lecture 11

public-key Cryptography

hybrid Setting :

public-key Enc scheme

Alice | Bob

(K_{pub}, K_{pri})

$\xleftarrow{K_{pub}}$

Choose random symmetric key

$y_1 = e_{pub}(K)$: plaintext $\xrightarrow[\text{ciphertext}]{y_1}$ $K = d_{pri}(y_1)$

key of symmetric Enc scheme

Symmetric key Enc

$y_2 = AES_K(x) \xrightarrow{y_2} x = AES^{-1}_K(y_2)$

agree on this "K" select

---

* one-way function

easy to compute $y = f(x)$

hard " " $x = f^{-1}(y)$

① Factoring Integers (RSA): given a composite integer finds its prime factors

② Discrete Logarithm (DH, Elgamal, DSA, ...)

given $a, y, m \longrightarrow$ find x s.t. $a^x = y \pmod m$

$\log_a y = x \pmod m$

③ Elliptic Curve (EC) → generalization of the 2nd problem

Euler's phi Function

set of "m" integers $\{0, 1, \dots, m-1\}$

How many numbers in the set are relatively prime to "m"?

Example $\{0, 1, 2, 3, 4, 5\} \longrightarrow m=6$

$\gcd(0,6)=6$

〃 $(1,6) = 1 \longleftarrow$ ①

〃 $(2,6) = 2$

〃 $(3,6) = 3$

〃 $(4,6) = 2$

〃 $(5,6) = 1 \longleftarrow$ ⑤

$\phi(6) = 2$

Example $\{0, 1, 2, 3, 4\} \longrightarrow m=5$

$\gcd(0,5) = 5$

〃 $(1,5) = 1 \longleftarrow$

〃 $(2,5) = 1 \longleftarrow$

〃 $(3,5) = 1 \longleftarrow$

〃 $(4,5) = 1 \longleftarrow$

$\phi(5) = 4$

$$m = p_1^{e_1} \cdot p_2^{e_2} \cdots p_n^{e_n} \longrightarrow \phi(m) = \prod_{i=1}^{n} \left(p_i^{e_i} - p_i^{(e_i-1)} \right)$$

$$m=6 \longrightarrow 6 = 2^1 \times 3^1 \longrightarrow \begin{matrix} p_1=2 & e_1=1 \\ p_2=3 & e_2=1 \\ n=2 & \end{matrix} \longrightarrow \phi(6) = \underbrace{(2^1 - 2^0)}_{i=1} \times \underbrace{(3^1 - 3^0)}_{i=2}$$

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

$$m=5 \longrightarrow 5 = 5^1 \longrightarrow \begin{matrix} p_1=5 & e_1=1 \\ n=1 & \end{matrix} \longrightarrow \phi(5) = \underbrace{(5^1 - 5^0)}_{i=1} = 5-1 = 4$$

$\boxed{m = \text{prime \#} \cdot \text{prime \#}}$

$e_i = 1 \longrightarrow \phi(m) = (p-1)(q-1)$

$\boxed{\text{Example}}$ $m = 899$

$$\phi(899) = (\underset{\text{prime}}{29} - 1)(\underset{\text{numbers}}{31} - 1) = 28 \times 30 = 840$$

$\boxed{\text{Euler's Theorem}}$ generalization of Fermat's little Theorem

given two relatively prime integers "a" & "m"

$$a^{\phi(m)} \equiv 1 \pmod{m}$$

$\boxed{\text{Example}}$ $m = 12$, $a = 5$ $\longrightarrow$ $\gcd(12, 5) = 1$

$$\begin{array}{r|l} 12 & 2 \\ 6 & 2 \\ 3 & 3 \\ 1 & \end{array}$$

e_1 e_2

$12 = \underset{p_1}{2}^{2} \cdot \underset{p_2}{3}^{1}$

$$\phi(12) = (2^2 - 2^1) \cdot (3^1 - 3^0) = (4-2)(3-1) = 4$$

verify theorem $\longrightarrow$ $5^{\phi(12)} \overset{?}{\equiv} 1 \pmod{12}$

$$5^4 \equiv 625 \equiv 1 \pmod{12}$$

1976 → problem came out

1977 → Ronald Rivest, Adi Shamir, Leonard Adleman

key Generation – RSA

1. Generate 2 large random primes p, q (same size)
2. Compute $n = pq$ an $\phi = (p-1)(q-1)$
3. public key → select random integer "e" $(1 < e < \phi)$ s.t. $\gcd(e, \phi) = 1$
4. private key → Use Ex. E Algo to compute another unique inter "d" $(1 < d < \phi)$ such that $e \cdot d \equiv 1 \pmod{\phi}$
5. Transmit (n, e) as public values & "d" is private key

RSA Alg.

Enc
1. obtain (n, e)
2. Message space $[0, n-1]$
3. Compute $c = m^e \pmod{n}$ $\boxed{\text{ciphertext}}$
4. send "c" to the other party

Dec
5. $m = c^d \pmod{n}$

Modular Exponentiation → lec02 (slid 23)

RSA Example

Alice

$x = \textcircled{4}$

Bob

$p = 3 \quad , \quad q = 11$

$n = p \cdot q = 3 \times 11 = 33$

$\phi(n) = (3-1)(11-1) = 20$

$e = 3 \curvearrowright \gcd(3, 20) = 1$

$d = e^{-1} \equiv 7 \pmod{20}$

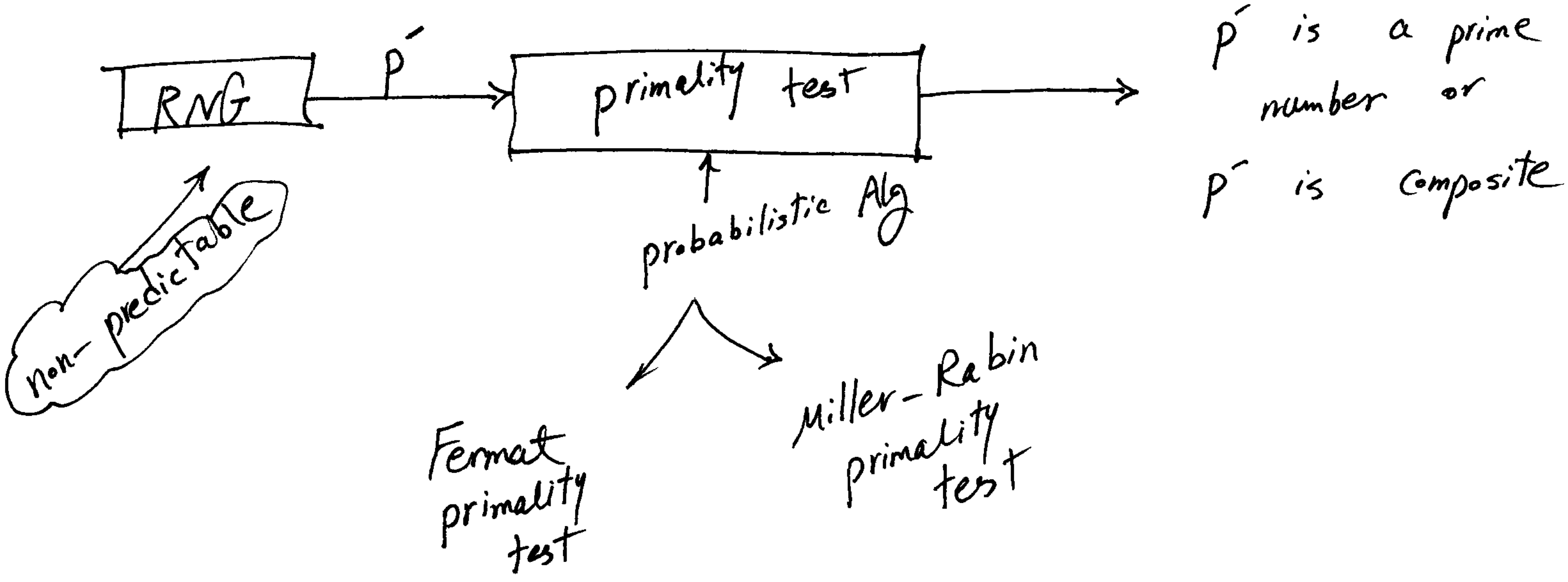
$$\xleftarrow{K_{pub}(\overset{\overset{n}{\downarrow}}{33}, \overset{\overset{key}{\downarrow}}{3})}$$

$y = 4^{3} \pmod{33} \equiv 31$

$$\xrightarrow[\text{cipher}]{31} \quad 31^{7} \equiv \textcircled{4} \pmod{33}$$

+ simple math for Enc & Dec

- if numbers are large, even by using fast algorithms for "Modular Exponentiation", it's very time consuming



Modular Exponentiation

Repeated square-and-multiply alg for exponentiation in $\mathbb{Z}_n$

Input: $a \in \mathbb{Z}_n$, $0 \leq k < n \longrightarrow$ binary representation $k = \sum_{i=0}^{t} k_i 2^i$

Output: $a^k \bmod n$

```
1. Set $b \leftarrow 1$. if $k=0$, Return (b)
2. Set $A \leftarrow a$
3. if $k_0 = 1$ then    $\overset{\text{variable}}{b} \leftarrow \overset{\text{Value}}{a}$
4. For $i = 1 \rightsquigarrow t$
    4.1  $A \leftarrow A^2 \pmod{n}$
    4.2  if $k_i = 1$ then    $b \leftarrow A \cdot b \pmod{n}$
5. return(b)
```

---

a^{26}

$26 = 2 \times 13 + 0$
$13 = 2 \times 6 + 1$
$6 = 2 \times 3 + 0$
$3 = 2 \times 1 + 1$
$2 \times 0 + 1$

$26 = (\underset{k_4}{1}\,\underset{k_3}{1}\,\underset{k_2}{0}\,\underset{k_1}{1}\,\underset{k_0}{0})_2$

1. $b = 1$, $k = 26 \neq 0 \rightarrow NO$
2. $A = a$
3. $k_0 = 0 \neq 1 \rightarrow NO$
4. $\underset{1}{i} \rightsquigarrow 4$

$\boxed{i=1}$ $A = A^2 \pmod{n}$

if $k_1 = 1$ ✓ $\longrightarrow$ $b = A^2 \cdot 1 = A^2 \pmod{n}$

$\boxed{i=2}$ $A = (A^2)^2 = A^4 \pmod{n}$

if $k_2 = 0 \neq 1$ X

$\boxed{i=3}$ $A = (A^4)^2 = A^8 \pmod{n}$

if $k_3 = 1$ ✓ $\longrightarrow$ $b = A^8 \cdot A^2 = A^{10} \pmod{n}$

$\boxed{i=4}$ $A = (A^8)^2 = A^{16} \pmod{n}$

if $k_4 = 1$ ✓ $\longrightarrow$ $b = A^{16} \cdot A^{10} = A^{26} \pmod{n}$

5. return $A^{26} \pmod{n}$

a^7

$7 = 2 \times 3 + 1$
$3 = 2 \times 1 + 1$
$1 = 2 \times 0 + 1$

$(111)_2$ → k_2 k_1 k_0

```
1.  b ← 1   if k=0   X
2.  A = a
3.  if $k_0 = 1$ ⟶ b = a
4.  i = 1 ~ 2
      [i=1]
        $A = A^2 \pmod n$
        if $k_1 = 1$ ✓ ⟶ $b = A^2 . a = A^3 \pmod n$
      [i=2]
        $A = (A^2)^2 = A^4 \pmod n$
        if $k_2 = 1$ ✓ ⟶ $b = A^4 . A^3 = A^7 \pmod n$
5.  return ($A^7 \pmod n$)
```

Fermat primality test alg.

Input: an odd integer $n \geqslant 3$ & security parameter $t \geqslant 1$ ↓ # of iteration of your loop/try

Output: answer n is "prime" or "composite"

```
1. For i = 1 ~ t do
     1.1  chose random integer a   $2 \leqslant a \leqslant \overbrace{n}^{\text{previous odd number}} - 2$
     1.2  Compute $r = a^{n-1} \pmod n$ using S-&-M algorithm
     1.3  If $r \neq 1$ then return "Composite"
2. Return("prime")
```

Fermat's Th

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

$\boxed{1}$ Execution

$n=25$ $t=3$

$i=1 \rightarrow 3$

$\boxed{i=1}$ $2 \leq a=9 \leq 23$

Compute $r=9^{24} \pmod{25} = 11$

if $\underbrace{r\neq 1}_{\checkmark}$ then retun "Composit"

$\boxed{2}$ Execution

$n=25$ $t=3$

$i=1 \rightarrow 3$

$\boxed{i=1}$ $2 \leq a=7 \leq 23$

Compute $r=7^{24} \pmod{25} = 1$

if $r\neq 1$ X

$\boxed{i=2}$ $2 \leq a=4 \leq 23$

Compute $r=4^{24} \pmod{25} = 6$

if $\underbrace{r\neq 1}_{\checkmark}$ return "Composite"

$\boxed{3}$ Execution

$n=25$ $t=2$

$i=1 \rightarrow 3$

$\boxed{i=1}$ $2 \leq a=7 \leq 23$

$r=7^{24} \pmod{25} = 1$

if $r\neq 1$ X not true

$\boxed{i=2}$ $2 \leq a=18 \leq 23$

$r=18^{24} \pmod{25} = 1$

if $r\neq 1$ X not true

return("prime")

Here, we set $t=2$ & after 2 iterations, the algorithm returns "prime" which is NOT correct because 25 is not prime. That is why it doesn't work properly all the time & it depends on your security parameter.

Miller_Rabin probabilistic primality test

Input: an odd integer $n \geq 3$ and $t \geqslant 1$ (security parameter)

Output: "Is $\underline{n}$ prime?"

1. write $\underbrace{n-1}_{even} = \underbrace{\underbrace{2^s}_{even} * \underbrace{r}_{odd}}_{even}$ s.t. "r" is odd →

$$n=25$$

$$n-1=24=2^{\overset{s}{3}} \times \underset{r}{3}$$

$$\begin{array}{r|l} 24 & 2 \\ 12 & 2 \\ 6 & 2 \\ 3 & 3 \\ 1 & \end{array}$$

```
2. For $i = 1 \rightsquigarrow t$ do                                   Loop
     2.1 Choose a random integer $a$, $2 \leqslant a \leqslant \overset{\text{previous odd number}}{\overbrace{n-2}}$
     2.2 Compute $y = a^r \pmod{n}$ using $\underline{\text{S-8-M algo.}}$
     2.3 If $y \neq 1$ and $y \neq n-1$ do
           $j = 1$
           while $j \leqslant s-1$ and $y \neq n-1$ do           Loop
               Compute $y \leftarrow y^2 \pmod{n}$
               if $y = 1$ then return "Composit"
               $j \leftarrow j+1$
           If $y \neq n-1$ then return "Composite"
3. Return "prime".
```