RSA Encryption Algorithm

The security depends on the difficulty of factoring large numbers:

$$\text{Given } n,$$
$$\text{find primes } p, q \text{ where } p \times q = n$$

Refer to `getLargePrime.py` for details about generating large primes.

The length of p and q should be 1024 bits.

Euler's Theorem:

$a^{\varphi(m)} \bmod m = 1$ (m must be a prime OR gcd(a,m) = 1, $\varphi(m)$ is called the Euler's Totient)

$\varphi(mn) = \varphi(m)\varphi(n)$ (gcd(m, n) = 1)

1. RSA for Encryption:

Key Generation:

$$\text{Generate large primes: } p, q$$
$$n = p \times q$$
$$\text{Calculate: } \varphi(n) = \varphi(p \times q) = (p-1)(q-1)$$
$$\text{Choose random } e \text{ that } \gcd(e, \varphi(n)) = 1$$

(Refer to `EulerTotient.py` for the method to get e)

$$\text{Compute } d \text{ where } de \bmod(\varphi(n)) = 1$$
$$\text{Public Key: } pk = (n, e)$$
$$\text{Secret Key: } sk = (n, d)$$

After key generation, $p, q, \varphi(n)$ have to be securely destroyed.

$\varphi(n)$ also need to be destroyed because $\varphi(n) = (p-1)(q-1), n = pq$,which is easy to reconstruct p and q.

Encrypt:

$$\text{Given message } m, \text{ public key } PK,$$
$$c = m^e \bmod n$$

(Length of message m is smaller than length of n)

Decrypt:

$$\text{Given cipher } c, \text{ secret key } SK,$$
$$m = c^d \bmod n$$

Correctness:

$$m = c^d \bmod n = (m^e \bmod n)^d \bmod n = m^{de} \bmod n = m^{k\varphi(n)+1} \bmod n (k \in Z^+)$$

If m and n are relative primes:

$$m^{k\varphi(n)+1} \bmod n = ((m^{\varphi(n)})^k \cdot m) \bmod n = ((m^{\varphi(n)})^k \bmod n \cdot m \bmod n) \bmod n$$
$$\because n > m,$$
$$\therefore m \bmod n = m,$$
$$m^{k\varphi(n)+1} \bmod n = ((m^{\varphi(n)} \bmod n)^k \bmod n \cdot m) \bmod n$$
$$\because m^{\varphi(n)} \bmod n = 1,$$
$$\therefore m^{k\varphi(n)+1} \bmod n = (1^k \bmod n \cdot m) \bmod n = m$$

If m and n are NOT relative primes:

$$\because m < n, n \text{ only has 4 factors: } \{1, p, q, n\}, \text{p q are co-primes,}$$
$$\therefore \text{either } m = cp \text{ or } m = cq (c \in Z^+)$$
$$\text{If } m = cp, \gcd(m, q) = 1$$
$$\text{then } m^{\varphi(q)} \bmod q = 1, m^{k\varphi(q)\varphi(p)} \bmod q = 1 \text{(easy to prove)}$$
$$\because \varphi(n) = \varphi(p)\varphi(q)$$
$$\therefore m^{k\varphi(n)} \bmod q = 1$$
$$\text{let } m^{k\varphi(n)} = rq + 1 (r \in Z^+)$$
$$m^{k\varphi(n)+1} \bmod n = (m(rq+1)) \bmod n = (cprq + cp) \bmod n = (crn + m) \bmod n = m$$

Therefore, no matter whether m and n are relative primes or not, the encrypted message can always be correctly decrypted.

2. RSA for Digital Signature:

The key generation procedures for RSA digital signature are very similar to RSA encryption.

Note: an RSA key pair should not be used for digital signature and encryption simultaneously. A simple example of an attack is someone might ask the victim to decrypt a message with the private key. If the attacker put H(m) as the message, the victim might be tricked to sign on a message without knowledge about the contents.