An Effective RC4 Stream Cipher

T.D.B Weerasinghe, Member IEEE

***Abstract*-RC4 is the most widely used stream cipher around. A lot of modifications of RC4 cipher can be seen in open literature. Most of them enhance the secrecy of the cipher and the security levels have been analyzed theoretically by using mathematics. In this paper, a new effective RC4 cipher is proposed and the security analysis has been done using Shannon's Secrecy theories where numerical values are obtained to depict the secrecy. The proposed cipher is a combination of Improved RC4 cipher proposed by Jian Xie et al and modified RC4 cipher proposed by T.D.B Weerasinghe, which were published prior to this work. Combination is done in such a way that the concept used in the modified RC4 algorithm is used in the Improved RC4 cipher by Jian Xie et al. Importantly, an immense improvement of performance and secrecy are obtained by this combination. Hence this particular modification of RC4 cipher can be used in software applications where there is a need to improve the throughput as well as secrecy.**

Index Terms—**Data encryption, RC4 modifications, Secrecy of ciphers, Stream cipher**

I. INTRODUCTION

RC4 is the most widely used stream cipher in the world. It is used in protocols like SSL, WEP, WPA, and applications like Skype, Remote Desktop and Microsoft Point-to-Point. There are many other applications which use RC4 as the encryption algorithm. It is used in hardware based encryption mechanisms as well. Due to its light weight it has become popular despite of various attacks on RC4 [2]. In open literature, there are a lot of publications, which describe various attacks on RC4 and a lot of them are theoretical. This is understood by the presentations by Klein A. [2], Fluhrer S. et al [7] and Paul G. et al [8]. There are a lot of publications on hardware implementations of RC4 to enhance performances. [4, 5, 6] Many known attacks on RC4 that unveil some part of the secret internal state, are based on fixing some elements of the S-box with the values of *i* and *j* that give information about the outputs at certain rounds with probability one or very close to one. This leads in distinguishing attacks on the cipher and helps to obtain the secret internal state with probability that is notably larger than expected. So, the correlations between the internal state and the external state violate the "randomness" feature of a cipher, at once. [3]

Thus, the objective of this research was, proposing an effective modification of RC4 which eliminates the common attacks on RC4 and simultaneously gives a higher throughput. Another important aspect of this work is obtaining a numeric result for the secrecy of ciphers that gives a good idea about the modifications, rather than theoretical analysis. So, the secrecy analysis is based on Shannon's theories of Secrecy where numerical values can be obtained in-order to emphasize the secrecy.

According to the literature survey, an improvement in performance is obtained by the *Improved RC4 cipher* proposed by Jian Xie et al which is based on RC4A [1]. RC4A has eliminated most of the known attacks on RC4 [3]. So, it was selected to be combined with another simple modification of RC4 algorithm, which is named *"A Modified RC4 cipher"* (proposed by T.D.B Weerasinghe [10]), thinking that, the particular combination will output an efficient and secure modification of RC4 especially for a software implementation. In this research, all source codes were written in Java SE with the help of Netbeans IDE 7.2.

II. MODIFIED RC4 CIPHERS USED IN THIS RESEARCH

A. *Improved RC4 Cipher*

This particular, *_Improved RC4 Cipher* was proposed by Jian Xie et al in [1]. It is an improved version of RC4A proposed by Souradyuti Paul in [3]. The specialty of this improved version of RC4 cipher is that it uses two keys. Improved RC4 cipher which can be considered as a modification of **RC4A** eliminated some statistical attacks which had been possible for RC4A. There are attacks on RC4 based on the relationships between the internal states of the S boxes. [1]

PSEUDO CODE I
KSA OF IMPROVED RC4 PROPOSED BY JIAN XIE ET AL: [1].

```
for i= (0 to N-1)
{
   S1[i]=i;
   S2[i]=i;
}
j1=j2=0;
for i=0 to N-l
{
   jl=( j1+S1[i]+kl[i]) mod N;
   swap(S1[i], S1[j]);
   j2=( j2+S2[i]+k2[i]) mod N;
   swap(S2 [i], S2[j]);
}
```

PSEUDO CODE II
PRGA OF IMPROVED RC4 PROPOSED BY JIAN XIE ET AL: [1].

```
i=jl=j2=0;
Loop
{
i=i+1;
   jl= jl+S1[i];
   swap(S1[i], S1[j]);
   j2= j2+S2[i];
   swap(S2[i], S2[j]);
   Output= S1 [(S1 [i]+ S1[j]) mod N];
   Output= S2[(S2 [i]+ S2[j]) mod N];
   swap(S1[S2[j1]], S1[S2[j2]]);
   swap(S2[S1[j1]], S2[S1[j2]]);
}
```

T.D.B Weerasinghe is with IFS R&D International (Pvt) Ltd, Sri Lanka as a Software Engineer. He is involved in research in Cryptography and Steganography apart from his work. He can be contacted through email: tharindu.weerasinghe@gmail.com.

More importantly, the outcome of the research done by Jian Xie et al in [1] shows that the above algorithm is faster than original RC4 and also it is secure than the original RC4.

B. Modified RC4 cipher, proposed earlier

Modification was done in the later part of the PRGA, or in other words, after major operations of PRGA. It is a simple alteration of introducing another XOR operation. If it is further explained; the plaintext message bit is XOR'd not only with the generated key but also with j.

PSEUDO CODE III
KSA OF THE MODIFIED RC4, SAME AS RC4

```
for i = 0 to 255
   S[i] = i;
j=0

for i = 0 to 255
   j = (j+S[i]+K[i mod l]) mod 256;
   swap S[i] and S[j];
```

PSEUDO CODE IV
PRGA OF THE MODIFIED RC4

```
i = 0, j=0;

for x = 0 to (M-1)
{
   i = (i+1) mod 256;
   j = (j+S[i]) mod 256;
   swap S[i] and S[j];
   GeneratedKey = S[ (S[i] + S[j]) mod 
256] ;
   Output = M[x] XOR GeneratedKey XOR j;
}
```

Where "M" is the plain text message and Output is the cipher text (C) of the plaintext message (M).

More importantly, the outcome of the research done by T.D.B Weerasinghe [10] shows that the above algorithm is faster than original RC4 and it is secure.

C. Algorithm developed in this research

So, in this research, the above two algorithms are combined in-order to obtain a faster and secure RC4 stream cipher. The combination is done in the PRGA. So, the KSA in Improved RC4 remains the same in this cipher too. The changed PRGA can be mentioned as follows:

PSEUDO CODE V
PRGA OF THE EFFECTIVE RC4, PROPOSED IN THIS PAPER

```
i=j1=j2=0;
Loop
{
   i=i+1;
   j1= j1+S1[i];
   swap(S1[i], S1[j]);
   j2= j2+S2[i];
   swap(S2[i], S2[j]);
   GeneratedKey1= S1 [(S1 [i]+ S1[j]) mod 
N];
   GeneratedKey2= S2[(S2 [i]+ S2[j]) mod 
N];
   swap(S1[S2[j1]], S1[S2[j2]]);
   swap(S2[S1[j1]], S2[S1[j2]]);
   Output = M[x] XOR GeneratedKey1 XOR 
j1;
   Output = M[x] XOR GeneratedKey2 XOR 
j2;
}
```

III. RESEARCH METHOD AND THEORIES USED

A. Performance Analysis

To analyze the performance of the algorithms, all 4 of them ***(original RC4, Modified RC4 proposed by T.D.B Weerasinghe, Improved RC4 proposed by Jian Xie et al and the new algorithm proposed in this paper)*** are used. Each was tested for data sizes ranging from 10KB to 100KB. Average encryption times were calculated after having obtained encryption times in several similar no. of experiments for each plaintext data size. With the average encryption times, the throughputs were calculated. With these two results, performance of each algorithm can be analyzed.

Since there were two algorithms (among the four that were tested and illustrated in this paper), a reasonable and impartial testing mechanism to analyze the performance, has been adhered. For that refer the sub-section C under this section (III).

B. Secrecy Analysis

The method of analyzing *Secrecy* was the same way as above. The only difference was some extra calculations and methods had to be used in-order to obtain the *secrecy of ciphers* according to the theories of *Shannon*. With this analysis, a basic idea on the security level of the algorithms has been obtained. Each algorithm was tested for data sizes ranging from 10KB to 100KB. Average secrecy values were calculated after having obtained secrecy values in several similar no. of experiments for each plaintext data size.

The theories behind these secrecy calculations are illustrated in the sub-section D. A reasonable and impartial testing mechanism to analyze the secrecy has been adhered. For that refer the sub-section C.

C. Information about the initial key(s)

With the help of **java.security.SecureRandom** and **java.math.BigInteger**, streams of random alpha - numeric characters are generated.

Example code:

```
new BigInteger(640, random).toString(32);
```

Since the key size is fixed to 128 bits, the first parameter of **BigInteger** is fixed to 640. The above code always generates a unique alpha-numeric random key.

Important:* All four algorithms should be tested under similar circumstances in-order to get an impartial result. *In each experiment the same random key(s) were used for all four algorithms.

The algorithm proposed by Jian Xie et al and the one presented in this paper use two random keys (k1 and k2). So when the encryption times of the original and the modified RC4 are concerned, a conflict occurred as there was a doubt, which key should be used to calculate the encryption times and the secrecy values. Average for those two keys had to be considered. If it is further explained with an example:

Original RC4 and Modified RC4 *(proposed by T.D.B Weerasinghe [10])* are tested for the keys, k1 and k2 in each experiment for each data size separately, in the same program. (In the performance analysis as well as the secrecy analysis) Since a generic result should be obtained the average of above results are considered. This is for the Original RC4 and the modified one as there is only one initial key is involved there not like in the improved RC4 and the one proposed here.

In all the experiments the key size used is fixed. i.e. 128 bits. Improved algorithm and its modification (the one which is introduced here) use two initial keys; they are also
128 bits, because the variable is the data size.

D. Theories and definitions related to Shannon's Secrecy of Ciphers

Definition 1: Entropy of a message:

Entropy of a message X is called H(X), which is the minimum number of bits required to encode all possible meanings of the message, assuming the occurrences of all messages are equally likely. [9, 10]

Mathematical equation: $H(X) = -\sum_{i=1}^{n} P(x_i)\log P(x_i)$

Definition 2: Uncertainty of a message:

Uncertainty of a message is the number of plaintext bits that must be recovered when the message is encrypted, in-order to obtain the plaintext. The uncertainty of a message is measured by its entropy. Higher the number, higher the uncertainty [9, 10]

Definition 3: Equivocation:

Equivocation is the uncertainty of a message which is reduced when there is additional information provided. [9, 10]

Mathematical equation:

$$H_Y(X) = \sum\{X,Y\}P(X,Y)\log_2[P_Y(X)]$$

$$H_Y(X) = \sum\{Y\}P(Y)\sum\{X\}P_Y(X)\log_2[P_Y(X)]$$

Definition 4: Secrecy of a cipher:

Now, all the above definitions and equations lead us to manipulate the secrecy of a cipher. It is calculated in terms of the key equivocation $H_c\,(K)$ of a key K for a given cipher text C; that is the amount of uncertainty in K given C: [9, 10]

$$H_c(K) = \sum\ \{C\}P(C)\sum\ \{K\}P_c(K)\log_2[P_c(K)]$$

These theories were used in my previous work [10] and all of them were derived from theories of Shannon related to entropy and secrecy. Claude Elwood Shannon [April 30, 1916 – February 24, 2001] is called the Father of Information Theory.

All the above definitions and equations are illustrated from the lecture notes of Dr.Issa Traore on Shannon's secrecy, University of Victoria, British Columbia, Canada, which were available in open literature at the time of research.

E. Method of calculating the secrecy of ciphers

$$H_c(K) = \sum\ \{C\}P(C)\underbrace{\sum\ \{K\}P_c(K)\log_2[P_c(K)]}_{\textbf{\textit{Part 1}}}$$

<u>How the secrecy calculation is done in this research:</u>

- First, Consider the ***Part 1***: It is the entropy of the key K, given the relevant cipher. (Cipher text C, has been obtained using this particular key K)
  - o Calculate how often each key byte is appeared in the key.
  - o And then calculate the probability of each byte appears (given the cipher) in the key and get the summation of Pc(K) * $\log_2 P_c(K)$.
- Then consider the other half: Calculating P(C) and then the summation.

- Calculate how often each cipher byte has appeared in the cipher text.
- And then calculate the probability of each byte appears in the key and get the summation (for all possibilities of the cipher bytes). Note that this cipher is obtained after the plaintext operations with the key; i.e. this cipher is correlated to the above key.
- Then get the multiplication of "***Part 1***" and P(C) is calculated and finally the summation of all possibilities is calculated.

F. *Method of calculating the secrecy of ciphers*

PSEUDO CODE VI
CALCULATION OF THE SECRECY OF A CIPHER

```
    double entropy = 0;
    double secrecy = 0;

    final int[] countedKey =
    countByteDistribution(key, start,
    key.length-1);


    final int[] countedCipher =
    countByteDistribution(cipher, start,
    cipher.length-1);

    for (int i=0;i<256;i++)
    {
           final double p_k = 1.0 *
           countedKey[i] / key.length; final
           double p_c = 1.0 *
           countedCipher[i] / cipher.length;
           if (p_k > 0)
           {
                  entropy += p_k *
           log2(p_k);
                  secrecy += -p_c *
           entropy;
           }
    }
    return secrecy;
}
```

IV. RESULTS

All experiments were done in an Intel® Core™ 2 Duo machine having 3.23 GB of RAM. (Operating System: Windows XP)

A. *Performance of ciphers*

Overall results of average encryption time and throughput over data size can be illustrated as follows:

TABLE I
AVERAGE ENCRYPTION TIME VS DATA SIZE

Data Size(KB)	Average Encryption Time (microseconds)			
	RC4	Modified RC4	Improved RC4	Efficient RC4
10	3140	3030.5	2616	952
20	6008.5	5385.5	3217	2019
30	7558.5	6962	3777	2821
40	8993	8591	4489	3836
50	10728.5	9882.5	6191	4382
60	11833	12147.5	7769	4940
70	13142.5	12837.5	8047	5818
80	14590.5	13954	8468	5722
90	16257.5	15949	8056	6123
100	17084	17142	8620	6329

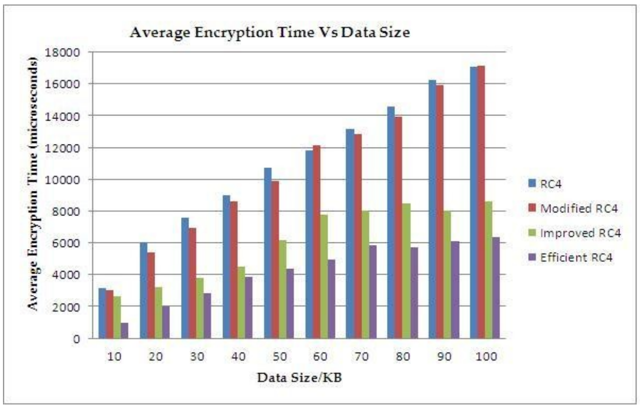

Fig. 1. Average Encryption Time (in microseconds) over Data Size. Average was taken out of similar no. of experiments for each data size. Efficient RC4 cipher proposed in this paper shows the lesser encryption time and the Original RC4 cipher shows the highest encryption time almost for every data size from 10 KB to 100KB.

TABLE II
AVERAGE THROUGHPUT VS DATA SIZE

Data Size(KB)	Average Throughput ()			
	RC4	Modified RC4	Improved RC4	Efficient RC4
10	3184.71	3299.79	3822.63	10504.20
20	3328.62	3713.68	6216.97	9905.89
30	3969.04	4309.11	7942.81	10634.53
40	4447.90	4656.04	8910.67	10427.53
50	4660.48	5059.45	8076.24	11410.31
60	5070.57	4939.29	7723.00	12145.75
70	5326.23	5452.78	8698.89	12031.63
80	5483.02	5733.12	9447.33	13981.13
90	5535.91	5642.99	11171.80	14698.68
100	5853.43	5833.63	11600.93	15800.28

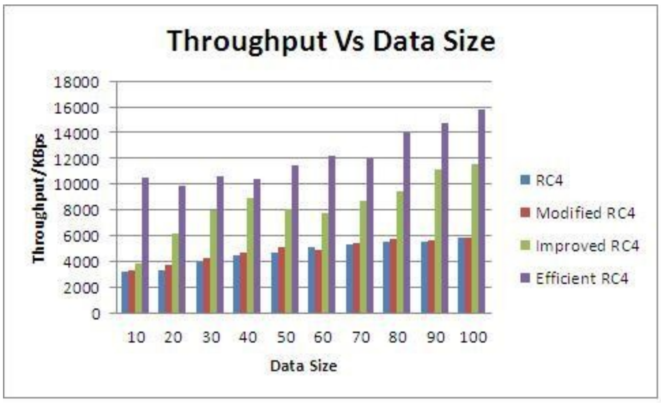

Fig. 2. Average Throughput (in KBps) over Data Size. Throughput was calculated by dividing the appropriate data size by the encryption time in seconds. Thus KBps values were obtained. The highest throughput was achieved by the Efficient RC4 proposed in this paper for each data size ranging from 10KB to 100KB.

B. *Secrecy of ciphers*

TABLE III
SECRECY OF CIPHERS VS DATA SIZE

Data Size(KB)	Secrecy of Ciphers			
	RC4	Modified RC4	Improved RC4	Efficient RC4
10	0.215607	0.216147231	1.618582374	1.618749911
20	0.216507	0.208852021	1.678654897	1.67857242
30	0.212457	0.208279356	1.648133091	1.648133091
40	0.2109	0.204443823	1.612979897	1.612939685
50	0.205461	0.204145458	1.599298717	1.599251074
60	0.211551	0.206477702	1.65231023	1.652282872
70	0.19737	0.204882622	1.5548991	1.5548991
80	0.1989	0.19447206	1.544407467	1.544353522
90	0.199106	0.198484092	1.611137465	1.6111198
100	0.195248	0.187029681	1.497070348	1.497047862

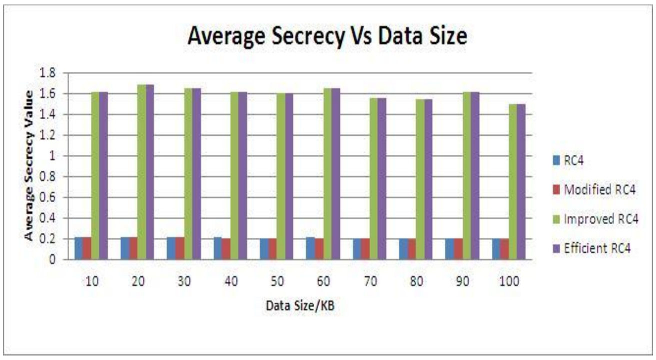

Fig. 3. Average Secrecy over Data Size. Secrecy is the representation of security level of each algorithm. Secrecy was calculated purely based Shannon's secrecy theories. The highest secrecy values were achieved by the Efficient RC4 proposed in this paper for each data size ranging from 10KB to 100KB.

V. CONCLUSION

Proposed algorithm in this paper is efficient; in other words, it is cost-effective than the original RC4 and other modifications of RC4 used in the research. Since there are numerical values to depict the security level of the ciphers, anyone can get a big picture of the secrecy of the relevant ciphers. Higher values of secrecy are shown by the new stream cipher, which means the randomness of the cipher is higher than that of others, which is feature of a good cipher. The reason behind having a higher secrecy can be the increased number of more operations and alterations in the PRGA.

The new algorithm can be implemented using parallelism because there are capable segments that can be parallelized. It will boost the performance of the algorithm further. But, in this research such a parallelism is not used. Even without parallelism it has produced a higher throughput, may be because there are similar steps where the Operating System itself can calculate quickly. Apart from that, if we are to elaborate more on the performance improvement then this cipher should be analyzed further!

ACKNOWLEDGMENT

My gratitude goes to *Dr. Manjula Sandirigama, Dr. Roshan Ragel, Dr. Disala Uduwawala, Dr. Janaka Wijekulasooriya, Prof. Kithsiri Liyanage and Dr. Jagath Kumara,* senior lecturers, Faculty of Engineering, University of Peradeniya who guided me to do research in scientific way during my post graduate studies (MSc.Eng) at the University of Peradeniya.

REFERENCES

[1] J. Xie, X. Pan, "An Improved RC4 Stream Cipher", *2010 International Conference on Computer Application and System Modeling, (ICCASM 2010)*, pp. (V7) 156-159, 2010.

[2] A. Klein, "Different attacks on the RC4 stream cipher", Lecture Notes, Department of Pure Mathematics and Computer Algebra, Ghent University, Belgium.

[3] S. Paul, B. Preneel, "A New Weakness in the RC4 Key stream Generator and an Approach to Improve the Security of the Cipher", *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004. Revised Papers,* vol.3017, no., pp.245,259, 2004.

[4] S. S. Gupta, A. Chattopadhyay, K.Sinha, S. Maitra, B. Sinha, "High Performance Hardware Implementation for RC4 Stream Cipher", *Computers, IEEE Transactions on,* vol.62, no.4, pp.730,743, April 2013 doi: 10.1109/TC.2012.19

[5] Z. Wang, T. Arslan, A. Erdogan, "Implementation o f Hardware Encryption Engine for Wireless Communication on a Reconfigurable Instruction Cell Architecture", *Electronic Design, Test and Applications, 2008. DELTA 2008. 4th IEEE International Symposium on*, vol., no., pp.148, 152, 23-25 Jan.2008 doi:10.1109/DELTA.2008.100

[6] P. Kitsos, G. Kostopoulos, N. Sklavos, O. Koufopavlou, "Hardware implementation of the RC4 stream cipher", *Circuits and Systems, 2003 IEEE 46th Midwest Symposium on,* vol.3, no., pp. 1363,1366 Vol. 3, 27-30 Dec. 2003 doi:10.1109/MWSCAS.2003.1562548

[7] S. Fluhrer, I. Mantin, A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", SAC2001 (S.Vaudenay, A.Youssef, eds.), vol.2259, no., pp.1,24, Springer, Verlag, 2001.

[8] G. Paul, S. Maitra, "RC4 State In formation at Any Stage Reveals the Secret Key", Proceedings of Selected Areas in Cryptography. LNCS, vol. 4876, no., pp. 260,377, Springer, Heidelberg 2007.

[9] I. Traore, "Introduction Cryptography: Mathematical Background, In the lecture notes, Department of Electrical and Computer Engineering, University of Victoria, British Colombia, Canada.

[10] T.D.B Weerasinghe, "Analysis of a Modified RC4 Algorithm", International Journal of Computer Applications, vol. 51, no. 22, pp. 13-17 Aug. 2012 doi:10.5120/8341-1617