Tutorial V group 2

17 October 2019 15:59

Assymetric Cryptography

→ RSA encryption

→ Diffie-Hellman key exchange

→ Knapsack cryptosystem

Basic Numbe Theory

$\mathbb{Z}_n \rightsquigarrow$ set of all remainders after division by n

$+ \bmod n$, $(\mathbb{Z}_n, +)$ is a group

$\mathbb{Z}_n^* \rightsquigarrow$ set of all non-zero remainders after division by n

$(\mathbb{Z}_n, \cdot \bmod n) \rightsquigarrow$ a group for prime n (inversions exist)

$\rightsquigarrow$ for general n monoid (not all inversions exist)

$\frac{a}{b} \bmod n = a \cdot b^{-1} \bmod n$ (b^{-1} exists iff $\gcd(b,n)=1$)

~~$\frac{5}{3} \bmod 7 \neq 1{,}666$~~

$= 5 \cdot 3^{-1} \bmod 7$

$3^{-1} = 5 \bmod 7$

$= 5 \cdot 5 \bmod 7$

$= 25$

$= 3 \bmod 7$

How to calculate inverses mod n?

Euclid's algorithm → algorithm to calculate $\gcd(a,b)$ for any $(a,b) \in \mathbb{Z}$

Bezout's identity → for a,b: $\gcd(a,b)=1$

$\exists\, x, y$ s.t.

$\exists\, x, y$ s.t.

$$ax + by = 1$$

Extended Euclid's algorithm $\to$ algorithm to calculate x, y from Bezóuts identity

$$ax + by = 1$$

$$ax = 1 - by \quad / \bmod b$$

$$\Downarrow$$

$$a^{-1} \equiv x \mod b$$

$$ax = 1 + 0 \mod b$$

Extended Euclid's algorithm

find $\gcd(a,b)$ example find $\gcd(96,18)$

$96 : 18 = 5$ rm 6 $\to$ the last non-zero remainder is gcd we are looking for

$18 : 6 = 3$ rm 0

Find $\gcd(17,3) = 1$

$17 : 3 = 5$ rm 2

$3 : 2 = 1$ rm 1

$2 : 1 = 2$ rm 0

$2 = 17 - 3 \cdot 5$

$1 = 3 - 2 \cdot 1$

$$1 = 3 - 2 \cdot 1$$

$$1 = 3 - (17 - 3 \cdot 5) \cdot 1$$

$$1 = 3 - 17 + 3 \cdot 5$$

$$1 = 3 \cdot 6 - 17 \cdot 1$$

$1 = 3 \cdot 6 - 17 \cdot 1$

$a \; x \quad b \; y = -1$

$a^{-1} = 6 \mod 17$

$b^{-1} = -1 \mod 3$

Modular exponentiation

$a^b \mod n$

$\cancel{2^{303} \mod 3 \equiv 2^6 \mod 3}$

incorrect

$2^{303} \mod 3$

$\parallel$

$2^{303 \bmod 2} \mod 3$

$2^1 \mod 3$

Euler's totient theorem

for $a, n : a < n, \; \gcd(a,n) = 1$

$$\underline{a^{\varphi(n)} \equiv 1 \mod n}$$

$\varphi(n)$ - Eulers theta function

- Euler's totient funcion

$=$ number of $a < n, \; \gcd(a,n) = 1$

$\varphi(p) = p - 1$ for prime

$$\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n) \cdot \frac{d}{\varphi(d)}$$

where $\gcd(m,n) = d$

$$\varphi(p \cdot q) = \varphi(p) \varphi(q) \cdot \frac{1}{\varphi(1)}$$

$$= (p-1)(q-1)$$

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

$$a^b \bmod n = a^{b \bmod \varphi(n)} \bmod n$$

$$\underbrace{\overbrace{a\cdot a\cdot a \cdots}^{\varphi(n)\ =\ 1}\ \overbrace{\cdots}^{\varphi(n)\ =\ 1}\ \overbrace{\cdots}^{\varphi(n)\ =\ 1}\ \cdots\ =1\ \cdots\ \overbrace{a}^{b \bmod \varphi(n)}}_{b\text{-times}}$$

For p prime you recover Fermat's little theorem

$\forall\ a < p$

$$a^{p-1} \equiv 1 \mod p \qquad p-1 = \varphi(n)$$

$$a^b = a^{b \bmod p-1} \mod p$$

Important problems in assymetric cryptography

Factorization easy: Given a,b find c, s.t. $c = a\cdot b$

hard: Given c find a,b s.t. $c = a\cdot b$

Essentially trying all divisors between 2 and $\sqrt{c}$ is optimal

$\boxed{2048\text{-bits}}$ $\quad c \approx 2^{2048} \quad$ $\boxed{\sqrt{c} \approx 2^{1024}}$

Number of protons in the Universe $\approx 2^{300}$

Discrete logarithm

easy: Given a, b and n calculate c

$$c = a^b \mod n$$

hard: Given c, a, n calculate b

$$c = a^b \mod n \qquad b \in \{1, \dots, \varphi(n)\}$$

$$b = \log_a c \mod n$$

RSA encryption

Private: p, q ~ two large primes

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

Public: e, $n = p \cdot q$

Encryption: of message $w < n$

$$c = w^e \mod n$$

Decrypt: of ciphertext c:

$$w = c^d \mod n$$
$$= (w^e)^d \mod n$$
$$= w^{e \cdot d} \mod n$$

$= w^{e \cdot d} \mod n$

$= w^{e \cdot d \mod \varphi(n)} \mod n$ **!** iff $\gcd(w, n) = 1$

$= w^1 \mod n$

$= w$

What can an adversary without the knowledge of p, q, d do?

1.) Factorize $n \Rightarrow$ know p and q $\Rightarrow \varphi(n) = (p-1)(q-1)$

hard

$\Downarrow$

calculate $d \equiv e^{-1} \mod n$

2.) Can I find an algorithm to calculate $\varphi(n)$ efficiently?

Than we can factor like this:

$p \cdot q = n$

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

Example

$p \cdot q = 1363$

$(p-1)(q-1) = 1288$

$pq - p - q + 1 = 1288$

$1363 - p - q + 1 = 1288$

$q = 76 - p$

$p \cdot (76 - p) = 1363$

$p^2 - 76p + 1363 = 0$

3.) $e, n \rightarrow d$ RSA problem

input output

This is hard (we do not know an efficient algorithm)

This is hard (we do not know an efficient algorithm)
but probably (we do not know an efficient reduction)
not as hard as factoring

Other RSA weaknesses

for known (w, c) pairs ($w^e = c \bmod n$) with modulus n
I can find other pairs

(w^2, c^2) is also a valid pair!

$$(w^2)^e = w^{2e} = w^e \cdot w^e = c \cdot c = c^2 \mod n$$

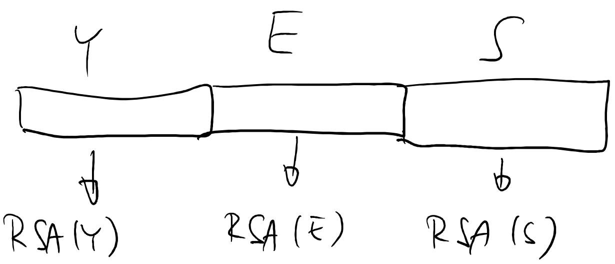
Whenever you see c^2 in a channel you know how to decrypt it without factoring n.

(w^n, c^n) is a valid pair

if (w_1, c_1) and (w_2, c_2) are valid pairs
then $(w_1 \cdot w_2, c_1 \cdot c_2)$ is a valid pair

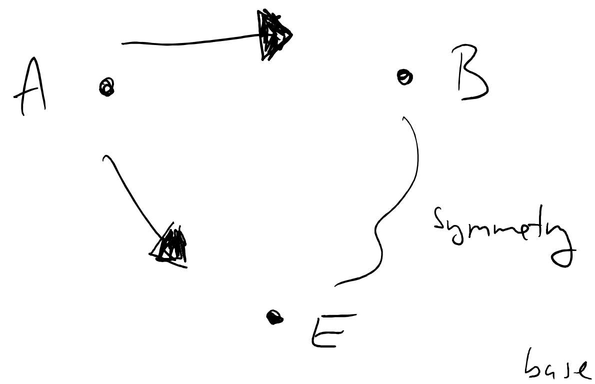
$$(w_1 \cdot w_2)^e = w_1^e \cdot w_2^e = c_1 \cdot c_2 \mod n$$

TEXT → ASCII



THIS IS MONOALPHABETIC SUBSTITUTION
(NOT SECURE)

DIFFIE HELLMAN key distribution



Prerequisities:

p - a large prime

q - a number of large order in $\mathbb{Z}_p^*$

base of logarithm

$1, 1^2, 1^3$. order $(1) = 1$

$\begin{array}{ccc} 1, & 1^2, & 1^3 \\ \| & \| & \| \\ 1 & 1 & 1 \end{array}$

$\begin{array}{ccc} -1, & -1^2, & -1^3 \\ \| & \| & \| \\ -1 & 1 & -1 \end{array}$ . . . order $(-1) = 2$.

$\log_{-1} c \bmod n =$

$= 2$

$A \rightarrow B$

$A = q^{x} \bmod p$

chosen at random

$B \rightarrow A$

$B = q^{y} \bmod p$

A calculates

$k = B^x \bmod p = q^{xy} \bmod p$

$\|$

B calculates

$k = A^y \bmod p = q^{yx} \bmod p$

What can the adversary do?

1.) calculate $\left.\begin{array}{l} x = \log_q A \\ y = \log_q B \end{array}\right| \rightarrow$ hard

then $k = q^{xy} \bmod p$

2.) given g^x and g^y and p calculate $g^{xy} \bmod p$

$\boxed{DH1}$ → believed to be hard.

KNAPSACK CRYPTOSYSTEM

NP-complete problem

given

$(x_1, \ldots, x_n)$ $x_i \in \mathbb{Z}_p^*$ (for large prime p)

and a constant C

find $b \in \{0,1\}^n$, such that

$\vec{x} \cdot \vec{b} = C \mod p$

encryption

$w \in \{0,1\}^n$

$X = (x_1, \ldots, x_n), p$ are a public key

$C = w \cdot X \mod p$

$C, (X, p) \Rightarrow$ decryption is an instance of subset sum problem which is generally np-hard

---

We want to transform a general instance of subset sum

We want to transform a general instance of subset sum problem to an easy one.

if X' is a superincreasing vector $\forall i \quad x_i > \sum_{j<i} x_j$

$x'_1, x'_2, x'_3 \ldots$

$x'_2 > x'_1$

$x'_3 > x'_1 + x'_2$

$x'_4 > x'_1 + x'_2 + x'_3$

(X', c) is an easy instance

Private: $u, X = (x_1, \ldots, x_n)$ superincreasing

$p > 2x_n > \sum_i x_i$

Public: $p, X' = u \cdot X \mod p$

this is not superincreasing anymore

encryption of w: $c = w \cdot X$

decryption of c: calculate $u^{-1} \cdot c = c' \mod p$

and solve subset sum instance (c', X)

$c' = w \cdot X \qquad / u.$

$c' \cdot u = w \cdot u \cdot X \mod p$

$C = w \cdot X'$