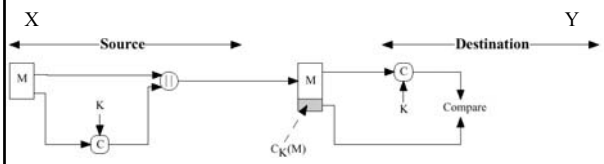
Forelesning 7

Digitale signaturer
og
autentisering

Hvorfor digitale signaturer?

- Eksempel: Xavier og Yuliang utveksler meldinger beskyttet av MAC med symmetrisk nøkkel



Hvorfor (forts.)

- Yuliang kan lage en ny melding med "gunstig" innhold, og hevde at den kom fra Xavier, siden Yuliang også har nøkkelen Xavier bruker til MAC
- Xavier kan nekte for at han har sendt en gitt melding, siden Yuliang er i stand til å "forfalske" meldingens MAC

Egenskaper til digitale signaturer

- Signaturen må kunne verifisere avsenderen av en melding og tidspunktet den ble sendt på
- Signaturen må kunne verifisere innholdet av meldingen på signerings-tidspunktet
- Signaturen må kunne verifiseres av en tredjepart for å avgjøre tvister

Krav til Digitale Signaturer

- Signaturen må være et bitmønster som avhenger av meldingen som signeres
- Signaturen må benytte noe informasjon som bare senderen har, for å unngå både forfalskning og fornektelse
- Det må være relativt enkelt å produsere signaturen
- Det må være relativt enkelt å gjenkjenne og verifisere signaturen

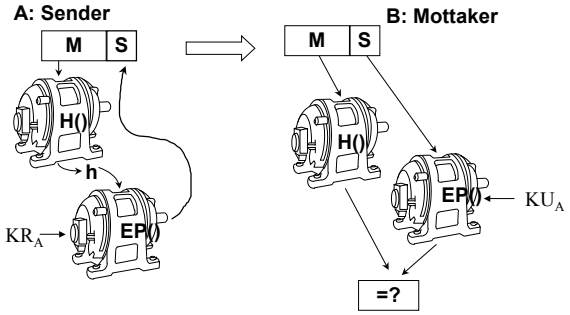
Krav forts.

- Det må være "umulig" å forfalske en digital signatur, enten ved å lage en ny melding for en eksisterende signatur, eller ved å lage en falsk signatur for en gitt melding
- Det må være praktisk mulig å beholde en kopi av den digitale signaturen på et lagringsmedium.

... og derfor bruker vi:

- En sikker (kryptografisk) hash-funksjon, der hashen blir kryptert med avsenderens private nøkkel, tilfredsstiller disse kravene!
- Informasjon som avsender, tidspunkt etc. må inngå i meldingen som det beregnes hash på

RSA Digitale signaturer eksempel



Direkte signaturer

- Involverer bare de to kommuniserende partene (avsender, mottaker)
- Forutsetter at mottakeren har den offentlige nøkkelen til avsenderen
- Svakhet: Avhenger av sikkerheten til avsenders hemmelige nøkkel (KR_a)

"Meklede" signaturer

- Enhver melding fra avsender X til mottaker Y går innom mekleren A
- A foretar diverse tester for å verifisere meldingens avsender og signatur
- Meldingen dateres, og sendes til Y med en indikasjon på at den er verifisert av A
- X kan nå ikke nekte for å ha sendt meldingen

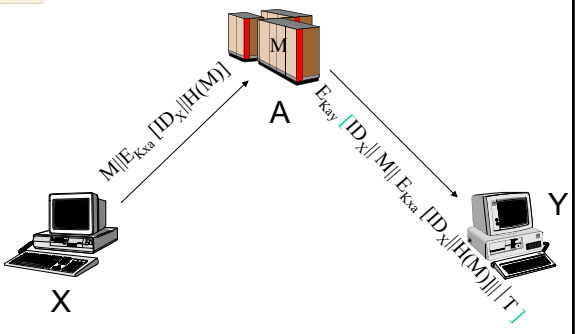
Eksempler - "Mekling"

a) Symmetrisk kryptering – Mekleren har tilgang til M

1) $X \rightarrow A$: $M||E_{Kxa}[ID_X||H(M)]$

2) $A \rightarrow Y$: $E_{Kay}[ID_X||M||E_{Kxa}[ID_X||H(M)]||T]$

Mekling forts.



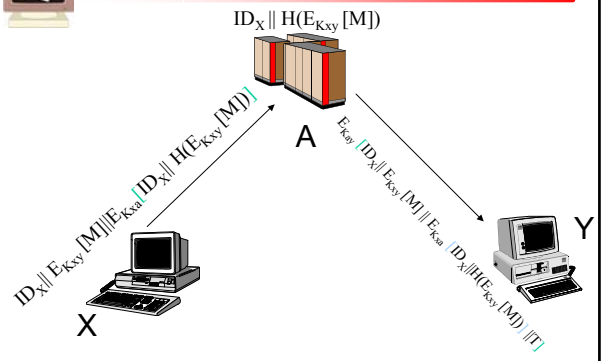
Eksempler forts.

b) Symmetrisk kryptering – Mekleren har ikke tilgang til M

1) $X \rightarrow A$: $ID_X || E_{Kxy}[M] || E_{Kxa}[ID_X || H(E_{Kxy}[M])]$

2) $A \rightarrow Y$: $E_{Kay}[ID_X || E_{Kxy}[M] || E_{Kxa}[ID_X || H(E_{Kxy}[M])] || T]$

Mekling forts.



Eksempler forts. forts.

c) Offentlig-nøkkel kryptering – Mekleren har ikke tilgang til M

1) $X \rightarrow A$: $ID_X || E_{KRx}[ID_X || E_{KUy}[E_{KRx}[M]]]$

2) $A \rightarrow Y$: $E_{KRa}[ID_X || E_{KUy}[E_{KRx}[M]] || T]$

Hva gjør mekleren?

- Bemerk at slik de foregående meklede mekanismene er beskrevet, gir de *ikke* bedre sikkerhet enn direkte signaturer (hva skjer hvis Xaviers nøkkel til mekleren blir stjålet?)
- Det må ergo i tillegg være et system som håndterer sjekk av om brukernes nøkler er gyldige, samt tvungen varsling av kompromitterte nøkler, etc. (Timestamp sentralt!)

Autentiseringsprotokoller

- Gjensidig Autentisering
  - Ekstern pålogging
  - Nedlasting av epost
  - Bruk av tjenester
- Enveis Autentisering
  - Mest brukt for epost etc.

Gjensidig autentisering

- Hvordan vet serveren at den snakker med den rette klienten, samtidig som at klienten vet at den snakker med den rette serveren?

Replay-angrep

- Enkel replay
- Gjentakelse som kan logges
  - Replay innen gyldig tidsrom (”vindu”)
- Gjentakelse som ikke kan detekteres
  - Opprinnelig melding blokkeres
- Retur uten modifikasjon
  - Ved symmetrisk kryptering

Mottiltak

- Timestamps
  - Synkroniserte klokker
- Challenge/Response
  - Nonce

Nøkkeldistribusjon & autentisering

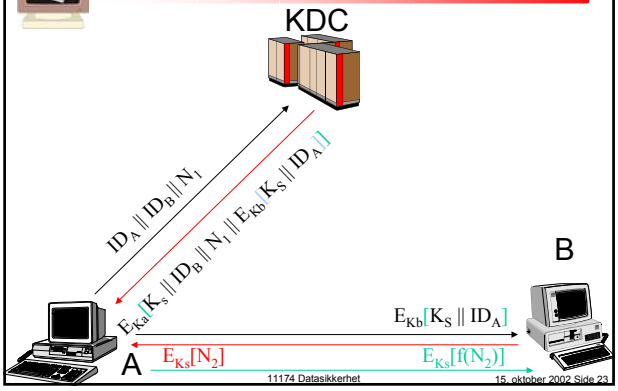
- Symmetrisk kryptering
  - Needham & Schroeder
  - Denning
  - Neuman
- Offentlig-nøkkel kryptering
  - Denning m/timestamp
  - Woo & Lam

Needham & Schroeder

1. $A \rightarrow KDC: ID_A \,||\, ID_B \,||\, N_1$
2. $KDC \rightarrow A: E_{Ka}[K_s \,||\, ID_B \,||\, N_1 \,||\, E_{Kb}[K_S \,||\, ID_A]]$
3. $A \rightarrow B: E_{Kb}[K_S \,||\, ID_A]$
4. $B \rightarrow A: E_{Ks}[N_2]$
5. $A \rightarrow B: E_{Ks}[f(N_2)]$

☞Replay av 3. mulig hvis tilgang til gammel K_S, og hvis 4. kan stoppes

Needham & Schroeder



Variant fra Denning

1. $A \rightarrow KDC: ID_A \,||\, ID_B$
2. $KDC \rightarrow A: E_{Ka}[K_s \,||\, ID_B \,||\, T \,|\, E_{Kb}[K_S \,||\, ID_A \,||\, T]]$
3. $A \rightarrow B: E_{Kb}[K_S \,||\, ID_A \,||\, T]$
4. $B \rightarrow A: E_{Ks}[N_1]$
5. $A \rightarrow B: E_{Ks}[f(N_1)]$

Timestamp skal sikre ”timelyness”

MEN: Avhengig av synkroniserte klokker

Neuman

Neuman's forslag mot supress/replay:

1. A → B: $ID_A \| N_a$
2. B → KDC: $ID_B \| N_b \| E_{Kb}[ID_A \| N_a \| T_b]$
3. KDC → A: $E_{Ka}[ID_B \| N_a \| K_S \| T_b] \| E_{Kb}[ID_A \| K_S \| T_b] \| N_b$
4. A → B: $E_{Kb}[ID_A \| K_S \| T_b] \| E_{Ks}[N_b]$

T_X - suggested expiration time (relativt til x' klokke)

Neuman forts.

- Kan opprette ny sesjon uten å gå innom KDC (forutsatt at tiden ikke er ute):

1. A → B: $E_{Kb}[ID_A \| K_S \| T_b] \| N_a'$
2. B → A: $N_b' \| E_{Ks}[N_a']$
3. A → B: $E_{Ks}[N_b']$

Offentlig-nøkkel-løsninger

- Denning
- Woo & Lam

Denning (m/timestamps)

1. A → AS: $ID_A \| ID_B$
2. AS → A: $E_{KRas}[ID_A \| KU_A \| T] \| E_{KRas}[ID_B \| KU_B \| T]$
3. A → B: $E_{KRas}[ID_A \| KU_A \| T] \| E_{KRas}[ID_B \| KU_B \| T] \| E_{KUb}[E_{KRa}[K_S \| T]]$

- AS leverer offentlige nøkler
- A velger sesjonsnøkkel

Woo&Lam (nonce-basert)

1. A → KDC: $ID_A \| ID_B$
2. KDC → A: $E_{KRauth}[ID_B \| KU_b]$
3. A → B: $E_{KUb}[N_a \| ID_A]$
4. B → KDC: $ID_B \| ID_A \| E_{KUauth}[N_a]$
5. KDC → B: $E_{KRauth}[ID_A \| K_{KUa}] \| E_{Kub}[E_{KRauth}[N_a \| K_S \| ID_B]]$
6. B → A: $E_{KUa}[E_{KRauth}[N_a \| K_S \| ID_B] \| N_b]$
7. A → B: $E_{Ks}[N_b]$

Revidert Woo&Lam

1. A → KDC: $ID_A \| ID_B$
2. KDC → A: $E_{KRauth}[ID_B \| KU_b]$
3. A → B: $E_{KUb}[N_a \| ID_A]$
4. B → KDC: $ID_B \| ID_A \| E_{KUauth}[N_a]$
5. KDC → B: $E_{KRauth}[ID_A \| K_{KUa}] \| E_{Kub}[E_{KRauth}[N_a \| K_S \| ID_A \| ID_B]]$
6. B → A: $E_{KUa}[E_{KRauth}[N_a \| K_S \| ID_A \| ID_B] \| N_b]$
7. A → B: $E_{Ks}[N_b]$

Enveisautentisering

Symmetrisk kryptering

1. A → KDC: $ID_A \| ID_B \| N_1$
2. KDC → A: $E_{Ka}[K_s \| ID_B \| N_1 \| E_{Kb}[K_s \| ID_A]]$
3. A → B: $E_{Kb}[K_S \| ID_A] \| E_{Ks}[M]$

Beskytter ikke mot replay!

Enveisautentisering, forts.

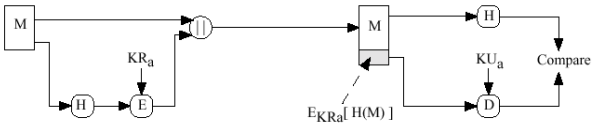
Offentlig-nøkkel kryptering

- Konfidensialitet:
  A → B: $E_{KUb}[K_s] \| E_{Ks}[M]$
- Integritet:
  A → B: $M \| E_{KRa}[h(M)]$
- Konf. & Int.:
  A → B: $E_{KUb}[M \| E_{KRa}[h(M)]]$
- Sertifikat:
  A → B: $M \| E_{KRa}[h(M)] \| E_{KRas}[T \| ID_A \| KU_a]$

Digital Signature Standard

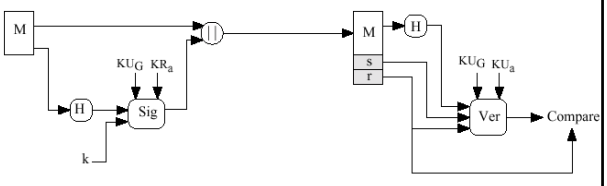
- DSS kan ikke brukes til kryptering eller nøkkelutveksling, men er likevel en offentlig-nøkkel-teknikk (avledet av ElGamal)
- DSS tar som input hash av melding, en hemmelig nøkkel KR_a, en "global" offentlig nøkkel KU_G og et tilfeldig tall k
- DSS genererer en signatur i 2 deler; s og r

RSA digitale signaturer



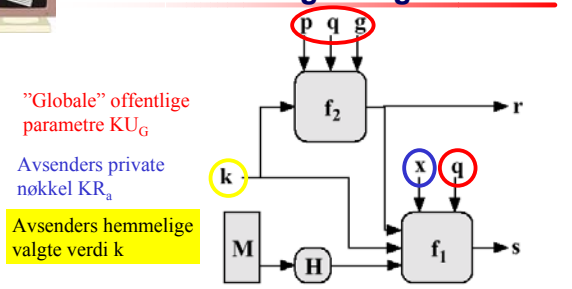
Her er verifisering det samme som dekryptering!
(Pluss nogo attåt)

DSS digitale signaturer



$KU_G = \{p,q,g\}$ "Globale" offentlige parametre
$KR_a = \{x\}$ Avsenders private nøkkel
$KU_a = \{y\}$ $(=g^x \bmod p)$ -"- offentlige nøkkel

DSS signering

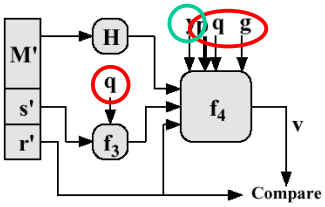


$s = f_1(H(M),k,x,r,q) = (k^{-1}(H(M)+xr)) \bmod q$
$r = f_2(k,p,q,g) = (g^k \bmod p) \bmod q$

DSS verifisering

”Globale” offentlige parametre KU_G

Avsenders offentlige nøkkel KU_a



$w = f_3(s',q) = (s')^{-1} \bmod q$

$v = f_4(y,p,q,g,H(M'),w,r')$

$= ((g^{(H(M')w) \bmod q} y^{r'w \bmod q} \bmod p) \bmod q$