Developed RC4 Algorithm

with Double Reverse for Better Data Protect

Fouad Sahib Muhamed Alkhazraji
Ministry of Higher Education and Scientific Research, Baghdad / Iraq
Fouadalkhazraji8@gmail.com

خوارزمية ارسي فور المطورة
مع العكس المزدوج لحماية افضل للبيانات

فؤاد صاحب محمد الخزرجي

وزارة التعليم العالي والبحث العلمي, بغداد \ العراق

Abstract

Computer security helps users to protect their data and Cryptographic techniques can be used for this purpose. One of the methods to secure data confidentiality is encryption whether stored on hard drives or transferred through media. RC4 (Rivest Cipher 4) is a cryptography Algorithm that uses the same key to encrypt and decrypt data. Stream cipher has been used widely in many protocols because it is flexible and fast, which RC4 is one type of it and XOR is a function that has been used to implement their steps. In this paper, Double Reverse (DR) has been implemented to make RC4 more complicated and harder to break by attackers. It reverses each bit of the key-stream and shifts the elements of the array entirely. In addition, we went through the basic ideas of Cryptography, Cryptanalysis, Symmetric key cryptography, Asymmetric key cryptography, and the Mechanism of RC4.

Keywords: Cryptography Algorithm (CA), RC4, Stream cipher, Double Reverse (DR) , XOR Cipher.

المستخلص

تقوم امنية الحاسبات بمساعدة المستخدمين على حماية بياناتهم حيث يمكن استخدام تقنيات التشفير لهذا الغرض. يعد التشفير من أفضل الطرق لتأمين سرية البيانات سواء تم تخزينها على الاقراص الصلبة أو من خلال نقلها عبر الوسائط. تعتبر (ار سي فور) واحدة من خوارزميات التشفير المتناظرة التي تستخدم نفس المفتاح للتشفير ولفك الشفرة. (الستريم سايفر) تستخدم في العديد من البروتوكولات وبنطاق واسع لسرعتها ومرونتها حيث ان (ار سي فور) احد انواعها. في هذا البحث تم استخدام طريقة (العكس المزدوج) لجعل طريقة (ار سي فور) اكثر تعقيدا ويصعب كسرها من قبل المهاجمين. تناول هذه البحث الأفكار الأساسية للتشفير ، وتحليل التشفير، وتشفير المفتاح المتماثل ، وتشفير المفتاح غير المتماثل ، وآلية (ار سي فور)

الكلمات المفتاحية: التشفير ، ار سي فور ، ستريم سايفر ، العكس المزدوج , شفرة اكس اور.

Introduction

Because of the information explosion in the world, its security has become a progressively significant task and a crucial strategic resource. Cryptography Algorithm (CA) is the way to transfer information securely over a channel to protect from competitors and delivers it to the right destination. In contrast, there is a parallel field called Cryptanalysis Algorithm which is used to break CA by using corresponding suitable techniques. Therefore, as strong as CA, it will be hard to break by Cryptanalysis. Cryptography and Cryptanalysis are fields of Cryptology. Although CA is basic for communication and network security, it has some gaps in its traditional techniques which attracted attack programs to break it. Currently, CA such as RSA (Rivest–Shamir–Adleman) and DES (Data Encryption Standard) algorithms which are supported by mathematical models are not trustworthy as before (Pujari, *et al.,*2018).

Symmetric key CA

Symmetric key CA is one of the main mechanisms in the security of CA and it is a trustworthy application for the future (Nan, *et al.,*2017). Symmetric key CA mechanisms are classified into stream ciphers and block ciphers. It is the mechanism of using the same key for encryption and decryption which is also named as secret key CA. The sender encrypts the message with the same key which is used by the receiver to decrypt it (Shrivastava, *et al.,*2016).

Asymmetric key CA

Asymmetric key CA is also named Public key CA which includes two different keys, one for encryption and one for decryption. The keys work in pairs of public and private keys. The public key can be given to others and the private one should be secret and no one has the permission to use it except the owner. When someone encrypts the message using the public key; it can be decrypted only with the private key, not the public one (Shrivastava,, *et al.*,2016). All senders have the public key and only the recipient has the private one. Symmetric key cryptography is faster than Asymmetric key cryptography and this is the main disadvantage of using Public-key cryptography (Wahab, *et al.*,2021). Figure -1 below shows the Classification of Cryptography.

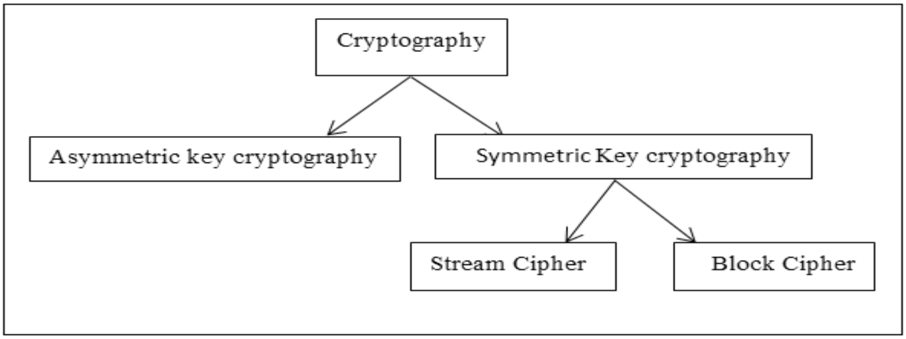


Figure –1 Classification of Cryptography

Stream Cipher

It is a type of symmetric-key cipher that merges plaintext digits with a sequence of numbers produced by a definite secret mathematical procedure to get an output cipher stream and because of that, they call it to state cipher. In both encryption and decryption, the XOR operator is used to merge the pseudorandom key-stream with the plaintext digit or output cipher stream. In encryption, the plaintext digit is combined with a pseudorandom key stream by XOR operator to generate the output cipher stream and in decryption, the output cipher stream is combined with the same pseudorandom key stream by the XOR operator to generate the original plaintext. With the stream cipher, the operation is encrypting each bit or byte at a time and using a feedback mechanism to change the key constantly (Shrivastava, *et al.,*2016). Because each bit or byte is independently encrypted or decrypted, stream ciphers are out of error expansion, and due to the preceding features; they are more vital for instant processing and communication. Also, they have been the best mechanism to use in the wireless field (El Hennawy, *et al.*,2015).

Block Cipher

With block ciphers the operation is encrypting one block of data each time without changing the key (Shrivastava, *et al.*,2016). They are algorithms use less memory and rearrange N-bit blocks of plain text data by emerging with the impact of the secret key to get N-bit blocks of encrypted data (El Hennawy, *et al.*,2015).

XOR Cipher

One of the most simple and fundamental encryption ciphers is the Exclusive OR (XOR) which is a symmetric encryption algorithm. It is a logical function that can be used for binary bits and the idea is derived from the Boolean algebra XOR function that returns (1) when the two arguments have different values and (0) when the two arguments have similar values. Its efficiency depends on the nature of the key and its length. It can achieve better security performance with a long random key and with large XOR keys, unpredictability can be increased and brute-force attacks can be confronted (Natsheh and Gale, 2015).

Stream Cipher Mechanism

According to (Polak and Boryczka , 2015) encryption and decryption in stream ciphers CA in the simplest way can be defined as follows:

ENCRYPTION:: EK (P) = C (1)

DECRYPTION:: DK(C) = P (2)

Where:

E represents encryption,

D represents decryption,

K represents key,

P represents plaintext,

C represents ciphertext.

By using stream ciphers there is no direct use of the key, the key generates key-stream (k) of a much longer period than itself. Then the keystream is added to the plaintext by using the (XOR) function. The processes of encryption and decryption are defined as follows:

ENCRYPTION: Pn Kn = Cn (3)

DECRYPTION: Cn Kn = Pn (4)

Where:

Pn – n_th bit of plaintext,

Kn – n_th bit of keystream,

Cn – n_th bit of ciphertext,

$\oplus$ – *eXclusive OR (XOR)*

For example:

```
P:      10011101011100011011011000110111…
K: ⊕ 11100111110101000011110100101100…
     ---------------------------------------------------------
C:      01111010101001011000101100011011…
```

RC4

RC4 is one of the most common stream ciphers of symmetric key CA which takes a lower area compared to others and it is fast with low complexity. As a type of stream ciphers, the RC4 algorithm includes a secret internal mechanism. It works by generating a stream of bits in a semi-random way and it consists of an array of 256 bytes, which is called the S-box or we can use M-box in this paper, and three 8-bit index pointers. The algorithm has two main parts. The first part for initializing the M -box uses a variable-length key, which is called the key scheduling algorithm (KSA). The second part for generating the bytes of the key-stream is called the pseudo-random generation algorithm (PRGA) (Taqieddin, *et al.*,2015). When the web security is based on RC4 structure, the plaintext is X-ored with a sequence of random bytes that have been generated by (KSA) and (PRGA). Actually, these bytes are not really random as they should be. Its weakness, the key is 64 bit (40 its are fixed) and the remaining 24 bits offer just 16 million possibilities which gives a 50% chance of reuse of the key after less than 5000 packets (Mekhaznia and Zidani, 2015). Although RC4 is widely used as a stream cipher in public domain and commercial software products, it has been deprecated because many cryptanalytic attacks proved its insecurity (Chakraborty, *et al.*,2021).

RC4 Mechanism is explained below:

The authors (Polak and Boryczka, 2015) mention that to start the algorithm, KSA takes an input a variable-length key and initializes the M-box array as below:

```
for i = 0, …, 255 { M[i] = i}
```

The next step is key-scheduling algorithm (KSA), each of the 256 entries in *M* are then swapped with the *j*-th entry in *M*, as below:

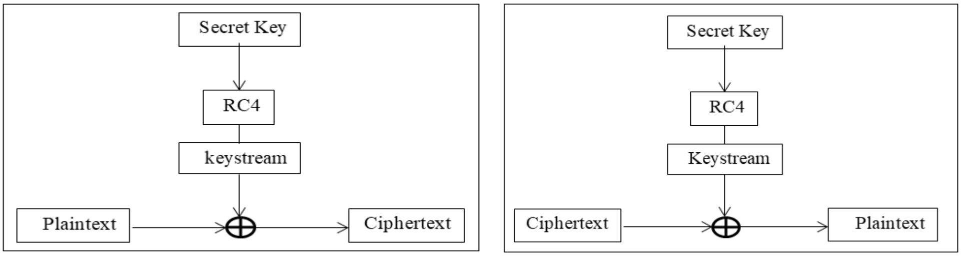
```
j = 0
 for i = 0, …, 255 {
        update j = ( j + M[i]+key[i] mod key-Length) mod 256
        swap M[i] with M[ji }
return M
```

PRGA:

After finishing KSA's iterations the next step of RC4 is implementing (PRGA) as below:

```
Initialize i = 0, j = 0
  While (Generating Output) {
  i = i+1, j = ( j+ M[i]) mod 256
  swap M[i] with M[j]
  k = (M[i]+M[j]) mod 256
 return M[k] }
```

Figure -2 below shows the Mechanism of RC4 for Encryption and Decryption.



(a) **(b)**

Figure -2 Mechanism of RC4 (a) Encryption (b) Decryption

Related Work

The authors (Saha, *et al.*,2019) refer that the RC4 method gets attacks at the step of the key scheduling Algorithm. Therefore, they modified RC4 as MRC4 at the first stage of (KSA) when initializing the key. Firstly, it creates an array called T' which is the size of 256 by copying the members of the key, and if the key size is less than 256, it repeats copying the members of the key till filling the array of T'. The next step is creating a T array by performing the Symmetric Random Function Generator (SRFG) method to each byte transferred from T' to T. SRFG gets two bytes as inputs and produces one byte as an output. Its inputs are two consecutive bytes transferred from T'. Each time performing SRFG, the second byte of its inputs will be the first input byte for the next SRFG. The last SRFG uses the first value of T' as a second input byte. The authors claimed that MRC4 is efficient against the attacks comparing to RC4 and it supports better confusion and diffusion. (Al-badrei and Alshawi, 2021) triy to overcome the weaknesses of RC4 algorithm by proposing a method called IRC4 which provides a high level of complexity and randomness. They claimed that IRC4 is more secure and stronger against attacks.

Proposed Algorithm

The mechanism in RC4 for encryption is producing one byte of keystream every time and this byte of key-stream is XORed with a byte of plaintext and as a result, it produces one byte of a ciphertext. For decryption it produces one byte of keystream every time and this byte of key-stream is XORed with a byte of ciphertext and as a result, it produces one byte of a plaintext (Polak and Boryczka, 2015). In our proposed algorithm there is a DR method implementing on the byte of key-stream before XORed it with the plaintext to get the ciphertext and it does the same thing for decryption. In encryption, when it gets the byte resulted from RC4 steps, it reverses each bit inside the byte by flipping zero to one and one to zero. Then it reverses the entire byte by shifting in horizontal way. It manipulates the locations of the bits inside the byte. The first position will be the last one, the last position will be the first one and so on till it reverses the entire byte. For decryption, it performs same steps of DR. Below is the general code for DR where it takes an array of binary numbers as an input and produces an array of binary numbers with different positions as an output. Here, we consider DR as the name of the array:

```
First step:
  for i from 0 , …, 7 {
          if DR[i] == 0
                  DR[i] = 1
          else
                  DR[i] = 0}
```

Second step:

```
start = 0
end = 7
while (start < end) {
 temp = DR[start]
 DR[start]  = DR [end]
 DR [end] = temp
 start++
 end-- }
```

Flow chart in Figure -3 explains the steps of DR mechanism as below:

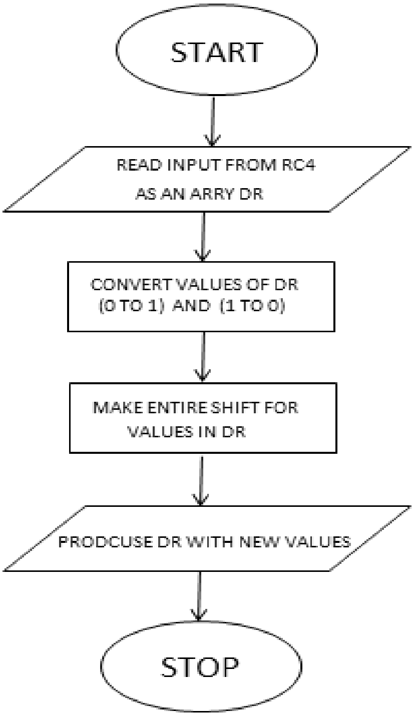


Figure - 3 Flow chart of DR (same steps for Encryption & Decryption)

Results and Discussion

To explain the idea of DR imagine we have this one byte of keystream:

1	1	0	0	1	1	1	0

The first step is replacing each zero with one and each one with zero to obtain this byte:

0	0	1	1	0	0	0	1

Then we make shifting for the entire byte to obtain this final byte:

1	0	0	0	1	1	0	0

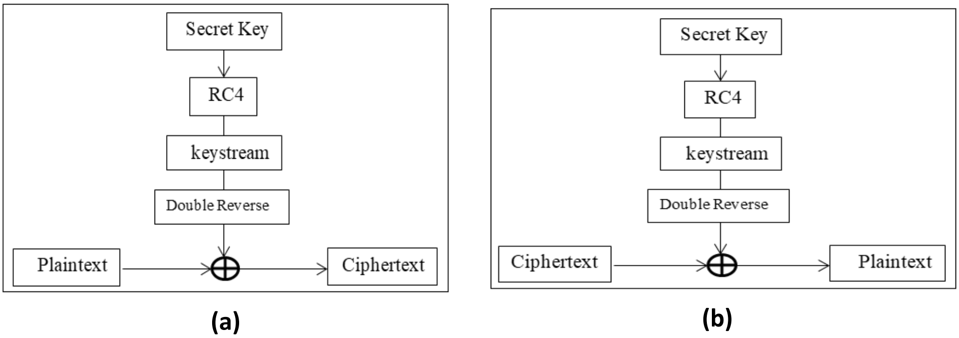
Now, we XORed this final byte with the current byte of plaintext to produce one byte of ciphertext as a process of encryption. To explain the second step of Double Reverse, imagine we have this array of numbers from one into 8 as shown below:

1	2	3	4	5	6	7	8

After making the entire shift for this array it will be as below:

8	7	6	5	4	3	2	1

But remember in our algorithm we deal with one and zero not with other numbers. For decryption, we do the same thing and it does not matter if we do the flipping or shifting as a first or second step because both mechanisms give us same result. Figure - 4 below shows how to use RC4 with Double Reverse for Encryption and Decryption



(a) (b)

Figure – 4 RC4 with DR (a) Encryption (b) Decryption

Conclusion

This paper went through the basic ideas of CA, Cryptanalysis, Symmetric key CA, and Asymmetric key CA. Stream cipher and RC4 have been discussed and their mechanisms have been explained. DR technique has been added to the steps of RC4 to make it more complicated and harder to break by attackers. It has two steps: the first one is replacing each zero with one and each one with zero within the byte which resulted from RC4. The second step is reversing the entire byte by shifting mechanism. It gives more power to the algorithm of RC4 and makes it more stochastic. It interfere with the key-stream to make it more complicated for analysts. So the attackers will face new difficulty to analyze the algorithm.

References

- Chandratop Chakraborty, Pranab Chakraborty, Subhamoy Maitra, (2021), ‘Glimpses are Forever in RC4 Amidst the Spectre of Biases, Discrete Applied Mathematics, Volume 298, pp. 84-102, ISSN 0166-218X. Available at: https://www.sciencedirect.com/science/article/pii/S0166218X21001335
- Eyad Taqieddin, Ola Abu-Rjei, Khaldoon Mhaidat, Raed Bani-Hani, (2015), ‘Efficient FPGA Implementation of the RC4 Stream Cipher Using Block RAM and Pipelining’, Procedia Computer Science, Volume 63, pp. 8-15, ISSN 1877-0509. Available at: https://www.sciencedirect.com/science/article/pii/S1877050915024345
- Hadia M.S. El Hennawy, Alaa E.A. Omar, Salah M.A. Kholaif, (2015) ‘LEA: Link Encryption Algorithm Proposed Stream Cipher Algorithm’, Ain Shams Engineering Journal, Volume 6, Issue pp. 57-65, ISSN 2090-4479. Available at: https://www.sciencedirect.com/science/article/pii/S2090447914001051
- Hasan H. Al-badrei and Imad S. Alshawi, (2021), ‘ Improvement of RC4 Security Algorithm," in Advances in Mechanics, Volume 9, Issue 3, pp. 1467-1476. Available at: https://faculty.uobasrah.edu.iq/uploads/publications/1639754875.pdf
- Iwona Polak and Mariusz Boryczka,, (2018), ’Tabu Search Against Permutation Based Stream Ciphers’, in INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, Vol. 64, No. 2, pp. 137–145. Available at: https://journals.pan.pl/publication/119361/edition/103842/international-journal-of-electronics-and-telecommunications-2018-vol-64-no-2-tabu-search-against-permutation-based-stream-ciphers-polak-iwona-boryczka-marcin?language=en
- Longmei Nan, Xiaoyang Zeng, Wei Li, Zhouchuang Wang, (2017) ‘Design and Implementation of Configurable SHIFT Instructions Targeted at Symmetrical Cipher Processing’, Procedia Computer Science. Volume 107, pp. 225-230, ISSN 1877-0509. Available at: https://www.sciencedirect.com/science/article/pii/S1877050917303587
- Manish Shrivastava, Shubham Jain, Pushkar Singh, (2016) ‘Content Based Symmetric Key Algorithm’, Procedia Computer Science,. Volume 85, pp. 222-227, ISSN 1877-0509. Available at: https://www.sciencedirect.com/science/article/pii/S1877050916305658
- O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein and H. F. A. Hamed, (2021) ‘Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques’, in IEEE Access, Vol. 9, pp. 31805-31815. Available at: https://ieeexplore.ieee.org/document/9356603

- Q.N. Natsheh, B. Li, A.G. Gale, (2016), ‘Security of Multi-frame DICOM Images Using XOR Encryption Approach’, Procedia Computer Science, Volume 90, pp. 175-181, ISSN 1877-0509. Available at: https://www.sciencedirect.com/science/article/pii/S1877050916311966
- R. Saha, G. Geetha, G. Kumar, T. -H. Kim and W. J. Buchanan, (2019), ‘ MRC4: A Modified RC4 Algorithm Using Symmetric Random Function Generator for Improved Cryptographic Features,’ in IEEE Access, Vol. 7, pp. 172045-172054 .
  Available at: https://ieeexplore.ieee.org/document/8915779
- Saswat K Pujari, Gargi Bhattacharjee, Soumyakanta Bhoi, (2018) ‘A Hybridized Model for Image Encryption through Genetic Algorithm and DNA Sequence’, Procedia Computer Science. Volume 125, pp. 165-171, ISSN 1877-0509.
  Available at: https://www.sciencedirect.com/science/article/pii/S1877050917327874
- Tahar Mekhaznia,and Abdelmadjid Zidani, (2015), ‘Wi-Fi Security Analysis’, Procedia Computer Science, Volume 73, pp. 172-178, ISSN 1877-0509 .
  Available at: https://www.sciencedirect.com/science/article/pii/S1877050915034705