7.2. Shamir's no-key protocol

Prop. 7.7. Let p be prime, $a, b \in \mathbb{Z}_{p-1}^*$. Then
$\forall m \in \mathbb{Z}_p: \quad m^{aba^{-1}b^{-1}} \equiv m \pmod p$.

Proof. $a^{-1}, b^{-1} \in \mathbb{Z}_{p-1}^*$ exist by definition.
$aa^{-1} \equiv 1 \pmod{p-1}$ and $bb^{-1} \equiv 1 \pmod{p-1}$,
i.e., $bb^{-1} = t(p-1)+1$ for some $t \in \mathbb{N}_0$
Hence, for all $m \in \mathbb{Z}_p$:

$$m^{aba^{-1}b^{-1}} \bmod p = (\underbrace{m^a \bmod p}_{c})^{bb^{-1}a^{-1}} \bmod p$$

$$= (\underbrace{c^{t(p-1)}}_{\equiv 1 \text{ (Fermat)}} \cdot c)^{a^{-1}} \bmod p$$

$$= m^{aa^{-1}} \bmod p \underset{\substack{\uparrow \\ \text{same argument}}}{=} m \bmod p. \qquad \blacksquare$$

A sends a message to B as follows:

- Initial setup: a prime p ~~and~~ published
- Protocol actions
  A and B chose secrete numbers $a, b \in \mathbb{Z}_{p-1}^*$ and calculate a^{-1}, b^{-1} in $\mathbb{Z}_{p-1}^*$.

RWTH AACHEN UNIVERSITY

$A \to B$: $c_1 = m^a \bmod p$ (A locks, sends to B)

$B \to A$: $c_2 = c_1^b \bmod p$ (B locks, returns to A)

$A \to B$: $c_3 = c_2^{a^{-1}} \bmod p$ (A unlocks, returns to B)

B deciphers $m = c_3^{b^{-1}} \bmod p$ (B unlocks, reads)

Observe: no authentication.

8. Public-Key Encryption

Idea: by Diffie & Hellman (76), earlier but unpublished paper by James Ellis (70), paper released by British Gov. 1997. (The possibility of non secret encryption.)

- All users share the same e, d (encryption, decryption)
- Each user has a pair of keys (K, L) such that

$$d(e(M,K), L) = M \quad \forall M \in \mathcal{M}$$

K is made public, L is private

- Requirements

(i) $C = e(M,K)$ "easy" given M and K, solving for M "infeasible".

(ii) $M = d(C,L)$ "easy" given C, L

Hence: $f_K(M) = e(M,K)$ is a one-way function with "trapdoor" L.

- Further requirements
  - (i) (K, L) easy to generate
  - (ii) There are sufficiently many pairs (K, L), exhaustive search impossible.

8.1. The RSA Cryptosystem

(Rivest, Shamir, Adleman, 1977)

Prior invented by Clifford Cox (1973), not published, released by Br. Gov. 1997.

RSA protocol

(i) Choose $p \neq q$ (large primes)
compute $n = p \cdot q$

(ii) Choose $d \in \mathbb{Z}^*_{(p-1)(q-1)}$, i.e., $\gcd(d, (p-1)(q-1)) = 1$
Compute $e = d^{-1} \bmod (p-1)(q-1)$.

(iii) Public key: (d^{-1}, n)
Private key: d, $\{p, q\}$

(iv) Message $m \in \{1, \ldots, n-1\}$
Encryption: $c = m^e \bmod n$
Decryption: $m = c^d \bmod n$

Questions:
1. Is m the original message?
2. Security?
3. Implementation?

RWTH AACHEN UNIVERSITY

Proposition 8.1 $p \neq q$ prime, $x, y \in \mathbb{N}$

$x \equiv y \pmod p$ and $x \equiv y \pmod q \iff x \equiv y \pmod{p \cdot q}$ $\perp$

Proof. $p \mid (x-y),\ q \mid (x-y) \iff p \cdot q \mid x-y$ $\square$

Prop. 8.2 $p \neq q$ prime, $n = p \cdot q$, $d, d^{-1} \in \mathbb{Z}^*_{(p-1)(q-1)}$,

$0 \leq m < n$, $c = m^{d^{-1}} \bmod n$.

Then $m = c^d \bmod n$. $\lrcorner$

Proof.

$d^{-1} d \equiv 1 \pmod{(p-1)(q-1)} \Rightarrow \exists t: t(p-1)(q-1) + 1 = d^{-1} d$

(i) $\gcd(m, p) = 1$

$$(m^{d^{-1}})^d \equiv m^{d^{-1} d} \equiv m^{t(p-1)(q-1)+1}$$

$$\equiv (m^{p-1})^{t(q-1)} \cdot m$$

$$\overset{\text{Fermat}}{\equiv} 1^{t(q-1)} \cdot m \equiv m \pmod p$$

(ii) $\gcd(m, p) = p$:

$p \mid m$, i.e., $m \equiv 0 \pmod p \Rightarrow m^{d^{-1} d} \equiv 0 \equiv m \pmod p$

Analogously: $(m^{d^{-1}})^d \equiv m \pmod q$

Using Prop. 8.1: $(m^{d^{-1}})^d \equiv m \pmod{p \cdot q}$. $\square$

Thinking the Future
Zukunft denken

Security of RSA

Relevant: chosen plaintext attack.

Known: d^{-1}, n, arbitrarily many (m, c).

a) Factoring of n, use p, q to compute

$$d = (d^{-1})^{-1} \bmod (p-1)(q-1) = (d^{-1})^{-1} \bmod \varphi(n)$$

Recall $\varphi(n) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1)$

But: Factoring infeasible

b) Computing square roots mod n allows factoring.

<u>Prop. 8.3:</u> $n = p \cdot q$, $p \neq q$ prime, x a nontrivial solution of $x^2 \equiv 1 \pmod{n}$, i.e., $x \not\equiv \pm 1 \pmod{n}$. Then

$$\gcd(x+1, n) \in \{p, q\}.$$

<u>Proof.</u> $x^2 \equiv 1 \pmod{n}$, $x \not\equiv \pm 1 \pmod{n}$

$\Rightarrow n \mid (x^2 - 1)$, $n \nmid (x-1)$, $n \nmid (x+1)$

$\Rightarrow pq \mid (x+1)(x-1)$, $p \cdot q \nmid (x-1)$, $p \cdot q \nmid (x+1)$

$\Rightarrow \gcd(x+1, n) \in \{p, q\}$ and
$\gcd(x-1, n) \in \{p, q\}$ $\boxtimes$

Hence: Computing square roots is no easier than factoring.

RWTH AACHEN UNIVERSITY

c) Computing $\varphi(n)$ without factoring n.

Any eff. alg. for computing $\varphi(n)$ yields an eff. alg. for factoring

<u>Proof.</u> $n = p \cdot q$ (p, q prime, unknown)

$\varphi(n) = (p-1)(q-1)$ (known)

$$\varphi(n) = (p-1)(q-1) = pq - p - q + 1 \Leftrightarrow p + q = n - \varphi(n) + 1 \quad (1)$$

$$(p-q)^2 - (p+q)^2 = -4pq \Leftrightarrow (p-q)^2 = (p+q)^2 - 4n \quad (2)$$

$$q = \frac{1}{2}\left((p+q) - (p-q)\right) \quad (3)$$

(1) yields $p+q$, from (2) obtain $p-q$, q follows by (3).

Hence, computing $\varphi(n)$ is no easier than factoring.

d) Computing $(d^{-1})^{-1}$ (without knowing $\varphi(n)$)

<u>Prop. 8.4.</u> Let $n = p \cdot q$, $p \neq q$ prime. Any eff. alg. for computing $b^{-1} \bmod \varphi(n)$ leads to an eff. probabilistic alg. for factoring n with error prob. $< \frac{1}{2}$.

<u>Proof.</u> Stinson, p. 139-141

Hence, computing $b^{-1} \bmod \varphi(n)$ is no easier than factoring.

Thinking the Future
Zukunft denken

RWTH AACHEN UNIVERSITY

Remarks

a) If d is known, n can be eff. factored (see Prop. 8.4). If the private key d is detected, it is not sufficient to compute a new d^{-1}, also change p, q.

b) Never let somebody observe your decryption process! (Side-channel attacks)

c) Conjecture of RSA (78)
An eff. alg. for breaking RSA yields an eff. alg. for factoring. (Still open)

8.1.2. Implementation of RSA

- Large primes p, q → Miller-Rabin
- Choice of $d \in \mathbb{Z}^*_{(p-1)(q-1)}$ → choose d prime, $d \geq \max\{p, q\}$,
  or start with d_0
  $d_0 = d_0 + 1$ until $\gcd(d_0, \varphi(n)) = 1$
  (Euclidean algorithm)
- Inverse $d^{-1} \pmod{\varphi(n)}$ → extended Eucl. alg.
- Exponentiation → square-and-multiply
- Table concerning RSA hardware, see Schneier p. 469
  (1995: 1 MB/sec.)
  (RSA ~ 1000 times slower than AES.)

Thinking the Future
Zukunft denken