ASANSOL ENGINEERING COLLEGE

RC5 Algorithm

Report accomplished by

Aishwarya Ghosh

Roll no.: 10800219074
Reg. no.: 036064 of 2019-20

Sub name: **Cryptography and Network Security**

Sub code: **PEC-IT801B**

Year: 2023

Department Of Information Technology

Table of Content

INTRODUCTION

In cryptography, RC5 is a symmetric-key block cipher notable for its simplicity. Designed by Ronald Rivest in 1994, RC stands for "Rivest Cipher", or alternatively, "Ron's Code" (compare RC2 and RC4). The Advanced Encryption Standard (AES) candidate RC6 was based on RC5.
RC5 is a symmetric key block encryption algorithm designed by Ron Rivest in 1994. It is notable for being simple, fast (on account of using only primitive computer operations like XOR shift, etc.) and consumes less memory.

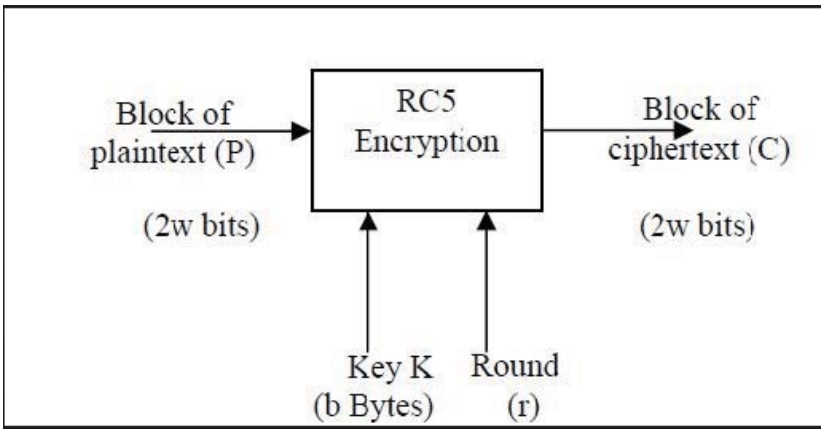

Figure 1: RC5 Encryption

Unlike many schemes, RC5 has a variable block size (32, 64 or 128 bits), key size (0 to 2040 bits) and number of rounds (0 to 255). The original suggested choice of parameters was a block size of 64 bits, a 128-bit key and 12 rounds.
Since at a time, RC5 uses 2 word blocks, the plain text block size can be 32, 64 or 128 bits.

For example:-

Key: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Plain Text: 00000000 00000000
Cipher Text: EEDBA521 6D8F4B15

RC5 is a block cipher and addresses two word blocks at a time.
Depending on input plain text block size, number of rounds and key size, various instances of RC5 can be defined and each instance is denoted as

RC5-w/r/b

Where, w = word size in bits,
r = number of rounds
and b= key size in bytes.

METHODOLOGY

In the RC5 algorithm, the input plain text block size, number of rounds and 8-bit bytes of the key can be of variable length.
The plain text message is divided into two blocks A and B each of 32 bits. Then two sub keys are generated S[0] and S[1]. These two sub keys are added into A and B respectively. This process produces C and D respectively and marks the end of the one-time operation.

Let us discuss the above-mentioned process in detail.

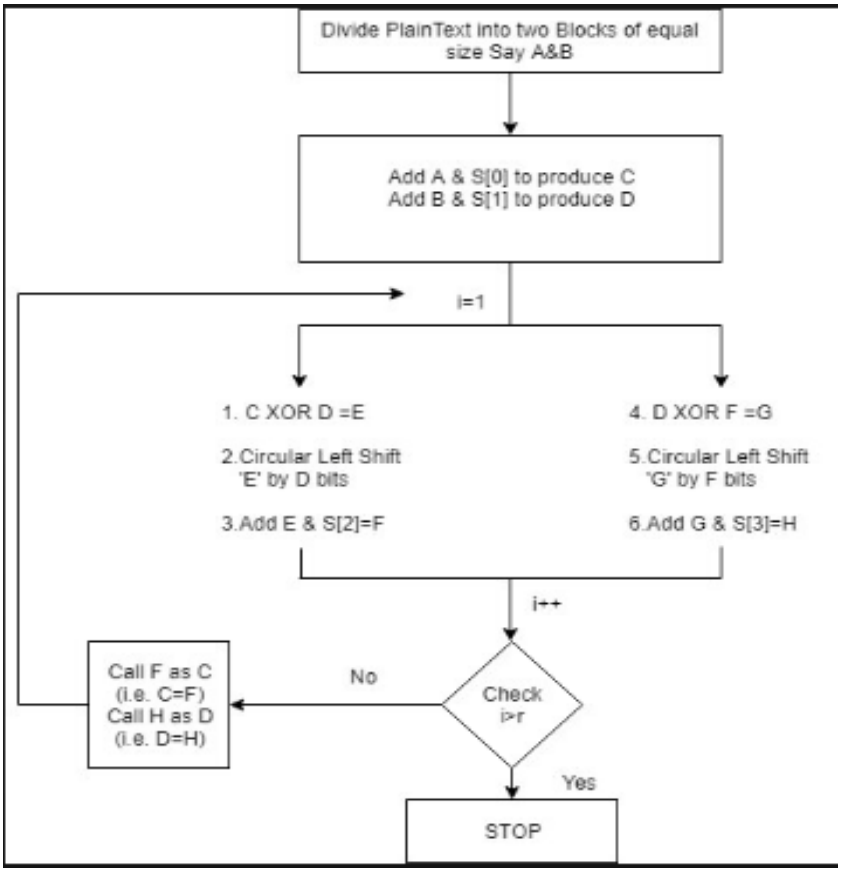

Figure 2: RC5 Algorithm

One Time Initial Operation:-

This process involves two steps:

- Step 1: Divide the plain text into two equal sizes block A and B.
- Step 2: Add S[0] to A and S[1] to B.

These operations are of mod 2^32 and generated C and D respectively**.**

Details of Round:-

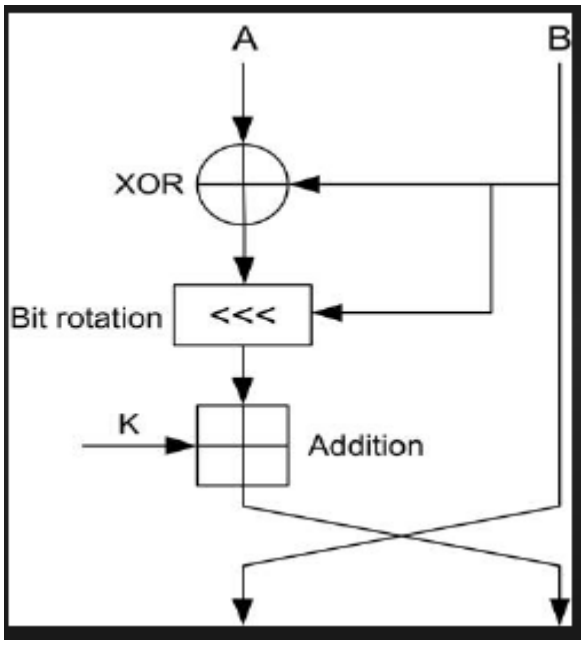

Figure 3: One Round in RC5

In this section, we will discuss the result for one round. The process for the first round will be the same for other rounds.

- **Step 1:** XOR C and D

This is the first step in every round where C and D are XOR to form E.

- **Step 2:** Circular left shift E

The output generated in step 1 i.e. E is now circular left-shifted by D positions.

- **Step 3:** Add E and next subkey

In this step, E is added to the next subkey to form the F.

- **Step 4:** XOR D and P

D and P are XORed to from G.

- **Step 5:** circular left shift G

The output generated in step 4 i.e. G is now circular left-shifted by F position.

- **Step 6:** Add G and next sub-key

In this step, G is added to the next sub-key to form the H.

- **Step 7:** Miscellaneous steps

In these steps, the checking is done to ensure that all-round is done properly.
This is performed using the following steps.

1. Increment i by 1
2. check if i < r

if i is less than r then rename F as C and H as D. If i is greater than r then the process will stop.

Subkey Creation:-

In RC5, subkey can be generated in two steps

- **Step 1:** Subkeys are generated
- **Step 2:** Subkeys which are generated in step 1 are mixed with the corresponding sub-portions of the original keys.

OUTCOMES

The outcome of the RC5 algorithm is cipher text, which is the encrypted form of the original plaintext. The encryption process involves several rounds of mixing and permutation operations on the plaintext, using a secret key that is known only to the sender and receiver. The resulting ciphertext is then transmitted over the network and decrypted by the receiver using the same key and decryption algorithm.

Encryption

We assume that the input block is given in two w-bit registers A and B. We also assume that key-expansion has already been performed, so that the array S[0...t - 1] has been computed. Here is the encryption algorithm in pseudo-code:

```
A = A + S[0];
B = B + Sill;
For i= l to r do
A = ((A | B) <<< B) + S[2. i];
B=((B|
```

The output is in the registers A and B.
We note the exceptional simplicity of this 5-line algorithm.
We also note that each RC5 round updates both registers A and B, whereas a "round" in DES updates only half of its registers. An RC5 "half-round" (one of the assignment statements updating A or B in the body of the loop above) is thus perhaps more analogous to a DES round.

Decryption

The decryption routine is easily derived from the encryption routine.

```
for i - r down to 1 do
B = ((B- S[2*i+ 1]) >>> A) ~A;
A=((A-S[2.i])>>>B)|
B = B – S[1];
d = d - S[0];
```

The security of the RC5 algorithm depends on the length of the key used, as well as the number of rounds performed during the encryption process. Generally, a longer key and more rounds result in stronger encryption, making it more difficult for an attacker to break the encryption and access the original plaintext. However, as with any encryption algorithm, there is always a risk that it may be vulnerable to attacks or exploits, and it is important to use strong encryption practices and keep software up-to-date to minimize these risks.

CONCLUSION

RC5 algorithm is designed to be simple, efficient, and secure, and has been widely used in both commercial and academic settings.

RC5 uses a variable block size, key size, and number of rounds, which allows it to be customized to meet specific security requirements. It operates on data blocks of 32, 64, or 128 bits, and can use keys of up to 2040 bits in length.

It allows a variable number of rounds and variable bit size key to add flexibility. Another advantage of using RC5 is that it requires less memory for execution. This feature enables RC 5 to be used for various purposes like desktop operation, smart cards, etc.

The security of RC5 is based on the difficulty of reversing the key schedule and the number of rounds used in the encryption process. While there have been some theoretical attacks against RC5, they have not been shown to be practical in real-world scenarios.

Overall, RC5 is a well-regarded encryption algorithm that continues to be used in various applications today. However, as with any cryptographic algorithm, its security must be evaluated periodically in light of new developments in cryptanalysis and computing technology.

REFERENCES

- https://en.wikipedia.org/wiki/RC5
- https://www.geeksforgeeks.org/rc5-encryption-algorithm/
- https://www.educba.com/rc5/
- https://link.springer.com/chapter/10.1007/3-540-60590-8_7