EA Based Dynamic Key Generation in RC4 Ciphering Applied to CMS

Ashraf Aboshosha[1], Kamal A. ElDahshan[2], Eman K. Elsayed[3], and Ahmed A. Elngar[2]

(Corresponding author: Ahmed A. Elngar)

NCRRT, Atomic Energy Authority, Cairo, Egypt[1]

Faculty of Science, Al-Azhar University, Cairo, Egypt[2]

(Email: elngar_7@yahoo.co.uk)

Faculty of Science (Girls), Al-Azhar University, Cairo, Egypt[3]

(Received Feb. 8, 2015; revised and accepted Mar. 25 & Apr. 21, 2015)

Abstract

RC4 is the most widely used stream cipher algorithm. It is used to protect valuable electronic information. However, RC4 encryption algorithm suffers from a secret key generation as a seed problems. This paper proposes an intelligent dynamic secret key generation as a seed by employing an Evolutionary Algorithm (EA). The proposed RC4-EA method tends to enhance the RC4 encryption algorithm with a high degree of a seed key randomness. The main advantage of the proposed RC4-EA method is that the generation of this secret key is done dynamically and randomly; this adds more strength of the RC4 encryption algorithm against breaking this cryptosystems. Several experiments on the proposed RC4-EA method are conducted. Where, the results of the experiments show the improvement of the encryption time and the throughput of the proposed encryption RC4-EA method. Also, the proposed validated RC4-EA encryption method is applied for data ciphering in Content Management System (CMS).

Keywords: Confidentiality, evolutionary algorithm (EA), RC4

1 Introduction

During the past decades, Internet Technology (IT) has influenced everyday life [5, 23]. Internet security issues have become more common nowadays; particularly on internet banking account, shopping online and content management systems (CMSs) due to the harmful impact on confidentiality, integrity and privacy [21]. Sensitive information should stay secured and such security should be transparent and ubiquitous whenever shared [24]. Cryptography plays a major role in helping to prevent eavesdropping of sensitive information [7]. Encryption is the process of transforming plaintext into cipher text in order to prevent any unauthorized recipient from retrieving the original data [10]. The encrypted data is sent over the public network and is decrypted by the intended recipient [25].

One of the cryptographic algorithms is RC4 stream cipher algorithm. RC4 is considered to be a perfect cipher algorithm, but it suffers from some drawbacks [1]. One of the main drawbacks is that; a nonrandom secret key k as a seed is exposed to the attacker [6, 22]. Since the same secret key when permutes with the exposed Initial Vector (IV), an attacker can re-derive the secret key by analyzing the initial words of the key streams with relatively little work [18].

The random number generator is deterministic and periodic, which means that the sequence of numbers will eventually repeat itself or reproduced at later date. So the random number generator is not suitable for applications where it is important that the numbers are really unpredictable, such as data encryption.

Evolutionary Algorithm (EA), which is based on a powerful principle of evolution: survival of the fittest which model the natural phenomena [3]. EA has been widely used in science for solving complex problems [4].

In this paper, a stream cipher RC4 employees an Evolutionary Algorithm (EA) based approach to generate a dynamic secret key as a seed permutes with Initial Vector (IV) to produce a final key stream for encrypting the data is proposed. The novelty in the proposed method is that; EA is used to generate a dynamic random secret keys as a seed used for RC4 encryption algorithm which leads to increase the security of the system.

In the proposed RC4-EA method, the plaintext is encrypted in the form of ciphertext. Where, EA based approach is used to generate the dynamic random secret key based on a normal biological evolution. Since we get the key it is sent securely to RC4 encryption algorithm which permute with the (IV) to generate the the final key stream. An XOR operation is performed on the final key stream with the plaintext to obtain the ciphertext then storzed in the database.

The advantage of the proposed RC4-EA method is to increase the security of the system, by generating the secret keys dynamically and randomly. Which leads to, overcome the drawback of a non-random secret key as a seed in the original RC4 encryption algorithm. Hence, the final key stream can not be cracked by the attacker.

The rest of this paper is organized as follows: Section 2 presents an overview of the used algorithms, including the RC4 Encryption algorithm, weaknesses and attacks over RC4, and Evolutionary Algorithm (EA). Section 3 introduces the proposed dynamic RC4-EA encryption method. Section 4 gives the implementation results and analysis. Section 5 presents an applicable case study of the proposed RC4-EA encryption method. Finally, Section 6 contains the conclusion remarks.

2 An Overview

2.1 RC4 Encryption Algorithm

RC4 is a stream cipher algorithm designed in 1987 by Ron Rivest for RSA Security. The details remained secret until 1994, when they were anonymously published on an internet newsgroup.

RC4 is the most widely used software stream cipher in the world. It is used to protect internet traffic as part of the SSL (Secure Socket Layer), TLS (Transport Layer Security) protocols [20]. RC4 also used for encryption in the Wired Equivalent Privacy (WEP) (part of the IEEE 802.11), and Wi-Fi Protected Access (WPA) (part of the IEEE 802.11 i) protocols [14].

The RC4 algorithm consists of a permutation of array S containing the numbers $0, ..., N-1$, and two indices i, j in the initial state, where $N = 256$. The algorithm can be broken into two main stages: the Key Scheduling Algorithm (KSA), which uses the secret key k as a seed to create a pseudo-random initial state, and the Pseudo Random Generation Algorithm (PRGA), which generates the pseudo-random stream [19]. The flow chart of the RC4 encryption algorithm is shown in Figure 1.

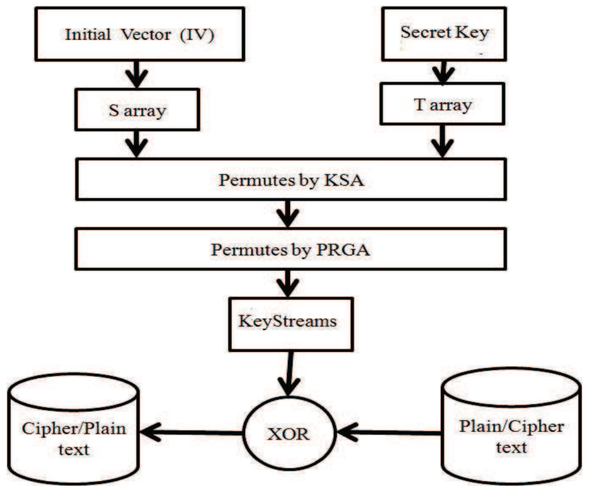


Figure 1: Encryption and decryption by RC4

2.1.1 Initialization

At the internal state, the entries of array S are set equal to the values $0, ..., N-1$ in ascending order; that is $S[0] = 0, S[1] = 1, \ldots, S[255] = 255$. Also, a temporary Vector T is created. If the length of the secret key k is 256 bits, then k is transferred to T. Otherwise, for a secret key k of length ℓ bits, the first element is copied from k to T till the length ℓ, then k is repeated as many times as necessary to fill out T [8].

```
$for\ \ i = 0\ \ to\ \ 255\ \ do$
    $S[i] = i$ ;
      $T[i] = k\ [\ i\ \ mod\ \ \ell\ ]$;
```

2.1.2 The Key Scheduling Algorithm

The Key Scheduling Algorithm (KSA) is used to generates a pseudo-random initial permutation of array S. Once the S array is initialized, S is shuffled using the secret key $T[i]$ to make it a permutation array [9]. The following actions are iterated 256 times after initializing i and j to 0:

```
  $Compute\ \ j = (\ j + S[i] + T[i]\ )\ \ mod\ \ 256$;
  $Swap\ \ (\ S[i],\ S[j]\ )$;
    $increment\ \ \ i$;
```

Once i has reached 256, the S array has been properly initialized [19]. The Key Scheduling Algorithm (KSA) is given in Algorithm 1.

Algorithm 1 A key scheduling algorithm (KSA)

```
Input:
$S$     $//before\ \ permutation$
$T$ $//A\ temporary\ vector\ of\ secret\ key\ k\ as\ a\ seed$
Output:
$array\ \ S$     $//after\ \ permutation$

1: for i = 0 to 255 do
2:     S[i] = i
3: end for
4: $j = 0$
5: for i = 0 to 255 do
6:     $j = (\ j + S[i] + T[i]\ )\ \ mod\ \ 256$
7:     $Swap\ \ (\ S[i],\ S[j]\ )$
8: end for
9: Return (S)
```

2.1.3 The Pseudo-Random Generation Algorithm (PRGA)

Once the S vector is initialized, the input secret key is no longer used [7]. The PRGA algorithm can generate key streams of any size. First, it initializes the two indexes

i, j to 0 and then starts the stream generation with $S[0]$ till $S[255]$. For each $S[i]$, $S[i]$ are swapped with $S[j]$ according to the following actions [12]:

1) Compute new value of i and j.
   $i = (i + 1) \; mod \; 256;$
   $j = (j + S[i]) \; mod \; 256;$

2) Swap $S[i]$ and $S[j]$ to have a dynamic state (it makes it harder to crack than if the state was computed only once and use for the generation of the whole key streams).
   $Swap \; (S[i], \; S[j]);$

3) Retrieve the next byte of the key stream from the array S at the index u.
   $u = S[(S[i] + S[j]) \; mod \; 256];$

The algorithm of the PRGA is given in Algorithm 2.

Algorithm 2 Pseudo-random generation algorithm (PRGA)

```
Input:
$S$    // $State\ of\ array\ S$
$u$              //$A\ temporary\ vector$
Output:
$K$              //$sequence\ of\ keystreams$

1: $i = 0$
2: $j = 0$
3: while $not\ end\ of\ sequence$ do
4:    $i = (i + 1)\ mod\ 256$
5:    $j = (j + S[i])\ mod\ 256$
6:    $Swap\ (S[i], S[j])$
7:    $u = S[(S[i] + S[j])\ mod\ 256]$
8:      $K = S[u]$
9: end while
10: Return (K)
```

2.1.4 Encryption and Decryption

Once the final key stream has been generated, the encryption and decryption process is the same as, the plaintext is XORed with the generated final key stream. If it is fed in plaintext, it will produce the cipher text, and if it is fed in a cipher text, it will produce the plaintext output [15].

2.2 Weakness And Attacks Over RC4

The cryptanalysis of the RC4 algorithm was divided into two parts, (1) analysis of the initialization of RC4 which focuses on the initialization of KSA, and (2) analysis of the output key streams generation which focuses on the internal state and the round operation of PRGA. To make the RC4 algorithm secure and capable to stand against attacks, lot of research are done over RC4 algorithm to enhancing its security.

Pardeep and Pushpendra in [16], reported that the RC4 algorithm was disclosed to the market and then experts start to analyze the RC4 algorithm and find out lots of weaknesses in both of two main stages of the algorithm KSA and PRGA.

Mantin and Shamir in [13], find out the weakness in the second round, where the probability of Zero output bytes are the major weakness of the RC4 algorithm.

Fluher et al. in [6], discovered the big weakness in the RC4, if anyone know the portion of the secret key then possible attacks fully over RC4.

Paul and Maitra in [17], generate the secret key by using the initial state table. They generated some equation on the bases of initial state table and they select some of the bytes of secret key on the bases of guess and the remain secret key find out by using the equation.

As the security of RC4 algorithm depends on the security of the secret key and the internal states of array S, thus many attacks focus on resuming the secret key of the internal states of the array S.

2.3 Evolutionary Algorithm

In real world applications Evolutionary Algorithm (EA), offers practical advantages to the researchers from facing difficult optimization problems [26]. These advantages are manifold, including the simplicity of the algorithm, its robust response to change circumstance, and its flexibility [2].

The EA can be applied to problems where heuristic solutions are not available or generally lead to unsatisfactory results. One of these problems is to implement diverse high-quality Random Number Generators (RNGs). As a result, EA has recently received increased interest, particularly with regard to the manner in which it may be applied for practical problem solving [11].

The conceptual base of EA to simulate the evolution of individual structures is via processes of selection, mutation, and reproduction. The processes depend on the perceived performance of the individual structures as defined by the problem [2].

Algorithm 3 Evolutionary algorithm (EA)

```
1: Starting with the source string.
2: Initialize a population of random parents.
3: Evaluate the population of the parents based on a
   fitness function.
4: while The parent is not yet the target. do
5:    Apply mutation to the selected parents.
6:    Evaluate the fitness of all parents in the population,
      and keep the most fit string as the new parent
7:    repeat until the fit parent converges to the target.
8: end while
9: Return the number of iteration of the best parent.
```

Algorithm 3 is starting with the Source string . A population of a random parents are initialized, then new parent is created by applying reproduction operator (mutation). The fitness of the resulting solution is evaluated towards the Target string. Then, a suitable selection strategy is applied to determine which parent is maintained into the next generation. The process is iterated until a candidate parent with sufficient quality is found. Finally, it returns the number of iteration of the best string closed to the Target string.

3 Proposed Dynamic RC4-EA Encryption Method

A dynamic RC4-EA method is used for encrypting and decrypting the plaintext. Where, the EA algorithm is adapted to generate a dynamic secret key as a seed used in the RC4 encryption algorithm. Then, XOR operation is performed with the final key stream generated from the RC4-EA method on the plaintext to obtain the ciphertext, which is then stored in the database. The proposed RC4-EA method is divided into the following phases.

3.1 Generate Dynamic Secret Key Phase

To generate the dynamic secret key, EA will start with some characters which can be view as string. It will randomly mutate these characters to evolve the string toward the target. The following fitness function is used to judge the new mutated string fitness.

```
$Fitness\ (source,\ target)$
$fitval\ =\ 0$ .
 $for\ i =\ 0\ to\ len\ (source)$
   $fitval\ += (ord(target[i])\ -\ ord(source[i]))^2$ .
     $return\ (fitval)$ .
```

The mutation function is given below, where only one character is mutated by one value at a time.

```
$mutate\ (source)$
$charpos\ =\ rand\ (\ 0\ ,\ len\ (\ source\ )\ -\ 1)$ .
$parts\ =\ list\ (\ source\ )$ .
$parts\ [charpos]\ =\ char\ (\ ord\ (part\ [\ charpos\ ])$
  $+\ \ rand\ (\ 0\ ,\ len\ (source\ -\ 1)))$ .
    $return\ \ ('\ \ '.join\ (\ parts\ )\ )$ .
```

Once reaching the best string; the number of iteration of the best string is used as the random secret key after multiply it n times to obtain the desirable length. Then this secret key is passed to the encryption plaintext phase.

The chart of an EA is illustrated in Figure 2

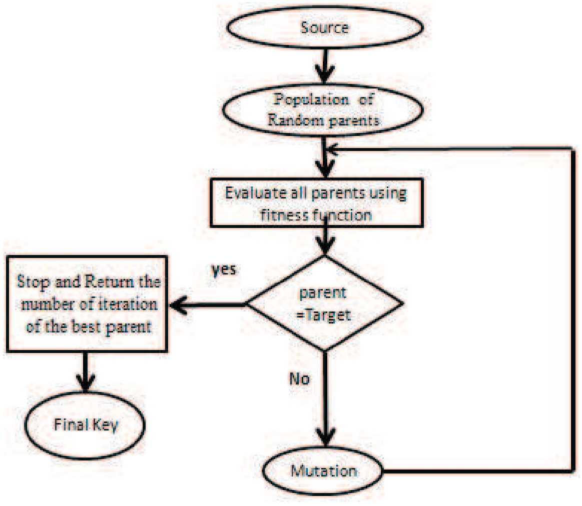

Source
Population of Random parents
Evaluate all parents using fitness function
parent =Target
yes
Stop and Return the number of iteration of the best parent
Final Key
No
Mutation

Figure 2: Generate dynamic secret key phase

3.2 Encryption Plaintext Phase

Figure 3 shows the proposed RC4-EA method for encrypting the plaintext. Where, the dynamic secret key is permute with the initial value (IV) in RC4 algorithm to generate the final key stream for encrypting the plaintext. To ensure the security, the sequence of key streams obtained is never used more than once. Instead of using the fixed secret key to generate key streams; we generate it dynamically and randomly. The randomness of the secret key will enhance the security of the system and prevent a hacker to crack final key stream which XORed with the plaintext. As we get the ciphertext, it is stored in the database.

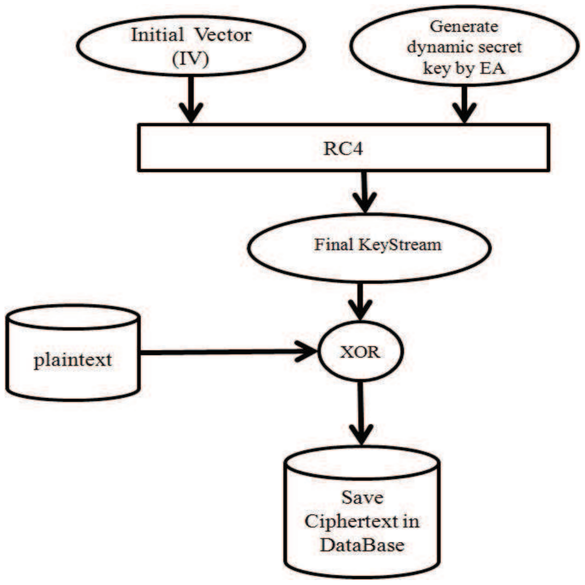

Initial Vector (IV)
Generate dynamic secret key by EA
RC4
Final KeyStream
plaintext
XOR
Save Ciphertext in DataBase

Figure 3: Encryption plaintext phase

4 Implementation Results and Analysis

The proposed RC4-EA encryption method is used for encrypting and decrypting the plaintext. All experiments have been performed using server 32 core AMD Opteron processor 6376 with 32 GB of RAM and 4 RAID 1s. The experiments have been implemented using PHP-MySql language environment.

4.1 Performance Evaluation

For the experiments, the performance evaluation of the encryption and decryption process is measured on different file sizes ranging from 20KB to 100KB. The performance evaluation metrics are:

1) **Encryption time:** It is the time that an encryption method takes to produce a ciphertext from a plaintext. As the encryption time decreases; the performance of the method increases.

2) **Throughputs:** The throughput of the encryption method is calculated as the total plaintext encrypted in KB divided by the encryption time in microseconds. As the throughput increases, the performance increases and the power consumption decreases.

4.2 Experiments and Analysis

To analyze the performance of the proposed RC4-EA encryption method, a performance comparison between the original RC4 encryption algorithm and the proposed RC4-EA method is conducted. Based on the key length, two different cases of experiments are evaluated:

1) With key length 128 bits and data size ranging from 20KB to 100KB.

2) With key length 256 bits and data size ranging from 20KB to 100KB.

Case 1: Key length 128 bits
The average encryption times in μs of the proposed RC4-EA encryption method and the original RC4 encryption algorithm were calculated over 10 different (random) key of length 128 bits. Where, the average encryption times are measured on different plaintext data sizes ranging from 20KB to 100KB. The comparison evaluation are shown in Table 1.

Table 1: Encryption time vs. data size with key length 128 bits

Data Size (KB)	Encryption Time RC4 (μs)	Encryption Time RC4-EA (μs)
20	1037.1208	905.9906
40	1085.9966	940.0845
60	1156.0917	978.9467
80	1192.0929	982.0461
100	1219.0342	988.9603

Table 2 shows the throughputs of the proposed RC4-EA encryption method compared with the original RC4 encryption algorithm.

From the results obtained with key length 128 bits, it is clear that the proposed RC4-EA encryption method shows an enhance in the encryption times and the throughputs compared to the original RC4 encryption algorithm, as shown in Figure 4 and Figure 5.

Table 2: Throughputs vs. data size with key length 128 bits

Data Size (KB)	Throughput RC4 (KB/S)	Throughput RC4-EA (KB/S)
20	19284.15	22075.28
40	36832.52	42549.36
60	51898.99	61290.36
80	67108.86	81462.57
100	82032.15	101116.29

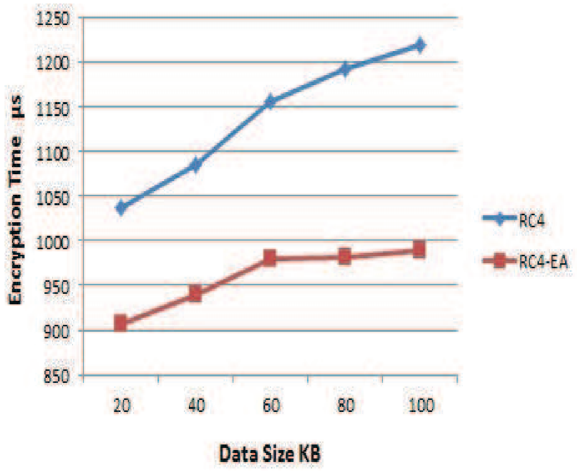


Figure 4: Encryption time of different data size with secret key of length 128 Bits

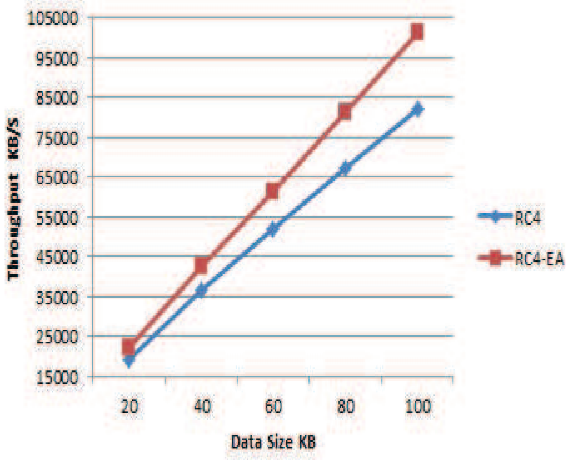


Figure 5: Throughputs for the encryption scheme of different data size with secret key of length 128 bits

Case 2: Key length 256 bits
Table 3 shows the average encryption times in μs of the proposed RC4-EA encryption method and the original RC4 encryption algorithm, with 10 different (random) key

of length 256 bits. The encryption time is calculated for different plaintext data size varying from 20KB to 100KB.

Table 3: Encryption time vs. data size with key length 256 bits

Data Size KB	Encryption Time RC4 (μs)	Encryption Time RC4-EA (μs)
20	1105.0701	1036.9219
40	1125.0973	1044.9886
60	1189.9471	1047.1344
80	1214.9811	1052.8564
100	1260.0422	1065.9695

The throughputs of the proposed RC4-EA encryption method compared with the original RC4 encryption algorithm with key length 256 bits are given in Table 4.

Table 4: Throughputs vs. data size with key length 256 bits

Data Size (KB)	Throughput RC4 (KB/S)	Throughput RC4-EA (KB/S)
20	18098.39	19287.85
40	35552.48	38277.92
60	50422.4	57299.23
80	65844.64	75983.77
100	79362.42	93811.31

From Figures 6 and 7, it is clear that the experiments results with key length 256 bits, show an enhance to the proposed RC4-EA encryption method for the encryption times and the throughputs compared to the original RC4 encryption algorithm.

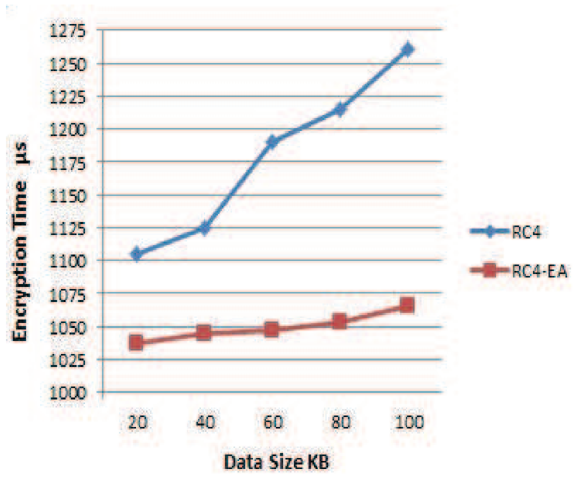


Figure 6: Encryption time of different data size with secret key of length 256 bits

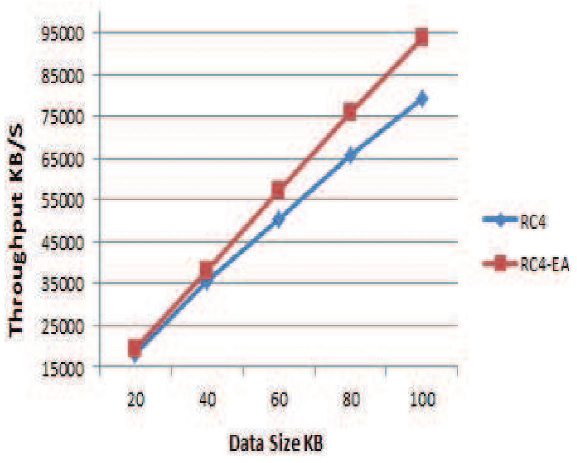
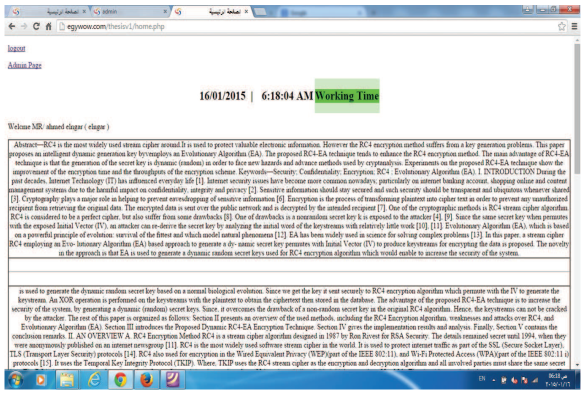


Figure 7: Throughputs for the encryption method of different data size with secret key of length 256 bits

5 Applicable Case Study Using RC4-EA Encryption Technique

The proposed RC4-EA encryption method is applied for data ciphering in Content Management System (CMS) in order to keep the data in high confidential authentication. Where the proposed RC4-EA method is used during the development of the web site (www.egywow.com/thesisv1).

Figure 8 and Figure 9 are samples of data in form of plaintext,and the same data in ciphertext form in (CMS) respectively .


admin
egywow.com/thesisv1/home.php
logout
Admin Page
16/01/2015 | 6:18:04 AM Working Time
EN
06:18 ص
٢٠١٥/٠١/١٦

Welcme MR/ ahmed elngar (elngar)

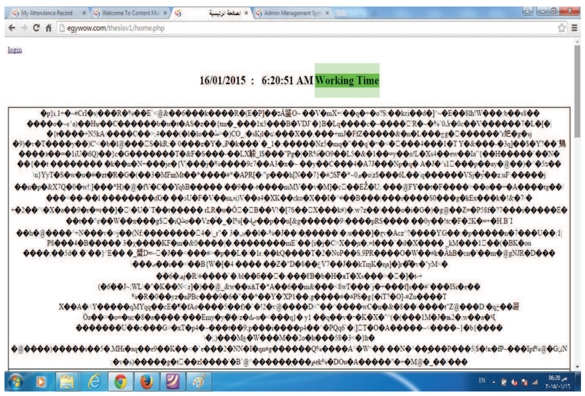
Abstract—RC4 is the most widely used stream cipher around.It is used to protect valuable electronic information. However the RC4 encryption method suffers from a key generation problems. This paper proposes an intelligent dynamic generation key byvemploys an Evolutionary Algorithm (EA). The proposed RC4-EA technique tends to enhance the RC4 encryption method. The main advantage of RC4-EA technique is that the generation of the secret key is dynamic (random) in order to face new hazards and advance methods used by cryptanalysis. Experiments on the proposed RC4-EA technique show the improvement of the encryption time and the throughputs of the encryption scheme. Keywords—Security; Confidentiality; Encryption; RC4 ; Evolutionary Algorithm (EA). I. INTRODUCTION During the past decades, Internet Technology (IT) has influenced everyday life [1]. Internet security issues have become more common nowadays; particularly on internet banking account, shopping online and content management systems due to the harmful impact on confidentiality, integrity and privacy [2]. Sensitive information should stay secured and such security should be transparent and ubiquitous whenever shared [3]. Cryptography plays a major role in helping to prevent eavesdropping of sensitive information [6]. Encryption is the process of transforming plaintext into cipher text in order to prevent any unauthorized recipient from retrieving the original data. The encrypted data is sent over the public network and is decrypted by the intended recipient [7]. One of the cryptographic methods is RC4 stream cipher algorithm. RC4 is considered to be a perfect cipher, but also suffer from some drawbacks [8]. One of drawbacks is a nonrandom secret key k is exposed to the attacker [4]. [9]. Since the same secret key when permutes with the exposed Initial Vector (IV), an attacker can re-derive the secret key by analyzing the initial word of the keystreams with relatively little work [10]. [11]. Evolutionary Algorithm (EA), which is based on a powerful principle of evolution: survival of the fittest and which model natural phenomena [12]. EA has been widely used in science for solving complex problems [13]. In this paper, a stream cipher RC4 employing an Evo- lutionary Algorithm (EA) based approach to generate a dy- namic secret key permutes with Initial Vector (IV) to produce keystreams for encrypting the data is proposed. The novelty in the approach is that EA is used to generate a dynamic random secret keys used for RC4 encryption algorithm which would enable to increase the security of the system.

is used to generate the dynamic random secret key based on a normal biological evolution. Since we get the key it sent securely to RC4 encryption algorithm which permute with the IV to generate the keystream. An XOR operation is performed on the keystreams with the plaintext to obtain the ciphertext then stored in the database. The advantage of the proposed RC4-EA technique is to increase the security of the system, by generating a dynamic (random) secret keys. Since, it overcomes the drawback of a non-random secret key in the original RC4 algorithm. Hence, the keystreams can not be cracked by the attacker. The rest of this paper is organized as follows: Section II presents an overview of the used methods, including the RC4 Encryption algorithm, weaknesses and attacks over RC4, and Evolutionary Algorithm (EA). Section III introduces the Proposed Dynamic RC4-EA Encryption Technique. Section IV gives the implementation results and analysis. Finally, Section V contains the conclusion remarks. II. AN OVERVIEW A. RC4 Encryption Method RC4 is a stream cipher algorithm designed in 1987 by Ron Rivest for RSA Security. The details remained secret until 1994, when they were anonymously published on an internet newsgroup [11]. RC4 is the most widely used software stream cipher in the world. It is used to protect internet traffic as part of the SSL (Secure Socket Layer), TLS (Transport Layer Security) protocols [14]. RC4 also used for encryption in the Wired Equivalent Privacy (WEP)(part of the IEEE 802:11), and Wi-Fi Protected Access (WPA)(part of the IEEE 802:11 i) protocols [15]. It uses the Temporal Key Integrity Protocol (TKIP). Where, TKIP uses the RC4 stream cipher as the encryption and decryption algorithm and all involved parties must share the same secret

Figure 8: Data before the encryption using the proposed RC4-EA encryption method

6 Conclusions

The major contribution of this paper is proposing RC4-EA encryption method for encrypting the plaintext. The proposed RC4-EA encryption method solves the deficiency of RC4 encryption algorithm which are caused by



16/01/2015 : 6:20:51 AM Working Time

[illegible]

Figure 9: Data after the encryption using the proposed RC4-EA encryption method

recovering the secret key. In the proposed RC4-EA encryption method, Evolutionary Algorithm (EA) is used for generating a dynamic random secret key as a seed for RC4 in KSA algorithm to create a pseudo-random initial state. Then, PRGA generates a pseudo-random output final key stream. This final key stream will XORed with the plaintext to generate the ciphertext.

The main advantage of the proposed RC4-EA method is increasing the security of the system by generating a dynamic random secret key. Hence, overcomes the drawback of a non-random secret key as a seed used in RC4. Which makes it difficult for the hacker to trace the plaintext and the secret key used to generate the final key stream. Several experiments with different secret key length (128 bits and 256 bits), were conducted to evaluate the proposed RC4-EA method. Experiment's results show that the proposed RC4-EA encryption method enhances the encryption times and the throughputs compared to the original RC4 algorithm. The proposed RC4-EA encryption method is applied for ciphering the data in Content Management System, to keep these data in high confidential authentication.

References

[1] Anonymous, *RC4 Source Code*, CypherPunks mailing list, Sept. 1994. (http://cypherpunks.venona.com/date/1994/09/msg00304.html)

[2] T. Back, "Parallel optimisation of evolutionary algorithms", in *International Conference on Evolutionary Computation The Third Conference on Parallel Problem Solving from Nature Jerusalem*, Israel, Springer, Berlin, vol. 866, pp. 418-427, 1994.

[3] T. Back, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, NewYork, 1st Edition, 1996.

[4] T. Back, H. P. Schwefel, "An overview of evolutionary algorithms for parameter optimization", *Evolutionary Computation*, vol. 1, n0000o. 1, pp. 1-23, 1993.

[5] D. S. Abd Elminaam, H. M. Abdual Kader, and M. M. Hadhoud, "Evaluating the performance of symmetric encryption algorithms", *International Journal of Network Security*, vol. 10, no. 3, pp. 213-219, 2010.

[6] S. Fluhrer, I. Mantin, A. Shamir, "Weaknesses in the key scheduling algorithm of RC4", in *Proceedings of Annual Workshop on Selected Areas in Cryptography*, Springer, Toronto, vol. 2259, pp. 1-24, 2001.

[7] A. Grosul, D. Wallach, "A related-key cryptanalysis of RC4", Department of computer science, Rice University, Technical report, pp. 01-358, June 2000.

[8] S. Gupta, A. Chattopadhyay, K. Sinha, S. Maitra, and B. Sinha, "High-performance hardware implementation for RC4 stream cipher", *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 730-743, 2013.

[9] M. M. Hammood, K. Yoshigoe and A. M. Sagheer, "RC4-2S: RC4 stream cipher with two state tables", *Information Technology Convergence*, Lecture Notes in Electrical Engineering, Springer Science - Business Media Dordrecht, vol. 253, pp. 13-20, 2013.

[10] Ch. Lin, Y. Li, K. Lv and C. C. Chang, "Ciphertext-auditable identity-based encryption", *International Journal of Network Security*, vol. 17, no. 1, pp. 23-28, 2015.

[11] N. Lourenco, F. Pereira, and E. Costa, "Evolving evolutionary algorithms", in *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO'12)*, ACM New York, NY, USA, pp. 51-58, 2012.

[12] J. Lv, B. Zhang and D. Lin, "Distinguishing attacks on RC4 and a new improvement of the cipher", *International Association for Cryptologic Research (IACR)*, 2013. (https://eprint.iacr.org/2013/176.pdf)

[13] I. Mantin, A. Shamir, "A practical attack on broadcast RC4", *FastSoftware Encryption*, LNCS 2355, pp. 152-164, 2001.

[14] Ch. N. Mathur and K. P. Subbalakshmi, "A light weight enhancement to RC4 based security for resource constrained wireless devices", *International Journal of Network Security*, vol. 5, no. 2, pp. 205-212, 2007.

[15] A. Mousa, and A. Hamad, "Evaluation of the RC4 algorithm for data encryption", *International Jornal of Computer Science and Applications*, vol. 3, no. 2, pp. 44-56, 2006.

[16] Pardeep and Pushpendra, "A pragmatic study over the different stream cipher and on different flavor of RC4 stream cipher", *International Journal of Computer Science and Network Security*, vol. 12, no. 3, pp. 37-42, 2012.

[17] G. Paul, S. Maitra, "RC4 state in formation at any stage reveals the secret key", in *Presented in the 14th Annual Workshop on Selected Areas in Cryptography*, SAC, Ottawa, Canada, LNCS vol. 4876, pp. 360-377, 2007.

[18] S. Paul, and B. Preneel, "A new weakness in the RC4 keystream generator", *Fast Software Encryption (FSE'04)*, Springer-Verlag, vol. 3017, pp. 245-259, 2004.

[19] K. Sharma, M. K. Ghose, D. Kumar, R. Singh, and V. Pandey, "A comparative study of various security approaches used in wireless sensor networks", *International Journal of Advanced Science and Technology*, vol. 17, pp. 31-44, Apr. 2010.

[20] S. O. Sharif, S. P. Mansoor, "Performance analysis of stream cipher algorithms", in *3rd International Conference on Advanced Computer Theory and Engineering (ICATE'10)*, vol. 1, pp. 522-525, 2010.

[21] A. K. Singha, S. G. Samaddar, S. R. Sahooc, and G. Mathewd, "Increasing robustness of RC4 family for automated selection of ciphersuites", in *International Conference on Communication Technology and System Design (ICCTSD'12)*, vol. 30, pp. 4552, 2012.

[22] L. Stosic, and M. Bogdanovic, "RC4 stream cipher and possible attacks on WEP", *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 3, pp. 110-114, 2012.

[23] P. Subsorn, S. Limwiriyakul, "A Comparative analysis of the security of internet banking in Australia: A customer perspective", in *2nd International Cyber Resilience Conference*, Western Australia, pp. 70-83, 2011.

[24] P. Subsorn, S. Limwiriyakul, "A comparative analysis of internet banking security in Thailand: A customer perspective", in Proceedings of Engineering, vol. 32, no. 37, pp. 260272, 2012.

[25] Q. Yu, C. Zhang, "RC4 state and its applications", in *Ninth Annual International Conference on Privacy, Security and Trust*, pp. 264-269, 2011.

[26] X. Yu, and M. Gen, *Introduction to Evolutionary Algorithms*, Springer London Dordrecht Heidelberg New York, ISBN: 978-1-84996-129-5, XVII, pp. 418, 2010.

Ashraf Aboshosha graduated with a B.Sc. in industrial electronics from Menoufia University, Egypt at 1990. At 1997 he received his M.Sc. in automatic control and measurement engineering. From 1997 to 1998 he was guest researcher at research centre Jlich (FZJ), Germany. From 2000 to 2004 he was a doctoral student (DAAD-scholarship) at Eberhard-Karls-University, Tbingen, Germany. Where he received his Doctoral degree (Dr. rer. nat.) at 2004. He is the CEO of ICGST LLC, Delaware, USA.

Kamal Abdelraouf ElDahshan is a professor of Computer Science and Information Systems at Al-Azhar University in Cairo, Egypt. An Egyptian national and graduate of Cairo University, he obtained his doctoral degree from the Universit de Technologie de Compigne in France, where he also taught for several years. During his extended stay in France, he also worked at the prestigious Institute National de Tlcommunications in Paris. Professor ElDahshan's extensive international research, teaching, and consulting experiences have spanned four continents and include academic institutions as well as government and private organizations. He taught at Virginia Tech as a visiting professor; he was a Consultant to the Egyptian Cabinet Information and Decision Support Centre (IDSC); and he was a senior advisor to the Ministry of Education and Deputy Director of the National Technology Development Centre. Prof. ElDahshan has taught graduate and undergraduate courses in information resources and centers, information systems, systems analysis and design, and expert systems. Professor ElDahshan is a professional Fellow on Open Educational Resources as recognized by the United States Department of State. Prof. Eldahshan wants to work in collaboration with the Ministry of Education to develop educational material for K-12 levels. Prof. Eldahshan is interested in training instructors to be able to use OER in their teaching and hopes to make his university a center of excellence in OER and offer services to other universities in the country.

Eman K. Elsayed Bachelor of Science from computer science Department, Cairo University 1994, Master of computer science from Cairo university 1999, and computer science PHD 2005 from Alazhar university. I Published eleven papers until 2010 in data mining, ontology and e-learning. I am a member in egyptian mathematical society and inteligent computer and information systems society.

Ahmed A. Elngar graduated with a B.Sc. in computer Science from computer science Department, Al-Azhar University 2004, Master of computer science in Intrusion Detection System (IDS) from Ain Shanm university 2012. Now he is a P.hD student at computer science Department, Al-Azhar University. Also he is a member in Egyptian Mathematical Society (EMS) and International Rough Set Society(IRSS).