Hybride Encryption

2.18 - 2.45

① Key generation

* p, q are 2 large prime numbers and p, q are same size.
* p, q are cannot be equal.
* Then compute n = p * q phi = (p-1) (q-1)
* select random integer "e" (1<e< phi) such that gcd(e, phi) = 1
* e chose randomly from range of 1 - phi
* g = gcd (e, phi) (great common divisa)
* d = Extend equlid algorithm to compute another unique integer "d" (1 < d < phi) such that e, d % = 1 (mod, phi)

d = mutiplicative Inverse (e, phi)

* Then return pubic keys and private keys.

n	e	d	n
↑	↑ Private	↑	↑
1 Public key	2 public key	1 Public key	2 private key

② Encryption

* First generating AES symmentric key
  ↑ using same key for encryption and decryption
* encode means string transform into byte (code) form what type we gaimen. usually we using 'utf-8' method
* Then plain text encrypt using AES algorithm.

~~increase the cy dete~~

* encrypt function (RSA) use for encrypt the key

③ Decryption

* First decrypt the encrypt key
* Then using that key and AES, decrypt the ~~encr~~ cipher text.

Process

n, e → 1st user Public key

n, e → 2nd user Public key

d, n → 1st user private key

d, n → 2^{n} user Private key

* key key create using secreate module. This key is the key which use for the encrypt plain text
* key Then key encrypt using RSA algorithm.

n → p * q

e → select the random integer for (1 - phi) range

g = gcd (e, phi) → 'g' is use for the execute the while loop then create e.

d = multiplicative Inverse (e, phi)

phi = (p-1) (q-1)

Why $a \equiv b \pmod n$ then

i.e a and b have the same "remainder" when they are divided by n

(3) $n \mid (a-b)$

↑ divides

$(a-b)$ is a multiple of n

(1) $a = kn + b$

n

Ex: $10 \equiv 14 \pmod 4$

(1)

$10 / 4 = 2$ R (2)

$14 / 4 = 3$ R (2)

$10 \equiv -2 \pmod 4$

2

(2) $10 = k \times 4 + 14$

$10 = -1 \times 4 + 14$

$k = -1$

(3) $4 \mid (10 - 14) \equiv -1$

* key size means number of bits in a key used by cryptography algorithm.

key size = 2^7, 2^8 128 - 256

is prime]

9 conditions

1. 0, 1, and negative numbers are not prim
2. low primes not use for this process (key bit size)
3. num in low primes return true
4. num % prime == 0 return false

→ Then recursive call miller ~~rabin~~. rabin function

Euclid extend algorithm

* This use to find the coprimes

* Coprime numbers is a set of numbers or integers which have as their highest common factors. Coprime numbers are also known as relatively prime or mutually prime numbers.

15 8

Factors 1, 3, 5, 15 1, 2, 4, 8

Common factor is 1 ∴ 15, 8 are coprime.

Modulo multiplicative inverse

* when get the a, x integers are $ax \equiv 1$ & with respect to the module (n). In the standard notation of modular arithmatic this congress is written as,

$$ax \equiv 1 \pmod{n}$$

$$ax \,\%\, n = 1$$

Example = $3 \pmod{26}$ - we want v such that

$$3v = 1 - 26w$$

$$3v \equiv 1 \pmod{26}$$

$3v + 26w = 1$

Euclid algorithm

$26 = 3 \times 8 + 2$

$3 = 1 \times 2 + 1$

$3 \times 9 - 26 \times 1 = 3v + 26w$

$9 \times 3 = 1 + 1 \times 26 \equiv 1 \pmod{26}$

$v = 9 \quad w = -1$

9 is a multiplicative inverse $\pmod{26}$

$2 = 26 - 8 \times 3$

$1 = 3 - 2 \times 1$

$= 3 - 1 \times (26 - 8 \times 3)$

$= 3 \times 9 - 26 \times 1$

...ller Rabin algorithm

This algorithm is use to testing the given number is prime or not.

* f(n) → Prime or → Composite

$a \equiv 0 \pmod n$ one of this means it is a divisor of n.

$1 < a < n-1$ Random

$a^{n-1} \equiv 1 \pmod n$

$a^{n-1} - 1 \equiv 0 \pmod n$

$\left(a^{\frac{n-1}{2}} - 1\right)\left(a^{\frac{n-1}{2}} + 1\right) \equiv 0 \pmod n$

$(a^2 - y^2) \longrightarrow \left(a^{\frac{n-1}{4}} - 1\right)\left(a^{\frac{n-1}{4}} + 1\right)\left(a^{\frac{n-1}{2}} + 1\right) \equiv 0 \pmod n$

$\left(a^{\frac{n-1}{2^k}} - 1\right)\left(a^{\frac{n-1}{2^k}} + 1\right) \cdots \cdots \left(a^{\frac{n-1}{2}} + 1\right) \equiv 0 \pmod n$

256 - 512

If n divise each any factor of these it is probely prime.

Final stage of the create the public key and private key

+ n = p * q

+ phi = (p-1) (q-1)

e → randomly from (1 - phi) range

gcd g = gcd (e, phi)

d = multiplicative Inverse ((e, Phi)

n e d ← p x q

↑ ↑ ↑

p x q randomly (1 - Phi)

↑

multiplicative (e, phi)

g = gcd(e, Phi)

highest common diveser of the e and phi

$$d \overset{?}{\equiv} e \bmod (Phi)$$

Encrypt. function	Decryption functi.
+ key encrypt using e ord (char) ** e % n	+ key decrypt using d char (chr + *d) % n