International Journal of Engineering & Technology, 7 (4) (2018) 3667-3673

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.20950
Research paper

Increasing security for RC4 algorithm by using PUF-based random number generators

Naji Mutar Sahib [1] *, Hazim Noman Abed [1], wasan Saad Ahmed [1]

[1] College of Science – University of Diyala
**Corresponding author E-mail: al.sahib@sciences.uodiyala.edu.iq*

Abstract

Rivest Cipher 4 (RC4) algorithm is the considerably stream cipher, and can be used of internet protocols. There is a quantity of weaknesses bytes even after the first 256 rounds (the size of the RC4 permutation) of the Pseudo random generation algorithm (PRGA). So far various modified RC4 research were presented but all of them have either ordinary privacy or accomplishment evaluation challenges. This paper introduces enhanced RC4 algorithm depending on Physical Unclonable Function (RC4 - PUF) which has solved both of these challenges. The principal development of the proposed RC4 - PUF methodology is that the changing of the s array reorganized to rely on the creation of a specific hardware random generator Method (HRGA) and the proposed methodology outcomes as follows:
Output = Plaintext XOR generated key XOR random values from HRGA.
The outcome of the tests demonstrates the refinement of the privacy of ciphers (average secrecy), randomness and accomplishment assessment (Encryption time and throughputs) over the variable key length and miscellaneous plaintext size of the proposed encryption RC4 – PUF methodology.

Keywords: *RC4; PUF; Average Secrecy; HRGA; Key Scheduling Algorithm.*

1. Introduction

RC4 is the considerably stream cipher, and can be used in different internet protocols for example wired equivalent privacy (WEP), Skype, Wireless protected access (WPA) and secure socket layer, Transport layer security (SSL/TLS) [1]. The significant operators in RC4 algorithm over such a substantial field of applications have been its speed and plainness; proficient implementation in both software and hardware were greatly simple to evolve.RC4 is very slight and rapid compared to other encryption algorithms. The Encryption Process of the RC4 algorithm was divided into two parts,
(1) assessment of the initialization of RC4 which focus on the initialization of key scheduling algorithm (KSA), and (2) evaluation of the output key streams generation which focus on the internal status and the round running process of PRGA. A great number of researches are done through RC4 algorithm to improving its security in order to make it protected and capable to face the attacks.
The necessity for random numbers in cryptographic operations is omnipresent. Initialization vectors block padding, nonces, challenges, and, surely, keys are some of the cryptographic objects where a string of unexpected bits is needed. Often the similar Random Number Generator (RNG) provides bits for all of the above with the using of a cryptographic system. a great number of the bits created by the RNG are sent in an obvious way and hence a passive attacker has simple chance to interpret the output of the RNG and can impact any deficiencies found there [2]. RNGs could be separated into two common categories:

- Pseudo Random Number Generators (PRNGs).
- True Random Number Generators (TRNGs).

RNGs can be used for cryptographic operations, on this account, can be well-considered a discriminating portion of the cryptographic system. A breakdown or weakness in the RNG may lead to an entire breakdown of the system. One of the principal methods used for designing a RNG is PUFs. A PUF is a procedure that produces a group of answers while stimulated by a group of challenges. It is a physical procedure because the challenge-response relation is specified by complicated characteristics of a physical material, for example the manufacturing variability of CMOS devices. Its unclonability is related to the circumstance that these characteristics could not be in a controlled way represented, making every equipment efficiently peerless. A PUF must be simple to assess which means the physical equipment must be able for assessing the procedure in a short time. It may also be difficult to define it. Hence from a restricted number of reasonable physical mensuration's or questions of selected Challenge-Response Pairs (CRP), a hacker who not anymore get the equipment, and who can only use a restricted number of means (money, time, raw material, etc...) could only pull an insignificant amount of information about the answer to a randomly selected challenge. PUFs shall be also exorbitantly difficult to clone, emulate, simulate, or prophesy.

2. Related work

Naji M. Sahib et al. in [3] we overbear on the weakness points in RC4 (Rivest Cipher 4) algorithm, there are an amount of mistakes in the key scheduling algorithm (KSA) of RC4. This strudy introduced enhanced RC4 key generation depending on multi-chaotic maps. The new pattern of KSA coined as enhanced KSA (IKSA), the permutation of the S array changed to rely on the random numbers generator depending on three disorganized maps ,And the suggested algorithm outputs as follows: Output = M

XOR produced key XOR Random value from IKSA (R3w) The enhanced RC4 with IKSA is proved for its secrecy, randomness and accomplishment over the variable key length and various plaintext size taking into account those of the original RC4.The results demonstrate that the enhanced RS4 with IKSA is superior than the original RC4 with KSA.

T.D.B Weerasinghr in [4] show the analysis of a simply modified RC4 algorithm, and tested an easy alteration of RC4 PRGA, where we can name it like this: Out Put=M XOR produced key XOR j.

S. M. Hameed et al. in [5] introduce a new pattern of KSA is proposed in a try to enhance the security of RC4 and dispose of the weakness connected to the initial permutation of the S array and the permutation operation of the S array.

Fluhrer, S. et al. in [6] we construed the KSA which concludes the initial status from a variable size key and define two important vulnerabilities of this operation. The first breakdown is in the existence of a great number of bits of the initial permutation (KSA output). The second breakdown is connected to key vulnerability, which can be applied when portion of the key exist to the KSA in observable to the Hacker.

Pardeep et al. in [7], told that the RC4 Method was revealed to the market and then specialists begin to construe the RC4 algorithm and discover a lot of breakdowns in both of two primary phases of the Method KSA and PRGA.

Mantin et al. in [8], discover the deficiency in the other phase, where the possibility of Zero output bytes is the primary deficiency of the RC4 Method.

Paul et al. in [9], produce the secret key with the use of the initial status table. They produced some equation on the basis of initial status table and they choose some of the bytes of secret key on the basis of presumption and the remaining secret key discovered with the use the equation.

A. Aboshosha et al. in [10] introduced the evolutionary algorithm depending on dynamic key generator in RC4 ciphering applied to CMS (RC4 – EA) This technique inclines to improve the RC4 encryption method with a great rank of a seed key coincidence.

3. Materials and methods

3.1. RC4 algorithm

Ron Rivest [11], one of the originators of RSA placed the RC4 algorithm in 1987. RC4 is an acronym for" Rivest Cipher 4", it is also famous as "Ron's Code 4". The algorithm is depending on using of a random permutation. The RC4 algorithm is easy and relatively slight to declare [12] [13].

Algorithm (1) (RC4 Stream Cipher Algorithm)
Input [plaintext] and [key]
Output [cipher text]
Step 1: /Initialize/
for i = 0 to 255
S[i] = i;
T[i] = K[i mod key];
Next i;
Step 2: / Perform IP of S /
Set j = 0;
For i = 0 to 255
j = (j +S[i] + T[i]) mod 256;
Swap (S[i], S[j]);
Step 3: /Stream Generation/
Set [i, j] = 0;
while (true)
i = (i + 1) mod 256;
j = (j + S[i]) mod 256;
Swap (S[i], S[j]);
t = (S[i] + S[j]) mod 256;
k = S[t];
Step 4: /The process/
Step 4.1: Encryption C = P K
Step 4.1: Decryption P = C K
Step 5: /End/

3.2. Physical unclonable function (PUF)

Physical Random Function (PUF) is a modern category of security, in which it has attracted a great deal of attentiveness. The new cryptographic diagram based on the use of one-way operations. These are procedures that are slight to work in the forward direction but impracticable to calculate in the converse direction without additional information [14]. PUFs are one-way procedures, which are slight to assess but challenging to reverse. These hardware one-way operations are cheap to fabricate, hard to mimeograph, afford no compact mathematical description [14]. PUFs are novelty primitives to conclude secrets from difficult hardware properties of ICs instead of saving the secrets in digital memory [15].

The secret key generation was first developed by use the PUFs in [16]. A ROPUF (Ring Oscillator Physical Unclonable Function) can be classified into an odd series of inverters. The RO frequency is created from the reversed signal that goes through the RO loop as demonstrated in Fig. 1 [11]. The presence of operation differences inside logic gates and wires leads to an uneven delay across the chip.

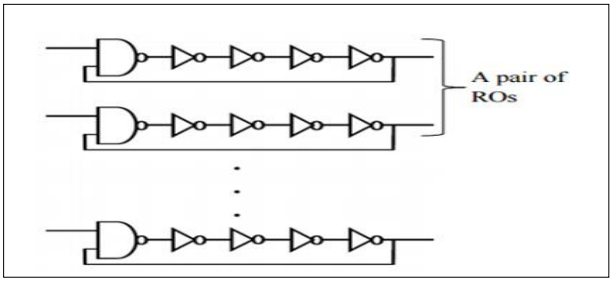


Fig. 1: Ring Oscillator Physical Unclonable Function (ROPUF).

A couple of ROs could generate two variable frequencies due to the existence of operation variations.

3.3. Design of ROPUF

a) Hardware Design

The hardware electronic circuit Fig 2. Includes a power supply and two integrated circuits, first one (IC2 7805) is (5v) regulator. whereas the second one (IC3 TC1262-3.3) is a programmable regulator.

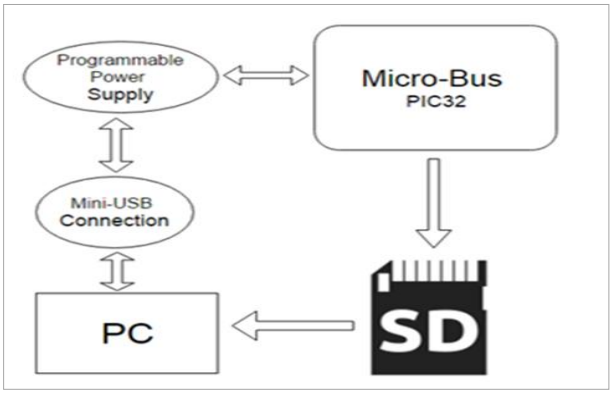


Fig. 2: Block Schema of the Hardware Design.

Fig.3. shows the programmable power supply (TC1262) which can be used to provide the variable voltage roughly (1.5-5 v) for the microcontroller and 3.3v to send Signal LED.

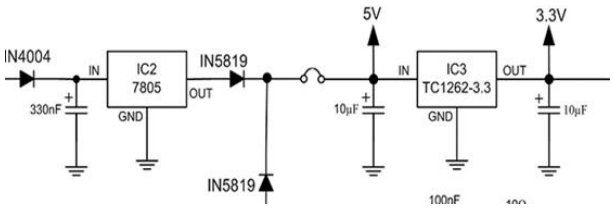


Fig. 3: Power Supply of the Hardware Design.

The Micro-Bus technology can be used as a variable technique to transmit data to SD and PC, in which it links (8) pins.
The algorithm of the written program inside microcontroller chip PIC32MX795F512L is demonstrated in algorithm (2). This Function is programmed in High-level Micro-C language.
Algorithm (2) Writing Random Numbers on PC and SD
Input: Libraries (SD, SPI, and UART); Counter (X).
Output: Writing random numbers on PC and SD.

1) First step: is calling the library of SD, SPI, and UART. Then a variable (x) is selected to be as a counter for counting the binary numbers. In this paper, the maximal value of (x) is 99393. surely, this value can be modified as needed.
2) Second step: is using the Real Time Clock (RTC) system in order to serially writing data on PC and in SD.
3) Third step: is inspecting the data that must be written on SD when these data are shown on PC. Otherwise, go to step (2).
4) Fourth step: is reiterating the condition of step (3) for 24 times which equal to a number of binary bits in the suggested system.
5) Fifth step: is ending write data on PC and SD card when x >99393.

3.4. Secrecy of ciphers

Secrecy of ciphers is calculated concerning the key equivocation (conditional entropy of key given cipher)

$$H\left(k/c\right) = \sum_{j=1}^{L} \sum_{i=1}^{n} q_i P_{ij} \log P_{ij} \quad (1)$$

Where

Qi = Pr (C = ci)

Pij = Pr (K=ki / C = ci)

4. Randomness test

Commonly, those sixteen tests are classified into two categories. The first category can be referred as non-parameterized tests and comprise:

- Frequency Test: defines the ratio of occurrence for ones and zeros are same in a produced order. It is recommended by NIST. The Frequency test can be carried out first, subsequently this test supplies the most elementary evidence for whether or not the existence of non-randomness in the generated orders such as non-conformity.
- Cumulative Sums-Forward Test: defines whether the maximum of the cumulative sums in an order possesses significant zeros or large ones at the first of the orders. This test can be well considered a random walk. The values of the random walk must be close to zero.
- Cumulative Sums- Reverse Test: defines whether the maximum of the cumulative sums in an order comprises many zeros or ones at the last of the orders. This test can be considered a random walk. The results of the random walk must be close to zero.
- Discrete Fourier Transform (DFT) Test: Defines the spectral frequency of the binary order that would be expected for a truly random order.
- Lempel-Ziv Compression Test: Defines whether an order is more compressed than a truly random order.
- Run Length Test: estimates the distribution of long runs of ones within an bit block to define if it approves with the hypothetical probabilities.
- Runs Test: Estimates whether the entire number of runs indicates that the frequency in the bit stream is quickly or weakly.
- Rank Test: the rank distribution is estimated for the corresponding random order due to the periodicity of repeated sub-orders.
- Random Excursions Variant Test: the distribution for the whole amount of bits across the random walks to finite state and determines whether it has over run the accurately random order.
- Random Excursions Test: estimates whether the distribution of the quantity of calls of a random walk to a certain state.
- The other category, which is parameterized test and contains:
- Serial Test: Calculates whether the amount of occurrences of m-bit overlapping arrays is approximately identical.
- Linear Complexity Test: estimates whether the induced order is enough complex which could be well-considered random or not.
- Overlapping Patterns of All One's Test: Defines the occurrence of m-bit periodic patterns.
- Non-periodic Patterns Test: Explains whether there are many appearances of non-periodic patterns.
- Approximate Entropy Test: defines whether that the outcome of an order more consistent by comparing with estimated values from a genuinely random order.
- Block Frequency Test: estimates the number of zeros and ones in every m-bit block are identical to have a random distribution.
- Universal Statistical Test: Calculates the compressibility by describing whether or not the binary order can be compressed without loss of information.

5. Proposed method (RC4-PUF)

The suggested algorithm is composed of three phases , while the main block scheme of the suggested Method is demonstrated in figure (4)

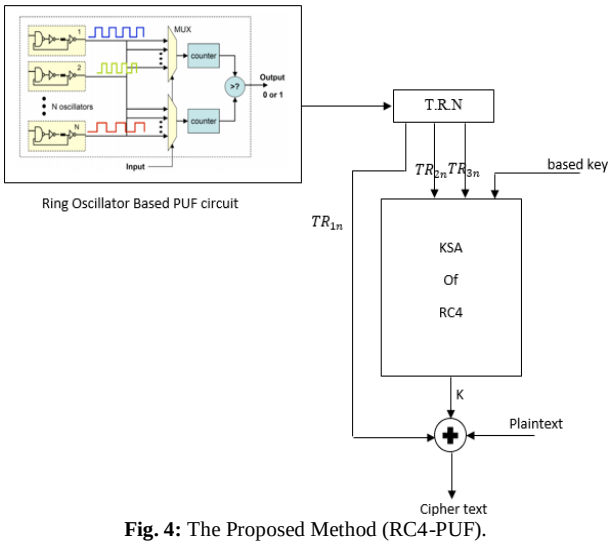


Fig. 4: The Proposed Method (RC4-PUF).

a) Ring Oscillator Based PUF circuit

Design of ROPUF relies on microcontroller chip (PIC32MX795F512L). this component is to produce true random numbers (TRN), We use each iteration (n=0 to 255 the number of

iterations) orders of 24 bits TR, n, TR_2n and TR_3n every has eight bits in order.

b) Key Scheduling for RC4-PUF

The modern version of KSA coined as development for KSA, the permutation of S modified to rely on true random numbers TR_2n and TR_3n:

For n=0 to 255

$j= (TR_3n + S[TR_2n]+T[TR_2n]) \bmod_{256}$

Swap (j, $S[TR_2n]$)

Next n

c) RC4-PUF encryption and Decryption

Encryption: Cipher = (plaintext ⊕ Generated key ⊕ TR_1n) $\bmod_{256}$

Decryption: plain = (Ciphertext ⊕ Generated key ⊕ TR_1n) $\bmod_{256}$

Algorithm 3 for RC4-PUF

Input [plaintext] and [key]

Output [cipher text]

Step 1: /Initialize /

for i = 0 to 255

S[i] = i;

T[i] = K[i mod key];

Next i;

Step 2: / Do IP of S /

For n=0 to 255

TR_2n = Location: generate from the TRN

TR_3n = Location: generate from the TRN

$j = (TR_3n + S[TR_2n] + T[TR_2n]) \bmod_{256}$

Swap (j, $S(TR_2n)$)

Next n;

Step 3: /Stream Generation/

Set [i, j] = 0;

While (true)

i = (i + 1) mod 256;

j = (j + S[i]) mod 256;

Swap (S[i], S[j]);

t = (S[i] + S[j]) mod 256;

k = S[t];

TR_1n: generate from the TRN

Step 4:/The process/

Encryption C = (M ⊕ K ⊕TR_{1n}) $\bmod_{256}$

$Decryption\ M = (C \oplus Generation\ key \oplus TR_{1n})$

Step 5:/End/

6. Results and discussion

diverse measures were used to show the capability of the suggested RC4-PUF Algorithm: the cipher secrecy, accomplishment of a given encryption Algorithm and the randomness of the National Institute of Standards and Technology (NIST) statistical test. The accomplishment of the suggested Algorithm RC4-PUF is analyzed under diverse key sizes and diverse plaintext sizes.

a) Average Secrecy of cipher

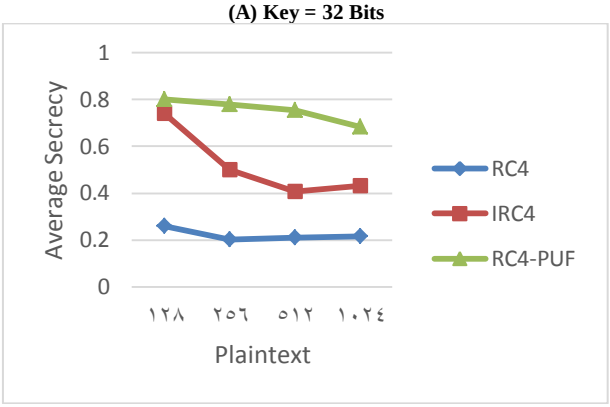
1) A variable plaintext size and fixed key length.

Table 1 and figures 5 (a), (b), (c) and (d) the suggested RC4-PUF Algorithm has superior average secrecy than the original RC4 algorithm with KSA and Enhanced RC4 algorithm with IKSA using a different plaintext sizes (128, 256, 512 and 1024 bits), and fixed key length for every stage (32, 64, 128 and 256 bits).

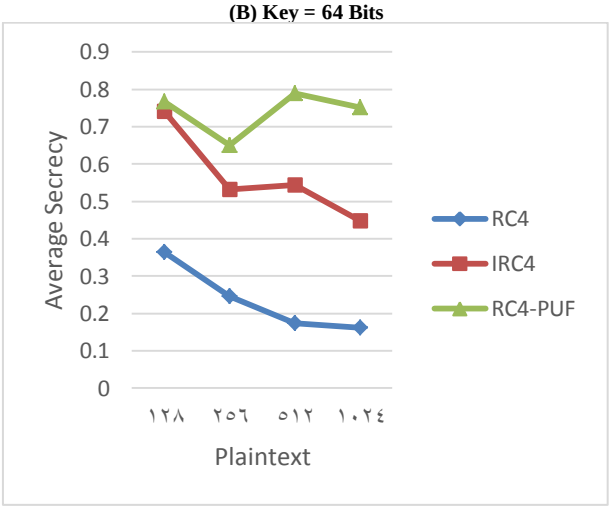
Table 1: Average Secrecy Value vs. Plaintext Size

Keys Length\Bits	Plaintext Size\Bits	Algorithms		
		Original RC4 With KSA	Improvement RC4 With IKSA	The pro-posed method RC4-PUF with TRNG
32	128	0.260459373	0.740856729	0.799913468
	256	0.203040633	0.498915456	0.779534601
	512	0.20944977	0.406738053	0.755014862
	1024	0.214365643	0.43235869	0.682594103
64	128	0.363815483	0.740856729	0.766035217
	256	0.245275562	0.531832803	0.650443915
	512	0.174139579	0.544481966	0.788801649
	1024	0.161067288	0.448314224	0.751880148
128	128	0.329087567	0.740856729	0.775550146
	256	0.249318629	0.531832803	0.783880148
	512	0.180433057	0.43880481	0.762227014
	1024	0.197202989	0.503334883	0.794476152
256	128	0.295187289	0.740856729	0.755501967
	256	0.247261403	0.74999756	0.747921083
	512	0.153576778	0.585268432	0.666604931
	1024	0.177807869	0.455159783	0.798641078

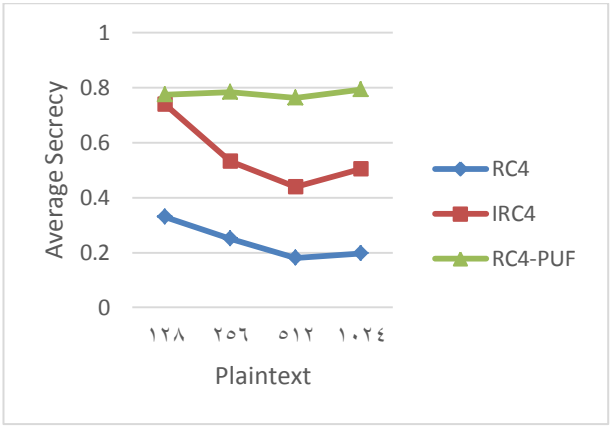
(A) Key = 32 Bits



(B) Key = 64 Bits



(C) Key = 128 Bits



(D) Key = 256 Bits

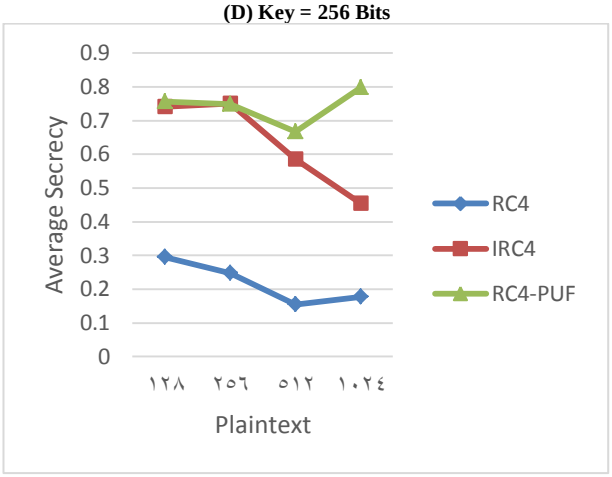


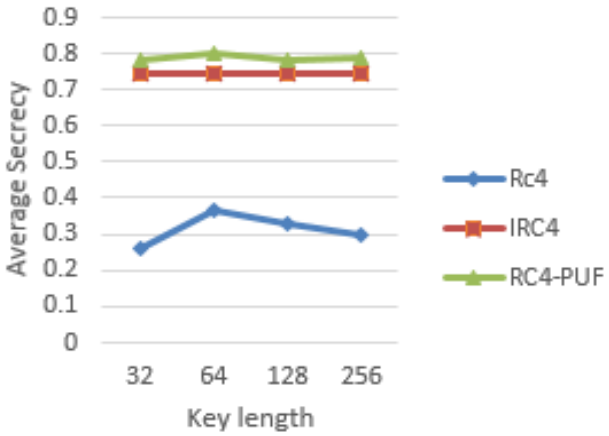
Fig. 5: Average Secrecy Value vs. Plaintext: (A) Key=32 Bits (B) Key=64 Bits (C) Key=128 Bits (D) Key=256 Bits.

2) Variable key length, fixed plaintext size

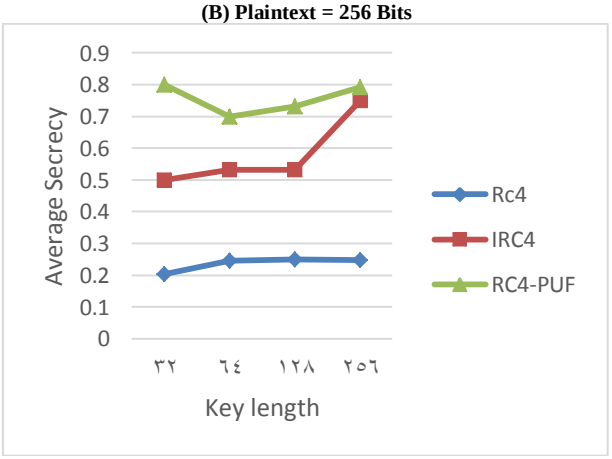
Table 2: Average Secrecy Value vs. Key length

Plaintext size/Bits	Keys Length/Bits	Algorithm		
		Rc4	IRC4	The pro-posed method RC4-PUF with TRNG
128	32	0.260459373	0.740856729	0.778256731
	64	0.363815483	0.740856729	0.799878230
	128	0.329087567	0.740856729	0.780865321
	256	0.295187289	0.740856729	0.789876592
256	32	0.203040633	0.498915456	0.799965021
	64	0.245275562	0.531832803	0.698941502
	128	0.249318629	0.531832803	0.730852613
	256	0.247261403	0.749997596	0.791803000
512	32	0.209449777	0.406738053	0.606732513
	64	0.174139579	0.544489666	0.653226723
	128	0.180433057	0.438884841	0.701104151
	256	0.153576778	0.585268432	0.695999423
1024	32	0.214365643	0.432358689	0.701200130
	64	0.161067288	0.448314224	0.648433015
	128	0.197202989	0.503348853	0.700047750
	256	0.177807869	0.455159783	0.687931365

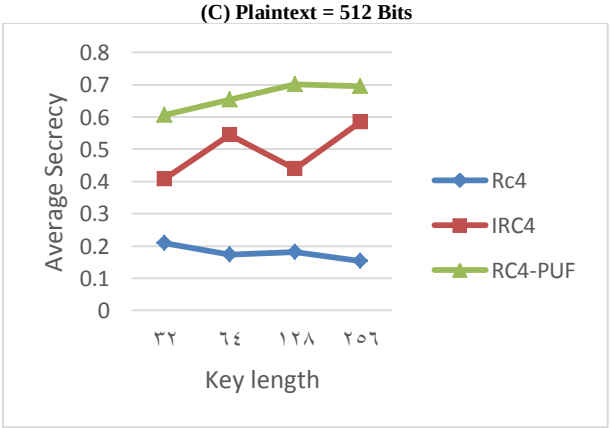
(A) Plaintext = 128 Bits



(B) Plaintext = 256 Bits



(C) Plaintext = 512 Bits



(D) Plaintext = 1024 Bits

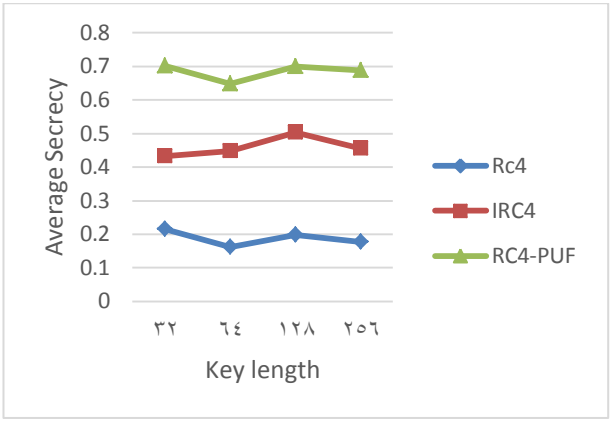


Fig. 6: Average Secrecy Value vs. Key Length: A) Plaintext = 128 Bits B) Plaintext = 256 Bits C) Plaintext =512 Bits D) Plaintext = 1024 Bits.

As demonstrated by the Table 2 and fig.6 (a), (b), (c) and (d), the suggested RC4-PUF Algorithm has superior average secrecy than the original RC4 algorithm with KSA and enhanced RC4 algorithm with IKSA, using a variable key length (32.64,128 and 256 bits), and fixed plaintext for every stage (128,256,512 and 1024 bits).

b) Encryption time

The time to generate a ciphertext from a plaintext using the suggested method RC4-PUF

Table 3: Encryption Time vs. Data Measurement with Key That Has Length 128 Bits

Data Size (KB)	Encryption Time RC4 (µs)	Encryption Time RC4-EA (µs)	Encryption Time RC4-PUF (µs)
20	1037.1208	905.9906	818.8915
40	1085.9966	940.0845	851.2348
60	1156.0917	978.9467	882.4369
80	1192.0929	982.0461	890.1278
100	1219.0342	988.9603	891.6669

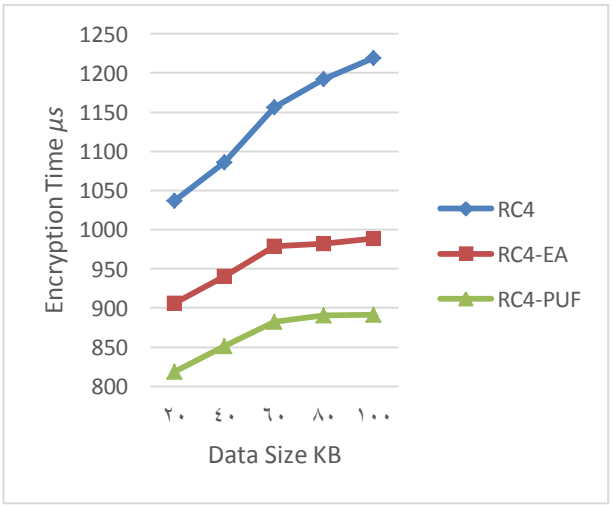


Fig. 7: Encryption Time of Diverse Data Size with the Use of Secret Key of Length 128 Bits.

Table 4: Encryption Time vs. Data Size with Key Length 256 Bits

Data Size (KB)	Encryption Time RC4 (µs)	Encryption Time RC4-EA (µs)	Encryption Time RC4-PUF (µs)
20	1105.0701	1036.9219	940.1653
40	1125.0973	1044.9886	954.7769
60	1189.9471	1047.1344	973.8752
80	1214.9811	1052.8564	980.8866
100	1260.0422	1065.9695	986.1634

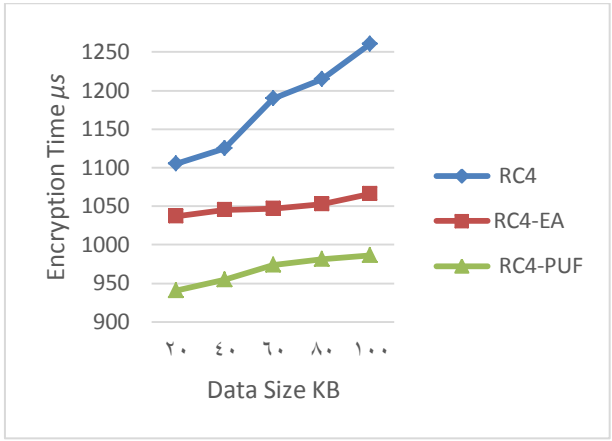


Fig. 8: Encryption Time of Several Data Measurement with Secret Key That Has Length 256 Bits.

As demonstrated by the tables and illustrations the ordinary encryption times are restrained on varied plaintext data size starting from 20 KB till 100 KB with key length that has128 bits and 256 bits.

An accomplishment comparison between the original RC4 encryption method, RC4-EA operation and the suggested RC4-PUF function is derived.

As the encryption time reduces, the accomplishment of the method raises.

c) Throughputs

Computed as the entire plaintext encrypted in KB categorized by the encryption time in microseconds.

As the throughputs raises, the accomplishment raises and the Energy usage reduces.

Table 5: Throughputs vs. Data Measurement with Key That Has Length 128 Bits

Data Size (KB)	Throughput RC4 (KB/S)	Throughput RC4-EA (KB/S)	Throughput RC4-PUF (KB/S)
20	19284.15	22075.26	29215.16
40	36832.52	42549.36	55330.80
60	51898.99	61290.36	73241.63
80	67108.86	81462.57	92116.53
100	82032.15	101116.29	117620.81

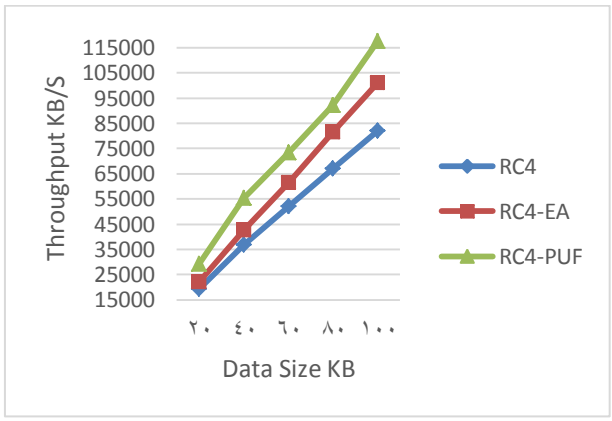


Fig. 9: Throughputs for the Encryption Pattern of Different Data Measurement with Secret Key That Has Length 128 Bits.

Table 6: Throughputs vs. Data Measurement with Key That Has Length 256 Bits

Data Size (KB)	Throughput RC4 (KB/S)	Throughput RC4-EA (KB/S)	Throughput RC4-PUF (KB/S)
20	18098.39	19287.85	21381.53
40	35552.48	38277.92	40716.72
60	50422.4	57299.23	65214.47
80	65844.64	75983.77	86823.34
100	79362.42	93811.31	103845.24

Fig. 10: Throughputs for the Encryption Pattern of Several Data Measurement with Secret Key That Has Length 128 Bits

As shown by the tables and figures. The throughputs of the proposed RC4-PUF encryption method compared with original RC4 and RC4-EA technique. The values came with key length 128 bits and 256 bits and variable plaintext data.

d) Randomness tests

The following multiple diverse trails are done with a view to doing the test the statistical attributes of the ciphertext generated from the suggested RC4-PUF technique with HRNG we used a large size binary order produced by every key. The movement was checked, and we afterwards computed the average of the p-values coming from these analyses. As demonstrated in the table 7.

Table 7: Randomness Test for RC4-PUF

Statistical test	p-value	Proportion/Threshold	Result
Frequency	0.578371	994/980	Pass
Block frequency	0.094185	992/980	Pass
Cumulative sums	0.838957	997/980	Pass
Runs	0.793138	995/980	Pass
Longest Run	0.268279	991/980	Pass
Rank	0.573641	994/980	Pass
FFT	0.139751	993/980	Pass
Non Over lapping Template	0.810725	994/980	Pass
Overlapping Template	0.923074	987/980	Pass
Universal	0.710936	994/980	Pass
Approximate Entropy	0.566284	993/980	Pass
Random Excursions	0.628072	626/615	Pass
Random Excursions Variant	0.499725	625/615	Pass
Serial	0.955507	990/980	Pass
Linear Complexity	0.900815	986/980	Pass

7. Conclusion

In this paper, the RC4-PUF technique with HRNG is suggested and accomplished the following:

1) Solve the vulnerability of the RC4 algorithm and another pattern that published.
2) Enhance the average secrecy of the algorithm and enhances its productivity by elevating encryption time and increasing efficiency.
3) Enhance the randomness of the ciphertext so that it fruitfully passed the NIST statistical tests for randomness.

The ordinary secrecy the accomplishment criteria secrecy and randomness were used to collate between the suggested technique with the original RC4 and other patterns from RC4 algorithm.

References

[1] A Book: Craincu, B., 2015 “On Invariance Weakness in the KSA Algorithm”. Procardia Technology, Elsevier, 19. pp: 850-857.

[2] P. Kohlbrenner, K. Gaj, (2004) “An embedded true random number generator for FPGAs”, 12th international symposium on Field programmable gate arrays, FPGA '04, Pages 71--78.

[3] Naji Mutar Sahib, Ali Hussein Fadel and Noora shihab Ahmed, (2018), "Improved Rivest Cipher 4 Algorithm Based on multi-chaotic Maps", Research journal of Applied Sciences, Engineering and Technology, 15(1), 1-6. https://doi.org/10.19026/rjaset.15.5285.

[4] Journal Articles: T.D.B Weerasinghe, 2012,” Analysis of a Modified RC4 Algorithm”. IJCA (0975-c xczs8887) Vol.51-No.22, p: 12-17.

[5] Journal Articles: Sarab M. Hameed, and Israa Nafea Mahmood, 2016,” A Modified Key Scheduling Algorithm of RC4”. Selected Areas in Cryptography.2259, Iraqi Journal of Science, (ISSN: 0067-2904), Vol. 57, No.1A, pp: 262-267.

[6] Journal Articles: Fluhrer.S, Mantin, I. and Shamir, 2001, A.” Weaknesses in the key scheduling algorithm of RC4”. Selected Areas cryptography, 2259, pp: 1-24.

[7] 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.

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

[9] 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.

[10] A. Aboshosha, K. A. ElDahshan, E. K. Elsayed and A. A. Elngar, 2015, "EA Based Dynamic Key Generation in RC4 Ciphering Applied to CMS", IJNS, Vol. 17, No. 4, pp, 405-412.

[11] A Book: Stallings W., 2011,” Cryptography and Network Security Principles and Practices, Fifth Edition”. Pearson Education, Inc. Pearson Prentice Hall, USA.

[12] A Book: Mao W., 2003, “Modern Cryptography: Theory and Practice”. Prentice Hall PTR.

[13] Journal Articles: Abdul M.S. Rahma, and Zainab M. Hussein, 2015,” Modified RC4 Dual key algorithm based on Irreducible Polynomial”. IJETTCS, Vol.4, Issue 2, p 79-85.

[14] 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.

[15] J.-W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, " A technique to build a secret key in integrated circuits with identification and authentication applications", Proceedings of the IEEE LSI Circuits Symposium, June 2004.

[16] Suh, G. Edward, and SrinivasDevadas, "Physical unclonable functions for device authentication and secret key generation.” Proceedings of the 44th annual Design Automation Conference. ACM, 2007.