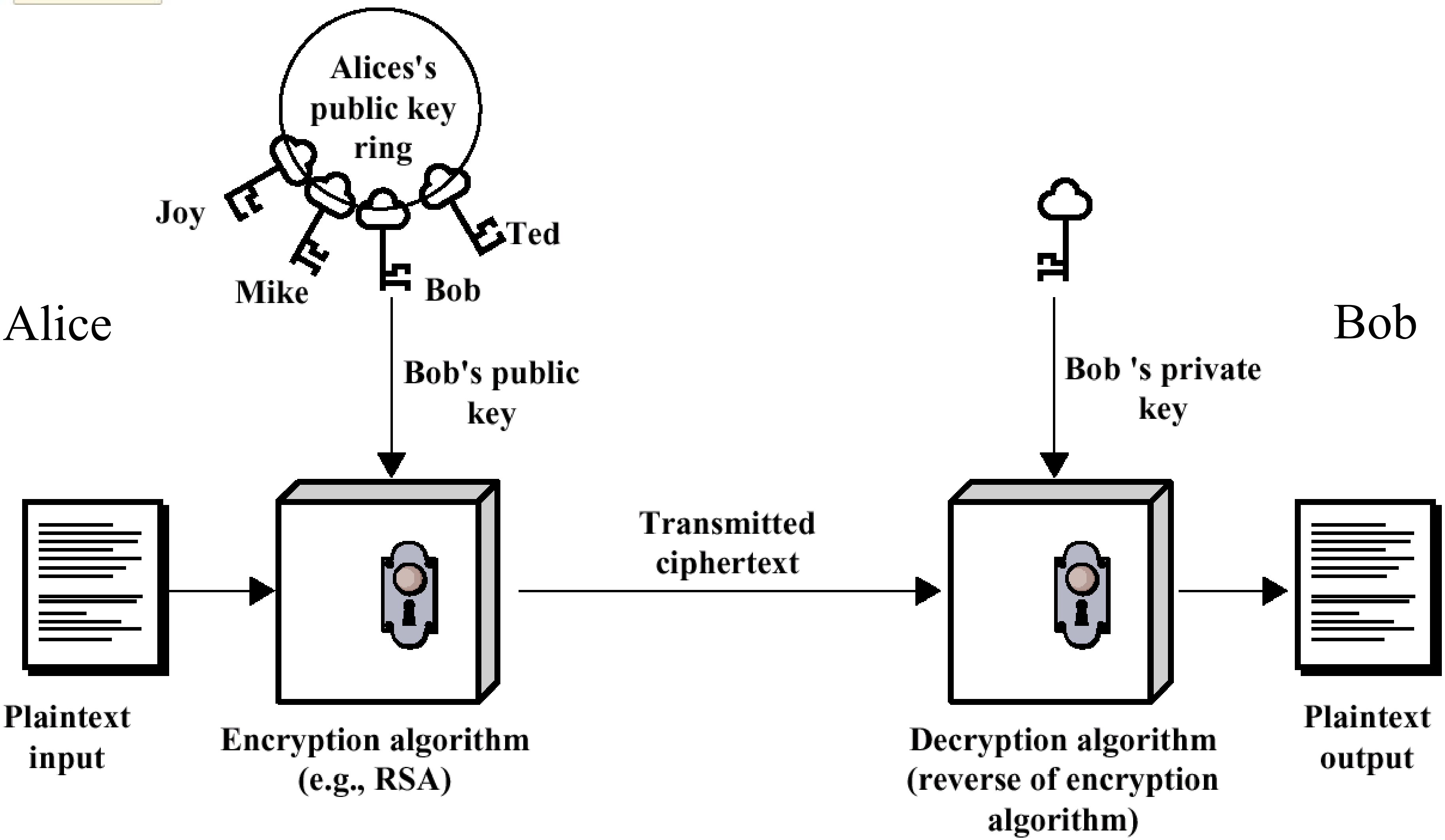
Forelesning 5

Offentlig-nøkkel-algoritmer

Finn-Erik Vinjes mareritt

- Offentlig-nøkkel-kryptografi?
- Offentlig nøkkel-kryptografi?
- Offentlig-nøkkel kryptografi?
- Offentlignøkkelkryptografi?
- ”Kryptografi hvor man bruker offentlige og private nøkler”?

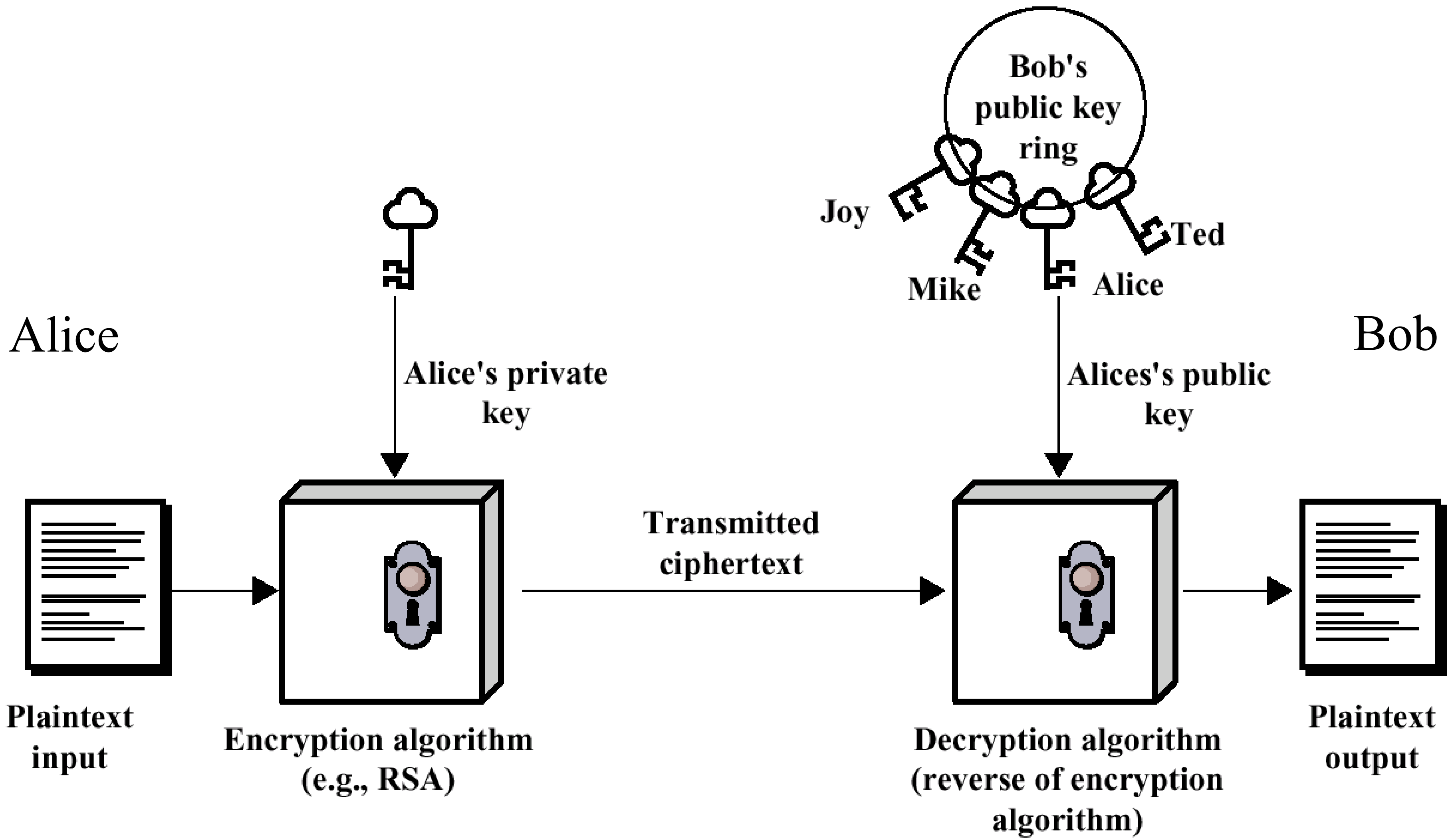
RSA Kryptering



Hvem har hvilke nøkler?

- I f.eks. RSA vil alle aktører ha et *nøkkelpar* bestående av en privat og en offentlig nøkkel
- Alice krypterer meldingen med Bobs offentlige nøkkel
- Bob bruker sin private nøkkel til å dekryptere meldingen
- Hvis Bob skal svare, må han kryptere med Alices offentlige nøkkel

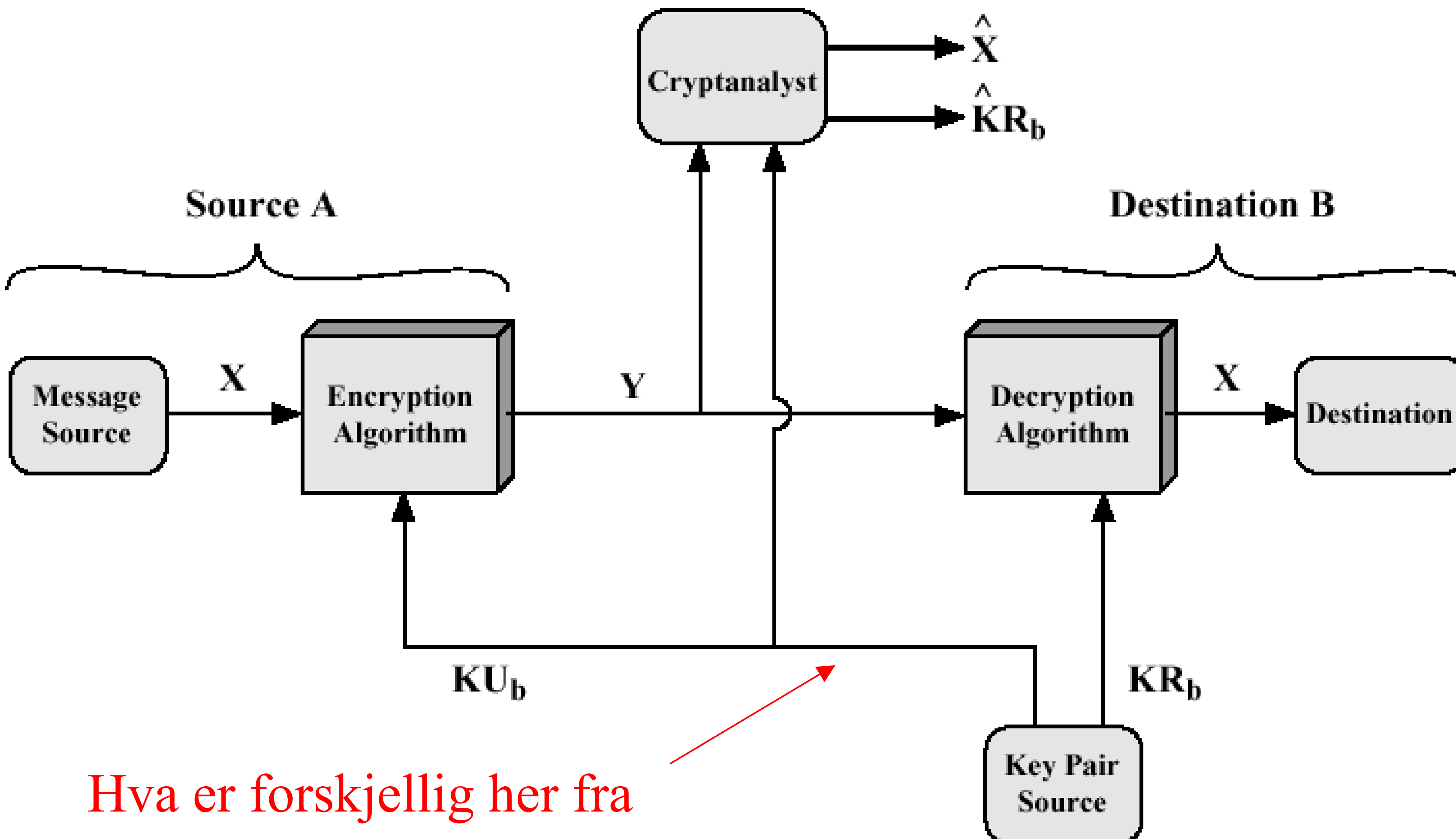
RSA autentisering



RSA autentisering

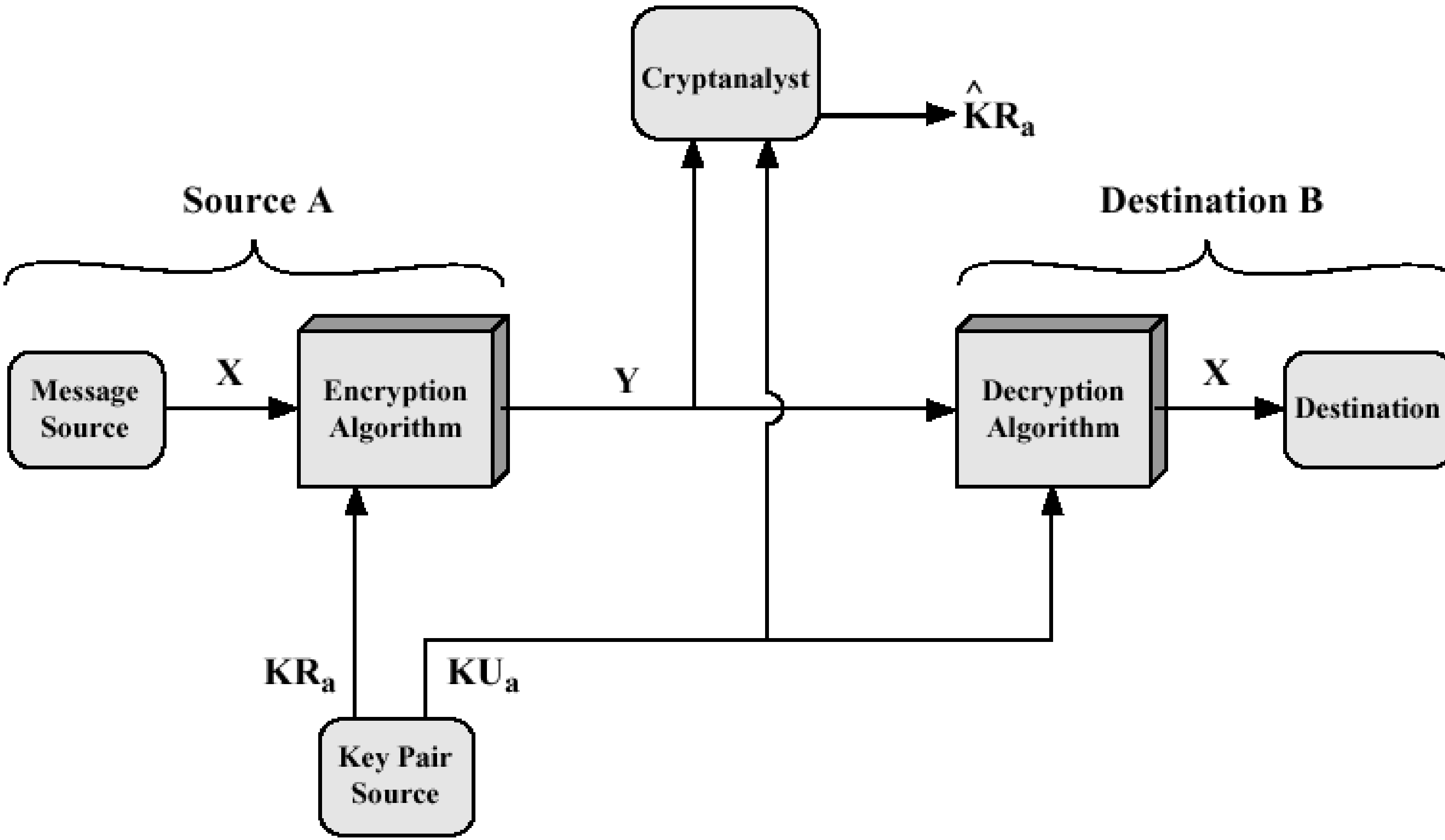
- Denne forenklede formen for autentisering medfører at mottageren ikke kan lese meldingen før den er autentisert
- I praksis gjør man det ikke slik!

Offentlig-nøkkel konfidensialitet



Hva er forskjellig her fra symmetrisk kryptering?

Offentlig-nøkkel autentisering



En liten advarsel

- Den foregående figuren er ikke så generell som vi kunne ønske!
- Det legges opp til at man signerer ved å *kryptere med den private nøkkelen* – dette er egentlig bare tilfelle for RSA; andre signaturalgoritmer (f.eks. DSA) har ikke en kryptering/dekrypterings-mulighet!

Notasjon for offentlige nøkler

- KU_a – offentlig (pUblic) nøkkel til A
- KR_a – privat (pRivate) nøkkel til A
- $E_{KUa}[]$ – kryptering med A's offentlige nøkkel
- $D_{KRa}[]$- dekryptering med A's private nøkkel
- $S_{KRa}[]$ – *signering* med A's private nøkkel (bemerk avvik fra Stallings, som bruker $E_{KRa}[]$ for signering)

Eulers totient-funksjon $\phi()$

- $\varphi(n)$ = ”Antall heltall mindre enn n som er innbyrdes primisk med n”
- Hvis n er et primtall, er $\varphi(n) = n-1$
- Hvis n er et produkt av j primtall $p_1 \ldots p_j$ er $\varphi(n) =$

$$\prod_{i=0}^{j}(p_i-1)$$

Eulers teorem

$$X^{\phi(n)} \equiv 1 \pmod{n}$$

RSA

- Velg to store primtall p,q
- $n = p \cdot q$
- $\varphi(n) = (p-1)(q-1)$
- Velg e slik at $\gcd(\varphi(n),e)=1, 1<e<\varphi(n)$
- $d = e^{-1} \bmod \varphi(n)$
  $(d \cdot e \equiv 1 \bmod \varphi(n))$
- Offentlig nøkkel: $KU=\{e,n\}$
- Privat nøkkel: $KR=\{d,n\}$

RSA kryptering

- Klartekst X må være mindre enn **n**
- Kryptering: $Y = E_{KU}[X] = X^e \bmod n$

- Dekryptering: $D_{KR}[X] = Y^d \bmod n$

$$= (X^e)^d \bmod n$$

$$= X^{ed} \bmod n = X \bmod n$$

RSA eksempel

e offentlig nøkkel
d privat nøkkel
$p = 251$
$q = 269$
$x = 21075$

$$n = p \cdot q = 251 \cdot 269 = 67519$$

$$(p-1)(q-1) = 250 \cdot 268 = 67000$$

$$e \cdot d = 50253 \cdot 27917 = 20939 \cdot 67000 + 1$$

$$x = 21075$$

$$y = 21075^{50253}{}_{\bmod 67519} = 48467$$

$$x = 48467^{27917}{}_{\bmod 67519} = 21075$$

Hvorfor virker RSA?

- Må vise at $Y^d = X$
- Har at
  $ed \equiv 1 \bmod \varphi(n) \Rightarrow ed = k\varphi(n)+1$
- $Y = X^e \bmod n$

Oppstilling

$$Y^d \pmod n = (X^e)^d \pmod n$$

$$= X^{e \bullet d} \pmod n$$

$$= X^{(k \bullet \varphi(n)+1)} \pmod n$$

$$= X^{(k \bullet \varphi(n))} \bullet X^{(1)} \pmod n$$

$$= (X^{\varphi(n)})^k \bullet X \pmod n$$

$$= (1)^k \bullet X \pmod n$$

$$= X \pmod n$$

QED!

Litt mer forvirring

- Under beskrivelsen av RSA benytter Stallings en annen notasjon enn i det generelle tilfellet:

M – Klartekst (tidligere X)

C – Chiffertekst (tidligere Y)

Ellers er det likt!

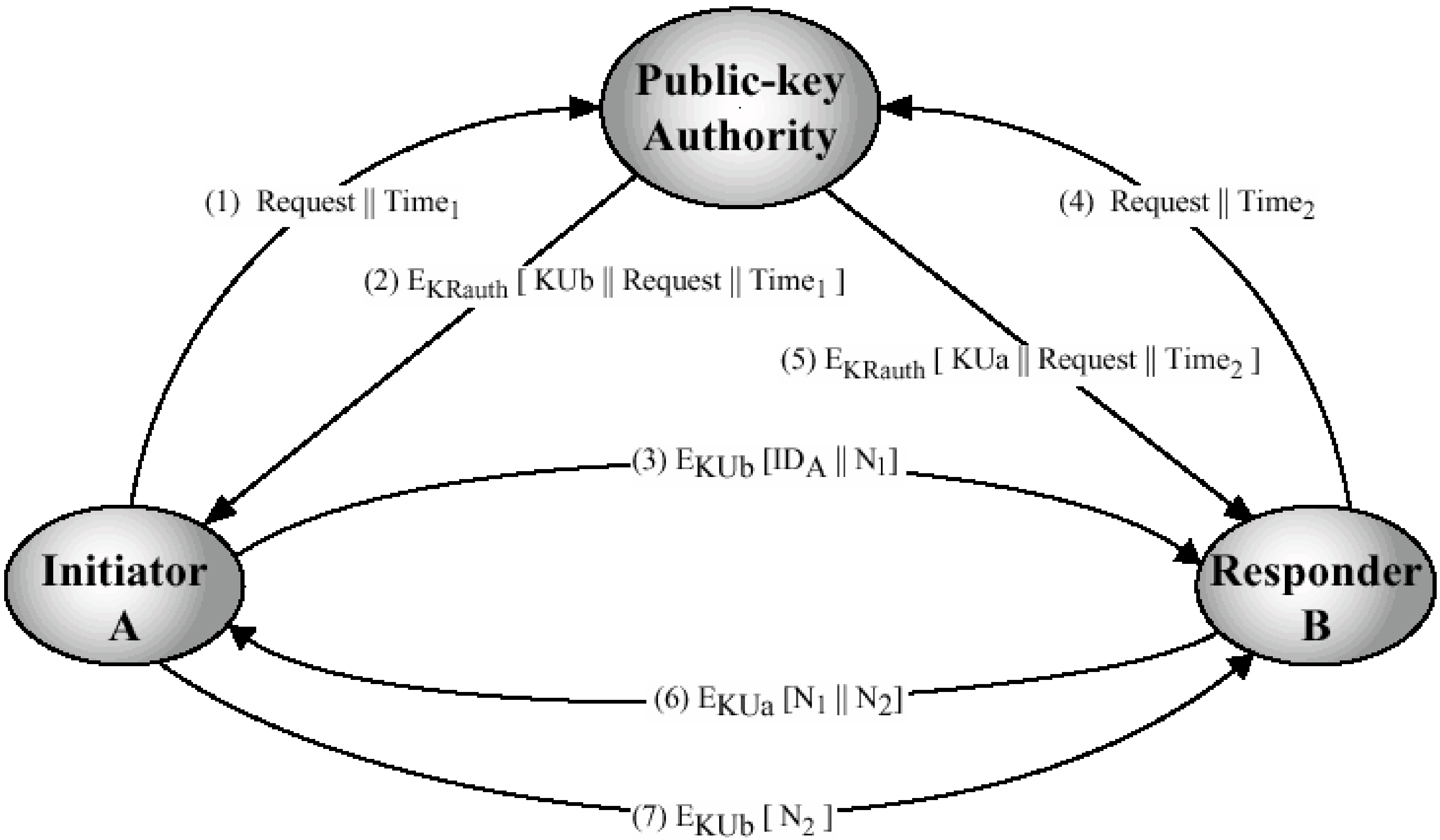
Distribusjon av offentlige nøkler

- Offentliggjøring
  - Newsgrupper
  - WWW
- Offentlig tilgjengelige kataloger
  - ”Telefonkatalogen”
- Autoritet (KDC)
- Sertifikater

Autoritet - notasjon

- KR_{auth} – privat nøkkel til KDC
- KU_a – Offentlig nøkkel til A
- KU_b – Offentlig nøkkel til B
- ID_A – identitet til A

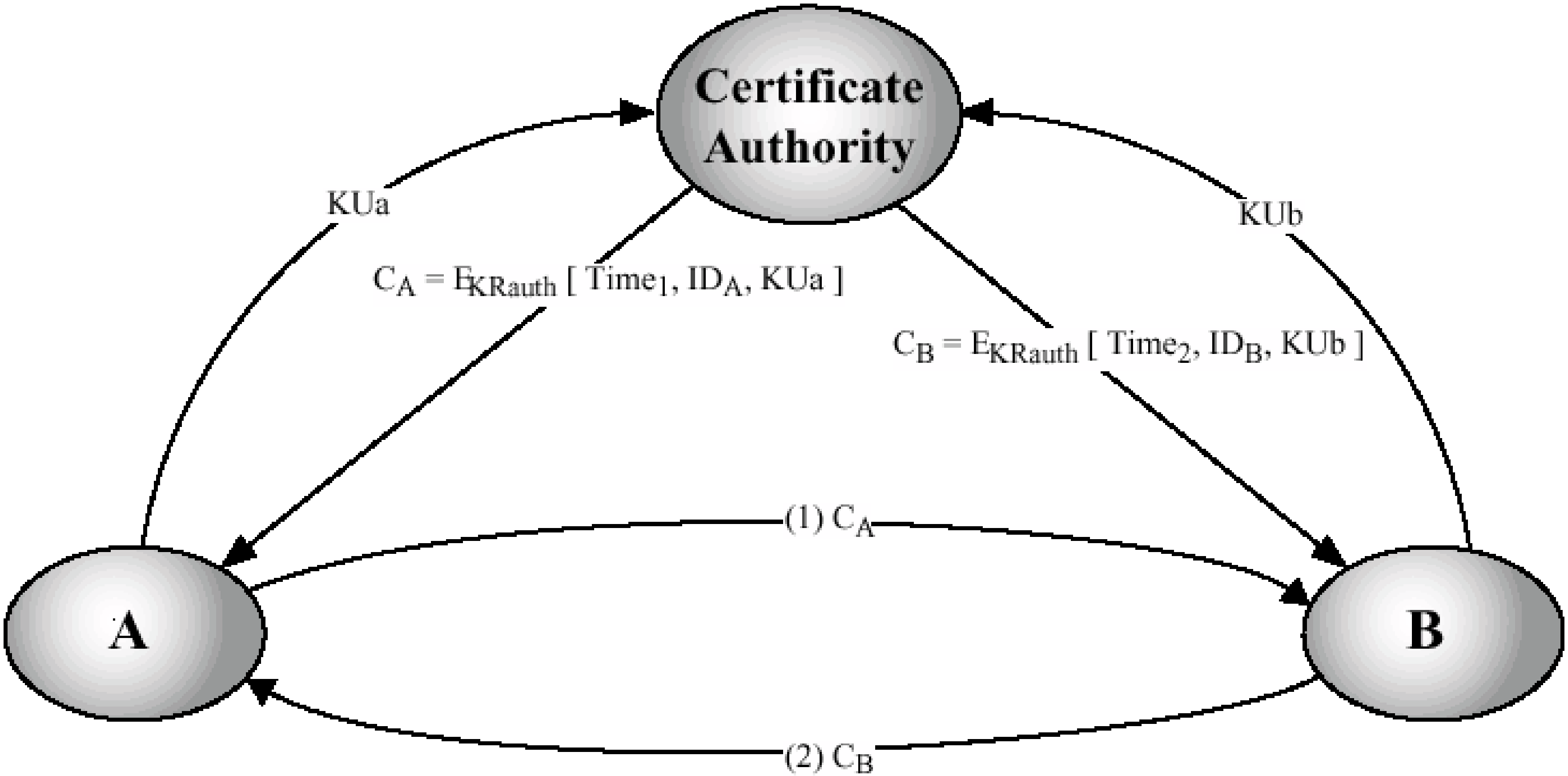
Distribusjon av offentlige nøkler



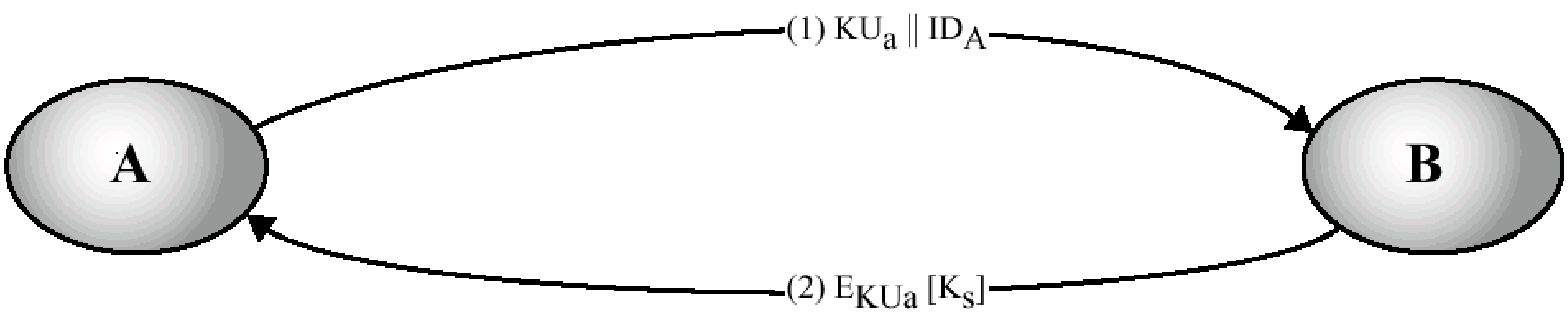
Sertifikater - notasjon

- C_A – sertifikat som inneholder A's offentlige nøkkel
- C_B – sertifikat som inneholder B's offentlige nøkkel

Distribusjon av sertifikater

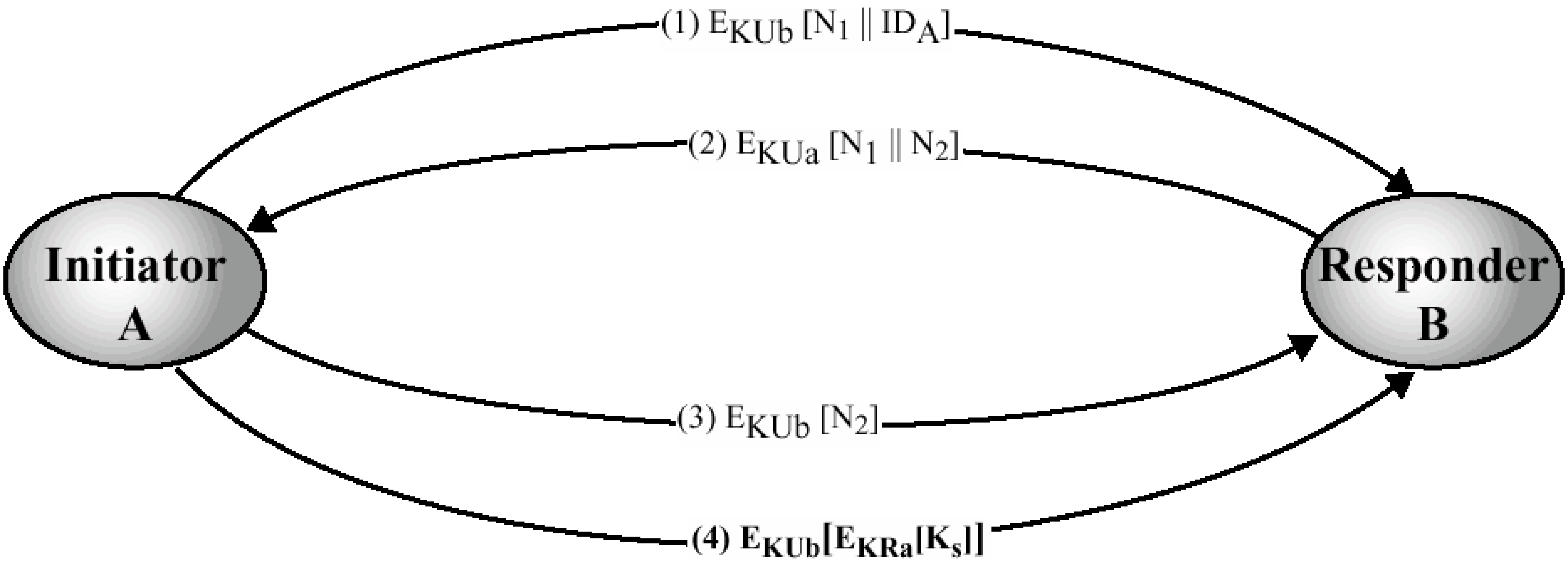


Distribusjon av sesjonsnøkkel



Fare for "Man-in-the-middle"!

Nok en distribusjon



Det evig tilbakevendende...

- Hvis man ikke vet hvem man snakker med, er man like langt
- Offentlige nøkler i seg selv er ikke noe vidundermiddel!

Diffie-Hellman, bakgrunn

- ”Primitiv rot”:
  - Hvis **a** er primitiv rot av primtall **p** $\Rightarrow$
  - a mod p, a^2 mod p, … , a^{p-1} mod p = {1,2, …, (p-1)} i en eller annen permutasjon
- For en integer **b** og primitiv rot **a** av primtall **p** finnes:
  - $b = a^i \bmod p, \qquad 0 \leq i \leq (p-1)$
  - **i** kalles ”diskret logaritme” av **b** for base **a**, mod **p**, eller

    $\mathrm{ind}_{a,p}(b)$

Diffie-Hellman

- q - primtall
- α - α < q , α primitiv rot av q
- Bruker A
  - Velg X_A $X_A < q$
  - Beregn $Y_A = \alpha^{X_A} \bmod q$
- Bruker B
  - Velg X_B $X_B < q$
  - Beregn $Y_B = \alpha^{X_B} \bmod q$

Diffie-Hellman, forts.

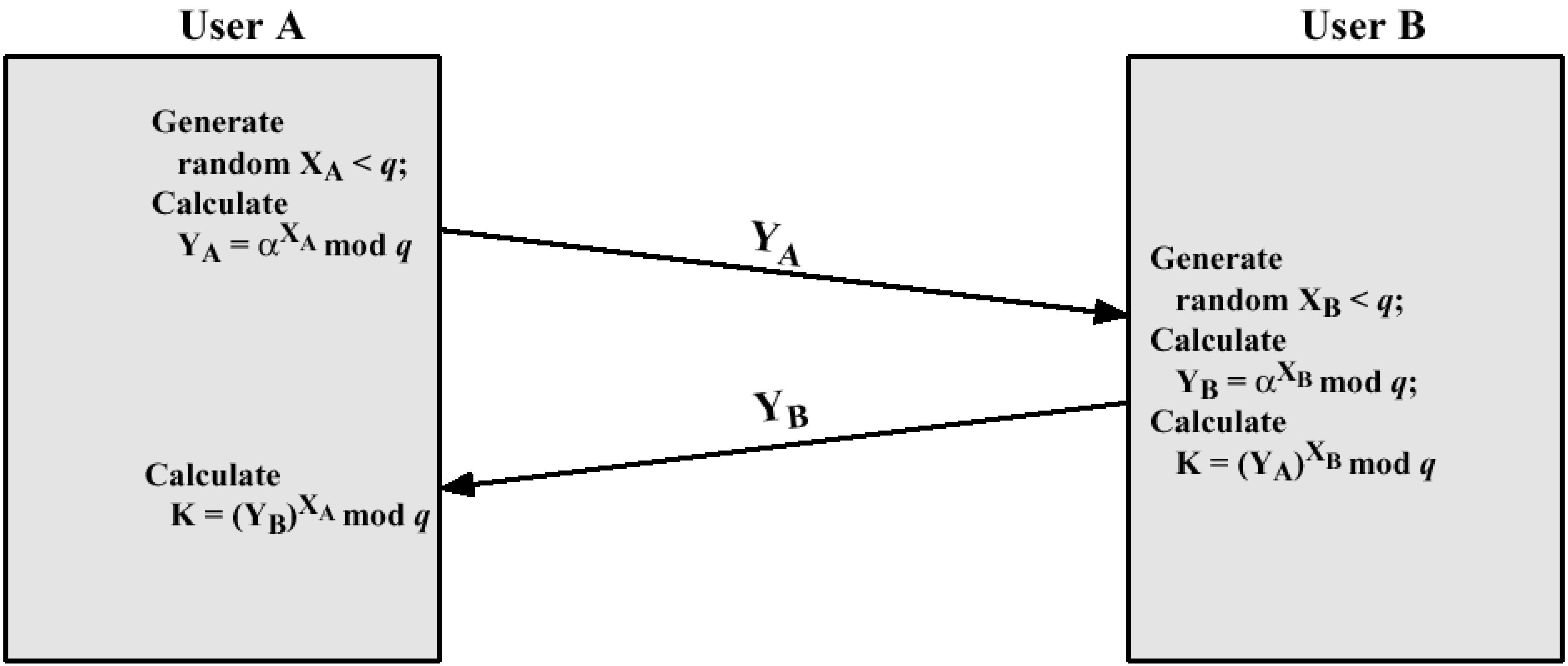
- Generering av hemmelig nøkkel av A

$$K = (Y_B)^{X_A} \bmod q$$

- Generering av hemmelig nøkkel av B

$$K = (Y_A)^{X_B} \bmod q$$

Diffie-Hellman illustrert



Diffie-Hellman, ”bevis”

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= \left(\alpha^{X_B} \bmod q\right)^{X_A} \bmod q \\
&= \left(\alpha^{X_B}\right)^{X_A} \bmod q \\
&= \alpha^{X_B X_A} \bmod q \\
&= \left(\alpha^{X_A}\right)^{X_B} \bmod q \\
&= \left(\alpha^{X_A} \bmod q\right)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$

Diffie-Hellman betrakninger

- Stallings kaller dette for ”key exchange” selv om det egentlig burde hete ”key agreement”
- Ingen av de to partene kan på forhånd bestemme den endelige nøkkelen
- Diffie-Hellman kan ikke brukes til å kryptere informasjon som skal gjenvinnes!

ElGamal

- Algoritme for kryptering med offentlige nøkler basert på samme grunnlag som Diffie-Hellman
- I PGP sies det ”Diffie-Hellman” når de egentlig mener ”ElGamal”

ElGamal algoritme

- p - primtall
- α - primitiv rot av p
- x - klartekst
- a - hemmelig nøkkel
- k - ”hemmelig” verdi (valgt av avsender)
- $\beta = \alpha^a \bmod p$
- Offentlig nøkkel er (β,α,p)

ElGamal, forts.

- $E_K(x,k) = (y_1,y_2)$
  - $y_1 = \alpha^k \bmod p$
  - $y_2 = x\beta^k \bmod p$
- $D_K(y_1,y_2) = y_2(y_1{}^a)^{-1} \bmod p$

- Betraktning: ElGamal medfører en 100% ekspansjon ved kryptering

ElGamal "bevis"

- $D_K(y_1,y_2) = y_2(y_1^a)^{-1} \bmod p$

  $= (x\beta^k \bmod p)((\alpha^k \bmod p)^a)^{-1} \bmod p$

  $= (x\alpha^{ak})((\alpha^k)^a)^{-1} \bmod p$

  $= x(\alpha^{ak})(\alpha^{-ak}) \bmod p$

  $= x$

ElGamal ”bevis”

- $D_K(y_1, y_2) = y_2(y_1^{a})^{-1} \bmod p$

$$= (x\beta^k \bmod p)((\alpha^k \bmod p)^a)^{-1} \bmod p$$

$$= (x\alpha^{ak})((\alpha^k)^a)^{-1} \bmod p$$

$$= x(\alpha^{ak})(\alpha^{-ak}) \bmod p$$

$$= x$$

Elliptic Curve Cryptography

- Dette er fryktelig vanskelig å forklare uten betydelige mengder matematikk
- Hovedpoenget er at man kan oppnå en gitt grad av sikkerhet med kortere nøkkellengder enn man måtte hatt for tilsvarende sikkerhet med RSA

Dagens website

- Handbook of Applied Cryptography

  http://cacr.math.uwaterloo.ca/hac/

  Alt du måtte ønske å vite om anvendt kryptografi – gratis!