Recall: Let p be prime, b be a primitive root mod p, $a \in \mathbb{Z}$, $a \not\equiv 0 \pmod{p}$. The <u>discrete logarithm</u> $\log_{b,p}(a)$ is an integer $d \in \{0, 1, \dots, p-2\}$ such that

$$b^d \equiv a \pmod{p}.$$

Note: Input of $\log_{b,p}(a)$ is a residue mod p
Output is a residue mod $p-1$.

Note: $\log_{b,p}$ is undefined for $a \equiv 0 \pmod{p}$.

Example: $p = 13$, $b = 2$.

a	1	2	3	4	5	6	7	8	9	10	11	12
$\log_{2,13}(a)$	0	1	4	2	9	5	11	3	8	10	7	6

In general for big primes p, computing discrete logs is a very hard problem (the <u>discrete log problem</u>).

§16. Diffie-Hellman key exchange and Elgamal cryptosystem.

Q: We want to establish a common secret key by communicating via non-secure channel.

$$\text{Alice} \longleftrightarrow \text{Bob}$$

Evesdropper

Algorithm (Diffie-Hellman key exchange) Example.

Step 1. Alice chooses:

Prime p — $p = 47$

Carefully chosen $b \in \{1, 2, \ldots, p-1\}$ — $b = 5$

~~Secret~~

Private key x — $x = 4$

She computes $k \equiv b^x \pmod p$ — $k \equiv 5^4 \equiv 14 \pmod{47}$

Step 2: Alice sends to Bob (p, b, k), keeping x in secret — $(47, 5, 14)$

Step 3: Bob chooses:

His own private key y — $y = 7$

He computes $c \equiv b^y \pmod p$ — $c \equiv 5^7 \equiv 11 \pmod{47}$

Step 4: Bob sends c to Alice, keeping y in secret.

Step 5: Both Alice and Bob agree on the same shared secret s

Alice computes: $s \equiv c^x \equiv b^{xy} \pmod p$ — $11^4 \equiv 24 \pmod{47}$

Bob computes: $s \equiv k^y \equiv b^{xy} \pmod p$ — $14^7 \equiv 24 \pmod{47}$

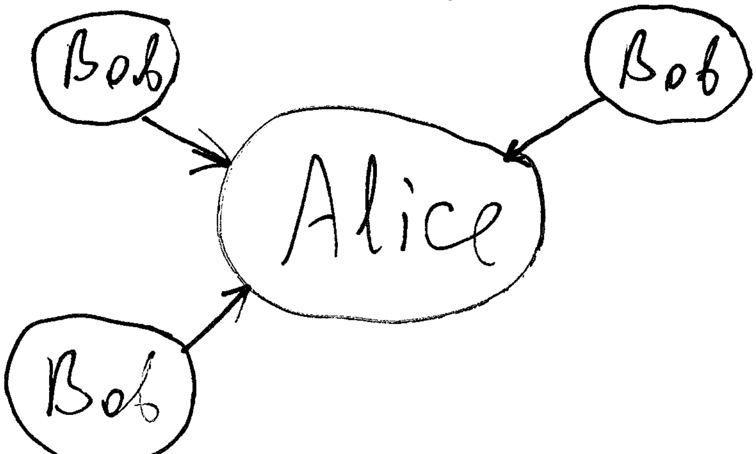
The problem for an Evesdropper (Diffie-Hellman problem): Given $p, b, k \equiv b^x \pmod p$ and $c \equiv b^y \pmod p$, find $s \equiv b^{xy} \pmod p$

(Note: $b^x \cdot b^y \equiv b^{x+y} \pmod p$, not b^{xy}).

It is believed (not proven) that this requires the solution of the discrete log problem: given $p, b, b^x \pmod p$, compute $x (= \log_{b,p}(b^x))$.

Common secrets can be used in some classical cryptosystems (DES, etc). Alternatively it can be used in open key cryptosystems like Elgamal.

Elgamal cryptosystem: everyone can encrypt the message, only Alice can decrypt it (as for RSA).



Algorithm (Elgamal): Example

Step 1: As before, Alice chooses (p, b, k), where $k \equiv b^x \pmod p$ — $(47, 5, 14)$, $x = 4$.

Step 2: Alice publishes (p, b, k), keeping x in secret.

Step 3: Bob encodes the message as $[M_1, M_2, \ldots, M_l]$ of — $[3, 13]$.

residues modulo p.

Step 4: Bob chooses his own private key y

He computes $c \equiv b^y \pmod p$ and (the common secret) $s \equiv k^y \pmod p$. $c = 11$, $s = 24$

Then he encrypts the message by replacing M_i by $sM_i \pmod p) = M_i'$. $M_1' \equiv 24 \cdot 3 \equiv 25$, $M_2' \equiv 24 \cdot 13 \equiv 30$.

Step 5: Bob sends the ciphertext to Alice:

$\langle c, [M_1', M_2', \dots, M_\ell'] \rangle$ $\qquad \langle 11, [25, 30] \rangle$.

Step 6: Alice computes:

$s \equiv c^x \pmod p$ $\qquad s \equiv 11^4 \equiv 24 \pmod{47}$

$t \equiv s^{-1} \pmod p$ $\qquad t \equiv 2 \pmod{47}$

$M_i \equiv t \cdot M_i' \pmod p$ $\qquad M_1 \equiv 2 \cdot 25 \equiv 3 \pmod{47}$, $M_2 \equiv 2 \cdot 30 \equiv 13 \pmod{47}$.

For security of these cryptosystems we need the computation of x (the discrete log) to be very hard.

We can try computing

$$b^0, b^1, b^2, \dots, \pmod p$$

until we find $k \equiv b^x \pmod p$. That requires up to $\mathrm{ord}_p(b)$ operations. So we want this number to be high.

$\Rightarrow$ b is a primitive root ($\mathrm{ord}_p(b) = p-1$, the highest possible).

p is Large (≈ 600 digits).