CRYPTO M2 (p-5)

techworldthink • March 19, 2022

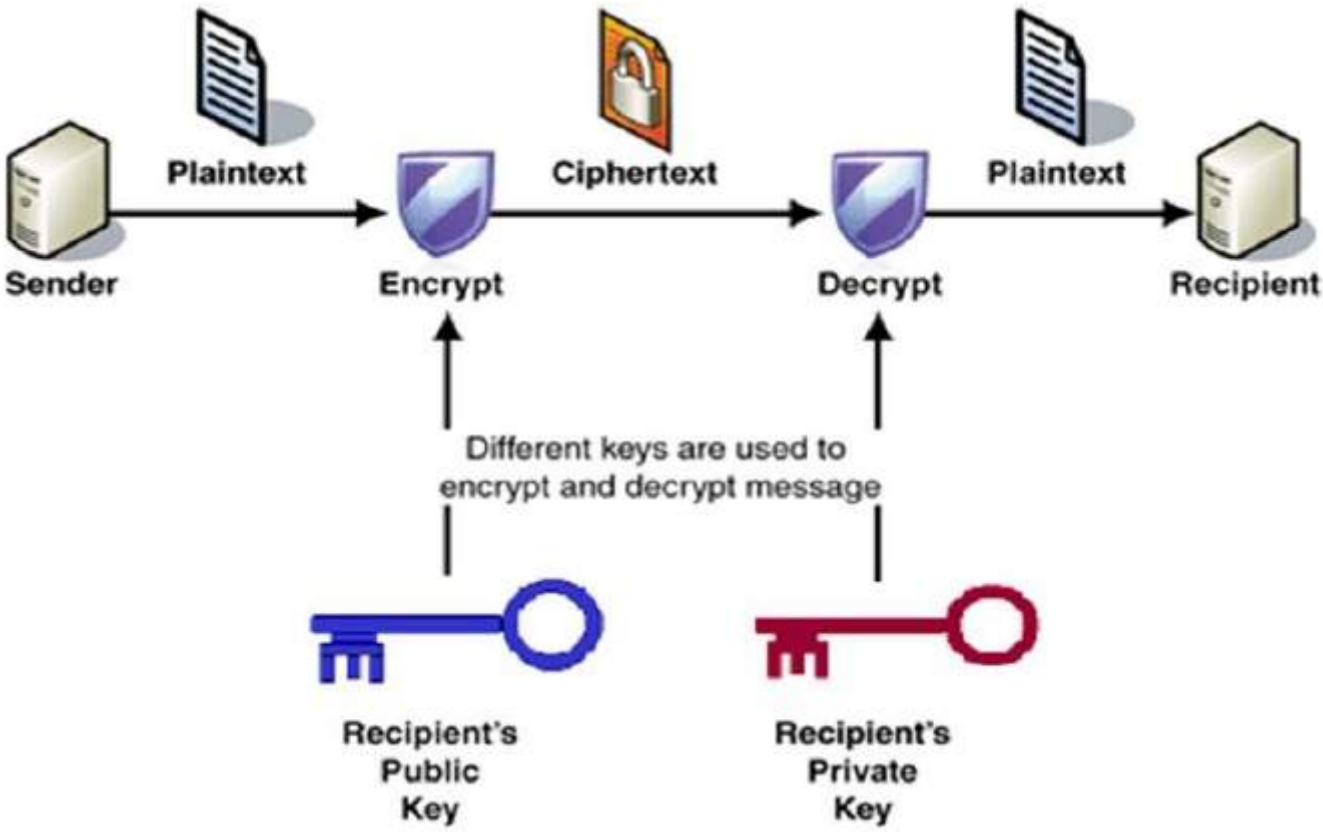
Public Key Cryptography

Unlike symmetric key cryptography, we do not find historical use of public-key cryptography. It is a relatively new concept.

Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations were involved in the classified communication.

With the spread of more unsecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems.

The process of encryption and decryption is depicted in the following illustration −



Public Key Cryptography

The most important properties of public key encryption scheme are −

- Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.
- Each receiver possesses a unique decryption key, generally referred to as his private key.
- Receiver needs to publish an encryption key, referred to as his public key.
- Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
- Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the ciphertext and the encryption (public) key.
- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys.

RSA Cryptosystem

This cryptosystem is one the initial system. It remains most employed cryptosystem even today. The system was invented by three scholars Ron Rivest, Adi Shamir, and Len Adleman and hence, it is termed as RSA cryptosystem.

We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below −

- Generate the RSA modulus (n)
- Select two large primes, p and q.
- Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.

- Find Derived Number (e)
- Number e must be greater than 1 and less than (p − 1)(q − 1).
- There must be no common factor for e and (p − 1)(q − 1) except for 1. In other words two numbers e and (p – 1)(q – 1) are coprime.
- Form the public key
- The pair of numbers (n, e) form the RSA public key and is made public.
- Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.
- Generate the private key
- Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
- Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1).
- This relationship is written mathematically as follows −

```
ed = 1 mod (p – 1)(q – 1)
```

Encryption and Decryption

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy.

Interestingly, RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n.

RSA Encryption

- Suppose the sender wish to send some text message to someone whose public key is (n, e).

- The sender then represents the plaintext as a series of numbers less than n.
- To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple mathematical step as −

```
C = P^e mod n
```

- In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n. This means that C is also a number less than n.
- Returning to our Key Generation example with plaintext P = 10, we get ciphertext C −

```
C = 10^5 mod 91
```

RSA Decryption

- The decryption process for RSA is also very straightforward. Suppose that the receiver of public-key pair (n, e) has received a ciphertext C.
- Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

```
Plaintext = C^d mod n
```

- Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29 −

```
Plaintext = 82^29 mod 91 = 10
```

RSA Analysis

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

- Encryption Function − It is considered as a one-way function of converting plaintext into ciphertext and it can be reversed only with the knowledge of private key d.
- Key Generation − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n. It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if a technique for factoring efficiently is developed then RSA will no longer be safe.

The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.

> ***Eg: Perform encryption and decryption using RSA Algorithm with parameters: P=17, q = 11, e = 7, M = 88. Explain the steps in detail.***

n = p.q = 187

φ(n)= (p-1)(q-1) = (16x10) = 160

e = 7

d = (e^-1) mod φ(n) => (7^-1) mod 160 = 23

> *d = (e^-1) mod φ(n)*

> *ed = 1 mod φ(n)*

> *ed mod φ(n) = 1*

> *7.x mod 160 =1*

> *7.x = 160+1*

> *so, x = 161/7=23*

M = 88

C = (M^e) mod n = (88^7) mod 187 = 11

M = (C^d) mod n = (11^23) mod 187 = 88

..... **Explanations** .....

p,q are 2 prime numbers

n = p*q

calculate euler totient, φ(n)= (p-1)(q-1)

which means, φ(n) *numbers are relatively prime to n*

select integer e as encryption key, gcd(φ(n),e)=1 , 1< e < φ(n)

calculate decryption key d , d = (e^-1) mod φ(n)

public key = {e,n}

private key = {d,n}

Ciphertext, C = (M^e) mod n

plaintext, M = (C^d) mod n

Diffie Hellman Key exchange

Diffie-Hellman key exchange, also called exponential key exchange, is a method of digital encryption that uses numbers raised to specific powers to produce decryption keys on the basis of components that are never directly transmitted, making the task of a would-be code breaker mathematically overwhelming.

To implement Diffie-Hellman, the two end users Alice and Bob, while communicating over a channel they know to be private, mutually agree on positive whole numbers p and q, such that p is a prime number and q is a generator of p. The generator q is a number that, when raised to positive whole-number powers less than p, never produces the same result for any two such whole numbers. The value of p may be large but the value of q is usually small.

Once Alice and Bob have agreed on p and q in private, they choose positive whole-number personal keys a and b, both less than the prime-number modulus p. Neither user divulges their personal key to anyone; ideally they memorize these numbers and do not write them down or store them anywhere. Next, Alice and Bob compute public keys $a*$ and $b*$ based on their personal keys according to the formulas

$a* = q\hat{}a \bmod p$

and

$b* = q\hat{}b \bmod p$

The two users can share their public keys $a*$ and $b*$ over a communications medium assumed to be insecure, such as the Internet or a corporate wide area network (WAN). From these public keys, a number x can be generated by either user on the basis of their own personal keys. Alice computes x using the formula

$x = (b*)\hat{}a \bmod p$

Bob computes x using the formula

$x = (a*)\hat{}b \bmod p$

The value of x turns out to be the same according to either of the above two formulas. However, the personal keys a and b, which are critical in the calculation of x, have not been transmitted over a public medium. Because it is a large and apparently random number, a potential hacker has almost no chance of correctly guessing x, even with the help of a powerful computer to conduct millions of trials. The two users can therefore, in theory, communicate privately over a public medium with an encryption method of their choice using the decryption key x.

The most serious limitation of Diffie-Hellman in its basic or "pure" form is the lack of authentication. Communications using Diffie-Hellman all by itself are vulnerable to man in the middle attacks. Ideally, Diffie-Hellman should be used in conjunction with a recognized authentication method such as digital signatures to verify the

identities of the users over the public communications medium. Diffie-Hellman is well suited for use in data communication but is less often used for data stored or archived over long periods of time.

> ***Eg : Apply Diffie-Hellman key exchange algorithm to compute the shared private key using the values P = 23, g = 9, a= 4, b= 3. Explain the steps in detail.***

private key of A = 4

private key of B = 3

prime no, P = 23

primitive root, G = 9

Share P&G (A->B or B->A)

find?

public key of A = (g^a) mod n = (9^4) mod 23 = 6

public key of B = (g^b) mod n = (9^3) mod 23 = 16

Exchange A&B public keys

find?

A, Secret key = (Bpublic ^ a) mod n = (16^4) mod 23 = 9

B, Secret key = (Apublic ^ a) mod n = (6^3) mod 23 = 9

> ***Extra notes***

- This algorithm is used to exchange the secret key between the sender and the receiver.
- This algorithm facilitates the exchange of secret key without actually transmitting it.

Let-

- Private key of the sender = Xs
- Public key of the sender = Ys
- Private key of the receiver = Xr
- Public key of the receiver = Yr

Step-01:

- One of the parties choose two numbers ‘a’ and ‘n’ and exchange with the other party.
- ‘a’ is the primitive root of prime number ‘n’.
- After this exchange, both the parties know the value of ‘a’ and ‘n’.

Step-02:

- Both the parties already know their own private key.
- Both the parties calculate the value of their public key and exchange with each other.

Sender calculate its public key as-

$$Y_s = a^{X_s} \bmod n$$

Receiver calculate its public key as-

$$Y_r = a^{X_r} \bmod n$$

Step-03:

- Both the parties receive public key of each other.
- Now, both the parties calculate the value of secret key.

Sender calculates secret key as-

$$\text{Secret key} = (Y_r)^{X_s} \bmod n$$

Receiver calculates secret key as-

$$\text{Secret key} = (Y_s)^{X_r} \bmod n$$

Primitive root ?

prime ,q = 7

primitive root ,p = 3

is 3 prime root ot 7?

check p^(1 to (q-1)) should have {1,2,3,.....,q-1}

3^1 mod 7 = 3

3^2 mod 7 = 2

3^3 mod 7 = 6

3^4 mod 7 = 4

3^5 mod 7 = 5

3^6 mod 7 = 1

={1,2,3,4,5,6} , so 3 is primitive root of 7

square root of large numbers

eg1:

31^500 mod 30 => (31-30)^500 mod 30 => 1

eg2:

242^329 mod 243 => (242-243)^329 mod 243 => (-1)^329 mod 243

=> 1^329 mod 243 gives 1 ,

so (-1)^329 mod 243 will be 243-1 -> 242