qround_chacha

Introduction to qround_salsa

Since qround_chacha is an improved version of qround_salsa, before analyzing qround_chacha, we should understand the algorithm principle of qround_salsa at beginning. The main process of the algorithm is as follows.

$$z_1 = y_1 \oplus ((y_0 + y_3) <<< 7)$$

$$z_2 = y_2 \oplus ((z_1 + y_0) <<< 9)$$

$$z_3 = y_3 \oplus ((z_2 + z_1) <<< 13)$$

$$z_0 = y_0 \oplus ((z_3 + z_2) <<< 18)$$

First of all, the data unit of the algorithm is word, a word is composed of 4 bytes, that is, 32 bits, and its input and output are 4 words. $\oplus$ means xor, + means addition modulo 2^{32}, and <<< means b-bit rotation of a 32-bit integer towards high bits. For computers, these operations are very easy to implement and very efficient. On the other hand, these seemingly simple operations combined can actually achieve the same level of security as any other operation option.

The first round of the algorithm is to add y_1 and y_3 modulo 2^{32}, then rotate the result to the left by 7 bits, and finally XOR with y1 to get z1. The next three operations are similar to the first round, but the initial word and the generated z are used for modulo addition. Also, the number of bits of the cyclic shift is also different. Algorithms are designed like this so that each bit in the plaintext affects many bits in the ciphertext, or each bit in the ciphertext is affected by many bits in the plaintext.

Comparing qround_chacha and qround_salsa

First of all, the data formats of qround_chacha and qround_salsa are the same, and the input and output are also the same. Also, the basic operation of the algorithm is the same. However, ChaCha applies the operations in a different order, and in particular updates each word twice rather than once. Specifically, ChaCha updates a, b, c, d as follows:

a += b; d $\oplus$=a; d <<<= 16;

c += d; b $\oplus$= c; b <<<= 12;

a += b; d $\oplus$= a; d <<<= 8;

c += d; b $\oplus$= c; b <<<= 7;

Although still the same basic operation, qround_chacha is more diffuse and efficient due to the different iterations of each round. The first round iterative algorithm of the two is similar, and the effect is almost the same. The difference is only reflected in the number of bits to rotate left. However, due to the assignment operation, at the end of the first round, the four initial values a and d of abcd have been confused and changed. From this point of view, qround_chacha seems to be twice as efficient as the previous algorithm.

An important indicator to measure a good cryptographic algorithm is diffusivity, that is, let each bit in the plaintext affect many bits in the ciphertext, or let each bit in the ciphertext be affected by many bits in the plaintext. Each assignment allows qround_chacha to diffuse more thoroughly through more iterations. The time complexity of the algorithm is almost unchanged, but the space overhead is slightly increased. These are the promotions of qround_chacha.