Tutorial V

17 October 2019 09:59

ASYMETRIC CRYPTOGRAPHY

→ RSA

→ Diffie-Hellman

→ Knapsack cryptosystem

Basics of number theory

$(\mathbb{Z}_n)$ ~ set of all remainders after division by n

$(\mathbb{Z}_n^*)$ ~ $\mathbb{Z}_n \setminus \{0\}$

$(\mathbb{Z}_n, +)$ ~ group

$(\mathbb{Z}_n^*, \cdot)$ ~ group if n is a prime

~ ring otherwise

$(\mathbb{Z}_n^*, \cdot)$ careful about division

$a^{-1} \bmod n$ exists if and only if $\gcd(a,n)=1$

a

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

$a \cdot b^{-1} \mod n$

How to calculate this?

$\gcd(b,n) = 1$

Example

$\frac{5}{3} \mod 7 \neq 1{,}66\ldots$

$5 \cdot 3^{-1} \mod 7$

$\quad\; \shortparallel$

$5 \cdot 5 \mod 7 \quad$ (because $3^{-1} = 5$ ie $3 \cdot 5 \equiv 1 \mod 7$)

How to find inverse mod n? Euclid's algorithm

+

Bezóut's identity

Euclid's algorithm 21

is an algorithm to find $\gcd(a,b)$ of arbitrary $a,b \in \mathbb{N}$

$\gcd(96, 18) = 6$

$96 : 18 = 5$ remainder $\boxed{6}$ → the last non-zero remainder

$18 : 6 = 3$ remainder 0

Find 3^{-1} mod 17 ie find b, such that $3b \equiv 1 \mod 17$

$\gcd(3, 17) = 1$

$17 : 3 = 5$ rm. 2

$3 : 2 = 1$ rm $\boxed{1}$

$2 : 1 = 2$ rm 0

Bezout's identity

for a, b : $\gcd(a,b) = 1$

$\exists x, y$ s.t. $\boxed{ax + by = 1}$

$\Downarrow$

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

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

($by \equiv 0$)

$ax \equiv 1 \mod b$

$1 = 3 - 2 \cdot 1$

$2 = 17 - 3 \cdot 5$

$1 = 3 - 2 \cdot 1$

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

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

$1 = 3 \cdot 6 - 17$

$1 = \underset{a}{3} \cdot \underset{x}{\underline{6}} - \underset{b}{17} \cdot \underset{y}{1} \quad / \mod 17$

$1 \equiv 3 \cdot 6 - 0 \quad \mod 17$

$a = 3$
$b = 17$
$x = 6$
$y = -1$

Modular exponentiation

$a^b \mod n$

$\cancel{2^{303} \mod 3 \stackrel{?}{=} 2^{101} \mod 3}$

this is wrong

$$2^{303} \equiv 2^{303 \mod \phi(3)} \mod 3$$

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

$$\equiv 2 \mod 3$$

Euler's totient theorem

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

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

$\phi(n) \to$ theta function

$\to$ Euler's totient function

= the number of $a < n$

s.t. $\gcd(a,n) = 1$

Number of coprimes to n

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

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

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

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

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

for p, q prime

$$a^b \equiv a^{b \bmod \phi(n)} \mod n \qquad \text{iff } \gcd(a,n)=1$$

$$a^b = \underbrace{\overbrace{a\cdot a\cdot a \cdots a}^{\phi(n)\text{ times} \;=\; 1} \mid \overbrace{\qquad\quad}^{\phi(n) \;=\; 1} \mid \overbrace{\qquad\quad}^{\phi(n) \;=\; 1} \mid \overbrace{a \ldots a}^{b \bmod \phi(n)}}_{b}$$

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

For primes Euler's totient theorem is Fermat's little theorem

$a < p$

$$a^{p-1} \equiv 1 \mod p$$

Important problems for assymetric cryptography

They are easy to calculate but their inverses are hard.

Factorization

has an efficient algorithm

Easy problem – multiplication: given a, b calculate $c = a \cdot b$

Hard problem – given c, find a, b, such that $a \cdot b = c$

Essentially trying to divide with all numbers between 2 and $\sqrt{c}$ is optimal.

$\boxed{\text{2048-bit numbers}}$ $\qquad c \approx 2^{2048}$ $\qquad \boxed{\sqrt{c} \approx 2^{1024}}$

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

Discrete logarithm

Easy: given a, b, n find $c \equiv a^b \mod n$

Hard: given a, c, n find b: $c \equiv a^b \mod n$

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

essentially try all $b \in \{1, \dots, \phi(n)\}$

RSA encryption

public key $= e, n$

private key $= d, p, q$

$d \equiv e^{-1} \mod \phi(n)$

$n = p q$, p, q are large primes

($n = p \cdot q$, p, q are large primes)

encrypt message $w < n$: $c = w^e \mod n$

decryption of cryptotext $c < n$: $w = c^d \mod n$

$$= (w^e)^d \mod n$$

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

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

$$= w^1 \mod n$$

$$= w$$

$\underline{\gcd(w,n) = 1}$

Why is this considered secure?

How to find w without knowing p, q, d

1.) $\underline{\text{Factorize } n}$ $\Rightarrow$ know $p \cdot q$ $\Rightarrow$ $\phi(n) = (p-1)(q-1)$

hard

$$d = e^{-1} \mod \phi(n)$$

2.) From $\boxed{e, n, c}$ calculate $\underline{\phi(n)}$ without factoring.

Algorithm to do this is as hard as factoring.

Algorithm to do this ...

$p \cdot q = 1363$

$(p-1)(q-1) = 1288 = \phi(n)$

$p(76-p) = 1363$

$pq - p - q + 1 = 1288$

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

$\Downarrow$

$1363 - p - q + 1 = 1288$

$75 - p + 1 = q$

$q = 76 - p$

3.) $\underset{p\ q}{\overset{\text{input}}{e, n}} \longrightarrow \overset{\text{output}}{d}$

Having e, d and n is not enough to factor n efficiently

This is hard (we don't know how to do this efficiently) but not as hard as factoring = RSA problem

Other RSA weaknesses

a valid pair c and modulus n $\quad w^e = c \bmod n$

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

(w, c) decrypts as, (w^2, c^2), ..., (w^n, c^n)

(w_1, c_1) (w_2, c_2) mod n

$\Downarrow$

$c_1 c_2$ decrypts as $w_1 \cdot w_2$

Homework

$c_1 c_2$ decrypts as $w_1 \cdot w_2$

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

the same messages have the same encryptions!

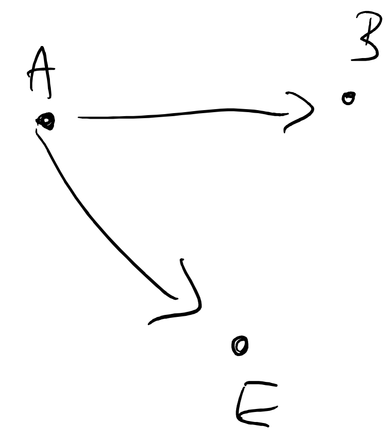
TEXT ⇝ ASCII

Y	E	S
ASCII	ASCII	ASCII
$e(Y)$	$e(E)$	$e(S)$

$\Downarrow$

monoalphabetic substitution

DIFFIE-HELLMAN ⇝ key distribution



Prerequisitio:

p – large prime

$q < p$ with large order $\mathbb{Z}_p^*$ — $\{1, \dots, p-1\}, \cdot \mod p$

⇝ basis of logarithm

order$(1) = 1$
$$\begin{array}{ccccc} 1, & 1^2, & 1^3, & \dots, & 1^{p-1} \\ \downarrow & \downarrow & \downarrow & & \downarrow \\ 1 & 1 & 1 & & 1 \end{array}$$

order$(-1) = 2$
$$\begin{array}{ccccc} -1, & -1^2, & -1^3, & \dots & -1^{p-1} \\ \downarrow & \downarrow & \downarrow & & \\ -1 & 1 & -1 & \dots & \end{array}$$

$A \to B$ $\quad A = q^x \mod p$ (x chosen randomly)

$A \to B \quad A = g^{x} \bmod p$

$B \to A \quad B = g^{y} \bmod p$

A calculates $\quad k = B^{x} = (g^{y})^{x} = g^{xy} \bmod p$

B calculates $\quad k = A^{y} = (g^{x})^{y} = g^{xy} \bmod p$

What can adversary do?

1.) calculate $\quad \log_g A \bmod p = x$

$\qquad\qquad\quad \log_g B \bmod p = y$

this is hard (g has high order)!

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

2.) given g^{x} and g^{y} calculate g^{xy} directly

$\boxed{DH1}$ → also believed to be hard.

Knapsack cryptosystem

Knapsack problem (Subset sum)

given a vector of numbers

$(x_1, \ldots, x_n)$ and a constant C,

find a bit vector $b_1, \dots, b_n$, such that
$\vec{x} \cdot \vec{b} = c$

Public key: p - a large prime

$$x = (x_1, \dots, x_n) \quad x_i \in \mathbb{Z}_p^*$$

Encryption of message

$$b \in \{0,1\}^n \qquad \underline{c = \vec{x} \cdot \vec{b}}$$

Easy instance of Knapsack problem:

Superincreasing vectors:

$$X = (x_1, \dots, x_n) \qquad x_i > \sum_{j<i} x_j$$

find b, such that $b \cdot x = c$

```
for i = n to i = 1
if x_i < c   b_n = 1
c = c - x_i
```

Private information

Superincreasing vector X' and $u < p$, such that

general vector

$\boxed{X} = u \cdot X'$ mod p

Solving $c = b \circ X$ is equivalent to solving

$u^{-1} c = b \cdot u^{-1} X$ mod p

$$c = \sum_{\substack{i \in [b] \\ \text{s.t. } b_i = 1}} x_i$$

$$u^{-1} c = \sum_{\substack{i \in [b] \\ \text{s.t. } b_i = 1}} u^{-1} x_i = \sum_{\substack{i \in [b] \\ \text{s.t. } b_i = 1}} x'_i$$

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

Solve Knapsack with c' and superincreasing X'.