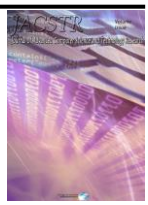

A PROPOSAL TO IMPROVE RC4 ALGORITHM BASED ON HYBRID CHAOTIC MAPS

Ahmed T. Sadiq[1], Alaa K. Farhan[1], Shaimaa A. Hassan[2]

[1]University of Technology, Baghdad, Iraq

[2]Middile Technical University, Baghdad, Iraq

drahmaed_tark@yahoo.com

ISSN: 2231-8852

ABSTRACT

The explosion of data has led to the unpredictable growth of the amount of digital data transmitted over the network, representing image, text, sound, audio, video, etc. Also, rapid developments of the telecommunication network, mobile phone, and the internet have led to increasing the necessity to develop security algorithms to keep pace with these developments. Cryptography has a long history that provides a way to store sensitive information or transmit it across insecure networks (i.e. the Internet) so that it cannot be read by anyone except the intended recipient. Stream Cipher is one type of symmetric encryption in which one bit or one byte of digital data stream is encrypted at a time. If the cryptographic keystream is random, then the cipher is unbreakable. However, both users must be provided with the keystream in advance via some independent and secure channel. In this paper, the RC4 algorithm is improved by using hybrid chaotic maps which consist of logistic and tent map to strengthen the randomization process. The experimental results show high average security of the proposed algorithm compared with the original RC4.

Keywords: RC4, Symmetric Key Algorithm, Stream Cipher, Chaotic Maps

1. Introduction

The rapid growth of internet and telecommunication network has increased the risk of unauthorised and unauthenticated access to data. This has led to developing various security techniques to protect data against attackers (Nisha and Madhu, 2013).

A chaotic system has received significant attention in previous years, and many researchers have found an excellent relationship between chaotic and cryptography (Kamel, 2010).

Chaos theory is one branch of mathematics that deals with the dynamical nonlinear system. Chaos theory is a Butterfly Effect, which means that it has a sensitive dependence on the initial conditions. For the system to have a Butterfly Effect, it must be nonlinear, and each state must be determined by the previous state (Boing, 2015).

In a chaotic stream cipher, the chaotic system is used to generate a sequence of the key stream that is XORed with the plaintext to produce ciphertext (Kamel, 2010; Forré, 1991).

Generally, in a stream cipher, a pseudo-random bit generator is used to generate a sequence of fixed length random key that it expanded and used to mask a sequence of plaintext bit by bit. The main problem is how to use a PRNG that generates a random number with excellent properties. A chaotic system can generate indistinguishable long period random numbers that are sensitive to a tiny change to the initial state (Kotulski and Szczepanski, 2000).

The rest of the paper is organised as follows: Section 2 contains the related works of the current study. Section 3 explains the chaotic maps used in proposed algorithms. In Section 4, the original RC4 algorithm is described including its main steps. The proposed algorithm and its main steps are presented in Section 5. In Section 6 the experimental results are given. Section 7 discusses the results of this research work. Concluding remarks were outlined in the last part.

2. Related Works

Many researchers have attempted to increase the security of RC4 algorithm. In Souradyuti and Bart (2004), a new statistical bias in the distribution of the first two output bytes of the RC4 keystream generator was presented. This paper also proposes a new pseudorandom bit generator, named RC4A, which is based on RC4's exchange shuffle model. It is shown that the new cipher offers increased resistance against most attacks that apply to RC4.

In Weerasinghe (2012), analysis of a modified RC4 algorithm was presented by improving RC4 key generation and pseudo-random number generation. In Maytham et al. (2015), RRC4 was proposed as improvement of RC4 to solve its weak keys problem by randomising the initial state *S*.

In Methaq and Fadya (2016), secret key and the CLM to produce a one-dimensional array of different numbers were used. Then the RC4 algorithms used to make some sort of random shuffling (relying on the contents of the array created by the CLM) to the array that is created by the RC4 first algorithm. After that, the second algorithm of RC4 used inside a loop to change the value of each colour (using the resultant array of the first RC4 algorithm) of a pixel until all the pixels of the image be changed. Moreover, by doing that we have produced a cipher image that is completely different and does not reveal any information of the plain image.

3. Chaotic Maps

The Chaos system is a dynamical, complex and a nonlinear system. The chaotic map is a map or evaluation function that represents the chaotic system behaviour. There are several chaotic maps, and the used maps in the proposed algorithm are discussed below (David, 1994).

3.1 Logistic Map

The logistic map is a simple formula that describes not only chaos but also how it is developed from ordered behaviour. Logistic function takes the form shown in Eq. (1) (David, 1994):

$$f(x) = ax(1-x) \quad \ldots\ldots\ldots\ldots \quad (1)$$

Where *a* is a real parameter between 0 and 4, and *f(x)* is a discrete dynamic of the population between 0 and 1 (Carlos, 2003). For iterating, an initial value x_0 is needed, and the result of the function will be the input of the next iteration (David, 1994).

3.2 Tent Map

Tent map is one form of chaotic maps that describe discrete, dynamical, nonlinear systems. Its name is due to the tent-like shape. It takes the form of Eq. (2):

$$T_r(x)=\begin{cases}2rx & ,if\ 0 \le x \le \frac{1}{2} \\ 2r(1-x), if\ \frac{1}{2} \le x \le 1\end{cases} \quad \ldots\ldots. (2)$$

Where *r* is a real value between 0 and 2, x_0 between 0 and 1 and $T_r(x)$ between 0 and 1 (Wadia and Mohammed, 2013; Katok and Hasselblatt, 1995).

4. RC4 Algorithm

Ronald Rivest designed RC4 as a popular form of RSA. This algorithm is a secret key that requires sharing the key securely between sender and recipient. The encryption and decryption of RC4 algorithm are identical in which the data stream is XORed with the generated key. The key used in the RC4 algorithm is a variable length from 1 to 256 bytes, and this key is used to give 256 bytes state table its initial values (John, 2003). This means that RC4 algorithm consists of two functions. The first function, key stream generator, is used to generate a sequence of bits that are combined with the plaintext using XOR operator to produce the cipher text. The second function is the key scheduling algorithm, in which it accepts the variable length key and uses it to initialize the state table of the key stream generator. The initialization of RC4 algorithm consists of initializing a 256-bit state table (**S)**, using the key (K**)** as a seed. Once the state table is setup, Key Scheduling Algorithm (KSA) shown in Algorithm (1) is used to produce the initial permutation on S. It then continues to be modified in a regular form as data is encrypted using Pseudo Random Number Generation Algorithm (PRNGA) shown in Algorithm (2) (Allam and Ahmad, 2006).

Algorithm (1): Rc4 Key Scheduling

```
Input: Key;
Output: State;
Begin
  For i = 0 to 255
      State[i] = i;
  End For
  Initialize j to 0;
  For i = 0 to 255
       j = (j+State[i]+Key[i mod l]) mod 256;
       Swap State[i] and State[j];
  End For
End.
```

Algorithm (2): RC4 Pseudo Random Generation

```
Input: State, Message;
Output: Cipher_message;
Begin
  Intialize i & j to 0;
  For x = 0 to messagelength -1
        i = (i+1) mod 256;
        j = (j+State[i]) mod 256;
       Swap State[i] and State[j];
       GeneratedKey = State[ (State[i] + State[j]) mod 256] ;
       Cipher-message = Message[x] XOR GeneratedKey;
  End For
End.
```

5. The Proposed Modified RC4 Based on Chaotic Maps

The main weakness of RC4 algorithm is its vulnerable to analytic attacks of the state table. The proposed algorithm is designed to overcome this issue by strengthening the randomization of the state table. This is done by using hybrid chaotic maps which consist of two maps, tent and logistic in the swapping process. To provide a good environment that chaotic maps can work, numbers between 0 to 255 are divided into the interval between 0 and 1. The key array is filled with selected key, and the state array is filled randomly using tent map as a pseudo random number generator based on the array of the selected key. Then the state array is randomised within itself using the logistic map as a pseudo-random number generator to produce the final key stream that is XORed with the plaintext to produce the ciphertext. The proposed algorithm consists of three algorithms; the first one is Fraction_code algorithm that takes integer values between 0 and 255 and converts them to 256 real ranges between 0 and 1 to support the values resulted from chaotic maps as shown in Algorithm (3). The second algorithm is a modified key scheduling algorithm (shown in Algorithm (4)) while the third one is a Pseudo random number generator in which is the same as Algorithm (2).

Algorithm (3): Fraction_code

```
Input: integer values between 1 and 256;
Output: Fraction _code as array of ranges between 0 and 1;
Begin
  For i=0 to 256
       Represent the value of i as a real value with three fraction digits;
      Put the result of above step in Fract_array[i];
   End For
   Fraction_code = Fract_array;
End
```

Algorithm (4): Modified Key Scheduling.

Input: Key, initial vector of Logistic and Tent Maps {X_0,Y_0, ,a, b} .
Output: State */* Array with permuted positions*
Begin
Construct Permuted_key [Key] from 0 to 255 by repeatedly
filling Key in it;
For i = 0 to 255
State[i] = i;
End For

For i = 0 to 255
X_i= Apply Logistic Map shown in Equation (1);
Y_i= Apply Tent Map shown in Equation (2);
For n = 0 to 255
If X_i ϵ Fraction_code[n] then State[i]=n;
If Y_i ϵ Fraction_code[n] then j=n;
End
j = (j + State[i] + Premuted_key[i]) mod 256;
Swap State[i] and State[j];
End For
End

6. The Experimental Results

The idea behind the proposed algorithm was to modify RC4 algorithm to check how it responds towards the average of security (conditional entropy) of the cipher generated. Since the average of security calculation is used to evaluate security level of a cipher, the focus was to check the average security of the generated ciphertext to evaluate the modified RC4 algorithm over the original one (Weerasinghe, 2012). The variations of the common security over the different message and different key sizes are examined to give the result of this study. The average of security is computed by taking the summation of multiplying the probability of every character in the ciphertext by the entropy of the key. The entropy of key is calculated using Eq. (3).

$$H(K) = -\sum_1^n p(K_i)\ log\ p(K_i) \quad \ldots\ldots\ldots \quad (3)$$

Where: $P(k_i)$= the probability of K_i character in the key.

Then the average security (AV_oS) is produced by taking the summation of multiplying the probability of every character in the ciphertext with the entropy of the key as shown in Eq. (4).

$$AV_o\ S(K/C) = \sum_{j=1}^{l} \sum_{i=1}^{n} q_i\ p_{ij}\ log\ p_{ij} \ldots\ldots\ldots\ldots\ldots \quad (4)$$

Where:

l= The length of the key.
n= The length of the ciphertext.
q_i= The probability of the ciphertext.
$P_{ij=}$ The probability of the plaintext.

Table (1) and Figure (1) shows the variation of the average security of modified Rc4 algorithm compared with the original RC4 using different key lengths. The experimental result shows how the modified RC4 is strong due to a great randomization process compared with the original one.

Table 1: The average security of modified RC4 algorithm compared with the original RC4 using different key length.

The Average of Security of Modified RC4	The Average of Security of Original RC4	Length of Key
0.431192035	0.170631683	8
0.541288138	0.199330496	16
0.478338426	0.210267288	24
0.576159712	0.239723168	32
0.419439555	0.199037247	40
0.59437982	0.171924614	48
0.576386444	0.210308525	94
0.576386444	0.210308525	188
0.495760175	0.202609747	255

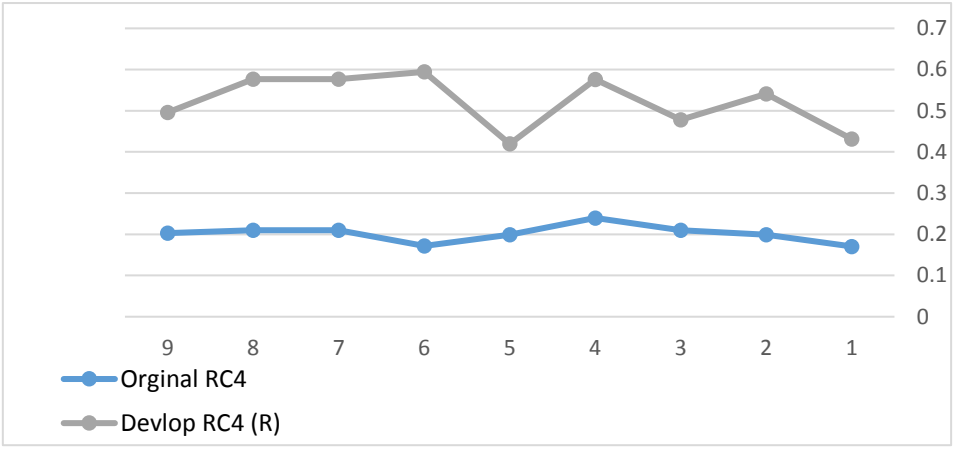

Figure 1: The average security of modified RC4 algorithm compared with the original RC4 using different key length.

7. Discussion

The proposed algorithm focuses on increasing the security of original RC4 algorithm by increasing the randomness. This is done by replacing the sequential increasing of the state table with random values generated from chaotic maps. The original RC4 is vulnerable to the analytical attack of the state table because the value of state [0] is not permuted. This problem is solved in the proposed algorithm because of a random permutation of the state table using values generated from two chaotic maps Logistic and Tent.

The average of security is used to measure the strength of the ciphertext. From experimental results, it has been noted that the average of security of the proposed algorithm using different key lengths is better than the original RC4. Also, the more powerful result can be generated when the number of samples is increased.

8. Conclusions

From the presented study, it has been concluded that the modified RC4 has a better average of security than the original one despite the little number of samples but the modified algorithm can give more powerful result and smooth curves when the number of samples is increased. Also, increasing the randomization of the modified RC4 has led to reducing the effect of the vulnerable attack on the state table. Finally, the confusion of the modified RC4 has been increased compared with the original RC4.

REFERENCES

1- Forré R. (1991). The Hénon Attractor as a Keystream Generator. In Advances in CryptologyEuroCrypt'91, vol. 0547, pp. 76-81, Berlin, Springer-Verlag.
2- David L. (1994). Chaos Theory and Strategy: Theory, Application. And Managerial Implications**.** Massachusetts Strategic Management Journal, Vol. 15, 167-178, U.S.A.
3- Katok A. and Hasselblatt B. (1995). Introduction to the modern theory of dynamical systems. Encyclopedia of Mathematics and its Applications. Cambridge University Press.
4- Kotulski Z. and Szczepanski J. (2000). On Constructive Approach to Chaotic Pseudorandom Number Generator. In Proceedings of Regional Conference on Military Communication and Information Systems, Zegrze, pp. 191-203.
5- Carlos G. (2003). Introduction to Chaos in Deterministic Systems. http://www.arxiv.org/
6- John J. (2003). RC4 Encryption Algorithm. VOCAL Technologies. Ltd. Custom Product Design Division 200, New York 14228. 716-688-4675. http://www.vocal.com/
7- Souradyuti P. and Bart P. (2004). A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher. Lecture Notes in Computer Science, Springer-Verlag, pp. 245–259.
8- Allam M. and Ahmad H. (2006). Evaluation of the RC4 Algorithm for Data Encryption. International Journal of Computer Science and Application. Vol.3. No.2.
9- Kamel F. (2010). Chaos-Based Key Stream Generator Based on Multiple Maps Combinations and its Application to Images Encryption. The International Arab Journal of Information Technology. Vol. 7. No. 3.
10- Weerasinghe T. (2012). Analysis of a Modified RC4 Algorithm. International Journal of Computer Applications (0975 – 8887). Volume 51– No.22.
11- Nisha K. and Madhu S. (2013). Chaotic Map based Block Encryption. International Journal of Computer Applications (0975 – 8887). Volume 71– No.16.
12- Wadia F. and Mohammed A. (2013). Some Dynamical Properties of the Family of Tent Maps**.** Int. Journal of Math. Analysis. Vol. 7. No. 29, 1433 – 1449.

13- Boing (2015). Chaos Theory and the Logistic Map.
14- Maytham M., Kenji Y. and Ali M. (2015). Enhancing Security and Speed of RC4. International Journal of Computing and Network Technology. V3, No. 2.
15- Methaq T. and Fadya F. (2016). An Efficient Image Encryption Technique using Chaotic Logistic Map and RC4 Stream Cipher. International Journal of Modern Trends in Engineering and Research.