritme?
  - Gode algoritmer kan være offentlige
  - *"Sunlight is the best disinfectant"*
  - Reverse engineering, etc.
  - "Gratis analyse" av kryptoanalytikere
  - Eksempler: Advanced Encryption Standard (AES)
- Generelt:
  - Kommersielle: Offentlige
  - Militære: Hemmelige

Kompleksitet vs. ”Brute-force”

- Algoritmen bør være effektiv å bruke
- Sikkerheten er avhengig av hvor komplekst koder er å knekke
- Eksempel: Kombinasjonslås
  - 3 tall, mellom 1 og 40 – 10 sekunder per forsøk
  - 4 tall mellom 1 og 40 – 13 sekunder per forsøk



Kompleksitet

- Kombinasjonslås
  - 3 tall, mellom 1 og 40 – 10 sekunder per forsøk
    - Antall kombinasjoner: $40^3 = 64.000$
    - 178 timer
  - 4 tall, 13 sekunder per forsøk
    - $40^4 = 2,560.000$
    - 9244 timer

Brute-force av kombinasjonslås med utnyttelse av svakheter:
https://youtu.be/09UgmwtL12c

Hemmelige koder

- Cæsarchiffer
  - Skifte bokstaver 3 ganger fremover
    - DOZEN → GRCHQ
- Dekoderringer
  - Substituer bokstaver *n* bokstaver videre (*n* = 1..25)
    - HAL → IBM (*n* = 1)
- Monoalfabetisk chiffer
  - Tilfeldig mapping (26! = $4.03291461 \times 10^{26}$)
    - 1 ms / forsøk → 10M år… *MEN*, bokstavfrekvenser...

Frekvensanalyse

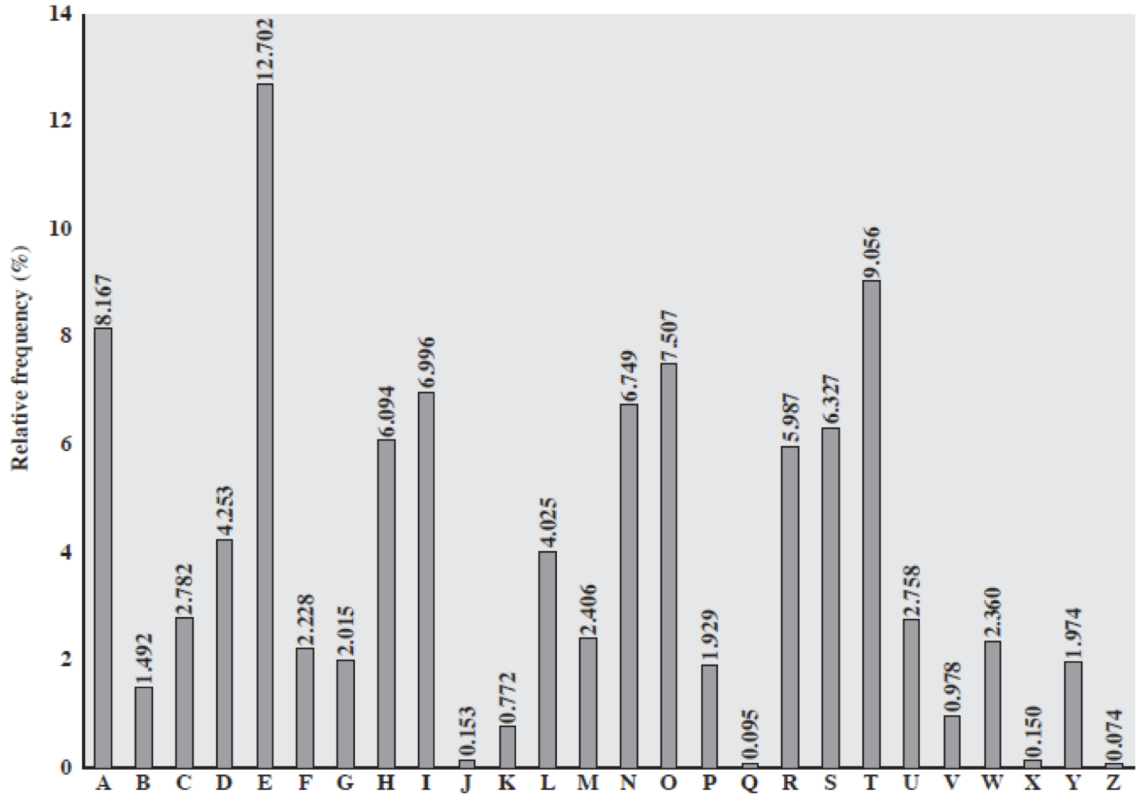
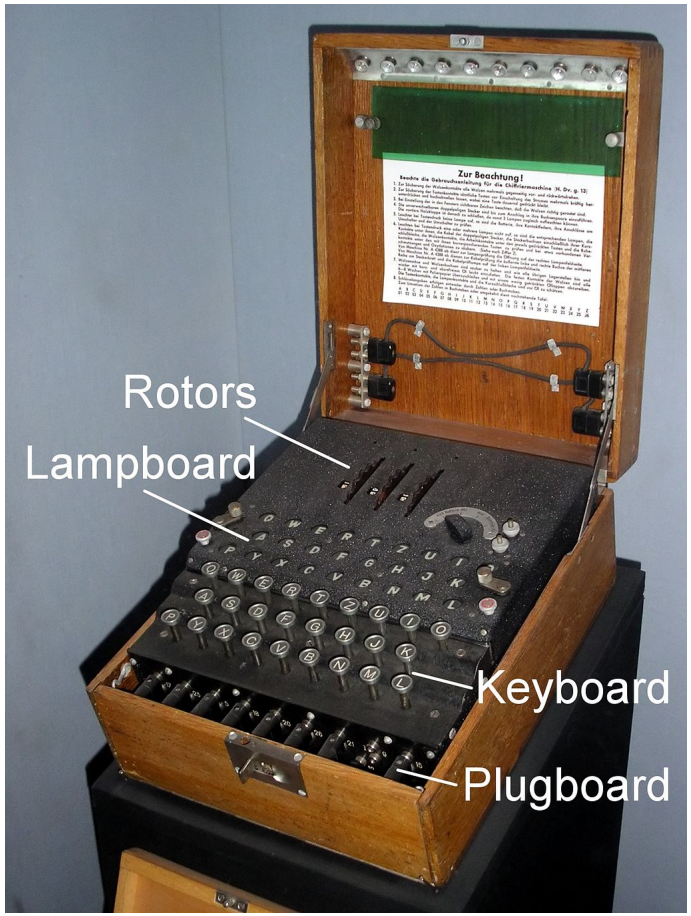


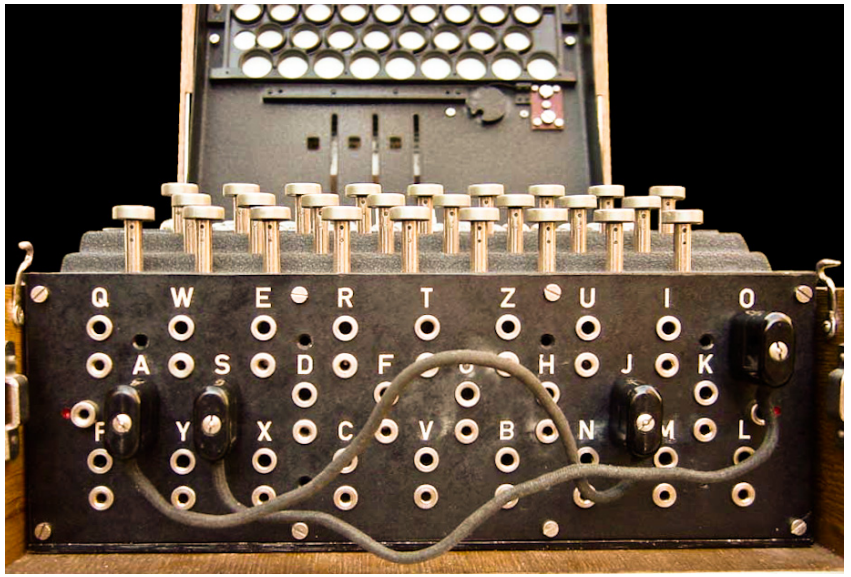
Figure 2.5 Relative Frequency of Letters in English Text

[Fra *Stallings, Cryptography & Network Security*]

Enigma



Enigma er basert på en polyalfabetisk chiffer



Images from Wikipedia: https://en.wikipedia.org/wiki/Enigma_machine

Hvordan ”knekke” en kryptering

- Kun Ciphertext / kryptert tekst
  - Prøve alle forskjellige nøkler (brute-force)
  - Trenger en lang nok chipertext
  - Implementasjonssvakheter
- Kjent plaintext / klar tekst
  - (plaintext, chipertext) par
- Deler av plaintext / klar tekst
  - Har chipertext, sammenligne med forventede verdier
  - *Brukt mot den Japanske marinen i WW2*
- Finne svakheter i krypteringsalgoritme

16. Oktober 2017

Sikkerhet i trådløse nettverk?

- Svakhet i protokoller og algoritmer
  - Key Reinstallation Attacks (KRACK)
    - Svakheter i ”handshake” mellom enheter
  - Svakheter i implementasjoner

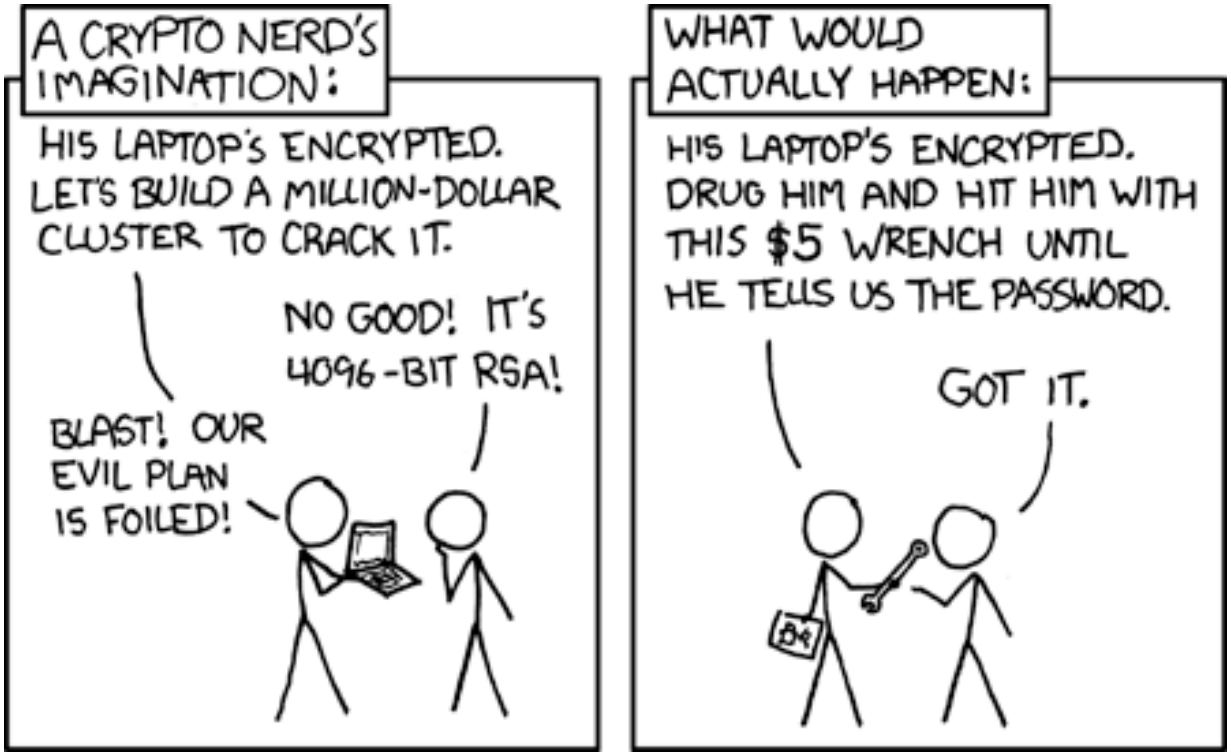
KRACK: Svakheter i WPA:
https://www.krackattacks.com/

- ”Brute-force”
  - Teste alle mulige kombinasjoner
  - Moderne GPUer kan teste over 1 million passord i sekundet
    - 8 siffer = 28 timer
    - 6 bokstaver = 4 dager
  - Ordbokangrep

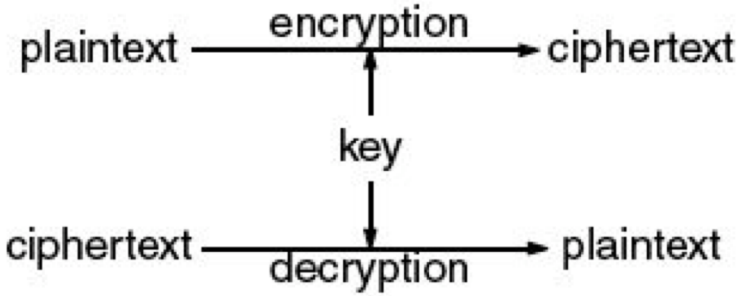
Password Calculator:
http://lastbit.com/pswcalc.asp

Forskjellige typer kryptering

- Symmetrisk kryptering
  - En nøkkel: Hemmelig (men ofte delt mellom sender og mottaker)
- Asymmetrisk kryptering
  - Offentlig nøkkelkryptering
  - To nøkler:
    - Privat
    - Offentlig
- Hash-algoritmer
  - Enveis identifikasjon
  - Ingen nøkkel



Symmetrisk kryptering



Også kjent som “vanlig kryptering”

- To typer:
  - Block cipher (DES, 3DES, AES)
  - Stream cipher (RC4)
- Per dags dato er «block cipher» mest brukt, og gjerne i blokker på 128-bits (AES)

Bruk av symmetrisk kryptering

- Overførsel over usikker kanal
  - Delt hemmelighet (sender, mottaker)
- Sikker lagring på utrygt media
- Autentifisering
  - *Sterk autentifikasjon*: bevise kunnskap uten å avsløre nøkkel

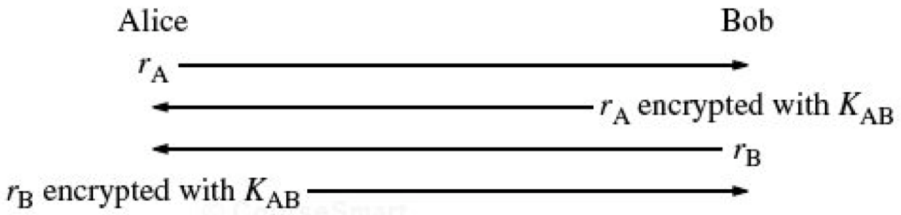
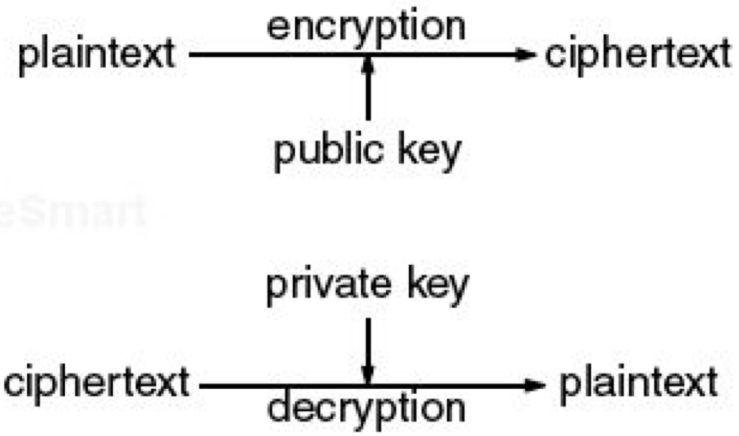


Figure 2-1. Challenge–response authentication with shared secret

Asymmetrisk kryptering



Også kjent som «offentlig nøkkelkryptering»

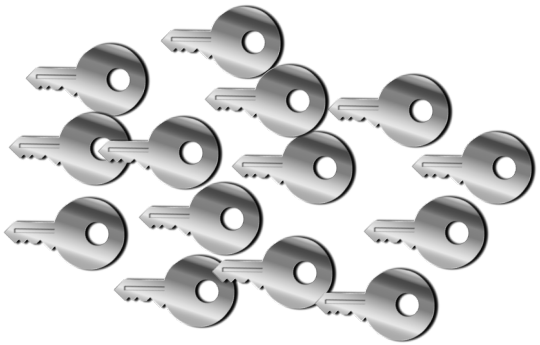
Offentlig nøkkel: publisert, gjerne åpnet på nettet for alle
Privat nøkkel: hemmelig og ikke offentlig,

Asymmetrisk kryptering

Privat nøkkel

Offentlig nøkkel

Den offentlige nøkkelen kopieres og gjøres lett tilgjengelig for dine kontakter



Asymmetrisk kryptering - kryptere

Dag ønsker å sende en melding til Håkon

Klartekst

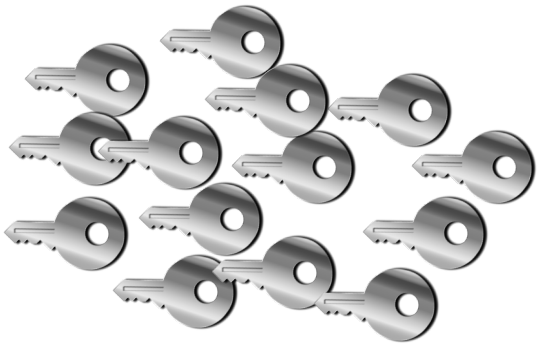
Hemmelig melding til Håkon

Kryptere meldingen med Håkon sin offentlige nøkkel

Chiffertekst

27fdbe64 8588b3d1
21ed8e8e ed01c340
982c50a4 8d87c7b2

Håkon sin offentlige nøkkel



Asymmetrisk kryptering - dekryptere

Håkon mottar chiffertekst fra Dag

Chiffertekst

27fdbe64 8588b3d1
21ed8e8e ed01c340
982c50a4 8d87c7b2

Håkon sin private nøkkel

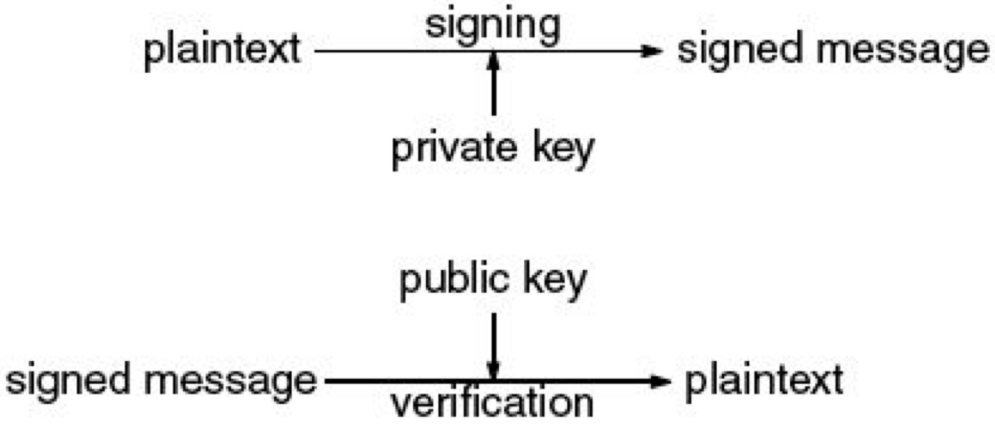
Klartekst

Hemmelig melding til Håkon

Utfordringer med asymmetrisk kryptering

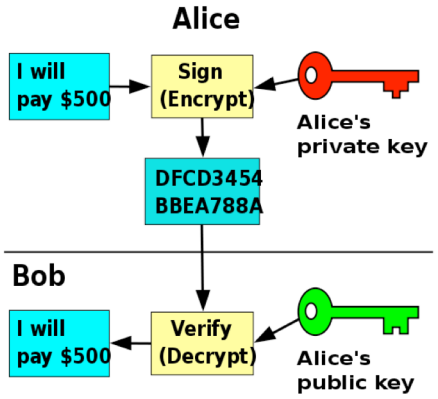
- Effektivitet
  - Asymmetriske krypteringsalgoritmer er mange ganger tregere enn symmetriske algoritmer.
- Hybrid modell
  - Offentlig nøkkel bruker for å bli enige om en midlertidig nøkkel.
    - Den vanligste teknikken er kjent som Diffie-Hellman nøkkelutveksling
  - Symmetrisk kryptering brukes under resten av kommunikasjonen.

Digital signatur

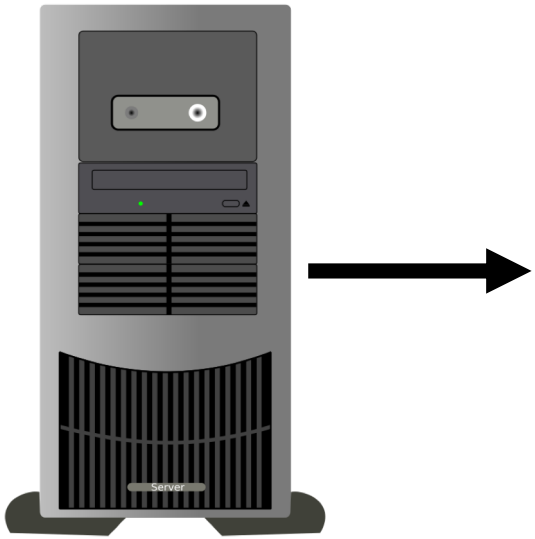


Asymmetri:
Signatur kan kun bli generert av eier eller noen som kjenner privat nøkkel.
Signatur kan bli *verifisert* av alle som har tilgang til offentlig nøkkel.

Ikke-benektelse:
Sender kan ikke bevise at beskjed (signatur) ikke ble sendt.
Nøkkel kan ha blitt stjålet / hacket.



Viktigheten av gode tilfeldige tall



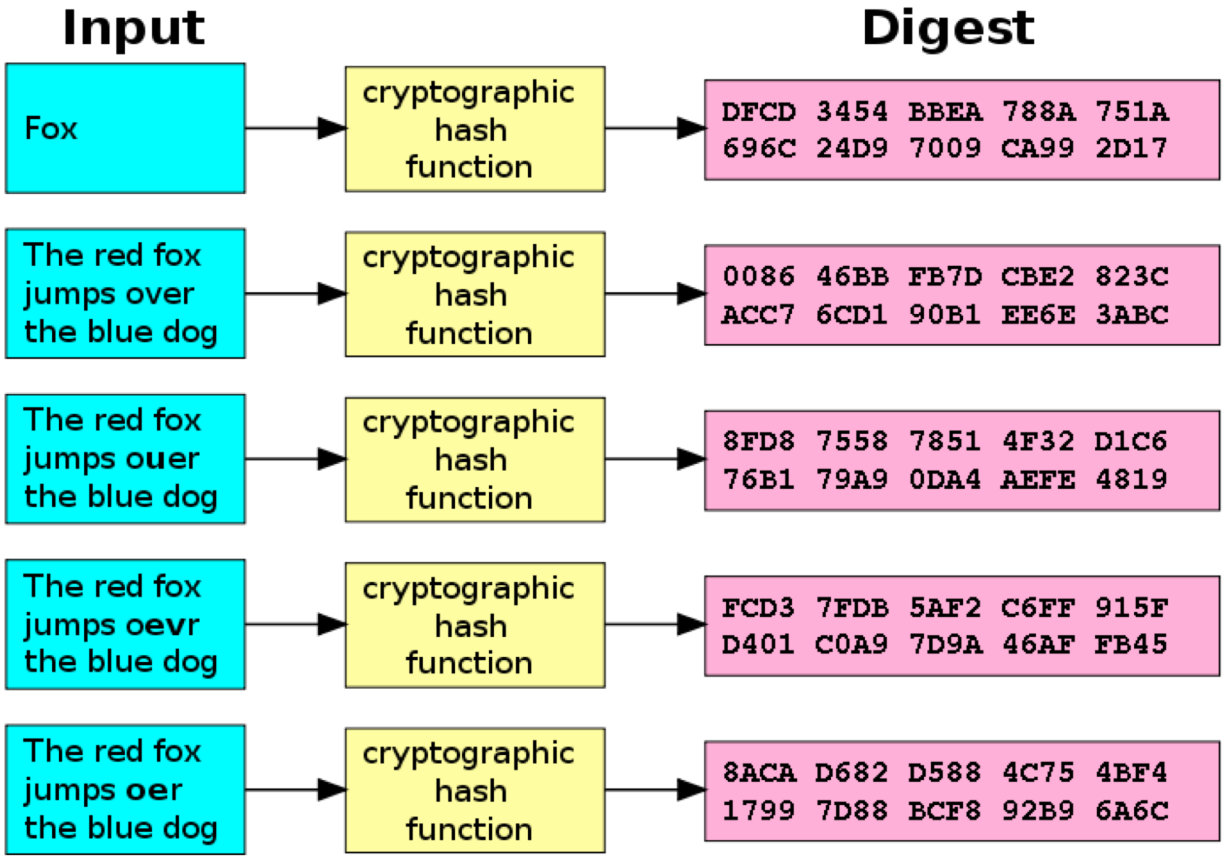

```
apetlund$ cat /dev/random | od -N 128 -H
0000000       4744bd30      f67d7c94      e4f7c2e0      9c434763
0000020       18a4cf29      564734ea      4891a4f1      7a683149
0000040       b8a2426f      7e2fa3cf      b0cc23ad      493f5132
0000060       634a1d63      d76a21fd      b840d3c1      ec8428f0
0000100       5fe82d93      4f70fb18      0e7ef382      219627e0
0000120       3b5b472a      b2d9d23d      5e2b09ce      f18f496e
0000140       918c5496      8c839333      fceabdd6      f4e444b8
0000160       d46308c1      5656abf3      cfe995b4      aaf9bc7c
0000200
```

Tilfeldig tallgenerator: NSA bakdør:
https://en.wikipedia.org/wiki/Dual_EC_DRBG

Kryptografiske enveisfunksjoner (hash)

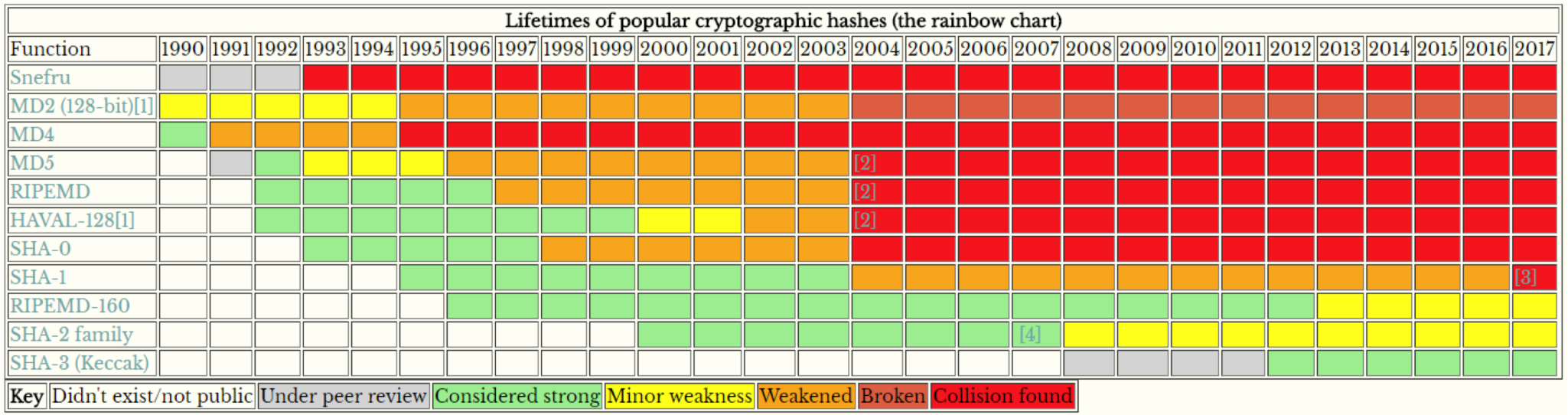
- Sjekke dataintegritet / ikke-reverserbar
- Kriterier for en god sjekksumalgoritme:
  - Enkelt å regne ut sjekksum for en gitt beskjed.
  - Ikke mulig å finne en beskjed for en gitt sjekksum.
  - Ikke mulig å endre en beskjed uten at sjekksummen blir endret.
  - Ikke mulig å finne to forskjellige beskjeder med same sjekksum

Sjekksumalgoritmer



http://en.wikipedia.org/wiki/Cryptographic_hash_function

Kryptografisk Sjekksumlivssykel



Lifetimes of popular cryptographic hashes (the rainbow chart)

Function	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Snefru																												
MD2 (128-bit)[1]																												
MD4																												
MD5															[2]													
RIPEMD															[2]													
HAVAL-128[1]															[2]													
SHA-0																												
SHA-1																												[3]
RIPEMD-160																												
SHA-2 family																		[4]										
SHA-3 (Keccak)																												

Key: Didn't exist/not public | Under peer review | Considered strong | Minor weakness | Weakened | Broken | Collision found

Status på Hash-algoritmer:
http://valerieaurora.org/hash.html

Ekstramateriale:

- *Bøker og artikler:*
  - W. Stallings. Cryptography and Network Security: Principles and Practice (6th Edition), 2013, Pearson
- *Sikkerhetskurs på IFI & UNIK:*
  - INF3510 – Informasjonssikkerhet
  - UNIK4250 – Sikkerhet i distribuerte systemer
  - UNIK4270 – Sikkerhet i operativsystemer og programvare

Takk for oss!


DILBERT By Scott Adams
TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
www.dilbert.com scottadams@aol.com
NINE NINE NINE NINE NINE NINE
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.
10/25/01 © 2001 United Feature Syndicate, Inc.