RSA Algorithm in Cryptography

RSA(**R**ivest-**S**hamir-**A**dleman) Algorithm is an **asymmetric** or **public-key cryptography** algorithm which means it works on two different keys: **Public Key** and **Private Key**. The Public Key is used for **encryption** and is known to everyone, while the Private Key is used for **decryption** and must be kept secret by the receiver. RSA Algorithm is named after Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman, who published the algorithm in 1977.

Example of Asymmetric Cryptography:
If Person **A** wants to send a message securely to Person **B**:

- Person A **encrypts** the message using Person B's **Public** Key.
- Person B **decrypts** the message using their **Private** Key.
-

RSA Algorithm
*RSA Algorithm is based on **factorization** of large number and **modular arithmetic** for encrypting and decrypting data. It consists of three main stages:*

1. ***Key Generation:** Creating Public and Private Keys*
2. ***Encryption:** Sender encrypts the data using Public Key to get **cipher text**.*
3. ***Decryption:** Decrypting the **cipher text** using Private Key to get the original data.*

1. Key Generation

- Choose two large prime numbers, say **p** and **q**. These prime numbers should be kept secret.
- Calculate the product of primes, **n = p * q**. This product is part of the public as well as the private key.
- Calculate Euler Totient Function**Φ(n)** as **Φ(n) = Φ(p * q) = Φ(p) * Φ(q) = (p – 1) * (q – 1).**
- Choose encryption exponent **e**, such that
    - $1 < e < \Phi(n)$, and
    - $\gcd(e, \Phi(n)) = 1$, that is e should be co-prime with Φ(n).
- Calculate decryption exponent **d,** such that
    - **$(d * e) \equiv 1 \bmod \Phi(n)$**, that is d is modular multiplicative inverse of **e** mod Φ(n). Some common methods to calculate multiplicative inverse are: Extended Euclidean Algorithm, Fermat's Little Theorem, etc.

- We can have multiple values of d satisfying **(d * e) ≡ 1 mod Φ(n)** but it does not matter which value we choose as all of them are valid keys and will result into same message on decryption.

Finally, the **Public Key = (n, e)** and the **Private Key = (n, d)**.

2. Encryption

To encrypt a message **M**, it is first converted to numerical representation using ASCII and other encoding schemes. Now, use the public key (n, e) to encrypt the message and get the cipher text using the formula:

C = Me mod n*, where C is the Cipher text and e and n are parts of public key.*

3. Decryption

To decrypt the cipher text **C**, use the private key (n, d) and get the original data using the formula:

M = Cd mod n, *where M is the message and d and n are parts of private key.*

Example of RSA Algorithm

Idea behind RSA Algorithm

The idea of RSA is based on the fact that it is difficult to factorize a large integer. The Public Key is (n, e), where n and e are publicly known, while the Private Key is (n, d). Since only the receiver knows the value of d, only they can decrypt the message. **But is it possible to find the value of d using n and e?**

We know that **(d * e) ≡ 1 mod Φ(n)**, so if we can calculate the value of **Φ(n),** we can find the value of **d**. But **Φ(n) = (p – 1) * (q – 1).** So, we need the value of p and q. Now, one might think that it's quite easy to find the value of p and q as **n = p * q** and n is already publicly known but RSA Algorithm takes the value of p and q to be very large which in turn makes the value of n extremely large and factorizing such a large value is computationally impossible.

Therefore encryption strength lies in the values of **p** and **q**. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken shortly. But till now it seems to be an infeasible task.

Note: If someone gets to know the value of **p** and **q**, then he can calculate the value of **d** and decrypt the message.

Advantages

- **Security:** RSA algorithm is considered to be very secure and is widely used for secure data transmission.
- **Public-key cryptography:** RSA algorithm is a public-key cryptography algorithm, which means that it uses two different keys for encryption and decryption. The public key is used to encrypt the data, while the private key is used to decrypt the data.
- **Key exchange:** RSA algorithm can be used for secure key exchange, which means that two parties can exchange a secret key without actually sending the key over the network.
- **Digital signatures:** RSA algorithm can be used for digital signatures, which means that a sender can sign a message using their private key, and the receiver can verify the signature using the sender's public key.
- **Widely used:** Online banking, e-commerce, and secure communications are just a few fields and applications where the RSA algorithm is extensively developed.

Disadvantages

- **Slow processing speed:** RSA algorithm is slower than other encryption algorithms, especially when dealing with large amounts of data.
- **Large key size:** RSA algorithm requires large key sizes to be secure, which means that it requires more computational resources and storage space.
- **Vulnerability to side-channel attacks:** RSA algorithm is vulnerable to side-channel attacks, which means an attacker can use information leaked through side channels such as power consumption, electromagnetic radiation, and timing analysis to extract the private key.
- **Limited use in some applications:** RSA algorithm is not suitable for some applications, such as those that require constant encryption and decryption of large amounts of data, due to its slow processing speed.
- **Complexity:** The RSA algorithm is a sophisticated mathematical technique that some individuals may find challenging to comprehend and use.
- **Key Management:** The secure administration of the private key is necessary for the RSA algorithm, although in some cases this can be difficult.
- **Vulnerability to Quantum Computing:** Quantum computers have the ability to attack the RSA algorithm, potentially decrypting the data.