RSA Encryption Algorithm

The security depends on the difficulty of factoring large numbers:



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

The length of p and q should be 1024 bits.

Euler’s Theorem:

(m must be a prime OR gcd(a,m) = 1,  is called the Euler’s Totient)

(gcd(m, n) = 1)

1. RSA for Encryption:

Key Generation:



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



After key generation, have to be securely destroyed.

also need to be destroyed because ,which is easy to reconstruct p and q.

Encrypt:



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

Decrypt:



Correctness:



If m and n are relative primes:



If m and n are NOT relative primes:



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.

