

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 $\gcd(e, \phi) = 1$

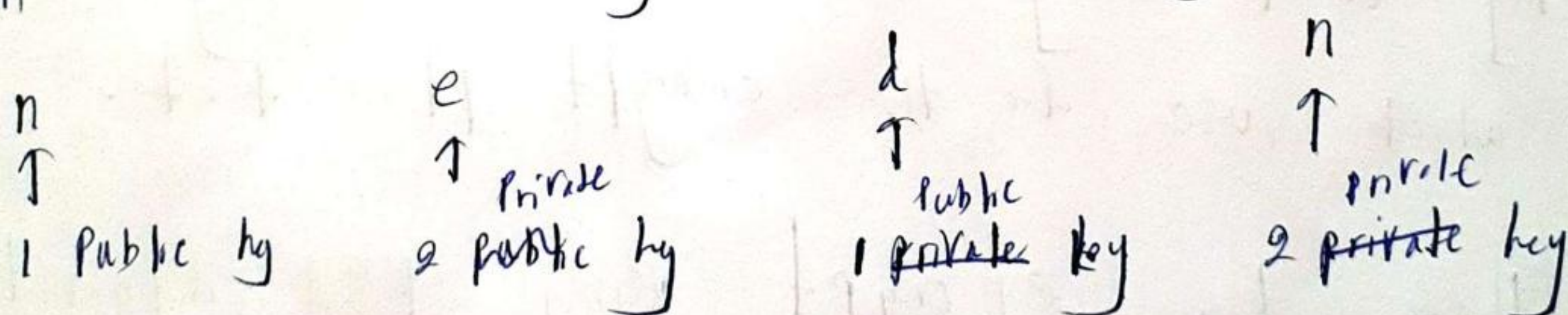
* e chose randomly ^{from} in range of $1 - \phi$

* $g = \gcd(e, \phi)$ (great common divisa)

* ~~Use~~ Extent equal algorithm to compute another unique integer "d" ($1 < d < \phi$) such that $e, d \equiv 1 \pmod{\phi}$

$d = \text{multiplicative Inverse}(e, \phi)$

* Then return public keys and private keys.



② Encryption

* First generating AES symmetric key

using same key for encryption and decryption

* encode means string transform into byte code for what type we given. usually we using 'utf-8' method.

* Then plain text encrypt using AES algorithm.

increase the cy better

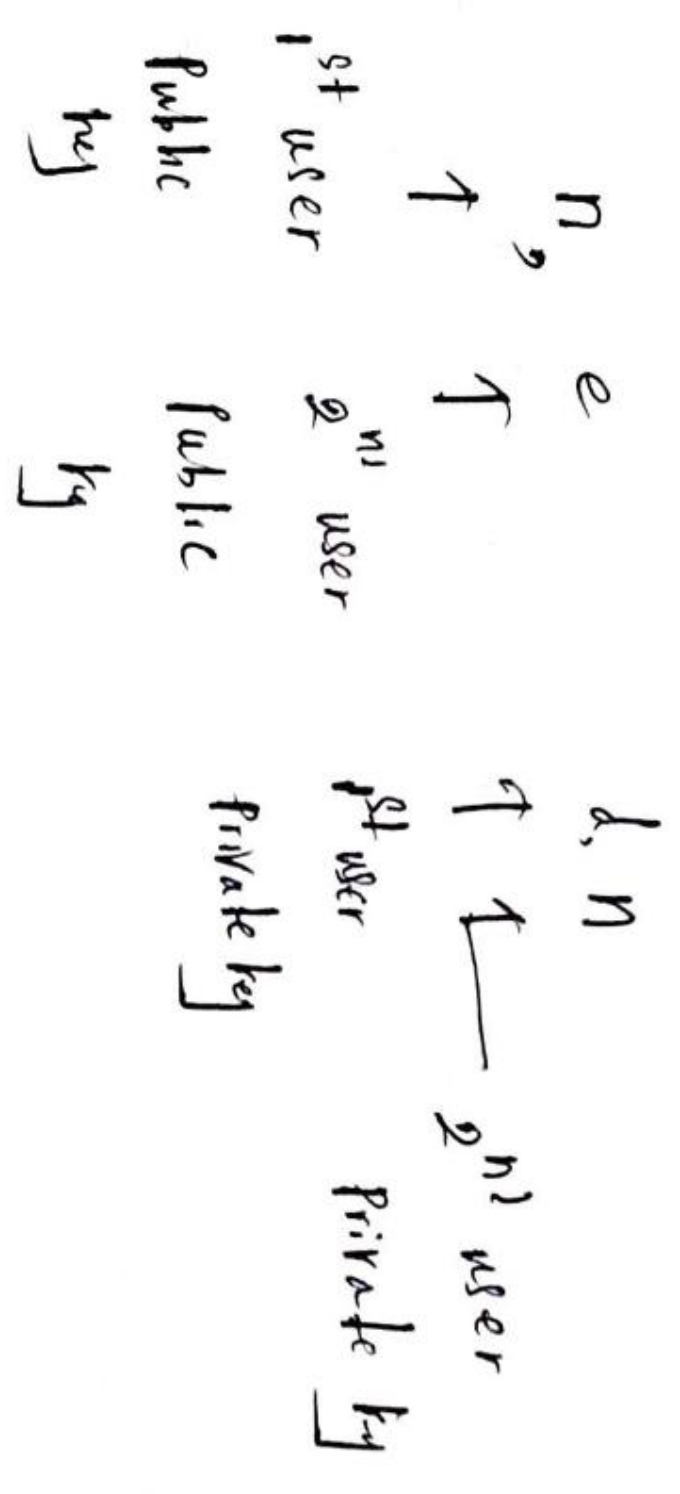
* encrypt a function (RSA) use for encrypt the key

*

③ Decryption

• First decrypt the encrypt key
• They using that key and AES, decrypt the ~~data~~ cipher text.

Process



* key key create using generate prime. This key is the key which use for the encrypt plain text.

* key Then key encrypt using RSA algorithm.

n ↓
 e ↓

select the
max integer
for $(1/2 - phi)$ range

$g = gcd(e, phi) \rightarrow g$ is use for the execute the while loop that create e .

$d = \text{multiplicative inverse}(e, phi)$

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

key ^{is} $a \equiv b \pmod{n}$ then

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

$$\textcircled{1} a = kn + b$$

$$\textcircled{2} n \mid (a-b)$$

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

Ex: $10 \equiv 14 \pmod{4}$

$$\textcircled{1} \begin{aligned} 10/4 &= 2 \text{ R } 2 \\ 14/4 &= 3 \text{ R } 2 \end{aligned}$$

$$10 \equiv -2 \pmod{4}$$

even note

$$\textcircled{2} \begin{aligned} 10 &= k \times 4 + 14 \\ 10 &= -1 \times 4 + 14 \\ k &= -1 \end{aligned}$$

$$\textcircled{3} 4 \mid (10-14) = -4$$

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

$$\text{key size} = 2^1, 2^8$$

128-256

is prime 1

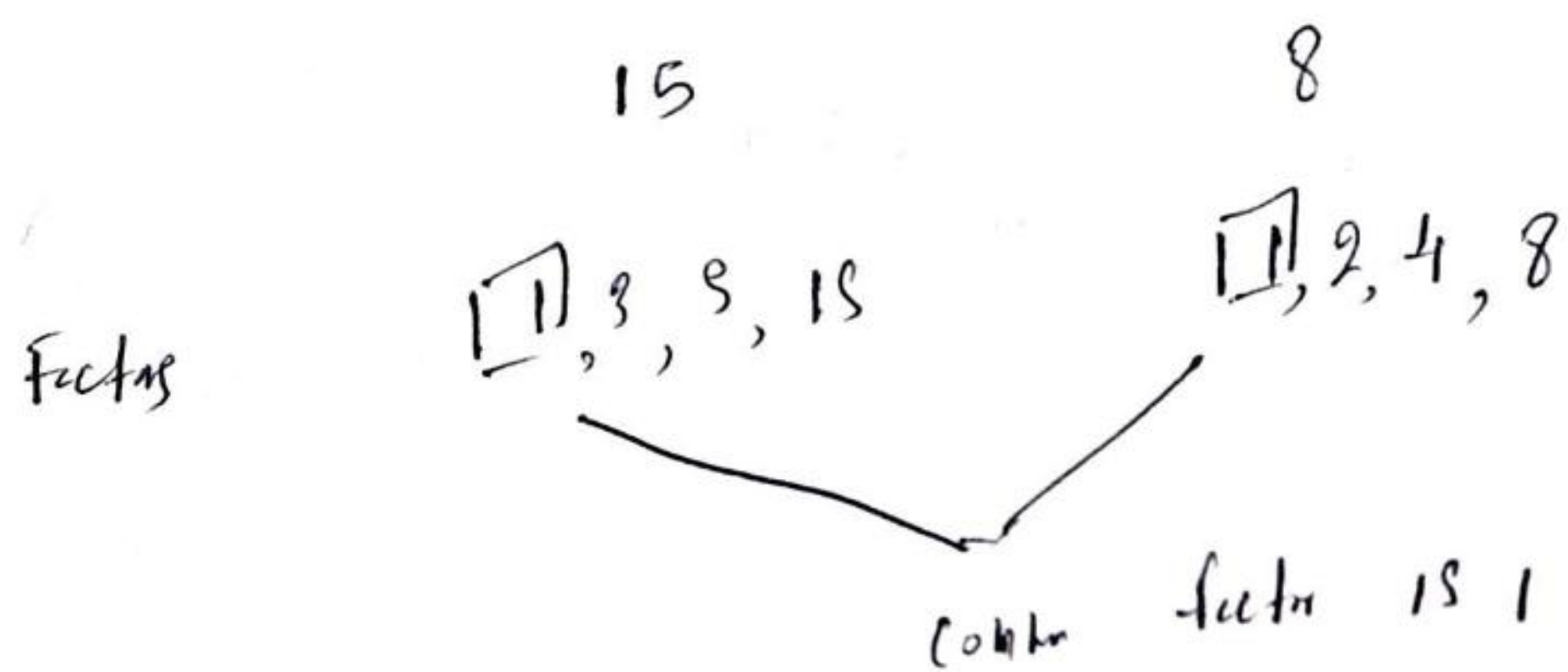
9 conditions

1. 0, 1, and negative numbers are not prime
 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 return ^{call} miller robin function

Euclid's algorithm

* This is used to find the coprimes

* Coprime numbers are a set of numbers or integers which have 1 as their ^{highest} common factor. Coprime numbers are also known as relatively prime or mutually prime numbers.



Modulo multiplicative inverse

* When given the a, x integers are $ax \equiv 1 \pmod{m}$ with respect to the module m . In the standard notation of modular arithmetic this congruence is written as,

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

$$\frac{ax}{m} \quad ax \% m = 1$$

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

$$3x = 1 - 26w$$

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

$3x + 26w = 1$

Using Euclid's algorithm

$$26 = 3 \times 8 + 2$$

$$2 = 1 \times 2 + 0$$

$$2 = 26 - 3 \times 8$$

$$1 = 2 - 1 \times 2$$

$$= 2 - 1 \times (26 - 3 \times 8)$$

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

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

~~$$3x + 26w = 1$$~~

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

$$x = 9 \quad w = -1$$

9 is a multiplicative inverse of 3

(mod 26)

Miller Rabin algorithm

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

$a \equiv 0 \pmod{n}$ one of this means n is a, liaser or of n .

* $f(n)$
 \nearrow Prime
 or
 \searrow Composite

$1 < a < n-1$ Random

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

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

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

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

$$\vdots$$

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

ps6-512

If n divide each any factor of these, it is probely Prime.

Final stage of the create the public key and private keys

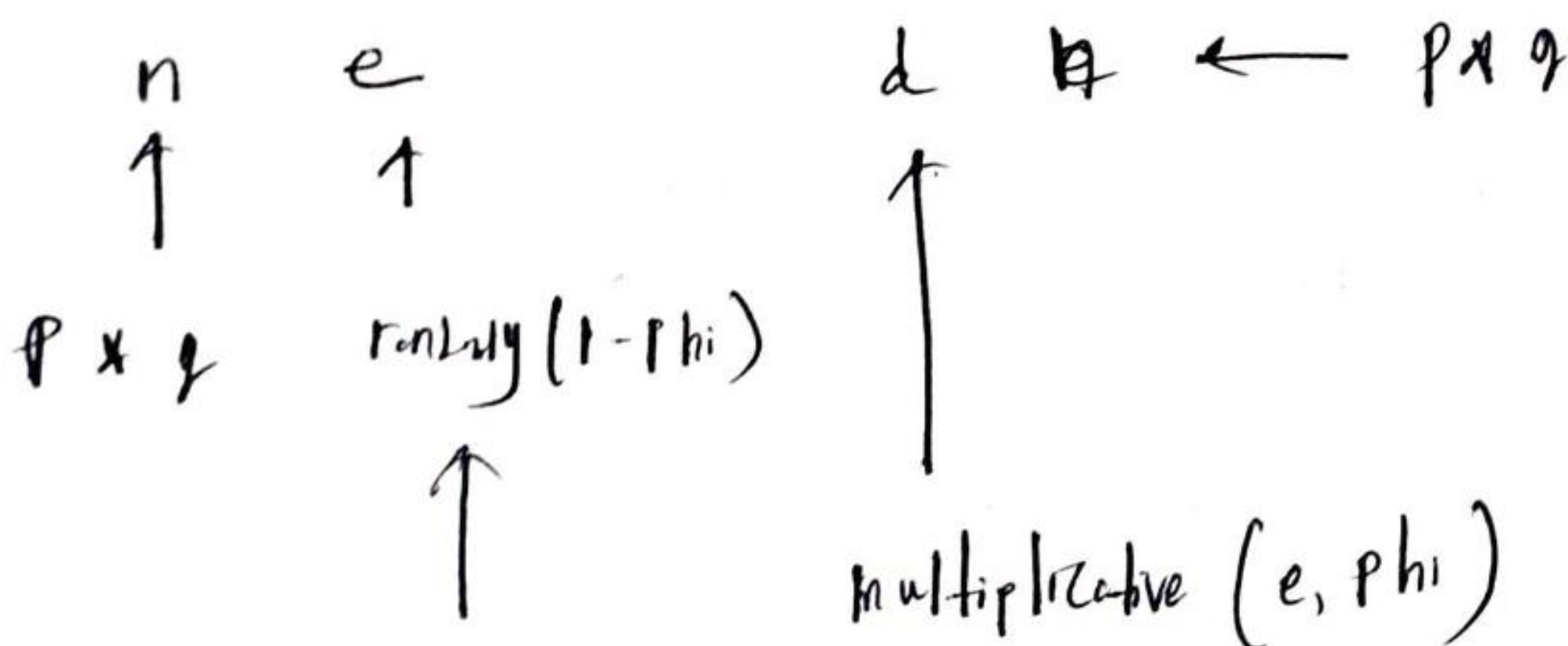
$$n = p \times q$$

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

$e \rightarrow$ randomly from $(1, \phi)$ range

$$g = \gcd(e, \phi)$$

$$d = \text{multiplicativeInverse}(e, \phi)$$



$$g = \gcd(e, \phi)$$

highest common divisor that of the e and ϕ

$$d \equiv e^{-1} \pmod{\phi}$$

Encryption function

* key encrypt using e
 $enc(char) = char^e \pmod{n}$

Decryption function

* key decrypt using d
 $dec(char) = char^d \pmod{n}$