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

- $P=D(k_{priv}, E(k_{pub}, P))$
- $P=E(k_{pub}, D(k_{priv}, P))$
- $S=D(k_{priv}, P)$ 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
  - \( n=p \times q \), where \( p \) and \( q \) are large primes
  - Choose \( e \) with \( \text{gcd}(e,(p-1)(q-1))=1 \).
  - Find \( d = e^{-1} \mod (p-1)(q-1) \).
    i.e.,
    \[ d \times e = 1 \mod (p-1)(q-1) \]
- To encrypt: \( C = P^e \mod n \)
- To decrypt: \( P = C^d \mod n \)

RSA Encryption Example

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

RSA Encryption: Another Example

- Choose
  - \( n=101 \times 199= 20099 \)
  - Choose \( e=13 \) and \( d=18277 \)
    \[ P=12346 \] (say)
- To encrypt: \( C = P^e \mod n \)
  \[ C = 12346^{13} \mod 20099 = 2704 \]
- To decrypt: \( P = C^d \mod n \)
  \[ P = 2704^{18277} \mod 20099 = 12346 \]
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 \( d \times e \equiv 1 \mod (p-1)(q-1) \) requires the exact knowledge of \( p \) and \( q \).
- (How? Extended Euclidean Algorithm to find \( d \), \( e \))
- Encryption: \( C = P^e \mod n \) requires only \( n \) and \( e \).
- Decryption: \( P = C^d \mod n \) requires \( d \) or an integer factorization of \( n \).
- (Need: modulus arithmetic, fast exponentiation, number theory, integer factorization)

Euclidean Algorithm

- Recursively find \( \gcd(a,b) \):
  - \( \gcd(a,b) = a \), if \( b = 0 \);
  - \( \gcd(a, b) = \gcd(b, a \mod b) \), 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
  - \( q_m \leftarrow r_{m-1}/r_m \),
  - \( r_{m+1} \leftarrow r_{m-1} \mod r_m \),
  - \( m = m+1 \)
  - \( m = m-1 \)
- \( r_m = \gcd(a,b) \).

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
  \[
  a \cdot s + b \cdot t = r.
  \]
- In particular, if \( \gcd(a,n) = 1 \), then
  \[
  a \cdot s = 1 \mod n.
  \]
- \( s \) is called \( a^{-1} \mod n \). How to find it?
  - 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
  \[
  a \cdot s + b \cdot t = \gcd(a,b).
  \]
- 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
  \[
  q = r_{i+1}/r_i; \ r_{i+1} = r_i - q \cdot r_{i+1};
  \]
  \[
  s_{i+1} = s_i - q \cdot s_{i+1}; \ t_{i+1} = t_i - q \cdot t_{i+1}; \ i = i+1;
  \]
- 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, \( i=1,2,...,m \).
- \( r_0 = a, r_1 = b, r_m = \gcd(a,b) \).
- Produce \((s_j, t_j)\) such that \( r_i = a \cdot s_i + b \cdot t_i \)
  - \( s_0 = 1, t_0 = 0 \)
  - \( s_1 = 0, t_1 = 1 \).
  - \( s_j = s_{j-1} - q_{j-1} \cdot s_{j+1} \) [same as \( r_j = r_{j-1} - q_{j-1} \cdot r_{j+1} \)]
  - \( t_j = t_{j-1} - q_{j-1} \cdot t_{j+1} \).
Application of Theorem 5.1

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

Example 5.1 \( \text{gcd}(75,28) \)

- \( r_0 = 75, s_0 = 1, t_0 = 0 \).
- \( r_1 = 28, q_1 = \frac{75}{28} = 2, s_1 = 0, t_1 = 1 \).
- \( r_2 = 75 \mod 28 = 19, q_2 = \frac{28}{19} = 1, s_2 = s_1 - q_1 \cdot s_1 = 1, t_2 = t_1 - q_1 \cdot t_1 = -2 \).

<table>
<thead>
<tr>
<th>( i )</th>
<th>( x_i )</th>
<th>( y_i )</th>
<th>( z_i )</th>
<th>( x_i \cdot y_i \cdot z_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>5</td>
<td>25</td>
<td>30</td>
</tr>
</tbody>
</table>

\( 75 \cdot 1 + 28 \cdot (-8) = 1, 28 \cdot 1 \mod 75 = -8 = 67. \)

The Chinese Remainder Theorem (CRT)

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

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

<table>
<thead>
<tr>
<th></th>
<th>( x \mod 5 )</th>
<th>( x \mod 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

Solving \( x = 2 \mod 5 \) and \( x = 1 \mod 3 \)

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

Solving \( x = 5 \mod 7, x = 3 \mod 11, \) and \( x = 10 \mod 13 \).

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

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

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

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

Summary

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