RSA Algorithm

Compiled by - Nazmus Sakib Akash

Cryptography & RSA Algorithm

Cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages.

- Cryptography can be broken down such as -
    1. Symmetric Cryptography : both the sender and receiver share the same key.
    2. Asymmetric Cryptography : uses pairs of keys: *public keys* which may be disseminated widely, and *private keys* which are known only to the owner. This is also known as *Public-key Cryptography.*
- RSA Algorithm is a type of Asymmetric Cryptography.

RSA (Rivest–Shamir–Adleman) Algorithm

- Ronald Rivest, Adi Shamir and Leonard Adleman proposed the asymmetric RSA cryptosystem in 1977.
- Until now, RSA is the most widely use asymmetric cryptosystem although elliptic curve cryptography (ECC) becomes increasingly popular
- In RSA, the asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers known as the "factoring problem".

RSA Algorithm Explained

Suppose, two people named Alice and Bob want to communicate where Bob wants to send a message to Alice. An eavesdropper by the name of Eve is trying to listen into Bob's message.

In RSA, there will be a pair of keys. Let's call the **encryption** key "**e**" which will be **the public key** while the **decryption** key is "**d**" which is **the private key**.

The public key in RSA is created based on *two large prime numbers*, along with an *auxiliary value*, **N** and sent from Alice to Bob (and Eve since she is snooping).

There is a Trapdoor One-Way Function working in the background.

Trapdoor One-Way Function

---------------------------------------> *Easy to compute*

$$m^e \pmod{N} = C$$

(Bob uses this formula to encrypt the message, **m** after receiving **e and N** from Alice and then sends encrypted message, **C** to Alice)

<--------------------------------------- *Difficult to compute*

$$?^e \pmod{N} = C$$

Even with the encrypted message, C and the public key, e with the auxiliary value, N, Eve cannot decrypt the message, m easily. Hence, a trapdoor is needed!

Public & Private Exponents

Since Bob got the public exponent, e and the auxiliary value N, he encrypted his message as such - $\boldsymbol{m^e \ (mod\ N) = C,}$ C being his encrypted message.
The public encryption key is {e, N}

Now, to decipher this, the value of **d** (the private key) is necessary. With the help of **d**, Alice will decrypt the encrypted message **C** as such - $\boldsymbol{C^d \ (mod\ N) = m}$
The private encryption key is {d, N}

So, from Bob and Alice's computations, we see that

$$m^{ed}(\text{mod } N) = m$$

We need to generate such exponents **e** and **d** and the auxiliary value, N

Key Generation Steps 1/5

The keys for the RSA algorithm are generated in the following way:

1. **Choose two distinct prime numbers P_1 and P_2**
   - For security purposes, the integers P_1 and P_2 should be chosen at random, and should be similar in magnitude but differ in length by a few digits to make factoring harder.
   - P_1 and P_2 are kept secret.

   For our problem, let's assume that $P_1 = 53$ and $P_2 = 59$.

2. **Calculate $N = P_1 * P_2$**

   N = 53*59 = 3127

Key Generation Steps 2/5

3. **Compute $\boldsymbol{\Phi(N) = (P_1-1) * (P_2-1)}$**

Note: Φ(N) is the PHI function of N and it refers to how many integers are less than or equal to N that do not share any common factors with N.

Derivation: For prime numbers - $\Phi(P) = P-1$

Also, $\Phi(A*B) = \Phi(A) * \Phi(B)$

So, $\Phi(N) = \Phi(P_1) * \Phi(P_2)$

=> $\mathbf{\Phi(N) = (P_1-1) * (P_2-1)}$

For our problem, $\Phi(N) = 52 * 58 = 3016$

Key Generation Steps 3/5

4. Select the public exponent *e ε {1, 2, …, Φ(N)-1} and e is coprime with N and Φ(N)*

In our problem, *e can be selected as 3*.

Φ(N) = 3016. So e = 3 lies between 1 and Φ(N) which fulfills the first condition.

Also, 3016 is not divisible by 3 so, e is coprime with Φ(N). Similarly, 3127 is not divisible by 3 and so, e is coprime with N. Thus, e = 3 fulfills the second condition.

Key Generation Steps 4/5

5. Choose the Private Exponent, d such that -

$$\mathbf{d = (k*\Phi(N) + 1) / e}$$

This is derived from Euler's theorem - $m^{\Phi(N)} = 1 \pmod N$

=> $m^{\Phi(N)*k} = 1 \pmod N$ [Since $1^k = 1$] => $m * m^{\Phi(N)*k} = m \pmod N$ [Since $1*m = m$]

$$\mathbf{=> m^{\Phi(N)*k+1} = m \pmod N}$$

Previously, we established $m^{ed} = m \pmod N$

So, $ed = k*\Phi(N) + 1$ => $\mathbf{d = (k*\Phi(N) + 1) / e}$

Key Generation Steps 5/5

Using $d = (k*\Phi(N) + 1) / e$ on our example -

$$d = (2 * 3016 + 1) \backslash 3 = 2011$$

For calculating d, we need the random value k. The value of k will be an integer value for which d is an integer and not a fraction.

Similarly, if we selected k = 5, we would find another possible value of d -

$$d = (5 * 3016 + 1) \backslash 3 = 5027$$

Implementation Details

- The RSA cryptosystem uses only one arithmetic operation (modular exponentiation) which makes it conceptually a simple asymmetric scheme
- Even though conceptually simple, due to the use of very long numbers, RSA is orders of magnitude slower than symmetric schemes, e.g., DES, AES
- When implementing RSA (esp. on a constrained device such as smart cards or cell phones) close attention has to be paid to the correct choice of arithmetic algorithms

Attacks and Countermeasures 1/3

There are two distinct types of attacks on cryptosystems

- **Analytical attacks** try to break the mathematical structure of the underlying problem of RSA
- **Implementation attacks** try to attack a real-world implementation by exploiting inherent weaknesses in the way RSA is realized in software or hardware

Attacks and Countermeasures 2/3

RSA is typically exposed to these analytical attack vectors -

Mathematical attacks

- The best known attack is factoring of N in order to obtain $\Phi(N)$
- Can be prevented using a sufficiently large modulus N
- The current factoring record is 664 bits. Thus, it is recommended that n should have a bit length between 1024 and 3072 bits

Protocol attacks

- Exploit the malleability of RSA, i.e., the property that a ciphertext can be transformed into another ciphertext which decrypts to a related plaintext – without knowing the private key
- Can be prevented by proper padding

Attacks and Countermeasures 3/3

Implementation attacks can be one of the following -

Side-channel analysis

- Exploit physical leakage of RSA implementation (e.g., power consumption, EM emanation, etc.)

Fault-injection attacks

- Inducing faults in the device while CRT is executed can lead to a complete leakage of the private key

Summary

- RSA is the most widely used public-key cryptosystem
- RSA is mainly used for key transport and digital signatures
- The public key e can be a short integer, the private key d needs to have the full length of the modulus n
- RSA relies on the fact that it is hard to factorize n
- Currently 1024-bit cannot be factored, but progress in factorization could bring this into reach within 10-15 years. Hence, RSA with a 2048 or 3076 bit modulus should be used for long-term security
- A naïve implementation of RSA allows several attacks, and in practice RSA should be used together with padding