A High Throughput Algorithm for Data Encryption

A.Rathika
Asst. Professor
Department of IT, VCET, Erode

Parvathy Nair
Department of IT
VCET, Erode

M.Ramya
Department of IT
VCET, Erode

ABSTRACT
RC4 encryption algorithm [3,12,16] was examined based on the text to be encrypted and the chosen key. It is a symmetric key algorithm and uses stream cipher i.e. data is being encrypted bit by bit. The same algorithm is used for both encryption and decryption. The data stream is XORed with the generated key sequence and the key stream is completely independent of the plaintext used. The weakness of the algorithm has been found out and is being rectified. We proposed a modification in RC4 algorithm that would result in saving the memory space and also augment the performance by reducing the time. The proposed algorithm explains that we can have the key as an image so that we can compress it and store. Hence the algorithm yields better results in hardware also.

Keywords: RC4 (Rivest Cipher 4) algorithm, Symmetric Key, Assymetric Key etc.,

1. INTRODUCTION
During the transmission of data, it is necessary to protect the information against unauthorized access or modification such as deletion or addition of some part into the information. Hence security plays a major role in transmitting the data. Hence in order to make the data to be more secured, we encrypt the data in the sender side and decrypt in the receiver side. Cryptography is a tool which is used to keep our information confidential and to ensure its integrity and authenticity.

Encryption[10,12,16] is the process of converting plain text into cipher text (to prevent unauthorized access).Decryption[10,12,16] is the reverse process of it. Both the sender and the receiver must know this key in order to encrypt and decrypt the data respectively.

Cryptographic algorithms can be divided into:

i) Private key/ Symmetric key algorithms
ii) Public key /Asymmetric key algorithms

Symmetric key algorithms[10,12] have the property that same keys are used for encryption and decryption. It is also called as private key encryption.

Asymmetric key algorithms[10,12] have the property of using different keys and hence the decryption key cannot be derived from the encryption key. It is also called as public key encryption.

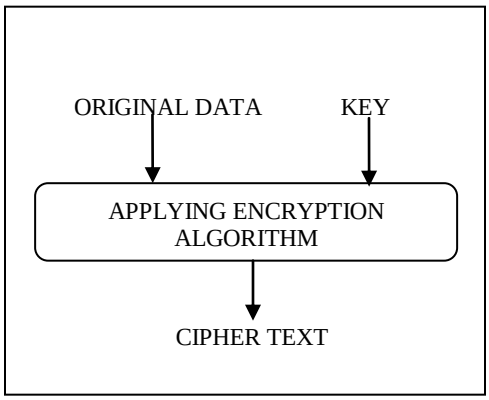


Figure 1. Process of Encryption

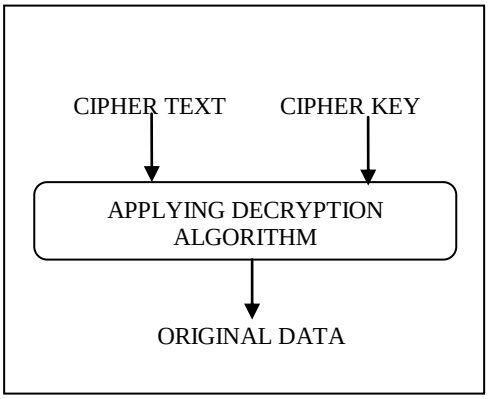


Figure 2. Process of Decryption

2. AN EXAMPLE FOR ENCRYPTION
Consider the original text as "WELCOME". If we are using substitution encryption[16], then let us assign a numerical equivalent numerical equivalent to each letter.

A	B	C	D	E	F	G	H	I
0	1	2	3	4	5	6	7	8

J	K	L	M	N	O	P	Q	R
9	10	11	12	13	14	15	16	17

S	T	U	V	W	X	Y	Z
18	19	20	21	22	23	24	25

Hence, the "WELCOME" is transmitted as "22 4 11 2 14 12 4" to be transmitted across the network.

Consider we are transforming the alphabet to that particular alphabet + 3. Then we would transmit "ZHOFRPH".

This is just an example for encryption in order to know how it works.

3. METHODS OF PRODUCING CIPHER TEXT

There are basically two methods of producing cipher text. They are:

1.Stream cipher

2.Block cipher

Stream cipher[9] is the method where each bit of data is sequentially encrypted using one bit of the key. To have this stream cipher more difficult to crack, one could use a key which varies in length. This masks the patterns which can be understandable, that makes more hard to crack. Stream ciphers keep some sort of memory, or state, as it processes the plaintext and uses this state as an input to the cipher algorithm.

Block ciphers[2,14] are designed to encrypt the data in chunks of specific size. This specification will identify how much data should be encrypted on each pass (called a block)and also the size of the key to be applied to each block. The encryption function is the same for every block. A block cipher can be represented by a bijective function f which accepts as input a block of plaintext of a fixed size, and a key, and outputs a block of ciphertext.

f(p,k)=c

4. THE RC4 ALGORTIHM

RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA security and hence it is called Rivest Cipher algorithm,[12,13,7]. It is a variable stream size key cipher with byte oriented operations. The algorithm is based on the use of a random permutation[14].

There are two steps in RC4 algorithm.

1.Key Scheduling Algorithm (KSA)[15] and

2.Pseudo Random Generation Algorithm (PRGA)

KSA which turns a random key (whose typical size is 40-256 bits) into an initial permutation S of {0,….,N-1}. Here S is initialized to be the identity permutation. This initializes indices I and j as zero and PRGA operation is applied.

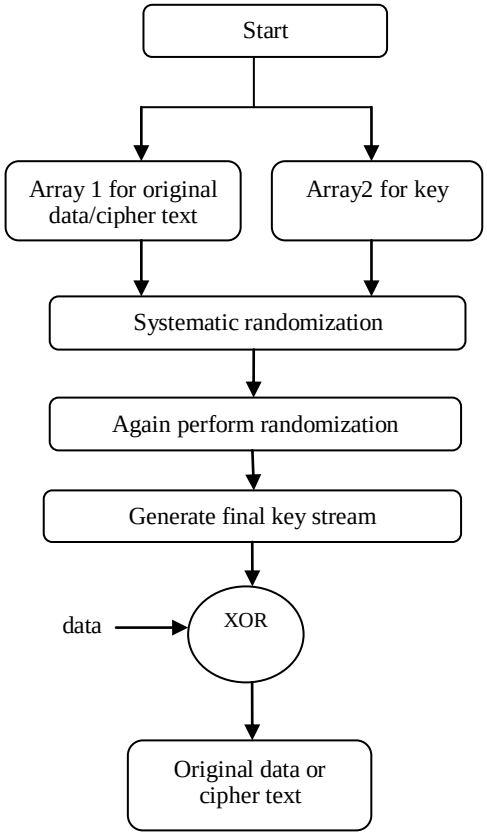
This can be given as:

```
for i=0 to N-1
  S[i]=i
j=0
for i=0 to N-1
  j+=S[i]+K[I mod length]
  swap(S[i],S[j])
```

PRGA uses the permutation obtained from KSA to generate a pseudo-random output sequence. Here two indices i and j are initialized to zero. Then I is incremented as counter, j is incremented pseudo randomly ,swap two values of S and output S.

This can be given as:

```
i=0
j=0
i=i+1
j=j+S[i]
swap(S[i],S[j])
output z=S[S[i]+S[j])
```



4.1 Weaknesses of RC4

A large number of weak keys[15.,1,4,6] are used. These weak keys have a small part of the secret key which determines a large number of bits of the initial permutation (KSA output). In addition, because of these weak keys initial outputs are disproportionally affected by a small number of key bits.

The attacker[11] can deduce the entire key even if part of the key presented to KSA[5] is exposed to the attacker. As the same secret key is used with different exposed values, the attacker is at ease of re-deriving the secret key by analyzing initial work.

RC4 is particularly slow in hardware.

5. PROPOSED ALGORITHM

In our algorithm, we propose that instead of using an array of key, we can use an image as the key. In case of software the image location is supplied and operations performed on the location value yield different images for encryption. For hardware we may use different images generated by the program graphics.
Features of the proposed algorithm include:

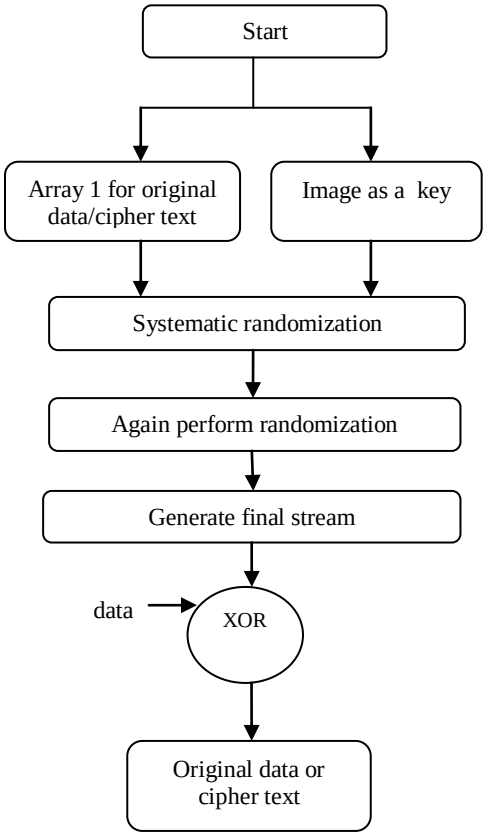
As the data is being hidden behind images, the chances of an attacker working out the original data from the image are less. The attacker would face difficulties in figuring out which image hide the data among the millions of images being sent. Compression techniques[16] can be used to transmit the encrypted images easily over the network.

The use of images already stored in the memory will result in better memory management as no additional space is being occupied for the encryption key and the images used are being dynamically used.

Making use of dynamic memory allocation results in faster compilation and the program generates the output in less timed compared to the original algorithm.

Randomization performed here is more efficient compared to the original algorithm as instead of a single key we are making use of dynamic images.

5.1 Flowchart



5.2 Algorithm

```
int i;

for i = 0 to 255:
S[i] = i;                    //where S is State Table
meslen=(Int32)messageStream.Length;
String im=meslen.ToString("x");
im=UnTrimColorString(Im,6);
int r=Int16.Parse(im.Substring(0,2),NumberStyles.HexNumber);
int g=Int16.Parse(im.Substring(2,2),NumberStyles.HexNumber);
int b=Int16.Parse(im.Substring(4,2),NumberStyles.HexNumber);
pixelColor=Color.FromArgb(r,g,b);
Image as a key
Perform randomization
Again randomize the first array filled with 0 to 255
Output data XOR im
```

6. COMPARISON CHART OF RC4 AND PROPOSED ALGORITHM

Table 1. Comparison for RC4 and Proposed algorithm

S.No	Parameters considered	RC4	Proposed
1	Execution Time	More	Comparatively less
2	Memory space occupied	More	Less
3	Execution speed in software	Fast	Very fast
4	Execution speed in hardware	Not fast	Fast

7. CONCLUSION

Thus the proposed algorithm was found to be comparatively faster and more efficient than the RC4 algorithm for data encryption. Use of image for encryption ensures lesser chances of cryptanalysis and hence better security can be provided to the data. In case of hardware using programs to get graphic images for encryption requires no additional memory for the images and thus efficient use of available storage space. As there are several image compression techniques available in both lossless and lossy image compression, we could use that compression technique for better memory usage. This results in efficient memory usage.

8. REFERENCES

[1] Allam Mousa and Ahmad Hamad ,"Evaluation of the RC4 Algorithm for Data Encryption", International Journal of Computer Science and Applications, Vol 3,No. 2, June 2006

[2] Bruce Schneier "Applied Cryptography-Protocols, Algorithms and Source code in C".

[3] Diffie.W and Hellman.M, "New Directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, 1976

[4] Ed Dawson, Helen Gustafson, Matt Henricksen, Bill Millan "Evaluation of RC4 Stream Cipher" Information Security Research Centre ,Queensland University of Technology

[5] Eli Biham and Yaniv Carmeli, "Efficient Reconstruction of RC4 Keys from Internal States", International Association for Cryptologic Research, 2008

[6] Erica Simcoe, Hirsh Goldberg, and Mehmet Ucal, Advisor: Dr. Sennur Ulukus, "An Examination of Security Algorithm Flaws in Wireless Networks"

[7] Guang Gong, Kishan Chand Gupta, Martin Hell and Yassir Nawaz, "Towards a General RC4-like Keystream Generator"

[8] Guy E. Blelloch, "Introduction to Data Compression", Computer Science Department Carnegie Mellon University, September 25, 2010.

[9] Menezes. A, Van.P, Oorschot, and Vanstone.S, "Handbook of Applied Cryptography", CRC Press, 1996

[10] Nadeem.A, "A performance comparison of data encryption algorithms," IEEE Information and Communication Technologies, 2006

[11] Nathaniel Couture Kenneth B. Kent, "The Effectiveness of Brute Force Attacks on RC4"

[12] PachghareV.K , Eastern Economy Edition, Cryptography and Information Security

[13] "RC4 Encryption Algorithm" http://www.vocal.com

[14] Rick Wash, "Lecture Notes on Stream Cipher and RC4"

[15] Scott Fluhrer, Itsik Mantin, and Adi Shamir, "Weaknesses in the key scheduling algorithm of RC4"

[16] Stallings.W, *Cryptography and Network Security*, Prentice Hall, 4th Ed, 2005.

ABOUT THE AUTHORS

Mrs. A. Rathika received her ME at Anna University of Technology, Coimbatore. Her research area of interest includes network security and information security. Currently she is working as an Assistant Professor at Velalar College of Engineering and Technology in the department of Information Technology.

Ms. Parvathy Nair pursuing her final year B.Tech in Information Technology department at Velalar College of Engineering and Technology, Erode. Her area of interest includes network security in Bio-Informatics.

Ms. M. Ramya pursuing her final year B.Tech in Information Technology department at Velalar College of Engineering and Technology, Erode. Her area of interest includes Cryptography and Network Security.