

Cryptography Mathematics and Basic Implementation

Christian Chinchole

July 7, 2023

1 RSA

The RSA algorithm is asymmetric, meaning it works with two keys: the public for encryption and the private for decryption.

Important variables are as follows:

- P, Q : are the primes.
- N : $p \cdot q$
- e : Encryption Exponent
- d : Decryption Exponent
- N, e pair: form the public key
- N, d pair: form the private key

1.1 How are the keys generated?

1. First find the LCM of $(p-1)(q-1)$
2. $d = e^{-1} \mod \text{LCM}$
3. $n = p \cdot q$
4. $dP = d \mod (p-1)$
 $dQ = d \mod (q-1)$
 $qInv = q^{-1} \mod p$

1.2 Pairwise Testing

The pairwise consistency test is used to check that the public and private exponent are suitable for encryption/decryption.

For k between $1 < k < (n-1)$:

$$k = (k^e)^d \mod n$$

1.3 Encryption

$$c = m^e \mod n$$

1.4 Decryption

The standard method: $m = c^d \mod n$

An exponentially faster method is to use the Chinese Remainder Theorem components:

$$\begin{aligned}m_1 &= c^{dP} \mod p \\m_2 &= c^{dQ} \mod q \\h &= (qInv)(m_1 - m_2) \mod p \\m &= m_2 + h.q\end{aligned}$$

2 Prime Generation

Referred to SP 186-56F.3

A prime number will be generated then checked that there exists a number to satisfy

$$p^{-1} \mod e$$

It is recommended that a length of n must be atleast 1024 bits meaning that p and q will each be 512 bits respectively. According to SP800-57, the strength will be associated to the length of bits. For example $1024 \rightarrow 80$, $2048 \rightarrow 112$, and $3072 \rightarrow 128$. Then for M-R testing, error trials will be calculated to satisfy $2^{(-strength)}$ so for 1024 it would be 2^{-80} . It is also accepted to use 2^{-100} for all prime lengths as it was often used in the past and considered acceptable in most applications.

2.1 Integer Trial Division

Referred to SP 186-56C.7

Trial Division is recommended to only be used for ≤ 10 digit numbers.

1. Start with a Prime p to be checked.
2. Make a table of primes less than \sqrt{p} (Typically done with sieving)
3. Divide p by every prime within the table, if it is divisible then fail and declare composite.
4. If it succeeds then call prime.

2.2 Sieving

Refereed to SP186-56C.8

Variables:

1. $Y_0, Y_0 + 1, \dots, Y_0 + J$ where J is the final addend
2. Find a factor base of all primes p_j from 2 to a limit L . L is arbitrary, but a good value is 10^3 to 10^5

Steps:

1. $S_j = Y_0 \mod p_j$ for all p_j in the factor base.
2. Initialize an array of length $J + 1$ to zero
3. Starting at $Y_0 - S_j + p_j$ let every p_j^{th} element of the array be set to 1. Do this for the entire length of the array and for every j .
4. When finished, every location in the array that has the value 1 is divisible by some small prime, and is therefore a composite.

Sieving with this method has an efficiency of approximately $M \log \log L$ where M is the length of the sieve interval. If $L = 10^5$ then the sieve will remove about 96% of all composites.

2.3 Probable Prime

1. Starting with a randomly generated number, n .
2. Perform trial division checking that the gcd of $n - 1$ and the primes is equivalent to 1 excluding 2.

This is done by:

1. Loop through from 1 to number of trial divisions.
2. Create an array of moduli from $\text{mod} = n \bmod \text{primes}[i]$
3. Loop through from 1 to number of trial divisions.
4. Check that the number's bit length is less than or equal to 31, the δ is not greater than unsigned long length and that the square of the prime is not larger than $n + \delta$
5. If $\text{mod}[i] + \delta \% \text{primes}[i] == 0$ then add 2 to δ . If $\delta > \text{maxdelta}$ then restart from n random generation. If not then redo the second loop.
6. Once this succeeds add the δ to n and confirm that the bit length of n is still the correct amount.