

# RSA Cryptosystem



密碼學與應用  
海洋大學資訊工程系  
丁培毅

# Naïve Public Key System

- Encryption and decryption algorithm are not the same
- Public/private key pair: private key is related to public key but can not be easily derived from public key
- Illustrating example:

$$m \in Z_{11}^*$$

$$m * 1 = m \pmod{11}$$

$$m * \underbrace{8}_{\text{encryption}} * \underbrace{8^{-1}}_{\text{decryption}} = m \pmod{11}$$

$8$  is the public key  
 $m * 8$  is the ciphertext  
 $8^{-1}$  is the private key (if nobody can derive this from the public key, then this system is secure)

# Knapsack (Subset Sum) PKC

- Merkel and Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks," IT-24, 1978
  - a good application of an **NP problem** on designing public key cryptosystem; no longer secure
- Super-increasing sequence:**  
 $\{a_1, a_2, \dots, a_n\}$  such that  $a_i > \sum_{j=0}^{i-1} a_j$  e.g. 1, 3, 5, 10, 20, 40
- Note:** 1. Given a number  $c$ , finding a subset  $\{a_j\}$  s.t.  $c = \sum_j a_j$  is an easy problem, e.g.  $48 = 40 + 5 + 3$
- 2. Sum of every subset  $S$ ,  $\sum_{j \in S} a_j < 2 \cdot a_M$  where  $a_M = \max_{j \in S} \{a_j\}$
- 3. Every possible subset sum is unique  
 pf: given  $x$ , assume  $x = \sum_{j \in S} a_j = \sum_{j \in T} a_j$ , where  $S \neq T$ , assume  $\max_{j \in S} \{a_j\} \neq \max_{j \in T} \{a_j\}$  ....

# Knapsack (Subset Sum) PKC

- choose a number  $b$  in  $Z_p^*$ , e.g.  $p = 101$ ,  $b = 23$ , and convert the **super-increasing sequence** to a **normal knapsack sequence**  
 $B = \{b_1, b_2, \dots, b_n\}$  where  $b_i \equiv a_i \cdot b \pmod{p}$   
 e.g.  $A = \{1, 3, 5, 10, 20, 40\} \Rightarrow B = \{23, 69, 14, 28, 56, 11\}$
- Since  $\gcd(b, p) = 1$ , this conversion is **invertible**, i.e.  
 $a_i \equiv b_i \cdot b^{-1} \pmod{p}$   
 e.g.  $b^{-1} \equiv 22 \pmod{101}$  such that  $b \cdot b^{-1} \equiv 1 \pmod{p}$
- Given a number  $d$ , finding a subset  $\{b_j\} \subseteq B$  s.t.  
 $d = \sum_j b_j \pmod{p}$   
 is an NP-complete problem, e.g.  $94 = 11 + 14 + 69$

## Knapsack (Subset Sum) PKC

### Encryption:

- \* **public key:** normal knapsack seq.  $B = \{23, 69, 14, 28, 56, 11\}$
- \* message  $m$ ,  $0 \leq m < 2^6$ , e.g.  $(60)_{10} = (111100)_2$
- \* sum up the corresponding elements of '1' bits, e.g.  
 $23 + 69 + 14 + 28 = 134$  is the ciphertext

### Decryption:

- \* **private key:**  $b^{-1} = 22$ ,  $p = 101$ ,  $A = \{1, 3, 5, 10, 20, 40\}$
- \* calculate  $134 * 22 \bmod 101 = 19$
- \* use the corresponding super-increasing knapsack seq.  $A = \{1, 3, 5, 10, 20, 40\}$  to decrypt as follows:
  - \*  $19 < 40$ , mark a '0'
  - \*  $19 < 20$ , mark a '0'
  - \*  $19 \geq 10$ , mark a '1' and subtract 10 from 19
  - \*  $9 \geq 5$ , mark a '1' and subtract 5 from 9
  - \*  $4 \geq 3$ , mark a '1' and subtract 3 from 4
- \* recovered message is  $(111100)_2 = (60)_{10}$

5

## Knapsack (Subset Sum) PKC

### Why does it work?

let the plaintext be  $(111100)_2$

ciphertext  $c = b_1 + b_2 + b_3 + b_4$

$$\equiv a_1 b + a_2 b + a_3 b + a_4 b \pmod{p}$$

decryption:  $c b^{-1} \pmod{p} \equiv a_1 + a_2 + a_3 + a_4 \pmod{p}$

is a subset sum problem of a super-increasing sequence

6

## RSA and Rabin

two important cryptosystems based on the difficulty of **integer factoring** (an NP problem) are introduced as follows:

### RSA's underlying problem

Solving e-th root modulo  $n$  is difficult

RSA function

$$y \equiv x^e \pmod{n}$$

### Rabin's underlying problem

Solving square root modulo  $n$  is difficult

Rabin function

$$y \equiv x^2 \pmod{n}$$

both functions are candidates for **trapdoor one way function**

7

## RSA and Rabin Function

Solving e-th root of  $y$  modulo  $n$  is difficult!!!

$$y \equiv x^e \pmod{n}, \text{ where } \gcd(e, \phi(n)) = 1$$

Why don't we take  $(e^{-1})$ -th power of  $y$ ?

$$\text{where } e^{-1} \cdot e \equiv 1 \pmod{\phi(n)}$$

$$\text{e.g. } n = 11 \cdot 13 = 143, e = 7$$

$$\phi(n) = 10 \cdot 12 = 120, e^{-1} = 103$$

Trouble: How do we know  $\phi(n)$ ?

Solving square root of  $y$  modulo  $n$  is difficult!!!

$$y \equiv x^2 \pmod{n}$$

Why don't we take  $(2^{-1})$ -th power of  $y$ ?

$$\text{where } 2^{-1} \cdot 2 \equiv 1 \pmod{\phi(n)}$$

$$\text{e.g. } n = 11 \cdot 13 = 143$$

$$\phi(n) = 10 \cdot 12 = 120, \gcd(2, \phi(n)) = 2$$

Remember solving square root of  $y$  modulo a prime number  $p$  is very easy

Trouble:  $d \cdot 2 \equiv 1 \pmod{\phi(n)}$  has no solution

8

## RSA Public Key Cryptosystem

- ✧ R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM, pp.120-126, 1978
- ✧ Based on the *Integer Factorization* problem
- ✧ Choose two large prime numbers:  $p, q$  (keep them secret!!)
- ✧ Calculate the modulus  $n = p \cdot q$  (make it public)
- ✧ Calculate  $\Phi(n) = (p-1) \cdot (q-1)$  (keep it secret)
- ✧ Select a random integer such that  $e < \Phi$  and  $\gcd(e, \Phi) = 1$
- ✧ Calculate the unique integer  $d$  such that  $e \cdot d \equiv 1 \pmod{\Phi}$
- ✧ **Public key:**  $(n, e)$                       **Private key:**  $d$

9

## RSA Encryption & Decryption

- ✧ Alice wants to encrypt a message  $m$  for Bob
- ✧ Alice obtains Bob's authentic public key  $(n, e)$
- ✧ Alice represents the message as an integer  $m$  in the interval  $[0, n - 1]$
- ✧ Alice computes the modular exponentiation
 
$$c \equiv m^e \pmod{n}$$
- ✧ Alice sends the ciphertext  $c$  to Bob
- ✧ Bob decrypts  $c$  with his private key  $(n, d)$  by computing the modular exponentiation
 
$$\hat{m} \equiv c^d \pmod{n}$$

10

## RSA Encryption & Decryption

- ✧ Why does RSA work?    **Is this really a problem???**

★ **Fact 1:**  $e \cdot d \equiv 1 \pmod{\Phi} \Rightarrow e \cdot d = 1 + k \Phi$

★ **Fact 2:**  $\forall m, \gcd(m, n) = 1, m^\Phi \equiv 1 \pmod{n}$   
(by Euler's theorem)

★ **From Fact 2:**  $\forall m, \gcd(m, n) = 1,$

$$c^d \equiv m^{ed} \equiv m^{1+k\Phi} \equiv m^{1+k(p-1)(q-1)} \equiv m \pmod{n}$$

- note: 1. This only proves that for all  $m$  that are not multiples of  $p$  or  $q$  can be recovered after RSA encryption and decryption.
2. For those  $m$  that are multiples of  $p$  or  $q$ , the Euler's theorem simply does not hold because  $p^\Phi \equiv 0 \pmod{p}$  and  $p^\Phi \equiv 1 \pmod{q}$  which means that  $p^\Phi \not\equiv 1 \pmod{n}$  from CRT.

11

## RSA Encryption & Decryption

- ✧ Why does RSA work?

★ **Fact 1:**  $e \cdot d \equiv 1 \pmod{\Phi} \Rightarrow e \cdot d = 1 + k \Phi$

★ **Fact 2:**  $\forall m, \gcd(m, p) = 1, m^{p-1} \equiv 1 \pmod{p}$   
(by Fermat's Little theorem)

★ **From Fact 2:**  $\forall m, \gcd(m, p) = 1$

note: this equation is trivially true when  $m = kp$

$$m^{1+k(p-1)(q-1)} \equiv m \pmod{p}$$

★ **From Fact 2:**  $\forall m, \gcd(m, q) = 1$

note: this equation is trivially true when  $m = kq$

$$m^{1+k(p-1)(q-1)} \equiv m \pmod{q}$$

★ **From CRT:**  $\forall m,$

$$c^d \equiv m^{ed} \equiv m^{1+k\Phi} \equiv m^{1+k(p-1)(q-1)} \equiv m \pmod{n}$$

12

## RSA Function is a Permutation

- ◇ RSA function is a permutation: (1-1 and onto, bijective)
  - ◇ Goal: “ $\forall x_1, x_2 \in \mathbb{Z}_n$  if  $x_1^e \equiv x_2^e \pmod{n}$  then  $x_1 = x_2$ ”
  - $\forall x \neq r \cdot p, x^{p-1} \equiv 1 \pmod{p}, \forall x \neq s \cdot q, x^{q-1} \equiv 1 \pmod{q}$
  - $\Rightarrow \forall k, \forall x \neq r \cdot p, x^{k\phi(n)} \equiv 1 \pmod{p}, \forall x \neq s \cdot q, x^{k\phi(n)} \equiv 1 \pmod{q}$
  - CRT  $\Rightarrow \forall k, \forall x, x^{k\phi(n)+1} \equiv x \pmod{p}, x^{k\phi(n)+1} \equiv x \pmod{q}$
  - $\Rightarrow \forall k, \forall x, x^{k\phi(n)+1} \equiv x \pmod{n}$
  - \*  $\gcd(e, \phi(n)) = 1 \Rightarrow$  inverse of  $e \pmod{\phi(n)}$  exists
  - $\Rightarrow$  let  $d$  be the inverse s.t.  $e \cdot d \equiv 1 \pmod{\phi(n)}$
  - \*  $\forall x_1, x_2 \in \mathbb{Z}_n$  if  $x_1^e \equiv x_2^e \pmod{n}$
  - $\Rightarrow (x_1^e)^d \equiv (x_2^e)^d \pmod{n}$
  - $\Rightarrow (x_1)^{e \cdot d} \equiv (x_2)^{e \cdot d} \pmod{n}$
  - $\Rightarrow x_1 \equiv x_2 \pmod{n}$
- Note: Euler Thm is valid only when  $x \in \mathbb{Z}_n^*$

13

## RSA Cryptosystem

- ◇ Most popular PKC in practice
- ◇ Tens of dedicated crypto-processors are specifically designed to perform modular multiplication in a very efficient way.
- ◇ **Disadvantage:** long key length, complex key generation scheme, deterministic encryption
- ◇ For acceptable level of security in commercial applications, 1024-bit (300 digits) keys are used. For a symmetric key system with comparable security, about 100 bits keys are used.
- ◇ In constrained devices such as smart cards, cellular phones and PDAs, it is hard to store, communicate keys or handle operations involving large integers

14

## Matlab examples

- ◇ rsatest.m
  - \* `maple('p := nextprime(1897345789)')`
  - \* `maple('q := nextprime(278478934897)')`
  - \* `maple('n := p*q');`
  - \* `maple('x := 101');`
  - \* `maple('e := nextprime(12345678)')`
  - \* `maple('d := e&^(-1) mod ((p-1)*(q-1))')`
  - \* `maple('y := x&^(e) mod n')`
  - \* `maple('xp := y&^(d) mod n')` extended Euclidean algo.
- Very likely to be relatively prime with  $(p-1)(q-1)$

15

## Rabin Cryptosystem (1/3)

- ◇ M.O. Rabin, “Digitalized Signatures and Public-key Functions As Intractable As Factorization”, Tech. Rep. LCS/TR212, MIT, 1979
- ◇ Choose two large prime numbers:  $p, q$  (keep them secret!!)
- ◇ Calculate the modulus  $n = p \cdot q$  (make it public)
- ◇ **Public Key**  $n$
- ◇ **Private Key**  $p, q$

16

## Rabin Cryptosystem (2/3)

- ✧ Alice want to encrypt a message  $m$  (with some fixed format) for Bob
- ✧ Alice obtains Bob's authentic public key  $n$
- ✧ Alice represents the message as an integer  $m$  in the interval  $[0, n - 1]$
- ✧ Alice computes the modular square
 
$$c \equiv m^2 \pmod{n}$$
- ✧ Alice sends the ciphertext  $c$  to Bob
- ✧ Bob decrypts  $c$  using his private key  $p$  and  $q$
- ✧ Bob computes the four square roots  $\pm m_1, \pm m_2$  using CRT, one of them satisfying the fixed message format is the recovered message

17

## Rabin Cryptosystem (3/3)

- ✧ The range of the Rabin function is not the whole set of  $Z_n^*$  (compare with RSA).
  - \* The range covers all the quadratic residues. (for a prime modulus, the number of quadratic residues in  $Z_p^*$  is  $(p-1)/2$ ; for a composite integer  $n=p \cdot q$ , the number of quadratic residues in  $Z_n^*$  is  $(p-1)(q-1)/4$ )
  - \* In order to let the Rabin function have inverse, it is necessary to make the Rabin function a permutation, i.e. 1-1 and onto. Therefore, the number of elements in the domain of the Rabin function should also be  $(p-1)(q-1)/4$  for  $n=p \cdot q$ . There are 4 possible numbers with their square equal to  $y$ , and we have to make 3 of them illegal.

18

## Number of Quadratic Residues

- ✧ For a prime modulus  $p$ : number of  $QR_p$ 's in  $Z_p^*$  is  $(p-1)/2$   
 pf: find a primitive  $g$ , at least  $\{g^2, g^4, \dots, g^{p-1}\}$  are  $QR_p$ 's  
 assume there are  $(p+1)/2$  QRs,  
 since there are exactly two square roots of a QR modulo  $p$   
 there are  $p+1$  square roots for these  $(p+1)/2$  QRs, i.e. there must be at least two pairs of square roots are the same (pigeon-hole), i.e. two out of these  $(p+1)/2$  QRs are the same, contradiction
- ✧ For a composite modulus  $p \cdot q$ : number of  $QR_n$ 's in  $Z_{p \cdot q}^*$  is  $(p-1)(q-1)/4$   
 pf: find a common primitive in  $Z_p^*$  and  $Z_q^*$   $g$ , at least  $\{g^2, g^4, \dots, g^{p-1}, \dots, g^{q-1}, \dots, g^{\lambda(n)}\}$  are  $QR_n$ 's, where  $\lambda(n) = \text{lcm}(p-1, q-1)$  can be as large as  $(p-1)(q-1)/2$ , this set has  $(p-1)(q-1)/4$  distinct elements  
 assume there are  $(p-1)(q-1)/4 + 1$   $QR_n$ 's in  $Z_n^*$ , since there are four square roots of a QR modulo  $p \cdot q$ , these  $QR_n$ 's have  $(p-1)(q-1) + 4$  square roots in total. There must be some repeated elements in this  $QR_n$ , therefore, there are at most  $(p-1)(q-1)/4$   $QR_n$ 's in  $Z_n^*$

19

## Matlab examples

- ```

✧ maple('p:= nextprime(189734535789)') % 189734535811 = 4 k + 3
✧ maple('p mod 4')
✧ maple('q:= nextprime(27847815934897)') % 27847815934931 = 4 k + 3
✧ maple('q mod 4')
✧ maple('n:=p*q');
✧ maple('x:=070411111422141711030000') % text2int('helloworld')
✧ maple('c:= x&^2 mod n')

```
- 
- ```

✧ maple('c1:= c mod p')
✧ maple('r1:= c1&^((p+1)/4) mod p') % maple('r1&^2 mod p')

✧ maple('c2:= c mod q')
✧ maple('r2:= c2&^((q+1)/4) mod q') % maple('r2&^2 mod q')

✧ maple('m1:= chrem([r1, r2], [p, q])') % 3704440302544264662351219
✧ maple('m2:= chrem([-r1, r2], [p, q])') % 70411111422141711030000
✧ maple('m3:= chrem([r1, -r2], [p, q])') % 5213281318342160554284041
✧ maple('m4:= chrem([-r1, -r2], [p, q])') % 1579252127220037602962822

```

20

## Security of the RSA Function

- ❖ **Break RSA** means ‘inverting RSA function without knowing the trapdoor’  $y \equiv x^e \pmod{n}$
- ❖ Factor the modulus  $\Rightarrow$  Break RSA
  - \* If we can factor the modulus, we can break RSA
  - \* If we can break RSA, we don't know whether we can factor the modulus... **open problem** (with negative evidences)
- ❖ Factor the modulus  $\Leftrightarrow$  Calculate private key d
  - \* If we can factor the modulus, we can calculate the private exponent d (the trapdoor information).
  - \* If we have the private exponent d, we can factor the modulus.

will be illustrated later after factorization 21

## Security of Rabin Function

- ❖ Security of Rabin function *is equivalent to* integer factoring
- ❖ inverting ‘ $y \equiv f(x) \equiv x^2 \pmod{n}$ ’ without knowing p and q  $\Leftrightarrow$  factoring n
  - \*  $\Leftarrow$ 
    - if you can factor  $n = p \cdot q$  in polynomial time
    - you can solve  $y \equiv x_1^2 \pmod{p}$  and  $y \equiv x_2^2 \pmod{q}$  easily
    - using CRT you can find x which is  $f^{-1}(y)$
  - \*  $\Rightarrow$ 
    - given a quadratic residue y if you can find the four square roots  $\pm x_1$  and  $\pm x_2$  for y in polynomial time
    - you can factor n by trying  $\gcd(x_1 - x_2, n)$  and  $\gcd(x_1 + x_2, n)$

22

## Basic Factoring Principle (1/4)

- ❖ Let n be an integer and suppose there exist integers x and y with  $x^2 \equiv y^2 \pmod{n}$ , but  $x \not\equiv \pm y \pmod{n}$ . Then ❶ n is composite, ❷ both  $\gcd(x-y, n)$  and  $\gcd(x+y, n)$  are nontrivial factors of n.

Proof:

let  $d = \gcd(x-y, n)$ .

Case 1: assume  $d = n \Rightarrow x \equiv y \pmod{n}$  contradiction

Case 2: assume d is 1 (the trivial factor)

$$x^2 \equiv y^2 \pmod{n} \Rightarrow x^2 - y^2 = (x-y)(x+y) = k \cdot n$$

$$d=1 \text{ means } \gcd(x-y, n)=1 \Rightarrow$$

$$n \mid x+y \Rightarrow x \equiv -y \pmod{n} \text{ contradiction}$$

Case 1 and 2 implies that  $1 < d < n$

i.e. d must be a nontrivial factor of n

23

## Basic Factoring Principle (2/4)

- ❖  $x^2 \equiv y^2 \pmod{p}$  implies  $x \equiv \pm y \pmod{p}$  since  $p \mid (x+y)(x-y)$  implies  $p \mid (x+y)$  or  $p \mid (x-y)$ ,  
i.e.  $x \equiv -y \pmod{p}$  or  $x \equiv y \pmod{p}$
- ❖  $x^2 \equiv y^2 \pmod{n}$   
 $pq \mid (x+y)(x-y)$  implies the following 4 possibilities
  1.  $pq \mid (x+y)$  i.e.  $x \equiv -y \pmod{n}$
  2.  $pq \mid (x-y)$  i.e.  $x \equiv y \pmod{n}$
  3.  $p \mid (x+y)$  and  $q \mid (x-y)$  i.e.  $x \equiv -y \pmod{p}$  and  $x \equiv y \pmod{q}$
  4.  $q \mid (x+y)$  and  $p \mid (x-y)$  i.e.  $x \equiv -y \pmod{q}$  and  $x \equiv y \pmod{p}$
  - \* Case 1 and case 2 are useless for factorization
  - \* Case 3 leads to the factorization of n, i.e.  $\gcd(x+y, n) = p$  and  $\gcd(x-y, n) = q$
  - \* Case 4 leads to the factorization of n, i.e.  $\gcd(x+y, n) = q$  and  $\gcd(x-y, n) = p$

24

## Basic Factoring Principle (3/4)

- ✧ This principle is used in *almost all factoring algorithms*.
- ✧ Why is it working?
  - \* take  $n = p \cdot q$  ( $p$  and  $q$  are prime) for example
  - \*  $x^2 \equiv y^2 \pmod{n}$  implies  $x^2 \equiv y^2 \pmod{p}$  and  $x^2 \equiv y^2 \pmod{q}$
  - \* we know 'x  $\equiv \pm y \pmod{p}$  are the only solution to  $x^2 \equiv y^2 \pmod{p}$ ' and 'x  $\equiv \pm y \pmod{q}$  are the only solution to  $x^2 \equiv y^2 \pmod{q}$ '
  - \* therefore, from CRT we know  $x^2 \equiv y^2 \pmod{n}$  has four solutions,
    - \*  $x \equiv y \pmod{p}$  and  $x \equiv y \pmod{q} \Rightarrow x \equiv y \pmod{n}$
    - \*  $x \equiv -y \pmod{p}$  and  $x \equiv -y \pmod{q} \Rightarrow x \equiv -y \pmod{n}$
    - \*  $x \equiv y \pmod{p}$  and  $x \equiv -y \pmod{q} \Rightarrow x \equiv z \pmod{n}$
    - \*  $x \equiv -y \pmod{p}$  and  $x \equiv y \pmod{q} \Rightarrow x \equiv -z \pmod{n}$
  - \* as long as we have  $z$  (where  $z \not\equiv \pm y$ ), we can factor  $n$  into  $\gcd(y-z, n)$  and  $\gcd(y+z, n)$

25

## Basic Factoring Principle (4/4)

- ✧ Ex: Consider the roots of  $4 \pmod{35}$ , i.e. solving  $x$  from  $x^2 \equiv 4 \pmod{35}$ 
  - \* try to take square root of both sides, we find  $x = \pm 2$  or  $\pm 12$
  - \* i.e.  $12^2 \equiv 2^2 \pmod{35}$ , but  $12 \not\equiv \pm 2 \pmod{35}$
  - \* therefore 35 is composite
  - \*  $\gcd(12-2, 35) = 5$  is a nontrivial factor of 35
  - \*  $\gcd(12+2, 35) = 7$  is a nontrivial factor of 35

26

## Miller-Rabin Test

Is  $n$  a composite number?

- ✧ Let  $n > 1$  be odd, write  $n-1 = 2^k \cdot m$  with  $m$  being odd
- ✧ Choose a random integer  $a$  with  $1 < a < n-1$
- ✧ Compute  $b_0 \equiv a^m \pmod{n}$ 
  - if  $b_0 \equiv \pm 1 \pmod{n}$ , stop,  $n$  is probably prime
- ✧ Compute  $b_1 \equiv b_0^2 \pmod{n}$ 
  - if  $b_1 \equiv 1 \pmod{n}$ , stop,  $\gcd(b_0-1, n)$  is a factor of  $n$
  - if  $b_1 \equiv -1 \pmod{n}$ , stop,  $n$  is probably prime
- ✧ Compute  $b_2 \equiv b_1^2 \pmod{n}$ 
  - .....
- ✧ Compute  $b_{k-1} \equiv b_{k-2}^2 \pmod{n}$ 
  - if  $b_{k-1} \equiv 1 \pmod{n}$ , stop,  $\gcd(b_{k-2}-1, n)$  is a factor of  $n$
  - if  $b_{k-1} \equiv -1 \pmod{n}$ , stop,  $n$  is probably prime
- ✧ Compute  $b_k \equiv b_{k-1}^2 \pmod{n}$ 
  - if  $b_k \equiv 1 \pmod{n}$ , stop,  $\gcd(b_{k-1}-1, n)$  is a factor of  $n$
  - otherwise  $n$  is composite (Fermat Little Thm,  $b_k \equiv a^{n-1} \pmod{n}$ )

n will pass Fermat test n is called pseudo prime with respect to base a

27

## Miller-Rabin Test Illustrated

$$\begin{aligned}
 n-1 &= 2^k \cdot m \\
 b_0 &\equiv a^m \pmod{n} \\
 b_1 &\equiv a^{2 \cdot m} \pmod{n} \\
 &\dots \\
 b_k &\equiv a^{2^k \cdot m} \equiv a^{n-1} \pmod{n}
 \end{aligned}$$

③ ① and ② are not true,  $b_i \equiv -1 \pmod{n}$ ,  $i=1,2,\dots,k$  all subsequent  $b_j \equiv 1 \pmod{n}$ , there is no chance to use Basic Factoring Principle, **abort**

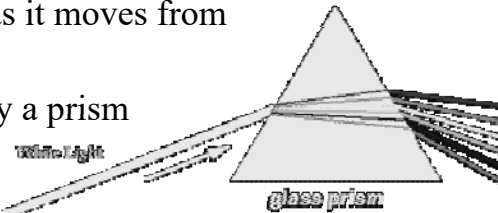
Consider 4 possible cases:

- ①  $b_0 \equiv \pm 1 \pmod{n}$  all  $b_i \equiv 1 \pmod{n}$ ,  $i=1,2,\dots,k$  there is no chance to use Basic Factoring Principle, **abort**
- ② ① is not true,  $b_{i-1} \not\equiv \pm 1 \pmod{n}$  and  $b_i \equiv 1 \pmod{n}$ ,  $i=1,2,\dots,k$  Basic Factoring Principle applied, **composite**
- ④ ①, ②, and ③ are not true,  $b_k \equiv a^{n-1} \pmod{n}$  if  $b_k \not\equiv 1 \pmod{n}$   $n$  is **composite** since if  $n$  is prime,  $b_k \equiv 1 \pmod{n}$  ( $b_k \equiv 1 \pmod{n}$  is covered by ②)

28

## Uncoordinated Behaviors

- Speed of light changes as it moves from one medium to another, e.g., refraction caused by a prism



- 趣味競賽: 兩人三腳, 同心協力, ...
- Squaring a number modulo a composite number (product of different prime numbers)

	$2^2$	$2^3$	$2^4$	$2^5$	$2^6$	$2^7$	$2^8$
mod 11	4	8	5	10	9	7	3
mod 13	4	8	3	6	12	11	9

29

## When/How does Basic Factoring Principle work in M-R test?

- When:

\* explicitly:  $b_{i-1} \not\equiv \pm 1 \pmod{n}$  and  $b_i \equiv b_{i-1}^2 \pmod{n}$

If  $n$  is not prime, sometimes  $b_k \equiv a^{n-1} \pmod{n}$  but often  $b_k \equiv a^{\text{r}\phi(n)} \pmod{n}$  as in universal exponent factoring

- How:

\* implicitly: let  $p \mid n$  and  $q \mid n$  ( $p, q$  be two factors of  $n$ )

$$b_{i-1}^2 \equiv 1 \pmod{p} \text{ and } b_{i-1}^2 \equiv 1 \pmod{q}$$

but either  $b_{i-1} \not\equiv 1 \pmod{p}$  or  $b_{i-1} \not\equiv 1 \pmod{q}$

\* catching the moment that  $b_0, b_1, \dots$  behave differently while taking square in  $(\text{mod } p)$  component and  $(\text{mod } q)$  component

30

## Miller-Rabin Test Example

- e.g.  $n = 561$

A Carmichael number: pass the Fermat test for all bases

$$n-1 = 560 = 16 \cdot 35 = 2^4 \cdot 35$$

let  $a = 2$

$$b_0 \equiv 2^{35} \equiv 263 \pmod{561}$$

$$b_1 \equiv b_0^2 \equiv 2^{2 \cdot 35} \equiv 166 \pmod{561}$$

$$b_2 \equiv b_1^2 \equiv 2^{2^2 \cdot 35} \equiv 67 \pmod{561}$$

$$b_3 \equiv b_2^2 \equiv 2^{2^3 \cdot 35} \equiv 1 \pmod{561}$$

561 is composite ( $3 \cdot 11 \cdot 17$ ),

$\text{gcd}(b_2-1, 561) = 33$  is a factor

Note:  $3-1=2, 11-1=2 \cdot 5, 17-1=2^4$

$$\phi(561) = 561(1-1/3)(1-1/11)(1-1/17) = 2 \cdot 10 \cdot 16$$

$$\text{gcd}(\phi(561), n-1) = 80, \text{ord}_{561}(2) \mid 80 \text{ in this case}$$

mod	3	11	17
	2	10	8
	1	1	13
	1	1	16
	1	1	1

$$\text{ord}_{17}(2) = 2^3$$

31

## Pseudo Prime and Strong Pseudo Prime

- If  $n$  is not a prime but satisfies  $a^{n-1} \equiv 1 \pmod{n}$  we say that  $n$  is a pseudo prime number for base  $a$ .

\* e.g.  $2^{560} \equiv 1 \pmod{561}$

- If  $n$  is not a prime but passes the Miller-Rabin test with base  $a$  (without being identified as a composite), we say that  $n$  is a strong pseudo prime number for base  $a$ .

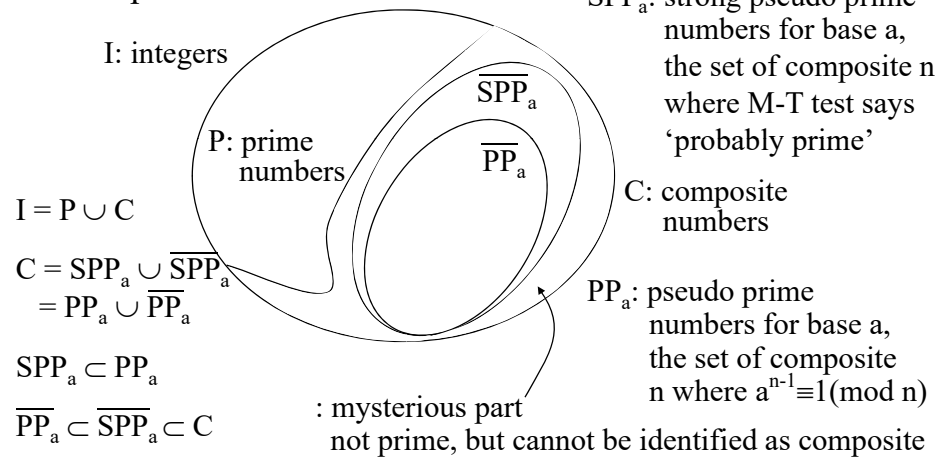
- Up to  $10^{10}$ , there are 455052511 primes, there are 14884 pseudo prime numbers for the base 2, and 3291 strong pseudo prime numbers for the base 2

32



## Fermat and Miller-Rabin Test

- Both of these two tests are for identifying subsets of composite numbers



33

## Composite Witness

- Note that the **M-R test** and probably together with the **Lucas test** leave the strong pseudo prime number *an extremely small set*.
- In other words, these tests are very close to a *real 'primality test'* between prime numbers and composite numbers.
- If you have an RSA modulus  $n=p \cdot q$ , you certainly can test it and find out that it is actually a composite number.
- However, these tests do not necessarily give you the factors of n in order to tell you that n is a composite number. The factors of n, i.e. p or q, are certainly a kind of witness about the fact that n is composite.
- However, there are other kind of witness that n is composite, e.g., " $2^{n-1} \pmod{n}$  does not equal to 1" is also a witness that n is composite.
- A composite number will be factored out by the M-R test only if it is a pseudo prime but it is not a strong pseudo prime number.

34

## Matlab Example

- `primetest(n)`
  - Miller-Rabin test for 30 randomly chosen base a
  - output 0 if n is composite
  - output 1 if n is prime
  - Matlab program can not be used for large n
  - use Maple `isprime(n)`, one strong pseudo-primality test and one Lucas test
- `primetest(2563)`  
ans= 0
- `factor(2563)`  
ans = 11 233

35

## Questions

- What is the probability that Miller-Rabin test fails???
- If n is a prime number, it will not be recognized as a composite number
- If  $n = p \cdot q$ , but
  - $b_k \equiv a^{n-1} \equiv 1 \pmod{n}$  meets Fermat test (pseudo prime number)
  - $0 < i < k$   $b_i \equiv 1 \pmod{n}$  and  $b_{i-1} \equiv -1 \pmod{n}$  meets Miller-Rabin test (strong pseudo prime number)
  - or  $\left. \begin{array}{l} b_i \equiv 1 \pmod{n} \equiv 1 \pmod{p} \equiv 1 \pmod{q} \\ b_{i-1} \equiv -1 \pmod{n} \equiv -1 \pmod{p} \equiv -1 \pmod{q} \end{array} \right\}$
- Note:  $a^{pq-1} \equiv 1 \pmod{n}$   
 $a^{(p-1)(q-1)} \equiv 1 \pmod{n}$   
 $a^{\text{lcm}(p-1, q-1)} \equiv 1 \pmod{n}$

36

## Note on Primality Testing

- ◇ Primality testing is *different* from factoring
  - ★ Kind of interesting that we can tell something is composite without being able to actually factor it
- ◇ Recent result (2002) from IIT trio (Agrawal, Kayal, and Saxena)
  - ★ Recently it was shown that deterministic primality testing could be done in polynomial time
    - ★ Complexity was like  $O(n^{12})$ , though it's been slightly reduced since then
  - ★ Does this mean that RSA was broken?
- ◇ Randomized algorithms like Rabin-Miller are far more efficient than the IIT algorithm, so we'll keep using those

37

## Finding a Random Prime

- ◇ Find a prime of around 100 digits for cryptographic usage
- ◇ Prime number theorem ( $\pi(x) \approx x/\ln(x)$ ) asserts that the density of primes around  $x$  is approximately  $1/\ln(x)$
- ◇  $x = 10^{100}$ ,  $1/\ln(10^{100}) = 1/230$   
if we skip even numbers, the density is about  $1/115$
- ◇ pick a random starting point, throw out multiples of 2, 3, 5, 7, and use Miller-Rabin test to eliminate most of the composites.
  
- ◇ `maple('a:=nextprime(189734535789)')`

38

## Factoring

- ◇ General number field sieve (GNFS): fastest  
$$e^{(1.923+O(1))(\ln(n))^{1/3} (\ln(\ln(n)))^{2/3}}$$
- ◇ Quadratic sieve (QS)
- ◇ Elliptic curve method (ECM), Lenstra (1985)
- ◇ Pollard's Monte Carlo algorithm
- ◇ Continued fraction algorithm
- ◇ Trial division, Fermat factorization
- ◇ Pollard's  $p-1$  factoring (1974), Williams's  $p+1$  factoring (1982)
- ◇ Universal exponent factorization, exponent factorization

39

## Simple Factoring Methods

- ◇ Trial division:
  - ★ dividing an integer  $n$  by all primes  $p \leq \sqrt{n}$  ... too slow
- ◇ Fermat factorization:
  - ★ e.g.  $n = 295927$  calculate  $n+1^2, n+2^2, n+3^2 \dots$  until finding a square, i.e.  $x^2 = n + y^2$ , therefore,  
 $n = (x+y)(x-y) \dots$  if  $n = p \cdot q$ , it takes on average  $|p-q|/2$  steps ... too slow
  - $$\boxed{\text{assume } p > q, n + y^2 = p \cdot q + ((p-q)/2)^2 = (p^2 + 2pq + q^2)/4 = ((p+q)/2)^2}$$
  - ★ in RSA or Rabin, avoid  $p, q$  with the same bit length
- ◇ By-product of Miller-Rabin primality test:
  - ★ if  $n$  is a pseudoprime and not a strong pseudoprime, Miller-Rabin test can factor it. about  $10^{-6}$  chance

40

# Universal Exponent Factorization

- \* if we have an exponent  $r$ , s.t.  $a^r \equiv 1 \pmod{n}$  for all  $a$   $\gcd(a,n)=1$
- \* write  $r = 2^k \cdot m$  with  $m$  odd
- \* choose a random  $a$ ,  $1 < a < n-1$
- \* if  $\gcd(a, n) \neq 1$ , we have a factor
- \* else
  - \* let  $b_0 \equiv a^m \pmod{n}$ , if  $b_0 \equiv \pm 1$  stop, choose another  $a$
  - \* compute  $b_{u+1} \equiv b_u^2 \pmod{n}$  for  $0 \leq u \leq k-1$ ,
  - \* if  $b_{u+1} \equiv -1$ , stop, choose another  $a$
  - \* if  $b_{u+1} \equiv 1$  then  $\gcd(b_u-1, n)$  is a factor (basic factoring principle)
- \* Question: How do we find a universal exponent  $r$  ??? Hard
- \* Note: if know  $\phi(n)$ , then any  $r = k \phi(n)$  will do, however, knowing factors of  $n$  is a prerequisite of know  $\phi(n)$
- \* Note: For RSA, if the private exponent  $d$  is recovered, then  $\phi(n) \mid d \cdot e - 1$ ,  $d \cdot e - 1$  is a universal exponent

$r$  must be even since we can take  $a=-1$   $(-1)^r \equiv 1 \pmod{n}$  requires  $r$  being even

$a \equiv \pm 1$  do not work

# Universal Exponent Factorization

- ◇ E.g.
  - $n=211463707796206571$ ;  $e=9007$ ;  $d=116402471153538991$
  - $r=e \cdot d - 1 = 1048437057679925691936$ ;  $\text{powermod}(2,r,n)=1$
  - let  $r=2^5 \cdot r_1$ ;  $r_1=32763658052497677873$
  - $\text{powermod}(2,r_1,n)=187568564780117371 \neq \pm 1$
  - $\text{powermod}(2,2 \cdot r_1,n)=113493629663725812 \neq \pm 1$
  - $\text{powermod}(2,4 \cdot r_1,n)=1 \Rightarrow \gcd(2 \cdot r_1 - 1, n) = 885320963$  is a factor
- ◇ Note:  $n = 211463707796206571 = 238855417 \cdot 885320963$ 
  - $238855417 - 1 = 2^3 \cdot 3 \cdot 73 \cdot 136333 = 2^{k_1} \cdot p_1$
  - $885320963 - 1 = 2 \cdot 2069 \cdot 213949 = 2^{k_2} \cdot q_1$
  - This method works only when  $k_1$  does not equal  $k_2$ .
- ◇ Exponent factorization even if  $r$  is valid for one  $a$ , you can still try the above procedure

# p-1 factoring (1/2)

- ◇ If one of the prime factors of  $n$  has a special property, it is sometimes easier to factor  $n$ .
    - \* e.g. if  $p-1$  has only small prime factors
    - \* Pollard 1974
  - ◇ Algorithm
    - \* Choose an integer  $a > 1$  (often  $a = 2$  is used)
    - \* Choose a bound  $B$
    - \* Compute  $b \equiv a^{B!} \pmod{n}$  as follows:
      - \*  $b_1 \equiv a \pmod{n}$  and  $b_j \equiv b_{j-1}^j \pmod{n}$  then  $b \equiv b_B \pmod{n}$
    - \* Let  $d = \gcd(b-1, n)$ , if  $1 < d < n$ , we have found a factor of  $n$ 
      - If  $B$  is larger than all the prime factors of  $p-1$  (very likely)  $\Rightarrow p-1 \mid B!$
      - therefore  $b \equiv a^{B!} \equiv (a^{p-1})^{k} \equiv 1 \pmod{p}$ , i.e.  $p \mid b-1$  Fermat Little's Thm
- If  $n=p \cdot q$ ,  $p-1$  and  $q-1$  both have small factors that are less than  $B$ , then  $\gcd(b-1, n)=n$ , (useless) however,  $b \equiv a^{B!} \equiv 1 \pmod{n}$  and we can use the Universal exponent method

# p-1 factoring (2/2)

- ◇ How do we choose  $B$ ?
  - \* small  $B$  will be faster but fails often
  - \* large  $B$  will be very slow
- ◇ In RSA, Rabin, Paillier, or other systems based on integer factoring, usually  $n=p \cdot q$ , we should ensure that  $p-1$  has at least one large prime factor.
  - \* How do we do this?
    - e.g. we want to choose  $p$  around 100 digits
      - > choose a prime number  $p_0$  around 40 digits
      - > look at integer  $k \cdot p_0 + 1$  with  $k$  around 60 digits and do primality test
- ◇ Generalization:
  - Elliptic curve factorization method, Lenstra, 1985
- ◇ Best records:  $p-1$ : 34 digits (113 bits), ECM: 47 digits (143 bits)

## Quadratic Sieve (1/4)

◇ Example: factor  $n = 3837523$

- \* form the following relations individual factors are small

$$9398^2 \equiv 5^5 \cdot 19 \pmod{3837523}$$

$$19095^2 \equiv 2^2 \cdot 5 \cdot 11 \cdot 13 \cdot 19 \pmod{3837523}$$

$$1964^2 \equiv 3^2 \cdot 13^3 \pmod{3837523}$$

$$17078^2 \equiv 2^6 \cdot 3^2 \cdot 11 \pmod{3837523}$$
- \* multiply the above relations make the number of each factors even

$$(9398 \cdot 19095 \cdot 1964 \cdot 17078)^2 \equiv (2^4 \cdot 3^2 \cdot 5^3 \cdot 11 \cdot 13^2 \cdot 19)^2$$

$$2230387^2 \equiv 2586705^2$$
- \* since  $2230387 \not\equiv \pm 2586705 \pmod{3837523}$ 
hope they are not equal
- \*  $\gcd(2230387 - 2586705, 3837523) = 1093$  is one factor of  $n$
- \* the other factor is  $3837523/1093 = 3511$

45

## Quadratic Sieve (2/4)

- ◇ Quadratic?  $x^2 \equiv$  product of small primes
- ◇ How do we construct these useful relations systematically?
- ◇ Properties of these relations:
  - \* product of small primes called factor base
  - \* make all prime factors appear even times
- ◇ Put these relations in a matrix

	2	3	5	7	11	13	17	19	add
9398	0	0	5	0	0	0	0	1	
19095	2	0	1	0	1	1	0	1	
1964	0	2	0	0	0	3	0	0	
17078	6	2	0	0	1	0	0	0	
8077	1	0	0	0	0	0	0	1	
3397	5	0	1	0	0	2	0	0	
14262	0	0	2	2	0	1	0	0	

46

## Quadratic Sieve (3/4)

- ◇ Look for linear dependencies mod 2 among the rows
  - \*  $1\text{st} + 5\text{th} + 6\text{th} = (6, 0, 6, 0, 0, 2, 0, 2) \equiv \mathbf{0} \pmod{2}$
  - \*  $1\text{st} + 2\text{nd} + 3\text{rd} + 4\text{th} = (8, 4, 6, 0, 2, 4, 0, 2) \equiv \mathbf{0} \pmod{2}$
  - \*  $3\text{rd} + 7\text{th} = (0, 2, 2, 2, 0, 4, 0, 0) \equiv \mathbf{0} \pmod{2}$
- ◇ When we have such a dependency, the product of the numbers yields a square.
  - \*  $(9398 \cdot 8077 \cdot 3397)^2 \equiv 2^6 \cdot 5^6 \cdot 13^2 \cdot 19^2 \equiv (2^3 \cdot 5^3 \cdot 13 \cdot 19)^2$
  - \*  $(9398 \cdot 19095 \cdot 1964 \cdot 17078)^2 \equiv (2^3 \cdot 3^2 \cdot 5^3 \cdot 11 \cdot 13^2 \cdot 19)^2$
  - \*  $(1964 \cdot 14262)^2 \equiv (3 \cdot 5 \cdot 7 \cdot 13^2)^2$
- ◇ Looking for those  $x^2 \equiv y^2$  but  $x \neq \pm y$

47

## Quadratic Sieve (4/4)

- ◇ How do we find numbers  $x$  s.t.  $x^2 \equiv$  product of small primes?
    - \* produce squares that are slightly larger than a multiple of  $n$ 
      - e.g.  $\lfloor \sqrt{i \cdot n} + j \rfloor$  for small  $j$
      - the square is approximately  $i \cdot n + 2j\sqrt{i \cdot n} + j^2$
      - which is approximately  $2j\sqrt{i \cdot n} + j^2 \pmod{n}$
- $$8077 = \lfloor \sqrt{17n} + 1 \rfloor$$
- $$9398 = \lfloor \sqrt{23n} + 4 \rfloor$$
- Probably because this number is small, the factors of it should not be too large. However, there are a lot of exceptions. So it takes time. Also, there are a lot of other methods to generate qualified  $x$  values.

48

## The RSA Challenge

- ◇ 1977 Rivest, Shamir, Adleman US\$100
  - \* given RSA modulus  $n$ , public exponent  $e$ , ciphertext  $c$ 
    - $n = 114381625757888867669235779976146612010218296721242362$
    - $562561842935706935245733897830597123563958705058989075$
    - $147599290026879543541$
    - $e = 9007$
    - $c = 968696137546220614771409222543558829057599911245743198$
    - $746951209308162982251457083569314766228839896280133919$
    - $90551829945157815154$
  - \* Find the plaintext message
- ◇ 1994 Atkins, Lenstra, and Leyland
  - \* use 524339 small primes (less than 16333610)
  - \* plus up to two large primes ( $16333610 \sim 2^{30}$ )
  - \* 1600 computers, 600 people, 7 months
  - \* found 569466 'x<sup>2</sup>≡small products' equations, out of which only 205 linear dependencies were found

49

## Factorization Records

Year	Number of digits
1964	20
1974	45
1984	71
1994	129 (429 bits)
1999	155 (515 bits)
2003	174 (576 bits)

Next challenge  
RSA-640

31074182404900437213507500358885679300373460228427  
27545720161948823206440518081504556346829671723286  
78243791627283803341547107310850191954852900733772  
4822783525742386454014691736602477652346609

50

## Security of the RSA Function

- ◇ **Break RSA** means 'inverting RSA function without knowing the trapdoor'
 

$y \equiv x^e \pmod{n}$
- ◇ Factor the modulus  $\Rightarrow$  Break RSA
  - \* If we can factor the modulus, we can break RSA
  - \* If we can break RSA, we don't know whether we can factor the modulus... **open problem** (with negative evidences)
- ◇ Factor the modulus  $\Leftrightarrow$  Calculate private key  $d$ 
  - \* If we can factor the modulus, we can calculate the private exponent  $d$  (the trapdoor information).
  - \* If we have the private exponent  $d$ , we can factor the modulus.

51

## Factoring reduces to RSA key recovery

- ◇ DeLaurentis, "A Further Weakness in the Common Modulus Protocol for the RSA Cryptosystem," Cryptologia, Vol. 8, pp. 253-259, 1984
  - \* If you have a pair of RSA public-key/private-key, you can factor  $n=p \cdot q$  with a probabilistic algorithm.
  - \* An example of the Universal Exponent Factorization method
- ◇ Basic idea: find a number  $b$ ,  $0 < b < n$  s.t.
  - $b^2 \equiv 1 \pmod{n}$  and  $b \neq \pm 1 \pmod{n}$  i.e.  $1 < b < n-1$
  - \* Note: There are four roots to the equation  $b^2 \equiv 1 \pmod{n}$ ,  $\pm 1$  are two of them, all satisfy  $(b+1)(b-1) = k \cdot n = k \cdot p \cdot q$ , since  $0 < b-1 < b+1 < n$ , we have either  $(p \mid b-1$  and  $q \mid b+1)$  or  $(q \mid b-1$  and  $p \mid b+1)$ , therefore, one of the factor can be found by  $\gcd(b-1, n)$  and the other by  $n/\gcd(b-1, n)$  or  $\gcd(b+1, n)$

52

## Factoring reduces to RSA key recovery

- Algorithm to find  $b$ :  $\Pr\{\text{success per repetition}\} = \frac{1}{2}$ 
  - Randomly choose  $a$ ,  $1 < a < n-1$ , such that  $\gcd(a, n) = 1$
  - Find minimal  $j$ ,  $a^{2^j} \equiv 1 \pmod{n}$  (where  $h$  satisfies  $e \cdot d - 1 = 2^h$ )
  - $b = a^{2^{j-1}h}$ , if  $b \neq -1 \pmod{n}$ , then  $\gcd(b-1, n)$  is the result, else repeat 1-3
- Note: If we randomly choose  $b \in Z_n^*$  and find out that  $b^2 \equiv 1 \pmod{n}$ , the probability that  $b=1$ ,  $b=-1$ ,  $b=c(\neq \pm 1)$ , or  $b=-c(\neq \pm 1)$  would be equal;  $\Pr\{\text{success}\} = \Pr\{a^{2^{j-1}h} \neq \pm 1\} = 1/2$
- Ex:  $p=131$ ,  $q=199$ ,  $n=p \cdot q=26069$ ,  $e=7$ ,  $d=22063$   
 $\phi(n)=(p-1)(q-1)=25740=2^2 \cdot 6435 \mid ed-1=154440=2^3 \cdot 19305$ ,  
choose  $a=3$ , try  $j=1$  ( $3^{2^1 \cdot 19305} \equiv 1$ ),  $b = a^{2^{j-1}h} = 3^{19305} = 5372 (\neq \pm 1)$   
 $p = \gcd(b-1, n) = \gcd(5371, 26069) = 131$ ,  $q = n/p = 199$

53

## Factoring reduces to RSA key recovery

- The above result says that “if you can recover a pair of RSA keys, you can factor the corresponding  $n=p \cdot q$ ” i.e. “once a private key  $d$  is compromised, you need to choose a new pair of  $(n, e)$  instead of changing  $e$  only”
- The above result suggests that a scheme using  $(n, e_1)$ ,  $(n, e_2)$ , ...  $(n, e_k)$  with a common  $n$  for each  $k$  participants without giving each one the value of  $p, q$  is insecure. You should not use the same  $n$  as some others even though you are not explicitly told the value of  $p$  and  $q$ .

54

## Factoring reduces to RSA key recovery

- The above result also suggests that if you can recover arbitrary RSA key pair, you can solve the problem of factoring  $n$ . Whenever you get an  $n$ , you can form an RSA system with some  $e$  (assuming  $\gcd(e, \phi(n))=1$ ), then use your method to solve the private exponent  $d$  without knowing  $p$  and  $q$ , after that you can factor  $n$ .
- Although factoring is believed to be hard, and factoring breaks RSA, breaking RSA does not simplify factoring. Trivial non-factoring methods of breaking RSA could therefore exist. (What does it mean by breaking RSA? plaintext recovery? key recovery?...)

55

## Deterministic Encryption

- RSA Cryptosystem is a deterministic encryption scheme, i.e. a plaintext message is encrypted to a fixed ciphertext message
- Suffers from chosen plaintext attack
  - an attacker compiles a large codebook which contains the ciphertexts corresponding to all possible plaintext messages
  - in a two-message scheme, the attacker can always distinguish which plaintext was transmitted by observing the ciphertext (does not satisfy the Semantic Security Notation)
- Add randomness through padding

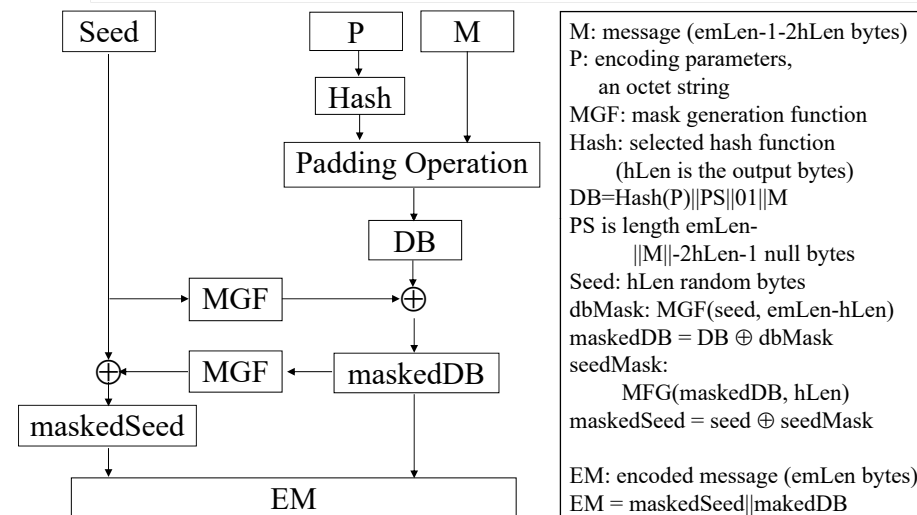
56

## RSA PKCS #1 v1.5 padding

- ◇ E.g.  $k=128$  bytes (1024 bits) PKCS#1 v1.5 RSA
  - \* plaintext message  $M$  (at most  $128-3-8=117$  bytes)
  - \* pseudorandom nonzero string  $PS$  (at least 8 bytes)
  - \* message to be encrypted  $m = 00\|02\|PS\|00\|M$
  - \* encryption:  $c \equiv m^e \pmod{n}$
  - \* decryption:  $m \equiv c^d \pmod{n}$
- ◇  $c$  is now random corresponding to a fixed  $m$ , however, this only adds difficulties to the compilation of ciphertexts (a factor of  $2^{64}$  times if  $PS$  is 8 bytes)

57

## PKCS #1 v2 padding - OAEP



58

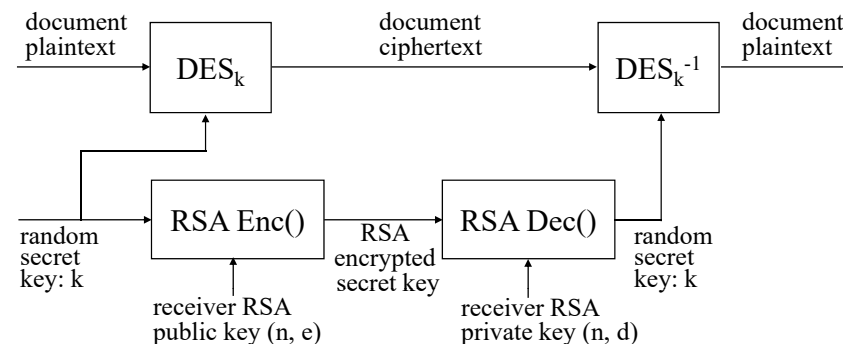
## PKCS #1 v2 padding - OAEP

- ◇ Optimal Asymmetric Encryption (OAE)
  - \* M. Bellare, "Optimal Asymmetric Encryption - How to Encrypt with RSA," Eurocrypt'94
- ◇ Optimal Padding in the sense that
  - \* RSA-OAEP is semantically secure against adaptive chosen ciphertext attackers in the random oracle model
  - \* the message size in a  $k$ -bit RSA block is as large as possible (make the most advantage of the bandwidth)
- ◇ Following by more efficient padding schemes:
  - \* OAEP<sup>+</sup>, SAEP<sup>+</sup>, REACT

59

## Digital Envelop

- ◇ Hybrid system (public key and secret key)
  - \* computation of RSA is about 1000 times slower than DES
  - \* smaller exponent is faster (but usually dangerous)



60

## RSA Fast Decryption with CRT

Public key (n, e)

$n = p \cdot q$ , p and q are large prime integers  
 $\gcd(e, \phi(n)) = 1$  s.t.  $\exists d, e \cdot d \equiv 1 \pmod{\phi(n)}$   
 $\phi(n) = (p-1)(q-1)$   $3 \leq e \leq n-1$

Private Key (n, d) or

(n, p, q, dp, dq, qInv)

$e \cdot dp \equiv 1 \pmod{p-1}$   
 $e \cdot dq \equiv 1 \pmod{q-1}$   
 $q \cdot qInv \equiv 1 \pmod{p}$

Encryption  $c \equiv m^e \pmod{n}$

Decryption  $m \equiv c^d \pmod{n}$  or

$$\begin{aligned} & \left. \begin{aligned} m_1 &\equiv c^{dp} \pmod{p} \\ m_2 &\equiv c^{dq} \pmod{q} \end{aligned} \right\} \text{CRT} \\ & h \equiv qInv \cdot (m_1 - m_2) \pmod{p} \\ & m \equiv m_2 + h \cdot q \pmod{n} \end{aligned}$$

$$\begin{aligned} m_1 &\equiv (m^e)^{dp} \equiv m^{e \cdot dp} \equiv m \pmod{p} \\ m_2 &\equiv (m^e)^{dq} \equiv m^{e \cdot dq} \equiv m \pmod{q} \\ m &\equiv m_2 \pmod{q} \text{ and} \\ m &\equiv m_2 + qInv \cdot (m_1 - m_2) \cdot q \equiv m_1 \pmod{p} \end{aligned}$$

61

## Multi-Prime RSA

RSA PKCS#1 v2.0 Amendment 1

the modulus n may have more than two prime factors

only private key operations and representations are affected (p, q, dp, dq, qInv) ( $r_i, d_i, t_i$ )

\*  $n = r_1 \cdot r_2 \cdot \dots \cdot r_k, k \geq 2$ , where  $r_1 = p, r_2 = q$

\*  $e \cdot d_i \equiv 1 \pmod{r_i - 1}, i = 3, \dots, k$

\*  $r_1 \cdot r_2 \cdot \dots \cdot r_{i-1} \cdot t_i \equiv 1 \pmod{r_i}, i = 3, \dots, k$

Decryption:

1.  $m_1 \equiv c^{dp} \pmod{p}$

2.  $m_2 \equiv c^{dq} \pmod{q}$

3. if  $k > 2$   $m_i \equiv c^{d_i} \pmod{r_i}, i = 3, \dots, k$

4.  $h \equiv (m_1 - m_2) qInv \pmod{p}$

5.  $m = m_2 + q \cdot h$

6. if  $k > 2, R = r_1$ , for  $k = 3$  to  $k$  do

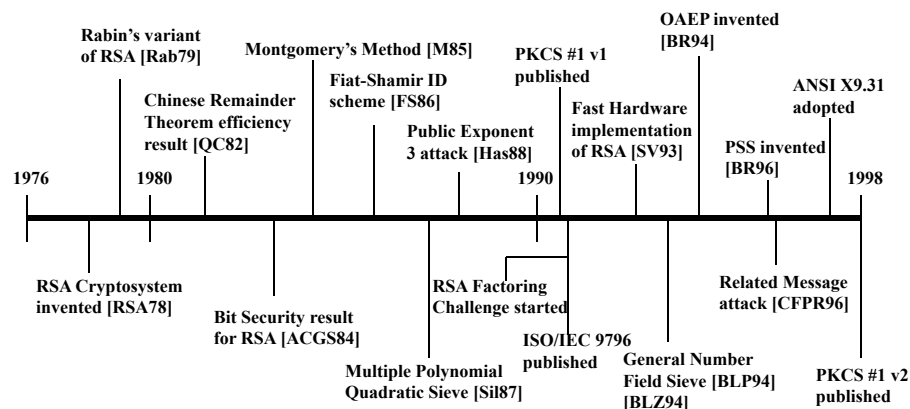
a.  $R = R \cdot r_{i-1}$

b.  $h \equiv (m_1 - m) \cdot t_i \pmod{r_i}$

c.  $m = m + R \cdot h$

advantages: lower computational cost for the decryption (and signature) primitives if CRT is used (also see 6.8.14) 62

## Factoring & RSA Timeline



63

## Alternative PKC's

ElGamal Cryptosystem (Discrete-log based)

\* Also suffers from long keys

NTRU (Lattice based)

\* Utilizes short keys

\* Proprietary (License issues prevent from wide implementation)

\* Recently, a weakness found in the signature scheme

Elliptic Curve Cryptosystems

\* Emerging public key cryptography standard for constrained devices.

Paillier Cryptosystem (High order composite residue based)

Goldwasser-Micali Cryptosystem (QR based)

\* very low efficiency

64