ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 7 Issue 6, December 2017

Enhancement of RC4 Algorithm using PUF

* Ziyad Tariq Mustafa Al-Ta'i, * Dhahir Abdulhade Abdullah, Saja Talib Ahmed
*Department of Computer Science - College of Science - University of Diyala - Iraq

Abstract: ***A new direction of Integrated circuits (ICs) is known as Physical Unclonable Functions (PUFs). PUFs are considered as a unique class of circuits; because there are benefits of the inherent variations in industrializing process to create unpredictable and unique secret codes. PUFs can be used as random numbers generators because they attempt to extract randomness directly from complex physical systems. Therefore, this paper presents an enhancement of RC4 algorithm by using PUF keys. In this paper, a Ring Oscillator PUF (ROPUF) is designed and built using the chip (PIC32MX795F512L). The random PUF keys are used to increase the security of RC4 in ciphering the secret text messages. The created output by the proposed random number generator (RNG) is successfully passed most of NIST tests, also have high entropy with (0.9999999) and have no correlation with value of (-0.1514535).***

Keywords: **Physical Unclonable Function (PUF), Microcontroller chip (PIC32MX795F512L), Cryptography, RC4 algorithm.**

I. INTRODUCTION

Information protection nowadays is one of the majority vital factors of information processing because of big use of The Internet. Today, there are many important technologies are utilized for information safety.The cryptography technology is one of the most important in which it is used to encode-decode the data. Randomness concept is utilized widely in this field; also the power and strong argument of any encoding algorithm, build upon the encoding secret code attributes; its length and randomness. The protection of all applications of this field depends essentially on making unpredictable secret code [1].

In the field of security, Physical Unclonable Function (PUF) is a simple physical unit to build but approximately hard to make the second one, even known the accurate developed process that produced it. The strong argument of PUF is its distinguishing characteristic; because it is so hard to make a copy of the circuit as it is not possible to control the developed process variations [2]. Also one of the significant characteristics of PUF is the Low cost of production RNs. Therefore, this paper focuses on developing a new security system depending on random secret codes which are generated by PUF. These random secret codes are utilized for ciphering.

This paper is organized as follows. In sections (2) and (3) some important background is presented. The proposed system is explained in section (4). Experimental results and analyses are shown in section (5). Finally, the conclusions are remarked in section (6).

II. RC4

Cryptography is the art of communication protection. This art is scrambling a message so it cannot be clear; it transforms a clear text into encoded text, via a secret code [3]. Current system intersects the disciplines of mathematics, computer science, and electrical engineering [4].

One of the significant techniques of the stream cipher is the (RC4) algorithm. Many websites incorporate RC4 to protect services such as electronic banking and social media[4].

RC4 is a stream cipher extensively deployed in software applications, its attributes are simplicity, efficiency and fast outcome -feedback. It is unlike each of AES and DES; it requires less memory, usually utilized as the default cipher for (SSL) and the (TLS) connections [5]. Its procedure uses a variable-length secret code to initialize a 256-byte state vector S. At all times, S contains a permutation of all 8-bit numbers (0-255). For encoding and decoding, a byte k is produced from S by selecting one entry in a methodical style. As each value of k is produced, the entries in S are once again permuted. The following are the essential phases of this algorithm [6].

A. Initialization of S

To start, the corresponding S-entries are assigned to the ascending order of the values (0-255); like S[0] = 0, S[1] = 1,..., S[255] = 255. Then a temporary vector, T, is also formed. K is transported to T if the length of the secret code K is 256 bytes. Otherwise, for a secret code of length (key-length) bytes, the first elements of T are copied from K, and then K is duplicated many times as needed to fill the T. Next T is utilized to produce the initial permutation of S. This involves starting with S[0] and going through to S[255], because the only process on S is a swap, permutation is the only effect on S. It still contains all the values (0-255).

B. Stream Generation

Once the S vector is initialized, the input secret code is no longer utilized. Stream generation involves cycling through all the items of S[i], and for each S[i], swapping with another byte in S according to a scheme dictated by the current configuration of S. After S [255] is reached, the process continues, starting over again at S [0].

C. XOR Plain text with the value of secret code

For encoding, XOR process is done for the value k with the next byte of clear text. For decoding, XOR process is done for the value k with the next byte of coded text. The phases of the RC4 algorithm are explained in the algorithm (1) [7]:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 7 Issue 6, December 2017

Algorithm (1) RC4 Algorithm

Input: Clear text data, secret code
Outcome: Coded text data
1) Get the data to be encoded and the chosen secret code.
2) Create two string arrays.
3) Initiate one array with numbers (0-255).
4) Fill the other array with the chosen secret code.
5) Randomize the first array depending on the array of the secret code.
6) Randomize the first array within itself to generate the final secret code stream.
7) XOR the final secret code stream with the data to be encrypted to give ciphertext.

III. PHYSICAL UNCLONABLE FUNCTION (PUF)

PUF (also called Physical Random Function) is a new class of security, in which it has attracted a great deal of attention. The modern cryptographic scheme depends on the use of one-way functions. These are functions that are simple to work in the forward direction but infeasible to compute in the reverse direction without additional information [8].

PUFs are one-way functions, which are easy to evaluate but difficult to invert. These hardware one-way functions are inexpensive to fabricate, difficult to duplicate, grant no compact mathematical representation [8]. PUFs are innovative primitives to deduce secrets from complex hardware characteristics of ICs rather than storing the secrets in digital memory [9].

The use of PUFs for the secret key generation was first proposed in [10]. The proposed PUF in this paper based on internal Ring Oscillator (RO). A ROPUF is composed of an odd series of inverters. The RO frequency is generated from the inverted signal that travels through the RO loop as shown in Figure (1) [10]. The presence of process variations inside logic gates and wires causes an uneven delay across the chip.

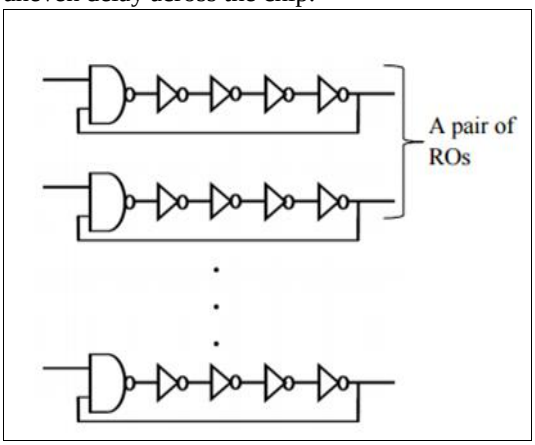

Fig (1) Ring Oscillator Physical Unclonable Function (ROPUF)

A pair of ROs could produce two different frequencies because of the presence of process variations.

IV. DESIGN OF THE PROPOSED SYSTEM

The proposed secure system contains HW and SW designs. The HW design involves ROPUF which is depended on microcontroller chip (PIC32MX795F512L). The SW design is about cipher text using RC4 technique.

A. General Block Diagram of the Proposed System

Figure (2) shows the general block diagram of the proposed system at the transmitter side. While the general block diagram of the proposed system at the receiver side is shown in figure (3).

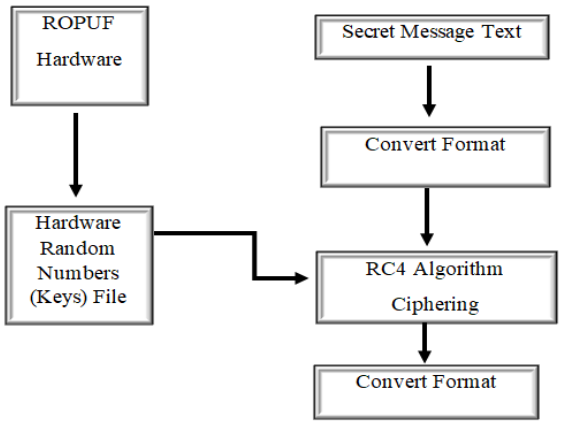

Fig (2) Block Diagram of the Proposed System at Transmitter Side

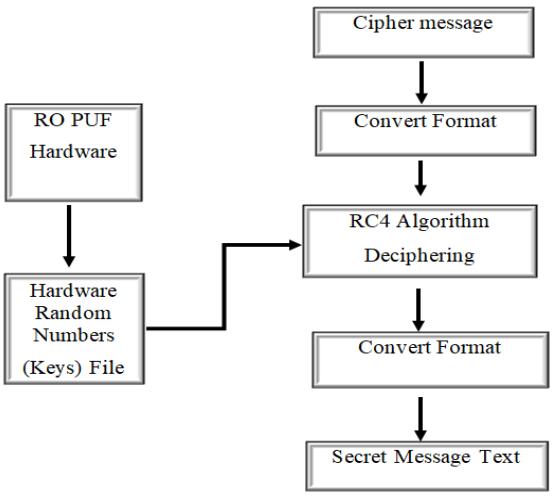

Fig (3) Block Diagram of the Proposed System at Receiver Side

IV. PROPOSED SYSTEM

A. Transmitter Side

1. HW Design: (ROPUF Design)

This is the first step at the transmitter side of the proposed system as in figure (2). The purpose of this step is to generate hardware random keys. The details of an electronic circuit for the hardware design is shown in figure (4). This electronic circuit contains a power supply and two integrated circuits, first one is (IC2 7805) which

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 7 Issue 6, December 2017

it is (5v) regulator. While the second one is (IC3 TC1262-3.3) which it is a programmable regulator.

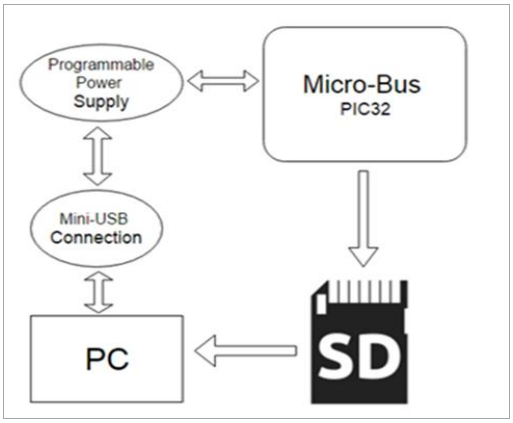

Fig (4) Block Diagram of the Proposed System

Figure (5) consists of the programmable power supply (TC1262) which is used to give the variable voltage approximately (1.5-5 v) for the microcontroller and 3.3v to signalization LED.

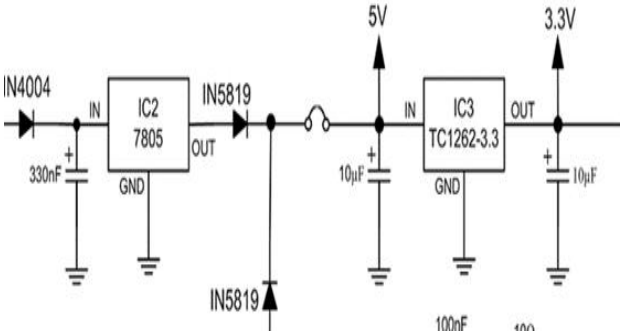

Fig (5) Power supply of the original circuit

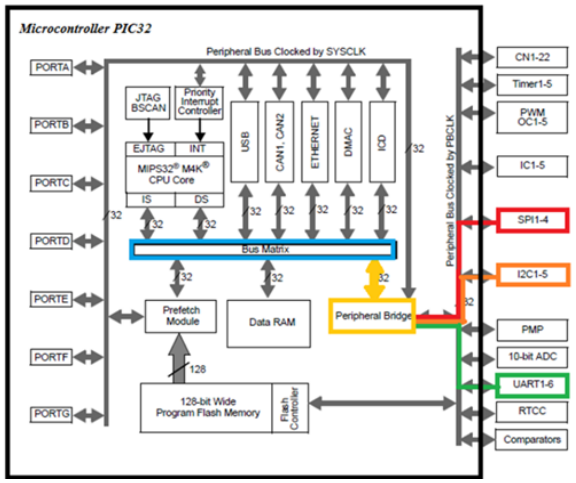

Fig (6) the internal block diagram of microcontroller and the Micro BUS [11]

The MICRO-BUS technique is used for variable way to send data to SD and PC which connects (8) pins. These pins have three variable types to send data: I2C, SPI, and UART as shown in figure (6).

-I2C: means the data are sent and received using (2) wires and it is called (inter to the integrated circuit).

-SPI: means the data are sent and received using (3) wires and it is called (Serial Peripherals interfacing).

-UART: means the data are sent and received asynchronously using (2) wires and it is called (Universal Asynchronous receiver and transmitter).

The flowchart of the written program inside microcontroller chip PIC32MX795F512L is shown in figure (7). This program is written in High-level Micro-C language. The steps of the proposed algorithm for microcontroller (PIC32MX795F512L) are shown in algorithm (2).

Algorithm (2) The Proposed Algorithm for Microcontroller (PIC32MX795F512L) Program

Input: Library functions SD, SPI, and UART.
Output: 99393HRN.
First step: Call the library functions (SD, SPI, and UART).
Second step: Choose a variable (x) as a counter, with maximum value equals to 99393 (this value can be changed).
Third step: Use the Real Time Clock (RTC) system in order to serially writing data on PC and in SD.
Fourth step: Check if the right data are written on SD when they are displayed on PC. Otherwise, go to the second step.
Fifth step: Repeat step (third and fourth) for 24 times which equal to a number of binary bits in RGB pixel in the proposed system.
Sixth step: Stop writes data on PC and SD card when counter (x)>99393.

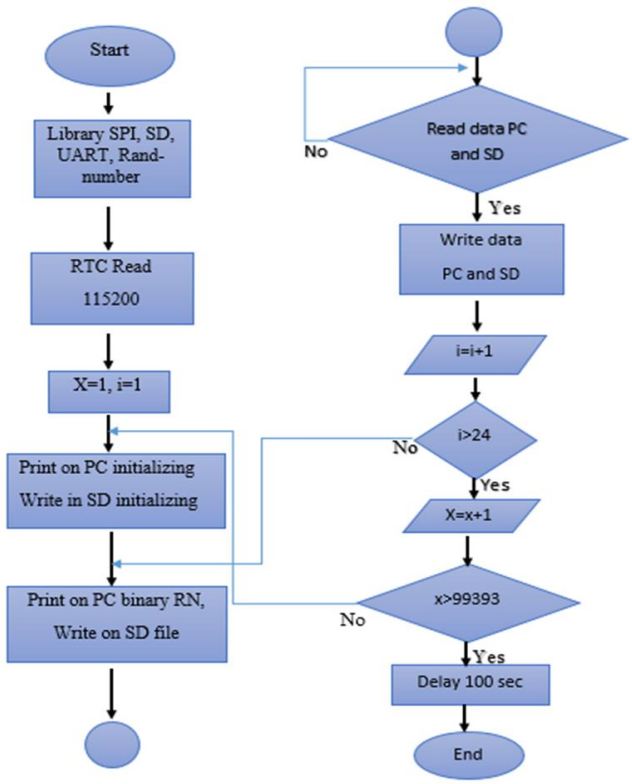

Fig (7) Flowchart of Microcontroller Program

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 7 Issue 6, December 2017

2. SW Design

i. Choose Secret Text Message

This is the first step in SW design and the second step at the transmitter side of the proposed system as in figure (2). In this step, a suitable secret text message is chosen.

ii. Convert Format of Secret Text Message

This is the second step in SW design and the third step at the transmitter side of the proposed system as in figure (2). In this step, the characters of the secret text message are converted to stream of numbers by using ASCII value for each character.

iii. Ciphering using RC4 Algorithm

This is the third step in SW design and the fourth step at the transmitter side of the proposed system as in figure (2). The RC4 algorithm is previously described in subsection (2.1). Figure (8) shows the RC4 Encryption depending on an algorithm (1).In RC4 cipher method with secret code, a random number generator is used to randomly generate secret code stream and XORing it with the secret message. In this approach, the random numbers are obtained from ROPUF in which previously created.

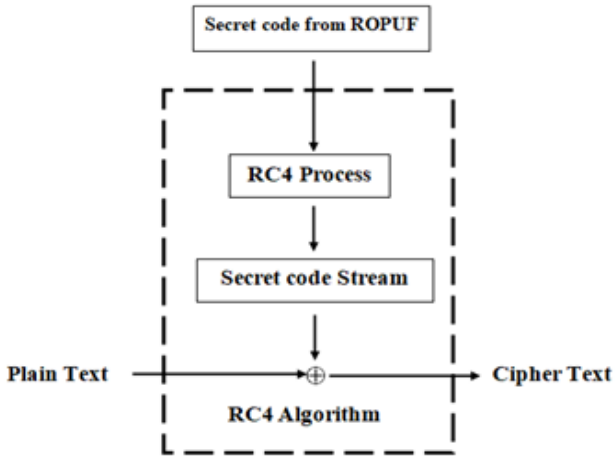

Fig (8) Block Diagram of RC4 Algorithm

The RC4 process consists of two parts: Key Scheduling Algorithm (KSA) and Pseudo-Random Generation Algorithm (PRGA). KSA part uses the secret code (hardware ROPUF key) to initialize and permutation of state vector S. This is done as shown in the pseudo code in figure (9).

```
For i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod(|K|)]);
j = 0;
for i = 0 to 255 do
    j = (j+S[i]+T[i])(mod 256)
swap (S[i], S[j])
```

Fig (9) Pseudo Code of KSA

Where:

[S], S is set equal to the values from 0 to 255

S[0] =0, S[1]=1,…, S[255]=255

[T], a temporary vector

[K], Array of bytes of secret code (ROPUF secret key)

|K| = Keylen, Length of (K)

Using T to produce initial permutation of S. The only operation on S is a swap;

S still contains a number from 0 to 255. After KSA, the input secret code and the temporary vector T will be no longer used.

PRGA part generates secret code stream k, one by one. This is done as shown in pseudo code at figure (10).

```
i = j = 0;
While (more_byte_to_encrypt)
i = (i + 1) (mod 256);
j = (j + S[i]) (mod 256);
swap(S[i], S[j]);
k = (S[i] + S[j]) (mod 256);
```

Fig (10) Pseudo Code of PRGA

The output byte is selected by looking up the values of S[i] and S[j], adding them together modulo 256, and then looking up the sum in S. This means, S[S[i] + S[j]] is used as a byte of the secret code stream, K. This is shown in figure (11).

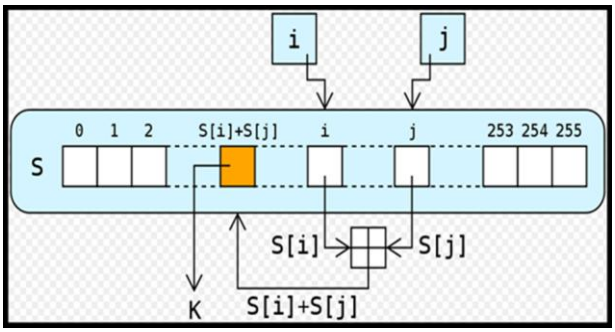

Fig (11) The method of getting the secret code stream (k)[12]

In the last stage of the RC4 algorithm, the secret message text is XORed with the k values bit by bit to generate the cipher text of the secret message text.

B. Receiver Side

The receiver side of the proposed system consists of software design only. This is because of the (ROPUF) hardware random numbers (secret codes) file (created in transmitter side subsection (4.2 .A.1) is handed down to the receiver side. Therefore, the software design of the receiver side of the proposed system includes the following subsections.

i. Convert Format

This is the first step at the receiver side of the proposed system as in figure (3). In this step, the ciphered message (which is received from transmitter side) are converted to ASCII values.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 7 Issue 6, December 2017

ii. Decipher the Ciphered Secret Message using RC4 Algorithm

This is the second step at the receiver side of the proposed system as in figure (3). The idea of deciphering is similar to the process of ciphering by the RC4 algorithm with HRN (secret code) which are handed out by the transmitter side to the receiver side. But in RC4 decipher algorithm, the XOR process is done between the HRN (secret code) and the ciphered secret message instead of the plaintext secret message. This process is clarified in figure (12).

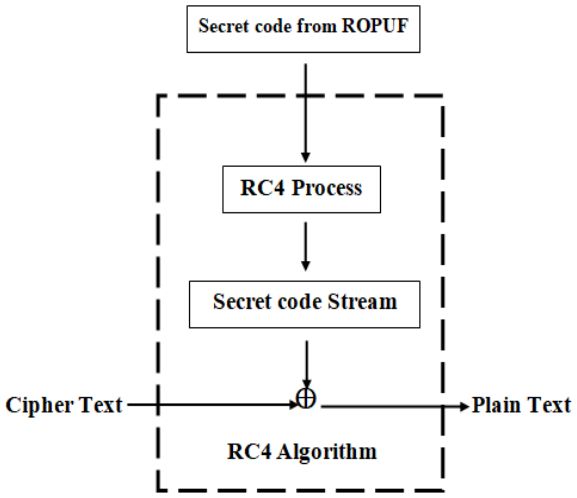

Fig (12) Block Diagram of RC4 Algorithm Deciphering

iii. Convert Format

This is the last step at the receiver side of the proposed system as in figure (3). In this step, the ASCII values of the deciphered secret message (which is obtained from subsection (3.2 B-ii) are converted to characters form. These characters are ordered sequentially, and then the secret text message is ready to be read.

IV. RESULT

This section is dedicated to present the results and tests that evaluate the performance of the proposed system. Since the proposed system consists of two phases (random key generation phase, and ciphering phase), therefore different measures are used to evaluate these phases.

1. Randomness Tests of Hardware Random Keys (PUF Keys)

A stream of random numbers is generated using hardware and software design in subsections (4.2 A-1,A-2) sequentially. An implementation of the microcontroller PIC32MX795F512L contains a PUF circuit which is accessible through software running on the chip.

The NIST statistical test suite is a set of algorithmic tests which attempt to identify sequences of binary numbers which do not behave in a truly random manner. To do this, these tests derive a p-value for every sequence of bits, which is, basically, the probability that the given sequence could have been generated by running a truly random number generator once.

Each test "passes" if the p-value is greater than some fixed confidence level, and "fail" if the p-value is less than the fixed confidence level. A thorough explanation of these testing algorithms can be found in a NIST Special Publication [13].

The statistical test suite also separates a given input string into many smaller substrings and performs the tests on each of these strings. Since it is possible for a truly random number generator to fail a given test sometimes, it is useful to discuss success in terms of the proportion of successful tests.

Table (1) shows the results of the tests, and the lowest proportion of successful tests found after running the test suite many times on the output of the proposed system. The proportion of successful tests is high enough to consider this a reasonably good random number generator.

Table (1) Results of NIST Statistical Test Suite on the Output of the Proposed System

Test name	Result	Lowest success ratio
Frequency Test	Success	99%
Approximate Entropy test	Success	100%
Block Frequency Test	Success	100%
Serial Test	Success	98%
Cumulative Sums Test (Forward)	Success	100%
Runs Test	Success	98%
Longest Run of One's test	Success	100%
FFT Test	Success	100%
Rank Test	Success	100%
Nonperiodic Templates Test	Success	79%
Overlapping Template of all One's Test	Success	100%
Lempel-Ziv compression Test	Success	100%

2. Cryptographic Results

Cryptographic Test Sample

As an example, a 32-bit (4 characters) which represents the word (Iraq), is selected as a plain text secret message.

Ciphering with RC4 Algorithm

1. The first step in cryptographic results is converting secret plain text message (Iraq) into ASCII values as shown in the table (2).

Table (2) the ASCII Values of Secret Plain Text Message

Plain Text	ASCII
Iraq	73, 114, 97, 113

Table (3) shows the (256) byte keys (hardware ROPUF random numbers) that are entered into RC4 process.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 7 Issue 6, December 2017

Table (3) 256- Byte Keys

256 Byte Keys
56,55,13,96,43,182,29,30,52,69,124,49,74,189,255,117,122,202,143,27,186,198,118,38,27,219,194,125,205,159,67,218,69,10,247,180,138,178,178,112,152,114,223,228,239,148,59,133,96,250,254,61,98,167,38,57,244,244,133,162,231,106,170,50,144,55,41,53,235,224,172,139,40,93,86,118,121,12,99,253,242,207,69,129,173,195,98,192,15,46,74,153,128,187,128,149,192,230,2,97,141,41,37,112,191,180,26,40,192,131,154,239,168,188,59,113,32,173,133,199,96,187,207,174,60,35,83,18,90,66,10,212,135,73,159,56,190,52,18,161,144,184,141,50,208,108,35,30,104,139,250,59,146,18,93,166,87,48,56,73,197,131,20,244,237,234,186,21,236,231,137,231,29,179,69,74,8,246,173,146,25,84,193,43,90,42,30,17,224,56,199,154,20,231,229,148,54,93,205,94,229,208,230,150,57,246,137,97,57,152,151,188,116,85,15,65,66,178,56,133,154,22,4,117,65,109,234,14,198,54,35,59,232,126,212,106,194,157,130,163,38,142,115,245,105,140,28,1,66,175,234,161,228,65,123,37,220,64,

Table (4) shows first permutation of keys in order to obtain the (S-Box):

Table (4) First Permuted 256-Byte Keys

Key Stream after first permutation (S-Box)
56, 209, 183, 1, 60, 204, 27, 14, 107, 166, 36, 96, 99, 128, 141, 4, 10, 116, 95, 61, 17, 122, 125, 62, 0, 5, 213, 188, 163, 229, 57, 63, 160, 202, 142, 249, 105, 77, 68, 153, 234, 3, 115, 250, 220, 40, 31, 243, 113, 198, 146, 78, 76, 192, 224, 137, 43, 173, 92, 73, 225, 67, 51, 205, 29, 55, 245, 114, 11, 65, 79, 207, 98, 255, 201, 138, 121, 26, 251, 159, 129, 161, 13, 195, 246, 18, 45, 134, 21, 22, 111, 211, 175, 179, 23, 39, 157, 52, 8, 90, 253, 139, 104, 9, 212, 110, 66, 80, 7, 244, 215, 177, 226, 74, 194, 193, 240, 20, 124, 151, 216, 50, 69, 156, 2, 102, 170, 187, 120, 145, 238, 87, 100, 162, 176, 228, 149, 140, 86, 136, 126, 106, 218, 152, 174, 247, 172, 199, 49, 44, 206, 239, 208, 88, 84, 35, 178, 72, 167, 135, 28, 33, 101, 54, 59, 169, 154, 197, 254, 190, 34, 242, 222, 70, 155, 108, 200, 182, 184, 186, 12, 16, 164, 37, 248, 219, 132, 85, 64, 75, 217, 89, 131, 109, 221, 189, 158, 237, 143, 48, 24, 53, 71, 235, 185, 223, 94, 227, 81, 112, 117, 196, 97, 168, 93, 252, 171, 25, 232, 38 ,123, 144, 30, 6, 130, 58, 32, 19, 103, 119, 231, 180, 133, 210, 181, 47, 148, 15, 41, 91, 214, 83, 236, 118, 203, 233, 46, 127, 150, 191, 82, 165, 241, 147, 42, 230

Table (5) shows the second permutation of the key stream for ciphering (4) characters.

Table (5) Four Second Permuted Key Stream

S-Box	Keystream
56	90
209	151
183	92
1	215

Table (6) shows the results of XORing the ASCII values of the secret text message in the table (2) with the key stream in the table (5). This result represents the RC4 ciphered secret message.

Table (6) XORed Process

Character	ASCII	Keystream	XOR ASCII with Keystream
I	73	90	19
R	114	151	229
A	97	92	61
Q	113	215	166

VI. CONCLUSION

In this research, a proposed random numbers generator is designed and implemented using ROPUF inside microcontroller chip PIC32MX795F512L to output Hardware Random Numbers (HRNs) which are used to enhance RC4 algorithm. The results in section (5) indicate that the security of RC4 algorithm is enhanced because of using :

1- Hardware random keys.

2- The used random keys did not need seed value.

3- The used hardware keys are completely random as shown in table (1).

4- The cost of obtaining HRNs is low.

VII.FUTURE WORK

1. Design other types of PUF such as :(Arbiter, design RO on FPGA, and others) for comparison purposes.
2. Re-design the proposed system to generate random numbers based on temperature, use these random numbers as a key in RC4 algorithm.

REFERENCES

[1] Abdulzahra, Hayfaa, R. O. B. I. A. H. Ahmad, and Norliza Mohd Noor. "Combining cryptography and steganography for data hiding in images." Applied Computational Science (2014): 128-135.

[2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. "Controlled physical random functions". In Proceedings of

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 7 Issue 6, December 2017

18th Annual Computer Security Applications Conference, December 2002.

[3] A. Rukhin, et al. "A statistical test suite for random and pseudorandom number generators for cryptographic applications". Booz-Allen and Hamilton Inc McLean VA, 2001.

[4] C. W. O'donnell, G. E. Suh, and S. Devadas. "PUF-based random number generation." In MIT CSAIL CSG Technical Memo 481 (2004).

[5] T. Dierks and C. Allen, "The TLS Protocol", Version 1.0, Internet Engineering Task Force, January 1999.

[6] W. Stallings. "The RC4 Stream Encryption Algorithm." Cryptography and network security (2005).

[7] A. Mousa, and A. Hamad. "Evaluation of the RC4 algorithm for data encryption." IJCSA 3.2 (2006): 44-56.

[8] Pappu Srinivasa Ravikanth, "Physical one-way functions", Ph.D. thesis, Massachusetts Institute of Technology, March 2001.

[9] 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", In Proceedings of the IEEE LSI Circuits Symposium, June 2004.

[10] 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.

[11] https://en.wikipedia.org/wiki/Harvard_architecture.

[12] https://en.wikipedia.org/wiki/RC4.

[13] Rukhin, Andrew et. al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications", Booz-Allen and Hamilton Inc McLean Va, 2001.