SECURITY FOR THE INTERNET OF THINGS

REPORT

YANG YUXIN

This course is mainly concerned about the security for the internet of things ,and compared the different method using in IOT.

In order to protect the security of information, we need to encrypt our data. First of all, we have to be able to generate random numbers and prime numbers, which is the basic knowledge of the security.

I.A SYMMETRICAL ENCRYPTION ALGORITHM FOR THE IOT

In the symmetric encryption algorithm, the data sender changes the plaintext (original data) and encryption key into complex encrypted ciphertext after being processed by a special encryption algorithm. After receiving the ciphertext, if the recipient wants to interpret the original text, it needs to decrypt the ciphertext using the encryption key and the inverse algorithm of the same algorithm to restore it to readable plaintext. In the symmetric encryption algorithm, only one key is used. Both sender and receiver use this key to encrypt and decrypt the data.

1. One Time Pad

According to Shannon's Theory ,when using One Time Pad for encryption, First of all ,we need to generate a random mask which is at least as long as our message by using the pseudo-random number generator, and the cryptogram is the bit-by-bit exclusive OR between the original message and this random sequence, then bit by bit exclusive OR between the cryptogram and this noise we can get the original message.

II. CONSTITUTION OF RANDOM GENERATORS

Random and pseudo-random number generators are one of the foundations of computer security.The first one is based on hardware,it’s not reproducible,the second one is based on algorithms.

1. LINEAR CONGRUENTIAL GENERATORS

The LCG algorithm is mathematically based on the formula:

$$X(n+1) = (a * X(n) + c) \% m$$

Where, the coefficients are:

Modulus m, $m > 0$

Coefficient a, $0 < a < m$

Increment c, $0 <= C < m$

Original value (seed) $0 <= X(0) < m$

The parameters c, m and a are sensitive, or directly affect the quality of pseudo-random number generation.

Generally speaking, m of high LCG is the exponential power of 2 (generally 2 ^ 32 or 2 ^ 64), because the modulo operation can truncate the rightmost 32 or 64 bits. Most compiler libraries use this theory to implement its pseudo-random number generator rand ().

The linear congruence method generates pseudo-random numbers, which roughly conforms to the uniform distribution. According to the central limit theorem, any distributed noise can become Gaussian noise by adding it repeatedly.

2. THE XORSHIFT

Xorshifts are linear operations. The left shift of a w-bit vector x by one bit,x1, can also be written as Lx where L is the w × w matrix with ones on its main superdiagonal and zeros elsewhere. Similarly, the right shift x1 can be written as Rx where R has ones on its main subdiagonal and zeros elsewhere. Matrices of the forms (I + La) and (I + Ra), where a $\in$ {1, . . . , w − 1}, are called left and right xorshift matrices, respectively. They represent left and right a-bit xorshift operations.

3. Blum Blum Shub

Blum Blum Shub takes the form

$$x_{n+1} = x_n^2 \bmod M$$

where M = pq is the product of two large primes p and q. At each step of the algorithm, some output is derived from xn+1; the output is commonly either the bit parity of xn+1 or one or more of the least significant bits of xn+1.

The seed x0 should be an integer that is co-prime to M (i.e. p and q are not factors of x0) and not 1 or 0.

The two primes, p and q, should both be congruent to 3 (mod 4) (this guarantees that each quadratic residue has one square root which is also a quadratic residue), and should be safe primes with a small gcd((p-3)/2, (q-3)/2) (this makes the cycle length large).

An interesting characteristic of the Blum Blum Shub generator is the possibility to calculate any xi value directly (via Euler's theorem):

$$x_i = \left(x_0^{2^i \bmod \lambda(M)}\right) \bmod M$$

III. OBTAINING LARGE PRIME NUMBERS

1. ERATOSTHENES' SIEVE

Although there are infinite numbers of prime numbers, people often ask about the number of prime numbers in a certain range. The sieve of Eratosthenes is a common algorithm to solve this problem. Eratosthene screening method is based on a basic property: any natural number greater than 1 is either a prime number itself or can be decomposed into the product of several prime numbers, and this decomposition is unique.

It is assumed that all numbers starting from the starting point (the starting point can be specified by the requirement) are prime numbers. Search forward from the starting point. If it is a prime number, mark its multiple (not exceeding the upper bound n) as a non prime number. For example, if 2 is a prime number, Mark 4, 6, 8,... The multiples of these 2 are non prime numbers, then mark the next... And so on.

Eratoseni screening method is simple to implement, but has the disadvantage of large demand for space. For this, we can use bitmap storage, which can greatly reduce the space demand.

2. Small theorem of Fermat

if P is a prime number and GCD (a, P) = 1 (A and P are coprime), then a ^ (p-1) ≡ 1 (mod p), i.e. (a ^ (p-1))% P = 1.

Take any prime number, such as 13. Consider a series of integers 1,2,3,4,5,6,7,8,9,10,11,12 from 1 to 12. Multiply these numbers by a number coprime with 13, such as 3, to get 3,6,9,12,15,18,21,24,27,30,33,36. For module 13, these numbers are congruent to 3,6,9,12,2,5,8,11,1,4,7,10. These residuals are actually the original 1,2,3,4,5,6,7,8,9,10,11,12, but in different order.

Multiply 1, 2, 3,..., 12, and the product is the factorial of 12!. Multiply 3, 6, 9,..., 36 and put forward the common factor 3. The product is 312 × 12!。 For module 13, these two products are congruent to 1,2,3,..., 12 series, although the order is not one-to-one correspondence, that is, 312x12! ≡12! mod 13。 Divide both sides by 12! 312 ≡ 1 mod 13. If we use p instead of 13 and X instead of 3, we get Fermat's small theorem XP-1 ≡ 1 mod P.

IV.RSA

In 1977, three mathematicians Rivest, Shamir and Adleman designed an algorithm to realize asymmetric encryption. This algorithm is named after the three of them and is called RSA algorithm. From then until now, RSA algorithm has been the most widely used "asymmetric encryption algorithm",whichi means:

1. Party B generates two keys (public key and private key). The public key is public and can be obtained by anyone, while the private key is confidential.
2. Party A obtains Party B's public key and uses it to encrypt the information.
3. Party B shall decrypt the encrypted information with the private key.

Step1 :Randomly select two unequal prime numbers P and Q.

Step2 :Calculate the product n of P and Q. The length of n is the key length. In practical application, the RSA key is generally 1024 bits, and 2048 bits in important occasions.

Step3:Calculate the Euler function of n φ (n).According to the formula:

$$\varphi(n) = (p-1)(q-1)$$

Step 4: Randomly select an integer e, provided that 1 < E< φ (n) , and E and φ (n) Coprime. In practical application, 65537 is often selected.

Step 5,:Calculate e for φ (n) Modular inverse element D.The so-called "modulo inverse element" means that there is an integer D, which can make ed be φ (n) The remainder of the division is 1.

$$ed \equiv 1 \pmod{\varphi(n)}$$

This equation is equivalent to

$$ed - 1 = k\varphi(n) \quad (k \in Z)$$

So, finding the modular inverse element D is essentially solving the following binary primary equation.

$$ex + \varphi(n)y = 1$$

This equation can be solved by "extended Euclidean algorithm".

Step 6: Encapsulate N and e into public keys and N and D into private keys. In summary is actually the process of calculating n, e, D

The function of PQ is to find n = = pq, and then find (p-1) (q-1) φ (n) , in φ (n) Random selection within the range is e, d = = k φ (n) Modular inverse element.

V.EL GAMEL

1. Introduction to ElGamal algorithm

ElGamal algorithm was proposed by tather ElGamal in 1985. It is an encryption system based on discrete logarithm problem. Like Ras algorithm, it can be used for both data encryption and digital signature. ElGamal algorithm is based on factor decomposition, while ElGamal algorithm is based on discrete logarithm problem. Compared with RSA algorithm, even if ElGamal algorithm uses the same private key to encrypt the same plaintext, the signatures obtained after each encryption are different, which effectively prevents replay attacks that may occur in the network.

2. Principle of ElGamal algorithm

1. ElGamal key generation

(1) Randomly select a large prime P, and P-1 is required to have a large prime factor. Then choose an primitive of module P α。 Add P and α Open.

(2) Randomly select an integer D as the key, $2 \leq D \leq P-2$.

(3) Calculate y= α^ D mod p, take y as the public key.

2. ElGamal encryption

(1) For plaintext m encryption, randomly select an integer k, $2 \leq K \leq P-2$

(2)C1 = α^ k mod p

(3)C2 = MY^k mod p

(4) Ciphertext is (C1, C2)

3. ElGamal decryption

Plaintext M can be obtained from ciphertext, M = C2 / C1 ^ D mod P