

Public-key Cryptography:

Recall: Symmetric-Crypto: A and B share a secret/private key k .
 If third person E has k \rightarrow E can decrypt messages.

\swarrow
 if k uses for identify. \rightarrow E can fake messages.

Weak Point:

- One has to exchange every time a key over a secure channel \Rightarrow key management can become costly.

Public-key Cryptography = asymmetric cryptography.
 (Used in PGP)

"If A sends a message to B, A looks up the public key of B in order to encrypt the message."

Every user has a pair of keys (e, d) with different, asymmetric properties:

- ▶ e public key, it allows any party to encrypt to this key.
- ▶ d private key, it allows the owner to decrypt messages sent to the matching public key e .

- ▶ There should be no way to compute d from e
 (The other way around is quite common)

Public-key cryptosystems consists of 3 algos

- (i) key generation (ii) encryption (iii) decryption

Mathematical Background:

(2)

Let $n \in \mathbb{N}$. We consider $\mathbb{Z}/n = \mathbb{Z}/n\mathbb{Z}$ = "integers modulo n "

~~We identify \mathbb{Z}/n with $\{0, 1, \dots, n-1\}$~~

For $a \in \mathbb{Z}$ we can compute (using XGCD) integers b, \tilde{b} such that

$$\gcd(a, n) = b \cdot a + \tilde{b} \cdot n$$

if $\gcd(a, n) = 1$

$$\Leftrightarrow 1 = b \cdot a + \tilde{b} \cdot n \Rightarrow 1 \equiv b \cdot a \pmod{n} \Leftrightarrow b^{-1} \equiv a \pmod{n}$$

In that case b is called the inverse of a modulo n .

Every a with $\gcd(a, n) = 1$ has an inverse modulo n .

Notation:

• $(\mathbb{Z}/n)^{\times}$ = invertible elements in $\mathbb{Z}/n = \{a \mid 1 \leq a < n, \gcd(a, n) = 1\}$
= multiplicative group mod n .

• $|(\mathbb{Z}/n)^{\times}| =: \phi(n)$ size of $(\mathbb{Z}/n)^{\times}$ (order of that finite group)
 ϕ is called Euler's totient, Euler Phi, Phi - function.

Examples:

(a) $(\mathbb{Z}/7)^{\times} = \{1, 2, \dots, 6\}$; $\phi(7) = 6$

Lemma: $\phi(p) = p-1$ for any prime number p .

(b) $(\mathbb{Z}/6)^{\times} = \{1, 5\}$; $\phi(6) = 2$

(c) $(\mathbb{Z}/10)^{\times} = \{1, 3, 7, 9\}$; $\phi(10) = 4$

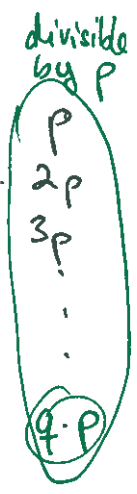
(d) $\phi(15) = |\{1, 2, 4, 7, 8, 11, 13, 14\}| = 8 = (3-1)(5-1)$

Lemma: Let p, q be two prime numbers. Then,

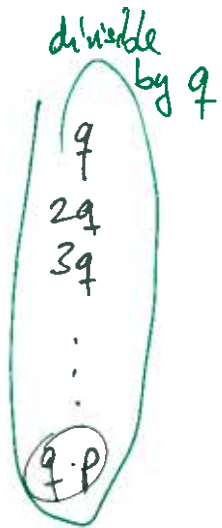
$$\phi(p \cdot q) = (p-1)(q-1)$$

Proof.

1	2	3	...
$p+1$	$p+2$	$p+3$...
$2p+1$	$2p+2$	$2p+3$...
...
$(q-1)p+1$



Copy and change $p \rightarrow q$



q of them!

p of them

These two are $p \cdot q - q - p + 1 = (p-1)(q-1)$ □

Lemma:

- $(\mathbb{Z}/n)^\times$ is a group under multiplication with $\phi(n)$ elements.

- $a^{\phi(n)} = 1$ and $a^{\phi(n)+1} = a$

\hookrightarrow holds in general for any $a \in \mathbb{Z}/n!$

RSA: (Rives, Shamir, Adleman 1977).

Let p, q be two prime numbers, $p \neq q$ and p, q both have about the same size.

Compute $n = p \cdot q$ and $\phi(n) = (p-1)(q-1)$. \rightarrow we can compute this only, because we have factorization

Fix integer e with $\gcd(e, \phi(n)) = 1$. that is, e has inverse. Let

d be an integer such that $e \cdot d \equiv 1 \pmod{\phi(n)}$.

- Forget about $p, q, \phi(n)$

- Publish the public key (e, n) , BUT: d is kept secret \leftarrow private key

Encryption: Encrypt $m \in \mathbb{Z}, m < n$ by computing $c \equiv m^e \pmod{n}$

Decryption: Decrypt c by computing $m' \equiv c^d \pmod{n}$.

System works, i.e. $m' = m$, since

$$m' \equiv c^d \equiv (m^e)^d \equiv m^{e \cdot d} \equiv m^{k \cdot \phi(n) + 1} \equiv m \pmod{n}$$

for some integer k . Remark: An RSA key can also be used to sign messages.

Signature: to sign message m with $m < n$ compute

$$s \equiv m^d \pmod{n}$$

Since d is a secret, nobody else can do this computation.

Verification: To verify that s is a signature on m compute

$$m' \equiv s^e \pmod{n}$$

and accept the signature as valid if $m = m'$.

~~set~~ school look

Remark: This RSA version does not ~~require~~ satisfy modern requirements.

- We want to be able to sign messages longer than n .
- In practice we sign the hash $h(m)$ instead of m . (Is also better for security)

Computational issue of m^e :

How to compute $m^e \pmod{n}$? (m^e gets very large for large e)

Idea: Reduce modulo n . ~~with every step~~ after any multiplication / squaring.

• $m^4 = m \cdot m \cdot m \cdot m = m^2 \cdot m^2 = (m^2)^2$
 2 mult. 2 squarings + 1 mult

• $m^9 = (m^2)^2 \cdot m$ • $m^6 = (m^2 \cdot m)^2$ • $m^{10} = ((m^2)^2 \cdot m)^2$
 3 squarings + 1 mult.

• $m^{11} = ((m^2)^2 \cdot m)^2 \cdot m$ • $m^{15} = (((m^2 \cdot m)^2 \cdot m)^2 \cdot m)$

In general there are at most $\lfloor \log_2(e) \rfloor$ squarings and at most that many multiplications.

To compute the pattern we look at the binary representation of the exponent:

$$4 = (100)_2, 9 = (1001)_2, 6 = (110)_2, 10 = (1010)_2, 11 = (1011)_2,$$

$$15 = (1111)_2.$$

Thus, ignoring the first position, we perform for any entry squaring and whenever the bit is one a multiplication.

We scan from left to right.

Let $e = \sum_{i=0}^{l-1} e_i 2^i$, $l = \lfloor \log_2 e \rfloor + 1$.

We compute $m^e \pmod n$ as follows:
Square-and-multiply algorithm:

1. $c \leftarrow m$
2. For $i = l-2$ to 0
 $c \leftarrow c^2 \pmod n$
 if $e_i = 1$: $c \leftarrow cm \pmod n$
3. return c .

Algo takes $l-1$ squarings and $\leq l-1$ multiplications.

Multiplications = $\#\{i \mid e_i \neq 0\} =$ Hamming weight of e

Remark: • One can choose e small ~~small~~ ^{since} it belongs to the public key - small d would not work.

- Decryption also needs multiplication and squares.
 - $d = 5, 17$ ^{small} would be easy to find by brute force
 - $d = \sqrt[3]{n}$ is dangerous by attack of Wiener.

• Common choices are $e = 3, e = 17, e = 2^{16} + 1 = 65537$
(These e 's have also small Hamming weight)

Problems using schoolbook RSA and small e:

(6)

Assume: A, B, C all use $e=3$ and somebody sends the same message to all of them
Denote by n_A, n_B, n_C the modulus of A, B, C, respectively.

We obtain

$$\left. \begin{aligned} c_A &\equiv m^3 \pmod{n_A} \\ c_B &\equiv m^3 \pmod{n_B} \\ c_C &\equiv m^3 \pmod{n_C} \end{aligned} \right\} (*)$$

One of $\gcd(n_A, n_B), \gcd(n_B, n_C), \gcd(n_A, n_C)$ is not 1.

(A) \Rightarrow we get one factor p_i of one of them

\Rightarrow ~~factorize~~ then we can factor the public key

~~we~~
We compute ϕ and $d \Rightarrow$ completely broken

(B) two parties share a key (e, n) and can read each others messages.

2) $\gcd(\dots) \geq 1$.

Then (*) is a system of congruences with moduli that are coprime.

~~For the~~
~~to show exist a solution modulo $n_A \cdot n_B \cdot n_C$ by the Chinese Remainder~~
~~Theorem.~~

So there exist a solution M modulo $N := n_A \cdot n_B \cdot n_C$ by the Chinese Rem Th.:

with ~~$M \equiv m^3 \pmod{N}$~~ $M \equiv m^3 \pmod{N}$

Since $m^3 < N$ we have $M = m^3 \Leftrightarrow \sqrt[3]{M} = m$.

In \mathbb{Z} we can efficiently compute cube roots.

Conclusion: We deduce m from the publicly available information.

Remark: Same approach works for e messages to recipients all using exponent e .