Übung Diffie-Hellman:

p 2048 Bit

gp: nextprime (random (2^2048))

a secret von Alice

$a = 3^{997}$

b secret von Bob

$b = 5^{503}$

$K = 25509 \ldots 516$

Grundrechenarten

$7 + 7 \equiv 1 \mod 13$ → $14 = 1 \cdot 13 + 1$

↑ andere Zahlen mod 13 sind äquivalent

$7 + 7 \equiv -12 \mod 13$ → $14 = 2 \cdot 13 - 12$

$7 \cdot 7 \equiv 10 \mod 13$

$7 + x \equiv 3 \mod 13$

$x \equiv 3 - 7 \equiv -4 \equiv 9 \mod 13$

Division

$$\frac{a}{b} \mod n \ ?$$

$$\equiv a \cdot \frac{1}{b} \mod n$$

Aufgabe: gegeben $n, b \in \mathbb{N}$

finde $x \equiv \frac{1}{b} \mod n$

$\Rightarrow x \cdot b \equiv 1 \mod n \Rightarrow \boxed{x \cdot b = 1 + y \cdot n}$

Rest

$$x \cdot b = 1 + y \cdot n$$

$$x \cdot b - y \cdot n = 1$$

Beispiel:

a) $x \cdot 2 - y \cdot 3 = 1 \Rightarrow (2,1)$

$(5,3)$ $(8,5)$

b) $x \cdot 6 - y \cdot 8 = 1$ keine Lösung

↳ Lösung nur, wenn $ggT(b,n) = 1$

Lösen der Gleichung

$$x \cdot b - y \cdot n = 1$$

mit Euklidischem Algorithmus

z.B. $b = 19$, $n = 23$

$$23 = 1 \cdot 19 + 4$$

$$\boxed{19} = \underline{4} \cdot \boxed{4} + \underline{3}$$

$$\boxed{4} = \underline{1} \cdot \boxed{3} + \underline{\boxed{1}} \rightarrow 1 = 4 - 1 \cdot 3$$

$$1 = 5 \cdot 23 - 6 \cdot 19$$

$$1 = 5 \cdot (23 - 1 \cdot 19) - 1 \cdot 19$$

$$\uparrow$$

$$1 = 5 \cdot 4 - 1 \cdot 19$$

$$1 = 4 - (19 - 4 \cdot 4)$$

$$\uparrow$$

$$1 = \underbrace{5}_{y} \cdot \underbrace{23}_{n} - \underbrace{6}_{x} \cdot \underbrace{19}_{b}$$

$\downarrow \mod n$

$$1 \equiv \cancel{5 \cdot 23} - 6 \cdot 19 \mod 23$$

$$\Rightarrow \frac{1}{19} \equiv -6 \equiv 17 \mod 23$$

Probe: $17 \cdot 19 \equiv 323 \equiv 93 \equiv 1 \mod 23$ ✓

Alternative Methode für "kleine" Zahlen

Ausprobieren!

$\frac{1}{7} \mod 12$

$7 \cdot 1 \equiv 1 \mod 12$? nein

$7 \cdot 2$ "

$7 \cdot 3$ "

$7 \cdot 4$ ⋮

ja!

$7 \cdot 7 \equiv 49 \equiv 1 \mod 12$

$\frac{1}{7} \equiv 7 \mod 12$

Potenzieren, effizient

$$g^a \equiv \underbrace{g \cdot g \cdot g \cdot \ldots \cdot g}_{a\text{-mal}} \mod p$$

Problem: a, g sind Eingabe

Eingabelänge Anzahl der Bits

$\log_2(a)$, $\log_2(g)$

$\log_2(p)$

$a = 2^{\log_2(a)}$

Trick:

mod n
$a \leq n$
$\mathcal{O}(\log^3 n)$

$$g^{16} = (((g^2)^2)^2)^2$$

$$g^{21} = g^{16} \cdot g^4 \cdot g^1$$

$$\underline{g}, g^2, \underline{g^4}, g^8, \underline{g^{16}}$$

$$\left.\#\text{Operationen} \leq 2 \cdot \log_2(a)\right\} \mathcal{O}(\log^3 a)$$

Operationen ⟷ Mult. / Div

$\mathcal{O}(\log^2(a))$

Parameterwahl für Diffie-Hellman

p, Standards beginnen bei 2048 Bit

g sollte ein Erzeuger sein
(oder große Ordnung mod p haben

p soll eine sichere Primzahl sein

Erzeuger, Ordnung

$p = 7$

$\mathrm{ord}_7\, 2 = 3$

$\mathrm{ord}_7\, 3 = 6$

3, 5 sind Erzeuger

i	0	1	2	3	4	5	6
$2^i \bmod 7$	1	2	4	1	2	4	1
$3^i \bmod 7$	1	3	2	6	4	5	1
$4^i \bmod 7$	1	4	2	1	4	2	1
$5^i \bmod 7$	1	5	4	6	2	3	1
$6^i \bmod 7$	1	6	1	6	1	6	1

kryptographisch relevant:

- Erzeuger = max. Ordnung $= p-1$
- "große" Ordnung, z.B. $\frac{p-1}{2}$

Übungsaufgabe:

wieviele Schlüssel kann man

- auf 10.000 CPUs / Cores
- mit 3,6 GHz
- in 1 Jahr ausprobieren, falls das Ergebnis in 1 Takt berechnet wäre

Primzahlen

allgemein:

$$\boxed{\begin{array}{l} \underline{\text{Fermat-Satz}} \\ p \text{ Primzahl}, \quad g \in \mathbb{N}, \\ ggT(g, p) = 1 \\ \Rightarrow \quad g^{p-1} \equiv 1 \mod p \end{array}}$$

Beispiel:

$p = 7, \quad g = 3 \Rightarrow 3^{p-1} = 3^6 \equiv 1 \bmod 7$

$p = 21, \quad g = 5$

$5^{21-1} = 5^{20} \equiv 4 \not\equiv 1 \bmod 21$

p Primzahl $\Rightarrow \quad \equiv 1 \bmod p$

keine Primzahl $\Leftarrow \quad \not\equiv 1$

$p = 561$

$2^{560} \equiv 1 \mod 561$

aber 561 keine Primzahl

Konsequenz:

Fermat-Test wird so weit verbessert, dass mehrmalige Anwendung mit mehreren g's eine Wahrscheinlichkeit zulässt, ob das vorliegende p Primzahl ist.

Miller-Rabin-Primzahltest