Advances in Mechanics
Volume 9, Issue 3, 2021
Page 1467-1476

Improvement of RC4 Security Algorithm

Hasan H. Al-badrei[1]
[1]Department of Computer Science, College of Computer Science and Information Technology, University of Basrah, Basrah, IRAQ(E-mail: hasanabohmod@gmail.com)

Imad S. Alshawi[2]
[2]Department of Computer Science, College of Computer Science and Information Technology, University of Basrah, Basrah, IRAQ(E-mail: emadalshawi@gmail.com)

Article history:*Received 20 October 2021* *Revised 22 November 2021* *Accepted 14 December 2021*

Abstract
Data confidentiality is the most important security service at present, and to ensure access to this service, efficient encryption algorithms are used to overcome unauthorized access to data, as attacks often exploit weaknesses in these algorithms. Encryption algorithms are used to protect the data, aimed to have a fast rate of execution, low level of complexity, and high level of security. To overcome these challenges, symmetric encryption algorithms are often used. The Rivest Cipher 4 (RC4) is the most common encryption algorithm for meeting the conditions for effective encryption. However, this algorithm has been proven to be vulnerable to a variety of assaults. This is accomplished by exploiting flaws in the critical phases of the generation process, as none of these algorithms are adequately utilized in randomization. Therefore, this paper aimed to improve the RC4 algorithm by overcoming its weaknesses. The proposed method, called Improvement RC4 (IRC4), improves the RC4 key generation based on multiple chaos maps. In addition, IRC4 is stronger against most attacks. This makes the proposed algorithm more secure.

Keywords: Key Scheduling, Random Number, RC4, Security Stream Cipher

1. Introduction

Currently, data and information protection are critical in order to prevent data from being compromised by others; Attacks should be avoided as many individuals are not responsible for the misuse of technologies especially data theft and eavesdropping[1]. Thus, the encryption methods can be used to data protection. With goals of data integrity, confidentiality, non-repudiation, and authentication, the cryptography is used to protect data by converting it to another language that cannot be identified[1]-[2].

Traditional encryption algorithms are complex and energy-intensive [REF]. As a result, symmetric key ciphering should be used in security systems. One of the most widely used algorithms is the Rivest Cipher 4 (RC4) symmetric stream cipher. The RC4 stream cipher algorithm provides fast encryption and decryption, low resource usage, easy to understand and implement, and low time and space complexity compared to other algorithms [3]. It consists of two phases: KSA, and PRGA. Both phases generate a keystream which is then used to encrypt the data [4]. The size of the encryption file generated by the RC4 algorithm is equal to the size of the original text file and for this reason; it does not reduce the storage space and does not require a long time to implement the encryption process. Therefore, the simple design and non-random behavior between the key, the plaintext, and the ciphertext, as well as statistical failures are some of the weak points of the algorithm [5]. In this paper, we introduced the improving of the RC4 algorithm, which called IRC4. IRC4 will enhance the key that generated by the PRGA algorithm, taking into account the preservation of the original structure of the algorithm. The proposed method leads to an increase the randomness in the resulting key and then to enhance the security of the resulting encryption when compared to the original algorithm variables.

The rest content of this paper is as follows: the related work is presented in Section 2. The RC4 is presented in Section 3. The improvement RC4 Algorithm is explained in Section 4. Evaluation performance is explained in detail in Section 5.

2. The Related Work

RC4 algorithm is one of the most commonly used stream ciphers in a variety of security protocols. It used to generate pseudo-random code; it accepts a secret key as input and uses a deterministic method to generate a stream of random bits. For that, an intruder's aims, in attack the RC4, is to look for non-random behavior in the internal state or the output keystream [REF].

Several researchers tried to improve the safety of RC4 by proposing many types of improvements. Zoltak [6] proposed the (VMPC) which is designed to be effective in program applications To overcome the KSA weakness which Fluhrer and et al. were defined in[7], In comparison to RC4, the structure of the (PRGA) in VMPC was more complicated, increasing the algorithm strength against assaults. Mironov [8]Introduced a new RC4 model and studied it using the random change principle, As a result of this analysis, recommends eliminating the 512 bytes found at the beginning to prevent the weakness that has resulted in a longer execution period. Preneel and Paul [9] proposed an enhancement over RC4 called (RC4A). after discovering a new weakness of statistical in the RC4 keystream generator's first two output bytes They said that the number of outputs necessary to differentiate the output of an RC4 random sequence with bias is 128 and that 256 should be used to overcome this bias. RC4A is thought to be resistant to most of RC4's flaws, notably the distribution flaw in the first two output bytes. However, Maximov [10] developed a differentiating attack on both VMPC and RC4A after a year that can differentiate the cipher output and random values. Hamad and Mousa[11] investigated the impact of several RC4 algorithm parameters by analyzed, such as execution time and file size, and found that the file size and the length of the encryption key had an impact on encryption and decryption speed. Pateriya and Pardeep [12]proposed the (PC-RC4) method as an enhancement to the RC4 to improve the work of both PRGA and KSA algorithms in the randomness, yet there is increases the time of execution. Hammod and et al proposed the RRC4 method, which enhanced the RC4's randomness Furthermore, an RC4 with two state tables (RC4-2S) developed the key obstetrics time while also surpassing the randomness of the keys produced[13].

3. RC4 ALGORITHM

RC4 it is a widely accepted and popular stream cipher devised in 1987 by Ron Rivest. The RC4 algorithm is one of the fastest encryption algorithms used for encryption within a lightweight, robust cipher in terms of memory footprint, power consumption, the flexible main size and CPU and is utilized in email in many popular protocols, such as WEP and TLS /SSL[14]. The security of the algorithm is based on a pseudorandom key scheduling procedure with a configurable key length from 1 to 256 bytes (8 bits to 2048 bits). This is used by initializing the initial vector (S) is completely independent of the plaintext[15]. RC4 algorithm includes two stages called KSA and PRGA algorithms. In the RC4 algorithm two variables, i and j are used. The variable i is a pointer that is increased by 1 at each step, while for the variable j it is a pseudorandom pointer whose content is updated based on the key K and the state vector S.

3.1 *KSA Algorithm*:

in this part of the RC4 Algorithm, it takes the key stored in K as input and is l bytes long, and used K to rearrange the values in the vector S [16]. The KSA sets i and j to zero, and S to change the identity. It then steps i across S looping N times(N=256), and updating j by adding the i-th entries of S and K.Each iteration ends with a two-byte operation in the vector S, indicating the current values of the variables i and j[17], the KSA steps is depicted in Algorithm 1.

Algorithm1. Key Schedule Algorithm(KSA)[16]

```
Input: Secret key K
K: key length
Output: Internal state S
j ⟵ 0
for i ⟵ 0 to N – 1 Do
    S [i ] ⟵ i
end for
for i ⟵ 0 to N - 1 Do
    j ⟵ (j + S [i] + K [i mod k]) mod 256
    Swap S [i] with S [j]
end for
Return (S)
```

3.2 PRGA Algorithm:

in this stage of the RC4 algorithm, setting both i and j to zero, and then does four actions in order: it increases i as a counter, adds S[i] to j, swap the two entries of S indicated by the present values of i and j, and outputs the value of S at index S[i] + S[j] as the value of z [18] the PRGA is shown in algorithm 2.

Algorithm2.Pseudo Random Generation Algorithm (PRGA)[16]

```
Input: Internal state S, generated by KSA
Output: keystream Z
i ⟵ 0
j ⟵ 0
for each new message byte Do
    i ⟵ (i+ 1) mod N
    j = (j + S [i]) mod N
    Swap S [i] with S [j]
    Z= S (S [i] + S [j]) mod N
end for
Return (Z)
```

The XOR-ed operation will be performed between the n bit represented by the z value with the n bit of the original message and the output will be a ciphertext of length n bit, on the other hand, to restore the original text from the ciphertext the XOR-ed operation is used between the z value and the ciphertext and the text length is n bits.

4. IMPROVED THE RC4 ALGORITHM

The proposed improvement aims to provide a high level of randomness and complexity to bypass RC4 vulnerabilities by introducing improved RC4 key generation as shown in the diagram below. The original key and id number and the output use as a key in the RC4 algorithm for both encryption and decryption, the schematics below are shown in Figure1.(a) and Figure1.(b) respectively for the optimization process. Note that both algorithms. Note that both the above algorithms are 1 and 2, which include the encryption code and decryption code respectively. Note that both the above algorithms are 1 and 2, which include the KSA code and PRGA code respectively.

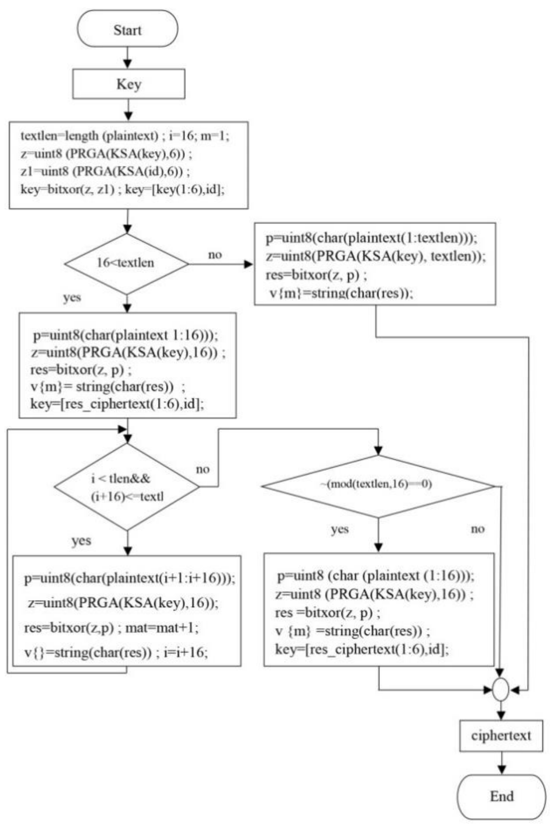


Figure1.(a). IRC4 Encrypt Algorithm

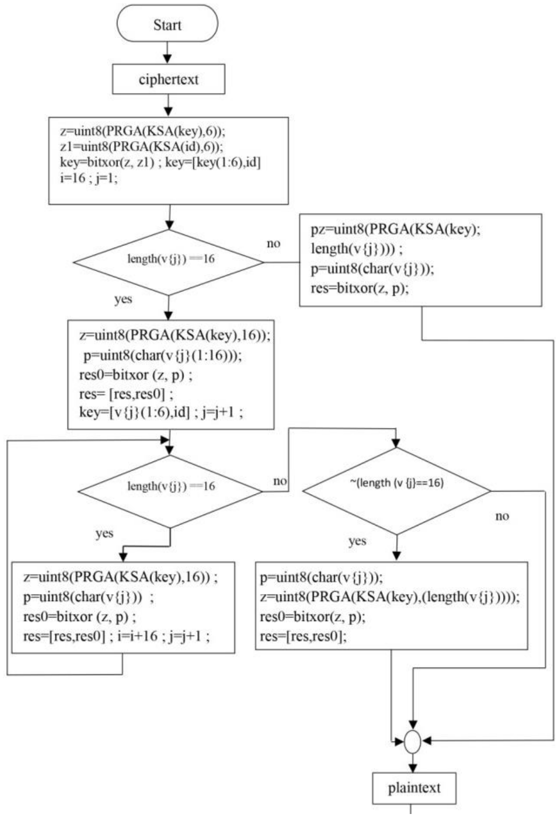


Figure1.(b). IRC4 Decrypt Algorithm

Evaluation Performance and Analyzation

The CRYPTX'98 mathematical analysis program[19] was used to examine the main streams. The frequency test, change point, binary derivative, sequence complexity and sub-block tests, and are all carried out. The suggested algorithm main streams examined own the characteristics of randomness bit streams according to CRYPTX 98. As for the other experiences of CRYPTX'98 aims to a collection of statistic characteristics that attackers can be used it, when it is used the same input plaintext and key directories, their effects compared to the encryption performance in the suggested development. If the algorithm creates a random stream, the p-values produced from a CRYPTX'98 test indicate the possibility of getting an outcome that differs from the test statistic. For the given metric, small p-values would allow non-randomness.

4.1 Frequency Test

A test for the bit stream is checked for an equal number of ones and zeros. In a long random sequence, the number of ones is roughly regularly distributed. That is, a sequence's number of ones and zeroes should be roughly equal. The frequency test estimates the sample stream's tail end probability for the number of ones[20]. The (Fig .5 and table 1) shows that the results of IRC4 the technique is superior to RC4.

(a) (b)

Fig.5 (a), (b) RC4 and IRC4 FREQUENCY TEST RESPECTIVELY.

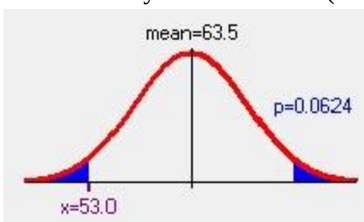
Table 1: The Results of Comparing of the Frequency Test

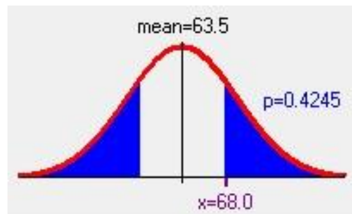
Test Parameter						
Algorithm	Total bits	Amount of the ones = (x)	(mean) = foreseeable ones	ratio of ones	(p-value)	is Satisfy
RC	128	67	64.0	0.5234	0.5959	yes
IRC	128	65	64.0	0.5078	0.8597	yes

4.2 The Binary Derivative

The second test is the binary derivative that is used in the measure of randomness of a string of binary created by a pseudorandom numbering generator used in the system of the cipher [21]. The results of the IRC4 technique are superior to RC4 which shows in Figs. (6 and 7), and Table s (2, and 3).

4.2.1 This sample represents 1st Binary Derivative test (D1) result for RC4 and IRC4 algorithms.





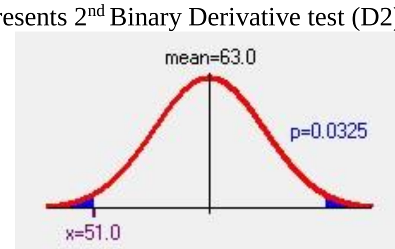
(a) (b)

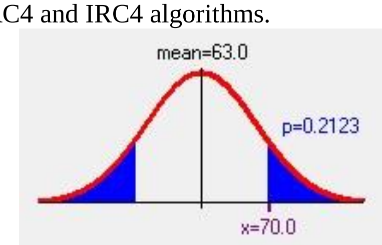
Fig.6 (a), (b) RC4 and IRC4 1st BINARY DERIVATIVE TEST RESPECTIVELY.

Table 2: The Results of Comparing of the First Binary Derivative Test (D1)

Test Parameter							
Algorithm	Total bits	Number of bits	Amount of the ones = (x)	(mean) = foreseeable ones	ratio of ones	(p-value)	Is Satisfy
RC	128	127	53	63.5	0.4173	0.0624	yes
IRC	128	127	68	63.5	0.5354	0.4245	yes

4.2.2 This sample represents 2nd Binary Derivative test (D2) result for RC4 and IRC4 algorithms.





(a) (b)

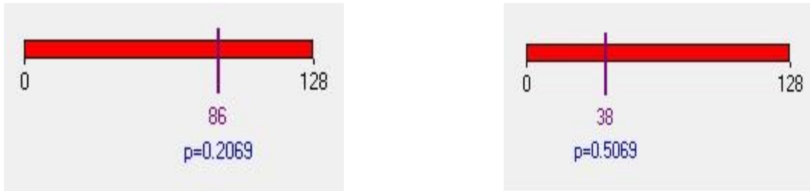
Fig.7 (a), (b) RC4 and IRC4 2nd BINARY DERIVATIVE TEST RESPECTIVLY.

Table 3: The Results of Comparing of the Second Binary Derivative Test (D2).

Test Parameter							
Algorithm	Total bits	Number of bits	Amount of the ones = (x)	(mean) = foreseeable ones	ratio of ones	(p-value)	Is Satisfy
RC	128	126	51	63	0.4048	0.1088	yes
IRC	128	126	70	63	0.5556	0.0325	yes

4.3 *Change Point Test*

This test looks for a significant change in the ratio of ones in the bit stream. In bit position in the bit sequence, the proportion of ones to that point is compared to the ratio of ones in the residual stream. The 'change point' is the area where the most change occurs. The test evaluates the significance of the ‘change[22].’ (Table 4 and Fig .8) shows that the results of IRC4 the technique is superior to RC4.



(a) (b)

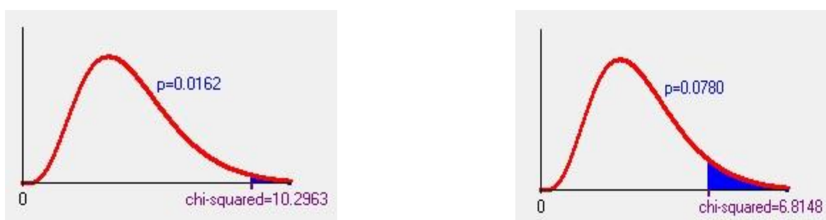
Fig.8 (a), (b) RC4 and IRC4 CHANGE POINT TEST RESPECTIVLY.

Table 4: The Results of Comparing of the Change Point Test

Test Parameter								
Algorit hm	Total bits	Amount of the ones = (x)	The Change point	Amount of the ones before	Ratio of ones before	Ratio of ones after	(p-value)	is Satisf y
RC	128	86	18	11	0.6111	0.409 1	0.137 3	yes
IRC	128	65	38	16	0.4211	0.544 4	0.506 9	yes

4.4 *Sub-block Test*

Tests of non-overlapping homogeneity sub-blocks of a given length, For sub-block sizes up to 16, the 'uniformity test' requires a sample of at least 5 * b * 2(b) bits, where b is the size of the sub-block. For sizes of sub-block bigger than 16, the 'repetition test' is applied. This test requires a specimen of b * 2(b/2+3) bits [23]. (Fig .9 and table 5) shows that the results of IRC4 the technique is superior to RC4.



(a) (b)

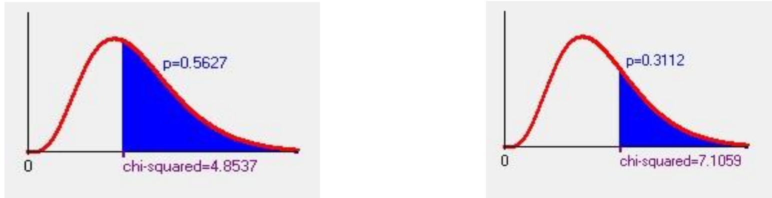
Fig.9 (a), (b) RC4 and IRC4 SUB-BLOCK TEST RESPECTIVELY.

Table 5: The Results of Comparing of the Sub-block Test.

Test Parameter						
Algorit hm	Total bits	Sub-block size	value of Chi-squared	Degrees freedom	(p-value)	is Satisf y
RC	128	2	10.2963	3	0.0162	yes
IRC	128	2	6.8148	3	0.0780	yes

4.5 *Runs Test*

The purpose of this test is to see if the numeral of runs of ones and zeros of matches what is anticipated from a random series. In particular, this test evaluates if the oscillation between such zeros and ones is too fast or too sluggish[24]. (Fig .10 and table 6) shows that the results of RC4 the technique is superior to IRC4. . But the values are close, as it is clear, and therefore doing not affect the strength of the encryption.



(a) (b)

Fig.10 (a), (b) RC4 and IRC4 RUNS TEST RESPECTIVELY.

Table 6: The Results of Comparing of the Runs Test.

Test Parameter								
Algorithm	Total bits	Number of runs	Number of blocks	The Number of gaps	The value of Chi-squared	Freedom Degree	(p-value)	is Satisfy
RC	128	54	27	27	4.8537	6	0.5627	yes
IRC	128	69	35	34	7.1059	6	0.3112	yes

4.6 Sequence Complexity Test

This check ensures that the stream has a sufficient number of new patterns. A stream is deemed non-random if the sequence complexity metric falls below a certain 'threshold' number, also, the value of An average of the complexity of the sequence for a stream in this length is counted[25]. (Fig .11 and table 7) shows that the results of RC4 the technique is superior to IRC4. But the values are close, as it is clear, and therefore doing not affect the strength of the encryption.

(a) (b)

Fig.11 (a), (b) RC4 and IRC4 SEQUENCE COMPLEXITY TEST RESPECTIVELY.

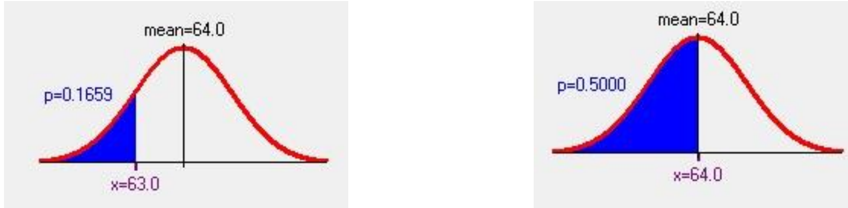
Table 7: The Results of Comparing of the Sequence Complexity Test

Test Parameter					
Algorithm	Total bits	complexity of Sequence	Value of Threshold	Value of Mean	is Satisfy
RC	128	19	18	21	yes
IRC	128	20	18	21	yes

4.7 Linear Complexity Test

The goal of this test is to see if the sequence is complicated enough to be deemed random or not. A larger (LFSR) 'longer linear feedback shift register' is used to delineated random series[26]. (table 8 and Fig .12) shows that the 'Linear Complexity Profile' test results of IRC4 the technique is superior to RC4, (Fig .13 and table 9) shows that the 'linear complexity-number of jumps' test results of IRC4 technique and RC4 is the same. And 'linear complexity-jump size' test results of IRC4 technique and RC4 is the same results.

4.7.1 This sample represents Linear Complexity Profile test result for RC4 and IRC4 algorithms.



(a) (b)

Fig.12 (a), (b) RC4 and IRC4 LINEAR COMPLEXITY TEST RESPECTIVELY.

Table 8: The Results of Comparing of the Linear Complexity Test

Test Parameter					
Algorit hm	Total bits	Linear Complex ity	Expected Linear Complex ity	(p-value)	is Satisfy
RC	128	63	64	0.1659	ye s
IRC	128	64	64	0.5000	ye s

4.7.2 This sample represents the Number of Jumps Linear Complexity test result for RC4 and IRC4 algorithms.

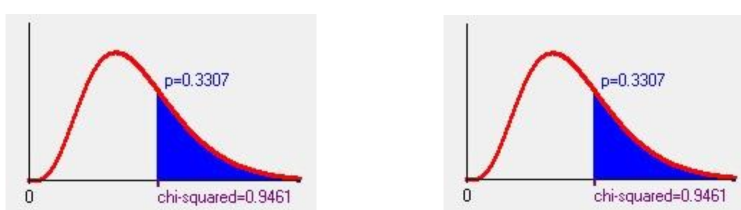
(a) (b)

Fig.13 (a), (b) RC4 and IRC4 NUMBER OF JUMPS OF LINEAR COMPLEXITY TEST RESPECTIVELY.

Table 9: The Results of Comparing of Number of Jumps of Linear Complexity Test

Test Parameter					
Algorit hm	Total bits	Number of Jumps	Expecte d Number of Jumps	(p-value)	is Satisfy
RC	128	34	32	0.6915	ye s
IRC	128	34	32	0.6915	ye s

4.7.3 This sample represents Linear Complexity Jumps Size test result for RC4 and IRC4 algorithms.



(a) (b)

Fig.14 (a), (b) RC4 and IRC4 LINEAR COMPLEXITY JUMPS SIZE TEST RESPECTIVELY.

Table 10: The Results of Comparing of the Linear Complexity Jump Size Test

Test Parameter					
Algorithm	Total bits	Number of Jumps	Expected Number of Jumps	(p-value)	is Satisfy
RC	128	0.9461	1	0.3307	yes
IRC	128	0.9461	1	0.3307	yes

5. **R**ESULTS **A**ND **D**ISCUSSION

To compare the work of both the original and the improved algorithms, the same original script was used for both RC4 and IRC4 algorithms and we found that the time taken to execute both the proposed and the original algorithms is almost the same, when comparing the ciphertext of the RC4 algorithm is less random and complex than the developed algorithm. Statistical randomness test was carried out using CRYPTX'98, after checking the values, the resulting P-value is matched, If the value is less than 0.01, the series is rejected and the series is considered non-random, so the obtained sequences are accepted and described as random and uniformly distributed Through the series, as shown in the above tables, we noticed that most of the tests of the proposed method gave better results than the original algorithm.

6. **C**ONCLUSIONS

RC4 stream encryption is a well-known encryption technology and one of the most popular encryption schemes for maintaining data security. Its implementation is simple and fast, but has flaws in keystream bytes, RC4 biases are now extracted for effective attacks. To provide a solution to bypass RC4 vulnerabilities by offering improved RC4 key generation. In this work, a new algorithm is proposed, Increases the randomness of the generated key by adding an ID number to the keystream before performing an XORed operation with the plaintext to generate a ciphertext as a slight modification has greatly enhanced the RC algorithm.

REFERENCES

1. Atikah, N., Ashila, M.R., Rachmawanto, E.H., and Sari, C.A.: ‘AES-RC4 Encryption Technique to Improve File Security’, in Editor (Ed.)^(Eds.): ‘Book AES-RC4 Encryption Technique to Improve File Security’ (IEEE, 2019, edn.), pp. 1-5
2. Mahmood, R.Z., and Fathil, A.F.: ‘High Speed Parallel RC4 Key Searching Brute Force Attack Based on FPGA’, in Editor (Ed.)^(Eds.): ‘Book High Speed Parallel RC4 Key Searching Brute Force Attack Based on FPGA’ (IEEE, 2019, edn.), pp. 129-134
3. Pu, C.-C., and Chung, W.-Y.C.: ‘Group key update method for improving RC4 stream cipher in wireless sensor networks’, in Editor (Ed.)^(Eds.): ‘Book Group key update method for improving RC4 stream cipher in wireless sensor networks’ (IEEE, 2007, edn.), pp. 1366-1371
4. Knudsen, L.R., Meier, W., Preneel, B., Rijmen, V., and Verdoolaege, S.: ‘Analysis methods for (alleged) RC4’, in Editor (Ed.)^(Eds.): ‘Book Analysis methods for (alleged) RC4’ (Springer, 1998, edn.), pp. 327-341
5. Jindal, P., and Singh, B.J.W.P.C.: ‘Optimization of the security-performance tradeoff in RC4 encryption algorithm’, 2017, 92, (3), pp. 1221-1250
6. Zoltak, B.: ‘VMPC one-way function and stream cipher’, in Editor (Ed.)^(Eds.): ‘Book VMPC one-way function and stream cipher’ (Springer, 2004, edn.), pp. 210-225
7. Fluhrer, S., Mantin, I., and Shamir, A.: ‘Weaknesses in the key scheduling algorithm of RC4’, in Editor (Ed.)^(Eds.): ‘Book Weaknesses in the key scheduling algorithm of RC4’ (Springer, 2001, edn.), pp. 1-24
8. Mironov, I.: ‘(Not so) random shuffles of RC4’, in Editor (Ed.)^(Eds.): ‘Book (Not so) random shuffles of RC4’ (Springer, 2002, edn.), pp. 304-319
9. Paul, S., and Preneel, B.: ‘A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher’, in Editor (Ed.)^(Eds.): ‘Book A New Weakness in the RC4

Keystream Generator and an Approach to Improve the Security of the Cipher' (Springer, 2004, edn.), pp. 245-259

10. Maximov, A.: 'Two linear distinguishing attacks on VMPC and RC4A and weakness of RC4 family of stream ciphers', in Editor (Ed.)^(Eds.): 'Book Two linear distinguishing attacks on VMPC and RC4A and weakness of RC4 family of stream ciphers' (Springer, 2005, edn.), pp. 342-358
11. Mousa, A., and Hamad, A.: 'Evaluation of the RC4 algorithm for data encryption', Int. J. Comput. Sci.
12. Pardeep, P.K.P., and Pateriya, P.: 'PC-RC4 algorithm: an enhancement over standard RC4 algorithm', International Journal of Computer Science and Network (IJCSN), 2012, 1, (3), pp. 1-6
13. Hammood, M.M., Yoshigoe, K., and Sagheer, A.M.: 'RC4-2S: RC4 stream cipher with two state tables': 'Information Technology Convergence' (Springer, 2013), pp. 13-20
14. Kareem, S.M., and Rahma, A.M.S.: 'A New Hybrid (MD5 and RC4) Cryptography Algorithm Using Multi-Logic States', in Editor (Ed.)^(Eds.): 'Book A New Hybrid (MD5 and RC4) Cryptography Algorithm Using Multi-Logic States' (IEEE, 2019, edn.), pp. 285-292
15. Saha, R., Geetha, G., Kumar, G., Kim, T.-H., and Buchanan, W.J.: 'MRC4: a modified rc4 algorithm using symmetric random function generator for improved cryptographic features', IEEE Access, 2019, 7, pp. 172045-172054
16. Wash, R.: 'Lecture notes on stream ciphers and RC4', Reserve University, 2001, pp. 1-19
17. Handa, D., and Kapoor, B.: 'PARC4: High performance implementation of RC4 cryptographic algorithm using parallelism', in Editor (Ed.)^(Eds.): 'Book PARC4: High performance implementation of RC4 cryptographic algorithm using parallelism' (IEEE, 2014, edn.), pp. 286-289
18. Paul, G., and Maitra, S.: 'RC4 stream cipher and its variants' (CRC press, 2011. 2011)
19. Dawson, E., Clark, A., Gustafson, H., and May, L.J.Q.U.o.T.: 'CRYPT-X'98,(Java Version) User Manual', 1999
20. Gustafson, H., Dawson, E., Nielsen, L., and Caelli, W.: 'A computer package for measuring the strength of encryption algorithms', Computers & Security, 1994, 13, (8), pp. 687-697
21. Davies, N., Dawson, E., Gustafson, H., and Pettitt, A.: 'Testing for randomness in stream ciphers using the binary derivative', Statistics and Computing, 1995, 5, (4), pp. 307-310
22. Pettitt, A.N.: 'A non-parametric approach to the change-point problem', Journal of the Royal Statistical Society: Series C (Applied Statistics), 1979, 28, (2), pp. 126-135
23. Dawson, E., Carter, G., and Gustafson, H.: 'Evaluation of the MUGI Pseudo-Random Number Generator July 31, 2002', 2002
24. Mood, A.M.: 'The distribution theory of runs', The Annals of Mathematical Statistics, 1940, 11, (4), pp. 367-392
25. Lempel, A., and Ziv, J.: 'On the complexity of finite sequences', IEEE Transactions on information theory, 1976, 22, (1), pp. 75-81
26. Massey, J.: 'Shift-register synthesis and BCH decoding', IEEE transactions on Information Theory, 1969, 15, (1), pp. 122-127