1 Public Key Cryptography (Chapter 6)

1.1 Prime Numbers

**Divisor:** A number $a$ that divides number $b$ without a remainder.

**Prime Number:** An integer greater than 1 that has no positive divisors other than 1 and itself.

**Greatest Common Denominator (GCD):** The greatest common denominator of integers $a$ and $b$ is the largest positive integer that divides both $a$ and $b$ given that at least one of the integers is nonzero. (Denoted $gcd(a, b)$)

**Relatively Prime:** $a$ and $b$ are relatively prime if and only if $gcd(a, b) = 1$

1.2 Euler’s $\phi$ function (a.k.a. totient function)

$\phi(n)$: The count of positive integers less than $n$ that are relatively prime to $n$.

1.2.1 $\phi$ function properties

Given $n$ is a prime number

$$\phi(n) = n - 1$$

Given $p$ is a prime number, $k$ is $\geq 1$

$$\phi(p^k) = p^k - p^{k-1}$$

(all numbers less than or equal to $p^k$ minus all multiples of $p$)

$$= p^k * (1 - \frac{1}{p})$$

Given $gcd(a, b) = 1$

$$\phi(a * b) = \phi(a) * \phi(b)$$
Given that $p_1, \ldots, p_r$ are prime numbers and $k_1, \ldots, k_r$ are all $\geq 1$

$$
\phi(n = p_1^{k_1} * p_2^{k_2} * \ldots * p_r^{k_r}) = p_1^{k_1} * (1 - \frac{1}{p_1}) * p_2^{k_2} * (1 - \frac{1}{p_2}) * \ldots * p_r^{k_r} * (1 - \frac{1}{p_r}) \\
= n * (1 - \frac{1}{p_1}) * (1 - \frac{1}{p_2}) * \ldots * (1 - \frac{1}{p_r})
$$

**Exercise:** Prove that $\phi(n = p * q) = (p - 1)(q - 1)$ where $p$ and $q$ are prime numbers

$$
\phi(n) = \phi(p * q) \\
= p * (1 - \frac{1}{p}) * q * (1 - \frac{1}{q}) \\
= p * q * (1 - \frac{1}{p}) * (1 - \frac{1}{q}) \\
= p * q * (1 - \frac{1}{p} - q - \frac{1}{p * q}) \\
= p * q - q - p - 1 \\
= p * (q - 1) - 1(q - 1) \\
= (p - 1) * (q - 1)
$$

### 1.3 Modular Arithmetic

Uses the non-negative integers less than a modulus (positive integer) $n$.

Written in the form $x \mod n = a$, where $a$ is the remainder when $x$ is divided by $n$.

Implications:

- $(x - a)$ is divisible by $n$
- $x = k * n + a$ for some integer $k$

**Additive Inverse** $y$ is the Additive Inverse of $x \mod n$ if $(x + y) \mod n = 0$.

**Multiplicative Inverse** $y$ is the Multiplicative Inverse of $x \mod n$ if $(x * y) \mod n = 1$. (Only exists if $x$ is relatively prime to $n$)

Example of Additive Inverse: $n = 20, x = 4, y = 16$ since $4 + 16 \mod 20 = 20$

Example of Multiplicative Inverse: $n = 20, x = 3, y = 7$ since $3 * 7 \mod 20 = 21 \mod 20 = 1$

When $n = p_1 * p_2 * \ldots p_r$ where $p_1, \ldots, p_r$ are distinct prime numbers

$$
x^y \mod n = x^y \mod \phi(n) \mod n
$$

Special case of this: if $y = 1 \mod \phi(n)$, then $x^y \mod n = x \mod n$ for any integer $x$
1.3.1 Chinese Remainder Theorem

If \( z_1, z_2, \ldots, z_k \) are relatively prime, then given

\[
\begin{align*}
  v \mod z_1 &= x_1 \\
  v \mod z_2 &= x_2 \\
  \vdots \\
  v \mod z_k &= x_k
\end{align*}
\]

for some value \( v \), let:

- \( m_i = z_1 \ast z_2 \ast \ldots \ast z_{i-1} \ast z_{i+1} \ast \ldots \ast z_k \)
- \( c_i = m_i \ast (m_i^{-1} \mod z_i) \) (where \( m_i^{-1} \) is the multiplicative inverse of \( m_i \mod z_i \))
- \( n = z_1 \ast z_2 \ast \ldots \ast z_k \)

then \( v \mod n = (x_1 \ast c_1 + x_2 \ast c_2 + \ldots + x_k \ast c_k) \mod n \)

**Example:** Given \( v \mod 5 = 2, v \mod 13 = 3 \), find \( v \)

\[
\begin{align*}
z_1 &= 5 & x_1 &= 2 \\
z_2 &= 13 & x_2 &= 3
\end{align*}
\]

So,

\[
\begin{align*}
m_1 &= 13 & c_1 &= 13 \ast 2 = 26 \\
m_2 &= 5 & c_2 &= 5 \ast 8 = 40 \\
n &= 65
\end{align*}
\]

which means

\[
v \mod 65 = (2 \ast 26 + 3 \ast 40) \mod 65
\]

\[
= 42 \mod 65
\]

1.4 RSA Public Key Encryption and Digital Signature Algorithm

Steps:

1. Choose two very large prime numbers \( p \) and \( q \) such that \( p \ast q \) is more than 1024 bits long
2. Assign \( n = p \ast q, \phi(n) = (p - 1) \ast (q - 1) \)
3. Choose \( e \) such that \( e \) is relatively prime to \( \phi(n) \)
4. Calculate \( d \) such \( d \) is the multiplicative inverse of \( e \mod \phi(n) \)

\(<e, n>\) becomes the public key

\(<d, n>\) becomes the private key

Given \( m \) is the plain text message to be sent and \( c \) is the cipher (encrypted) text message:

- \( c = m^e \mod n \)
- \( m = c^d \mod n \)

Note that no fixed block size or key size is used, however, the length of \( m \) in bits must be less than the length of \( n \) in bits (i.e., \( m < n \))
1.4.1 Why does \( m \) need to be less than \( n \)?

Assume Bob’s public key is \( K^+_b = <e, n> \) and the private key is \( K^-_b = <d, n> \)
Given message \( m: c = m^e \mod n \)
To decrypt \( c \), we compute \( c^d \mod n \)
\[
c^d \mod n = (m^e \mod n) \mod n \\
= (m^e - k * n)^d \mod n \quad \text{(where \( k \) is some integer)} \\
= (m^{e*d} - (k * n)^d ) \mod n \\
= m^{e*d} \mod n \\
\text{(other term is cancelled out since it has an \( n \) term in it)} \\
= m^e \mod \phi(n) \mod n \\
= m \mod n \quad \text{(since \( d \) is the multiplicative inverse of \( e \mod \phi(n) \))} \\
= m, \text{ if } m < n
\]

Although the public key \( <e, n> \) is known and \( d \) can be obtained using Euclid’s algorithm, it is still very difficult to break RSA. The reason is \( \phi(n) \) is still unknown since \( p \) and \( q \) are unknown and computationally infeasible to determine given only \( n \)

1.4.2 Vulnerabilities of RSA

1. As the public key is known to everyone, there is a problem when only a limited, known set of messages (e.g., names of cities) are to be encrypted. Trudy can encrypt the entire list using the public key and determine what the encrypted message is that Bob sends to Alice.

2. A common choice for \( e \) is 3 (\( e \) can be small, but then \( d \) must be large).
   In this case, \( c = m^3 \mod n \). If \( m \) is very small compared to \( n \), \( c = m^3 \).
   Trudy can get \( m \) by computing the cube root of \( c \).

3. If a person is sending the same message to multiple people, the Chinese Remainder Theorem can be used to calculate \( m \).

4. The message can be modified. If \( m^e \mod n \) is being sent, then an adversary can do this: \( (m^e \mod n)^2 \mod n = (m^2)^e \mod n \)
   This allows the adversary to change the message being sent. This can also be applied to signing a message. An adversary is able to modify what a person has signed. This is not likely to happen commonly because the chance of the modified message making sense is very small.

The first three vulnerabilities can be solved by using a large random number for padding.

1.4.3 Example of using the Chinese Remainder Theorem to Break RSA:

Message \( m \) is sent to three people with public keys \( <3, n_1> \), \( <3, n_2> \), \( <3, n_3> \).
The adversary has access to:

\[ m^3 \mod n_1 \]
\[ m^3 \mod n_2 \]
\[ m^3 \mod n_3 \]

Using the Chinese Remainder Theorem, the adversary finds \( m^3 \mod n_1 \times n_2 \times n_3 \) since \( n_1 \), \( n_2 \), and \( n_3 \) are likely to be relatively prime to each other. Given that \( m < (n_1, n_2, n_3) \), \( m^3 \mod n_1 \times n_2 \times n_3 = m^3 \), then \( m \) can be discovered by taking the cube root of the quantity obtained using the Chinese Remainder Theorem.