

Föreläsning 16

Anteckning... TMV200 Föreläsningar

Skapad: 2020-12-11 07:47

Ändrad: 2020-12-11 09:45

Författare: hchristian.johansson@gmail.com

TMV200 11/12-20

Idag: Public Key Cryptography. —

— { Diffie-Hellman
RSA (del av kursen 5.7 i boken.)

Public Key Cryptography

Alice och Bob vill ha ett hemligt sätt att kommunicera.

Traditionella chiffer

A → B

B → C

C → D

D → E

⋮

HEJ BOB → IFK CPC

Symmetrisk

Om man vet hur man krypterar ett meddelande, så vet man också hur man avkrypterar

(lika svåra - algoritmskt samma
...ut)

komplexitet).

Genom åren blev dessa metoder klart mer avancerade
(Enigma, 2:a världskriget).

Nackdelar: Alice och Bob behöver komma överens
om krypteringsmetoden: kräver
kommunikation som kan avlyssnas.

Opraktiskt t.ex på Internet.

På 1970-talet kom två metoder för att nudd
det här problemet. Båda bygger på kongruens-
räkning.

Diffie-Hellman (-Merkle) key exchange

För avancerade (symmetrisk) krypteringsmetoder så
råder det inte att man känner till metoden
"i princip" för att kunna knäcka den, den
behöver exakta parametrar.

DH: Sätt att etablera hemlig information endast
genom kommunikation över en öppen kanal.

Metod:

- 1) Alice och Bob väljer tillsammans ett
(stort) primtal p . Alla andra tal nedan är
i $\{1, \dots, p-1\}$.

- 2) Sen väljer de ett g som har egenskapen att

$$g^x \not\equiv 1 \pmod{p} \quad \text{om } 1 \leq x < p-1.$$

(Kom ihåg: Fermats lilla sats säger att $g^{p-1} \equiv 1 \pmod{p}$)

(g kallas för en primrot modulo p)

- 3) Alice väljer själv ett tal a och beräknar $C \equiv g^a \pmod{p}$.
Hon skickar C till Bob.

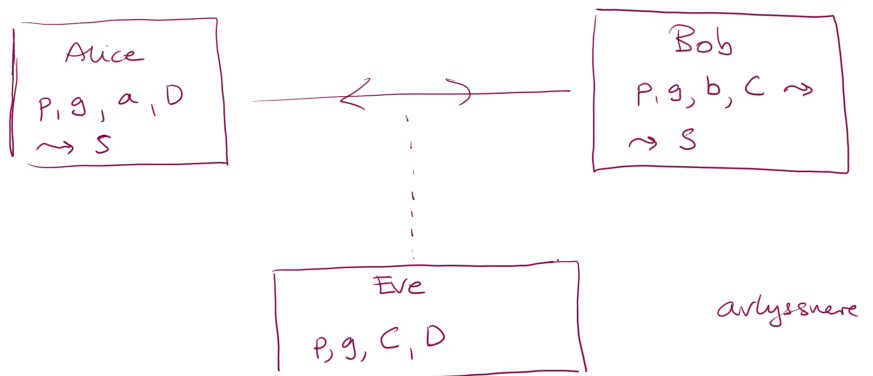
- 4) Bob väljer själv ett tal b och beräknar $D \equiv g^b \pmod{p}$.
Han skickar D till Alice.

- 5) Alice beräknar $D^a \pmod{p}$ och Bob beräknar $C^b \pmod{p}$.

Observera att

$$D^a \equiv (g^b)^a \equiv (g^a)^b \equiv C^b \pmod{p}.$$

Så de får samma tal S . Det här är den hemliga informationen ('nyckeln').



Kan inte Eve beräkna S utifrån (p, g, c, D) ?

Behöver a (eller b), dvs att beräkna a
utifrån g och $C \equiv g^a \pmod{p}$.

Kallas för ett diskret logaritm - problem

(Om $C = g^a$ vanliga tal, så är $a = \log_g C$).

Det är beräkningsmässigt komplext om p är stort
och a, p osv är valda på lämpligt sätt.

Så: Eve kan i teorin lista ut S , men i
praktiken kommer det ta så lång tid att
det blir irrelevanter.

RSA

Rivest - Shamir - Adleman (MIT), 1977.

(Clifford Cocks, 1973, GCHQ, hemligstämpelat
HUs 1997)

Här skickas krypterad information direkt, istället
för byte av en nyckel.

Alice vill kunna ta emot krypterade meddelanden:

1) Alice väljer två (stora) primtal p & q

$1 \leq i \leq p-1$

och beräknar

$$N = pq$$

$$\Phi(N) = (p-1)(q-1)$$

- 2) Alice väljer sedan ett (väl valt)
 $a \in \{1, \dots, N-1\}$ med $\text{sgd}(a, \Phi(N)) = 1$
och beräknar ett b så att

$$ab \equiv 1 \pmod{\Phi(N)}$$

(med Euklides algoritmen, t.ex.).

- 3) Alice lägger ut informationen

$$(N, a) \quad (\text{public key})$$

Hon håller b hemligt (och $\Phi(N), p, q$!)

- 4) Bob vill skicka ett meddelande till Alice.

Han omvandlar det till siffror enligt
någon (offentlig) metod $\leadsto$ tal x .

Han beräknar sedan

$$y \equiv x^a \pmod{N}$$

med Alices allmänna nyckel (N, a) ,
och skickar y till Alice.

- 5) Alice beräknar x (arkrypterar)
genom

$$y^b \equiv (x^a)^b = x^{ab} = x^{1+k\Phi(N)} \equiv x \pmod{N}$$

$\uparrow$
Eulers sats

$$x^{\Phi(N)} \equiv 1 \pmod{N}$$

och kan läsa meddelandet.

Om Bob också har en allmän nyckel (N, a) så kan Alice skicka meddelanden till Bob också.

Knäcka RSA:

Behöver $b \leftarrow \Phi(N) \Leftrightarrow$ faktorisering
 $(p-1)(q-1)$
 $N = pq$

Att primtalsfaktorisera tal är beräkningsmässigt krångligt.

Ex Talet RSA-250 har 250 siffror
 (829 i binärt). Det faktorerades 2020
 med motsvarande 2700 core years på
 en 2.1 GHz Intel Xeon processor.

Tal med minimum 2048 ^{siffror} i binärt rekommenderas
 för praktiskt användning av RSA.

(mer om du inte vill att NSA skall lyssna,
 antagligen...).

Anmärkingar: 1) Det jag presenterat idag är
 grandidéerna, är ett matematiskt
 perspektiv. Mycket mer sofistikerad
 talteori/matematik & datavetenskap
 krävs för en säker implementering.

2) RSA används i praktiken ofta
 som Diffie-Hellman (dvs som ett

sätt att överföra en nyckel till ett symmetriskt krypter) eftersom den är ganska långsam.

- 3) Om Bob har en egen RSA-setup, kan han signera sitt meddelande så att Alice, med hjälp av Bobs public key, kan verifiera att meddelandet kommer från Bob. (står lite om det i boken).

Annat exempel på metod för public key cryptography

Elliptic curve cryptography

elliptisk kurva, bygger på geometri hos elaster
$$y^2 = x^3 + ax + b$$

(populär idag!).