

Chapter 5

The RSA Cryptosystems

COMP 7120-8120

Lih-Yuan Deng

lihdeng@memphis.edu



Outline

- Introduction to Public Key Cryptography
- The RSA Cryptosystem
- More Number Theory
 - Primality Testing,
 - Square Roots Modulo n ,
 - Factoring Algorithms
- Other Attacks on RSA
- The Rabin Cryptosystem
- Semantic Security of RSA



Public Key Encryption

- Diffie and Hellman (1976): first proposed a public key encryption system. (public key e_K for encryption, private key d_K for decryption)
- symmetric key vs. asymmetric key ?
- Motivation of public key encryption ?

Public Key Encryption

Motivation

- For a symmetric key encryption, any pair of users need to share the key.
- System with n users need $n(n-1)/2$ keys to keep track or memorize. (why ?)
 - Difficult to maintain the security of so many keys.
- Public key cryptosystem: each user needs one private key, one public key shared by anyone.



Characteristics

- Asymmetric encryption system.
 - Public key (shared key) for encryption.
 - Secret key for decryption.
 - P=plaintext, two keys k_{pub} , k_{priv} .
 - $C=E(k_{pub}, P)$ ciphertext.
 - $P=D(k_{priv}, C)$ plaintext.

$$P=D(k_{priv}, E(k_{pub}, P))$$

$$P=E(k_{pub}, D(k_{priv}, P))$$



Characteristics

$$P = D(k_{\text{priv}}, E(k_{\text{pub}}, P))$$

$$P = E(k_{\text{pub}}, D(k_{\text{priv}}, P))$$

- $S = D(k_{\text{priv}}, P)$ is sometimes called the **signature** of message P (why?).
 - Only owner of P with k_{priv} can produce S .
- The number of keys needed is minimized.



RSA Encryption

- Rivest-Shamir-Adelman (RSA) Encryption (1977)
- Choose
 - $n = p \cdot q$, where p and q are **large primes**
 - Choose e with $\gcd(e, (p-1)(q-1)) = 1$.
 - Find $d = e^{-1} \bmod (p-1)(q-1)$. i.e.,
$$d \times e = 1 \bmod (p-1)(q-1)$$
- To encrypt: $C = P^e \bmod n$
- To decrypt: $P = C^d \bmod n$



RSA Encryption Example

- Choose
 - $n = 101 \times 199 = 20099$
 - Choose $e = 13$ and $d = 18277 = e^{-1} \pmod{n}$
 $d \times e = 1 \pmod{19800}$
 - $P = 12345$ (say)
- To encrypt: $C = P^e \pmod{n}$
 $C = 12345^{13} \pmod{20099} = 7220$
- To decrypt: $P = C^d \pmod{n}$
 $P = 7220^{18277} \pmod{20099} = 12345$



RSA Encryption: Another Example

- Choose
 - $n = 101 \times 199 = 20099$
 - Choose $e = 13$ and $d = 18277$
 - $P = 12346$ (say)
- To encrypt: $C = P^e \bmod n$
 $C = 12346^{13} \bmod 20099 = 2704$
- To decrypt: $P = C^d \bmod n$
 $P = 2704^{18277} \bmod 20099 = 12346$



RSA Encryption and mathematical tools required

- $n = p \times q$, where p and q are huge numbers so that impossible to factor n
- Solution of d and e in
$$d \times e = 1 \pmod{(p-1)(q-1)}$$
requires the exact knowledge of p and q .
- (How ? Extended Euclidean Algorithm to find d, e)
- Encryption: $C = P^e \pmod n$ requires only n and e .
- Decryption: $P = C^d \pmod n$ requires d or an integer factorization of n .
- (Need: modulus arithmetic, fast exponentiation, number theory, integer factorization)



Euclidean Algorithm

- Recursively find $\text{gcd}(a,b)$:
 - $\text{gcd}(a,b) = a$, if $b=0$;
 - $\text{gcd}(a, b) = \text{gcd}(b, a \% b)$, if $b \neq 0$.

Algorithm 5.1 (page 164)

- $r_0 \leftarrow a, r_1 \leftarrow b, m \leftarrow 1$.
- while $r_m \neq 0$, do
 - $q_m \leftarrow r_{m-1}/r_m$,
 - $r_{m+1} \leftarrow r_{m-1} \% r_m$,
 - $m = m + 1$
- $m = m - 1$
- $r_m = \text{gcd}(a,b)$.



Euclidean Algorithm Example

- E.g. $\gcd(54,30)=6$.
 - $54 \div 30 \Rightarrow q=1, r=24$
 - $30 \div 24 \Rightarrow q=1, r=6$
 - $24 \div 6 \Rightarrow q=4, r=0$.
 - $\gcd(54,30)=\gcd(30,24)=\gcd(24,6)=6$.



Euclidean Algorithm

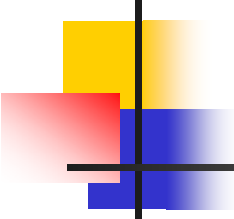
- Let $r = \gcd(a,b)$. Then we can find s and t such that

$$a s + b t = r.$$

- In particular, if $\gcd(a,n)=1$, then

$$a s = 1 \pmod n.$$

- s is called $a^{-1} \pmod n$. How to find ?
 - Extended Euclidean Algorithm.
 - Fermat's Little Theorem.



Extended Euclidean Algorithm (Algorithm 5.2, page 166)

- Input: $a \geq b \geq 0$.
- Output: (s, t) such that
$$a s + b t = \gcd(a, b).$$
- Initialization:
 - $i=0; r_0=a; r_1=b; s_0 = 1; t_0 = 0; s_1 = 0; t_1 = 1;$
- While $(r_i \neq 0)$ do
 - $q = r_{i-1}/r_i; r_{i+1} = r_{i-1} - q r_i;$
 - $s_{i+1} = s_{i-1} - q s_i; t_{i+1} = t_{i-1} - q t_i; i=i+1;$
- Return $(s=s_{i-1}, t=t_{i-1})$



Theorem 5.1 (page 165)

- Given q_i (quotients) and r_i (remainders) from Euclidean Algorithm, $i=1,2,\dots,m$.
- $r_0 = a, r_1 = b, r_m = \gcd(a,b)$.
- produce (s_i, t_i) such that $r_i = a*s_i + b*t_i$
 - $s_0 = 1, t_0 = 0$
 - $s_1 = 0, t_1 = 1$.
 - $s_j = s_{j-2} - q_{j-1} * s_{j-1}$. [same as $r_j = r_{j-2} - q_{j-1} * r_{j-1}$]
 - $t_j = t_{j-2} - t_{j-1} * q_{j-1}$.



Application of Theorem 5.1

- We can find s and t such that
 - $r = a s + b t, r = \gcd(a,b)$.
- Corollary 5.2 (page 166). In particular, if $r = 1$ (a and b are relative prime),
 - $1 = a s + b t$.
 - $s = a^{-1} \bmod t$ (why ?)
 - $t = b^{-1} \bmod s$

Example 5.1 gcd(75,28)

- $r_0 = 75, s_0 = 1, t_0 = 0.$
- $r_1 = 28, q_1 = 75/28 = 2, s_1 = 0, t_1 = 1.$
- $r_2 = 75 \% 28 = 19, q_2 = 28/19 = 1,$
- $s_2 = s_0 - q_1 * s_1 = 1, t_2 = t_0 - q_1 * t_1 = -2.$

i	r _i	q _i	s _i	t _i	a*s _i +b*t _i
0	75		1	0	75
1	28	2	0	1	28
2	19	1	1	-2	19
3	9	2	-1	3	9
4	1	9	3	-8	1
5	0				(Thm 5.1)

$75*3+28*(-8) = 1, 28^{-1} \text{ mod } 75 = -8 = 67.$



The Chinese Remainder Theorem (CRT)

- Suppose m_1, m_2, \dots, m_r that are pairwise relative prime. That is,
 - $\gcd(m_i, m_j) = 1, i \neq j.$
- For any integers a_1, a_2, \dots, a_r , find x satisfy the r equations:
 - $x = a_i \pmod{m_i}, i=1,2,\dots,r.$
 - Theorem 5.3 (page 170).



$\chi(x)$ function

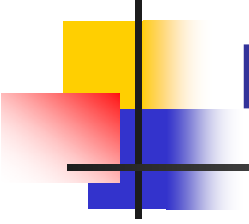
- Define $\chi(x) = (x \bmod m_1, x \bmod m_2, \dots, x \bmod m_r)$.
- e.g. 5.2. (page 168) $r=2$, $m_1 = 5$, $m_2 = 3$.
- Find x such that
 - $x = 2 \bmod 5$ and $x = 1 \bmod 3$.
 - table of ($x \bmod 5$, $x \bmod 3$) given below. $x = \chi^{-1}(2,1)$.
 - How to find $\chi^{-1}(a_1, a_2)$ in general ?

0	0	0	1	1	1	2	2	2
3	3	0	4	4	1	5	0	2
6	1	0	7	2	1	8	3	2
9	4	0	10	0	1	11	1	2
12	2	0	13	3	1	14	4	2

Solving

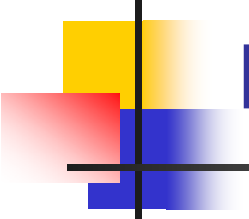
$$x = 2 \pmod{5} \text{ and } x = 1 \pmod{3}$$

- $x = 2s + 1t \pmod{15}$, such that
 - $s = 1 \pmod{5}$ and $s = 0 \pmod{3}$
 - $t = 0 \pmod{5}$ and $t = 1 \pmod{3}$.
- Why ?
 - $x = 2s + 1t \pmod{5} = 2s \pmod{5} = 2$.
 - $x = 2s + 1t \pmod{3} = 1t \pmod{3} = 1$.
- How ?
 - $(s,t) = (3 \cdot (3^{-1} \pmod{5}), 5 \cdot (5^{-1} \pmod{3}))$ (why?)
 - $(s,t) = (3 \cdot 2, 5 \cdot 2) = (6, 10)$
- $x = 2s + 1t \pmod{15} = 2 \cdot 6 + 10 \pmod{15} = 7$.



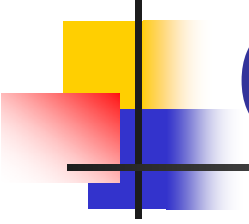
Solving $x = 5 \pmod{7}$, $x = 3 \pmod{11}$, and $x = 10 \pmod{13}$.

- $x = 5s + 3t + 10u \pmod{7*11*13}$
 - $s = 1 \pmod{7}$, $s = 0 \pmod{11}$, $s = 0 \pmod{13}$.
 - $t = 0 \pmod{7}$, $t = 1 \pmod{11}$, $t = 0 \pmod{13}$.
 - $u = 0 \pmod{7}$, $u = 0 \pmod{11}$, $u = 1 \pmod{13}$.
- $s = (11*13*((11*13)^{-1} \pmod{7}))$ (why ?)
- $t = (7*13*((7*13)^{-1} \pmod{11}))$
- $u = (7*11*((7*11)^{-1} \pmod{13}))$



Solving $x = 5 \pmod{7}$, $x = 3 \pmod{11}$, and $x = 10 \pmod{13}$.

- $x = 5s + 3t + 10u \pmod{7 \cdot 11 \cdot 13}$
 - $s = (11 \cdot 13 \cdot ((11 \cdot 13)^{-1} \pmod{7}))$
 - $143 \pmod{7} = 3, 3^{-1} \pmod{7} = 5.$
 - $s = 11 \cdot 13 \cdot 5 = 715$
 - $t = (7 \cdot 13 \cdot ((7 \cdot 13)^{-1} \pmod{11}))$
 - $91 \pmod{11} = 3, 3^{-1} \pmod{11} = 4.$
 - $t = 7 \cdot 13 \cdot 4 = 364$
 - $u = (7 \cdot 11 \cdot ((7 \cdot 11)^{-1} \pmod{13}))$
 - $77 \pmod{13} = 12, 12^{-1} \pmod{13} = 12.$
 - $u = 7 \cdot 11 \cdot 12 = 924.$
- Hence, $x = 5s + 3t + 10u \pmod{7 \cdot 11 \cdot 13} = 894.$
- Example 5.3 (page 170).



The Chinese Remainder Theorem (CRT) [Theorem 5.3 (page 170)]

- Let m_1, m_2, \dots, m_r , $\gcd(m_i, m_j) = 1, i \neq j$.
- $M = m_1 * m_2 * \dots * m_r$,
- $M_i = M/m_i, i=1,2,\dots,r$.
- Solution for x to satisfy r equations:
 - $x = a_i \pmod{m_i}, i=1,2,\dots,r$.
- $x = \sum a_i s_i \pmod{M}$, where
 - $s_i = M_i * (M_i^{-1} \pmod{m_i})$



Summary

- Public Key Encryption
- RSA Cryptosystem
- Mathematical tools required:
 - Euclidean Algorithm
 - Extended Euclidean Algorithm
 - Chinese Remainder Theorem
 - Fermat Theorem, Euler Theorem.
 - ...