جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة ديالى / كلية العلوم / قسم علوم الحاسبات

"إنشاء نظام أمني باستخدام ROPUF بالاعتماد على رقاقة المتحكم الدقيق"

رسالة مقدمة

الى/ جامعة ديالى /كلية العلوم / قسم علوم الحاسبات كجزء من متطلبات نيل شهادة الماجستير في علوم الحاسبات.

من قبل

سجى طالب أحمد

بكالوريوس علوم حاسبات / الجامعة المستنصرية للعام ٢٠٠٢-٢٠٠٣

بأشراف

أ. د . ظاهر عبد الهادي عبد الله

أ. د. زياد طارق الطائي

١٤٣٩/ ديالى 2018 /

الملـــــــخص

مع التطورات الأخيرة في تكنولوجيا الحوسبة وتدخلها في حياتنا اليومية، ازدادت الحاجة إلى الاتصال السري والشخصي. وتتطلب السرية في الاتصالات الرقمية تبادل المعلومات السرية بين كيانين باستخدام الاتصالات الحاسوبية.

تستخدم أساليب وتقنيات مختلفة لتوفير السرية في الاتصالات. في هذه الأطروحة، تم تصميم نظام أمني متعدد المستويات مع مفاتيح مادية عشوائية. المستوى الأول هو تشفير رسالة نصية سرية باستخدام خوارزمية RC4. المستوى الثاني هو إخفاء الرسالة المشفرة في صورة باستخدام خوارزمية LSB. ويتم تعزيز كلا المستويين باستخدام الأرقام المادية العشوائية (مفاتيح) (HRN_s).

يتم إنشاء هذه المفاتيح المادية العشوائية من قبل النظام المادي الذي هو حلقة المذبذب للوظيفة المادية الغير قابلة للتغيير(ROPUF).إن الـ(PUF) هي فئة فريدة من الدوائر التي تستفيد من الاختلافات الكامنة في عملية التصنيع لإنشاء أرقام عشوائية فريدة من نوعها وغير قابلة للتغير (مفاتيح). في هذا العمل، تم تصميم وتنفيذ (ROPUF) باستخدام رقاقة متحكم (PIC32MX795F512L) من أجل توليد ملف بالارقام العشوائية(HRN_s) (المفاتيح). يتم استخدام هذا الملف كتيار من المفاتيح لـ(RC4) لتشفير رسالة نصية سرية، ومن ثم تستخدم المفاتيح العشوائية لإخفاء رسالة سرية مشفرة باستخدام (LSB) داخل أغطية الصور.

وقد تم اختبار (HRN_s) من قبل حزمة اختبارات(NIST). وقد اجتازت (HRN_s) معظم اختبارات (NIST) مع معدلات نجاح عالية. وقد فحصت هذه المفاتيح أيضا وأثبت أنها غير متوقعة مع قيمة عالية من الانتروبـــــــي (0.9999) للبت الواحد وفريــــــدة من نوعها مع عدم وجود ارتبـــــــاط وبقيمة (0.1514-) .

وقد حل النظام مشاكل الـ security والـ stability الموجودة في مثل هذه التصاميم، وقد حقق أمنية عالية من ناحية الـ S/W وتعقيداً أقل من ناحية الـ H/W. كما وقد استطاع النظام المقترح من ايجاد حلول لنقاط الضعف الموجودة في خوارزمية الـ RC4 Standard.

وتستخدم أربعة صور كأغطية للاخفاء بخوارزمية (LSB) مع سبع رسائل سرية مختلفة الاحجام. وتراوحت قيمة الـ (PSNR) للإخفاء حوالي من (75.19813053) إلى (95.32650278)، أي ان الاخفاء ذو جودة عالية. وكذلك، الرسائل السرية المستخرجة صحيحة تماما.

1.1 Introduction

Information protection, nowadays, is one of the majority vital factors of Information processing and correspondence; the reason goes back to a big increase of the (WWW) and the copyright rules. Two important technologies cryptography and steganography are used for Information safety of digital reality for today. The two technologies have their boundaries and this is the reasons for many types of research that combine these two techniques for increasing the digital information protection [1].

These two technologies are known very well and depended on techniques that encryption\decryption or hiding the data. The first technology is concerned with encryption and decryption the data depending on an undisclosed key. The second technology is depended on hides and covers the information to protect it. However, the systems using the first technology can be generally classified into two types; the first used one Key and the second used two keys, one known and the second is secret [2].

A randomness concept is used widely in this field; also the power and strong argument of any encryption algorithm, build upon the encryption Key attributes; its length and randomness. The protection of all application of this field depends essentially on making unpredictable Key [2].

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 [3].

Also one of the significant characteristics of PUF is the Low cost of production RN_s. Therefore, this thesis focuses on developing a new security system depending on random Keys which are generated by PUF. These random Keys are used for encryption and hiding.

1.2 Background

This part is concerned with selected topics, which are considered as the background for this thesis.

1.2.1 Cryptography

It is the art of communication protection. The aim of using its techniques is to protect against illegal parties and preventing them from doing any change.

This art is scrambling a message so it cannot be clear; it transforms a clear text into encrypted text, via a Key [4]. Current system intersects the disciplines of math, computer science, and electrical engineering [5].

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 [5].

1.2.2 Rivest Cipher 4 (RC4)

This algorithm is a stream cipher extensively deployed in SW applications, its attributes are simplicity, efficiency and fast output - feedback cipher.

It is unlike each of AES and DES; fast SW, requires less memory, usually used for (SSL) and the (TLS) connections as the default cipher [6].

In 1987, Ron Rivest designed this algorithm for (RSA) Security. It is using in (WEP) protocol, and in (WPA) protocol that is part of the (IEEE 802.11 wireless LAN standard). It has many characters, as shown below:

1- It is using a variable-length Key.

2- It is depending on the random concept.

3- For each byte produced, (8-16) machine processes are required.

4- It can be implemented so fast in SW.

5- It has been preserved as a commercial secret by RSA Security.

6- It was published on the Internet in (1994) [7].

The steps of the algorithm are as follows:

1- It is using a variable-length Key to reset a state vector (S) which consist of (256 bytes).

2- S-state contains a permutation of all (8-bits) numbers (0-255).

3- A byte k is produced from (state-S) through choosing (one) entry in a systematic manner, which is using for encryption and decryption,

4- For each produced k, the entries in S are once again permuted.

The following are the essential phases of this algorithm [8].

1. Initialization of S:

There are many points for starting, these are:

1- The corresponding S-items are assigned to the ascending order of the values (0-255); like S[0] = 0; S[1] = 1;...; S[255] = 255.

2- A temporary vector (T) is also created.

3- If the length of the Key K is (256 bytes), then (K) is transported to (T). Otherwise, for a Key of length (key-length) bytes, the first items of T are copied from (K). (K) is duplicated many times as needed to fill the T.

4- The following T is used to produce the initial permutation of S. This requires starting with S[0] and going to S[255], because the only process on S is a swap, permutation is the only effective process on S. It still contains all the values (0-255).

2. **Generating Stream :**

The input Key is used once the (S-vector) is initialized. Generating a stream involves cycling through all items S[i]. According to a plan dictated by the current structure of S, each item in S[i], will be swapping with another byte in the same S. The process continues, starting over at S[0], when S [255] is reached.

3. **XOR Plain text with the value of Key:**

For encryption, the value of (k) is XORing with the next byte of plain text. For decryption, the value of (k) is XORing with the next byte of the cipher text.

The Phases of The RC4 Algorithm Are Explained In The Algorithm (1.1) [9]:

Algorithm (1.1) RC4 Algorithm

```
Input: Plain text , Keys.
Output: cipher text.
1. Obtain the data to be encrypted and the chosen Keys.
2. Produce two string arrays.
3. for i= 0 to 255
    {   Start one array with numbers (0-255).
        Fill in the other array with the chosen Keys.    }
5. The first array will be randomized and depended on the array of Keys.
6. To produce the final Key stream, the first array will be random with itself
7. The final Key stream will be XORing with the data to be encrypted to obtain ciphertext.
```

Several attributes of the (RC4) algorithm can be briefed as:

1. Symmetric stream key.
2. The length of the Key is variable.
3. Simple and Very fast in SW.
4. Used for secured communications as in the encryption of traffic to and from secure websites using the SSL protocol [9].

1.2.3 RC4 Weakness Point

The initial state from a variable size key, describes tow significant weakness. The first weakness is in the existence of a large number of bits of the initial permutation(KSA). The second weakness is related to key vulnerability. Which applied when part of the key presented to the KSA in exposed to the attacker.

1.2.4 Steganography

It means “covering writing” in Greek. It is the art of hiding data by embedding message within another file which called cover; this group is known as stego-file, in other words[10].

Many algorithms have been proposed recently to hide information into images and preserve their quality. In this Master thesis we focus on image steganography algorithms. An image consists of light luminance or pixels represented as an array of values at different points. A pixel consists of one byte or more. For example in 8-bit images each pixel consists of 1 byte (i.e., 8 bits). While each pixel in a 24-bit image is represented as three bytes representing the Red, Green and Blue (RGB) colors. Any variation of the bits can lead to a different color.

Several algorithms have been produced in this field; one of them is Least Significant Bit (LSB).

1.2.5 Least Significant Bit (LSB)

Least Significant Bit (LSB) insertion method is a common and easy technique which is using to insert data into an image file. In this method the LSB of a byte is replaced with an M_s bits. This technique works good for image steganography[10]. To the human eye the stego image will look identical to the carrier image. For hiding information inside the images, the LSB (Least Significant Byte) method is usually used. To a computer, an image file is simply a file that shows different colors and intensities of light on different areas of an image. The best type of image file to hide information inside is a 24 Bits BMP (Bitmap) image. When an image is of high quality and resolution it is easier to hide information inside image[10]. Although 24-Bit images are best for hiding information due to their size. When using a 24-Bit image, one can store (3-bits) in each pixel by changing a bit of each of the red, green and blue color components. The problem states from the fact that modifying the three colors of a pixel produces a major distortion in the resulting color. So the one method that would introduce more efficiency and less distortion is Enhanced Least Significant Bit.

Enhanced LSB algorithm works in the spatial domain. It improves performance of LSB by hiding information in only one of the three colors that is blue color of the carrier image by using HRNs which are generated from ROPUF based on microcontroller chip (PIC32MX795F512L). That means the choosing process for the pixels is Random depending on the HRNs.

1.3 Related Works

- J.-W. Lee et.al. (2004) had used SPRF$_S$ to produce RN$_S$. It Does not require a secret starting point; This HRN$_s$ producer was depending on the (meta-stability inherent) in (Silicon PUFs (SPUFs)) and had been shown to generate outputs which pass most randomized statistical tests [11].
- C. W. O'Donnell, et.al. (2004) described how to use PUFs to create a candidate HRNG. The small argument is used to assist the tenability of this algorithm and offer an overview estimate of the "randomness" of digits made using PUFs through a sequence of algebraic examinations. It is clear that PUF-based RNG is a low-cost and feasible alternative to more complex and costly HRNG [5].
- B. Sunar et.al. (2007) used several distinct RO$_S$ which have been organized on a chip design; Each (RO) contains some inverters. These samples are obtained from (RO$_S$) to the (XOR tree) at regular frequency around the clock [12].
- Mehdi Ayat et.al. (2011) suggested a new structure PUFs to create a candidate HRNG. So far, several RNG$_s$ depend on RO were introduced [13].
- Ali Sadr and Mostafa Zolfaghari-Nejad et.al. (2012) suggested a new structure in which an arbiter depend on PUF has been employed as a (NFSR) to produce TRN$_s$. The output flow rate is 10 million bits per second. The suggested (RNG) is able to pass all (NIST) tests and the (Entropy) of the output stream is (7.999837 bits per byte). The suggested circuit has a very low resource utilization of (193 slices) that makes it appropriate for lightweight applications [14].
- Roel Maes et.al. (2012) developed and well estimated on a substantial set of FPGA devices. It used an extremely enhanced ring oscillator PUF (ROPUF) design [15].

- Dongfang Li et..al. (2015) developed a high-security and high-throughput hardware true random number generator, called PUFKEY, which consists of two kinds of Physical Unclonable Function (PUF) elements. Combined with a conditioning algorithm, true random seeds are extracted from the noise on the start-up pattern of SRAM memories [16].
- Bc. Filip Kodýtek, CSc (2016) proposed improvements of the PUF scheme in order to enhance its statistical properties. Since the proposed PUF is depended on ROs whose frequencies are dependent on various physical conditions, the objective was to eliminate this dependence. Therefore, the improvements were mainly proposed to decrease the influence of physical conditions on the stability of the PUF outputs [17].
- Taner Tuncer, Erdinc Avaroğlu et.al. (2017) designed RNG of LFSR based stream encryption algorithms and their HW implementations are presented. LFSR based stream encryption algorithms have been applied on (FPGA) by using (VHDL) language. The outputs of this FPGA passed the (NIST) tests, and it can be used as the standard for many applications in cryptographic [18].

1.4 Cryptographic Random Number Generators

TRN and PRN are so important in many applications of cryptographic. For example, any cryptosystems use Keys that must be generated in a random style [5]. Random Numbers Generators (RNG_S) are classified into Pseudo Random Numbers Generators ($PRNG_S$) and True Random Numbers Generators ($TRNG_S$).

There are two disadvantages of PRNGs [19]:

First: This category needs some input that specifically controls the result. For safe use of PRNGs, these inputs (starting points) should remain confidential.

Second: Only a fixed bits number can be generated by PRNGs, before they cycle and duplicate themselves. This may be a problem, for applications that need very long periods of random bits.

These two disadvantages of $PRNG_s$ are not the same for $HRNG_S$, where it can get a periodic (random) outputs, without the need for input [19]. This is achieved by using internal unpredictability in complex physical systems to generate result bits. Therefore, the security of any HRNG is directly linked to the infeasibility of modeling and the style of the basic physical system.

$PRNG_S$ produce a string depending on a mathematical procedure by using (one-way) functions.

Therefore, the security of (PR) string depends on the complexity of its algorithms and their functions, while the outputs of PUF are $TRN_{S,}$ Because PUF works into the random variation during an IC manufacture process. Unlike PR sequences, TR sequences are totally unpredictable [13].

1.5 Random Tests

Many tests are used in this thesis in order to test the random of the HRNs.

A. NIST Test Suite [20]

This test is a set of procedures tests, which aims to detect a sequence of binary numbers that do not really work random.

It is a method, which is basically the probability that a certain string can be generated. For each test, if the (p-value) is greater than some constant level of confidentiality (α), then it is (passes). If ($P\text{-}value \geq \alpha$), then the string seems to be random, otherwise the string seems to be non-random. For the tests, a significance level (α) can be chosen. Usually, (α) is chosen in the range [0.001- 0.01] .

- When the value of (α) = (0.001), it means that one string in (1000) strings will be rejected by the test. A (P-value ≥ 0.001), would mean that the string would be considered to be random with a confidence of (99.9%). A (P-value < 0.001), would mean that the string is non-random with a confidence of 99.9% .
- When the value of (α) = (0.01), it means that one string in (100) strings will be rejected by the test. For (P-value ≥ 0.01), a string would be considered to be random with a confidence of (99%). For (P-value < 0.01), a string would be considered to be random with a confidence of (99%).

1. Frequency (Monobit) Test:

The purpose of this test is to determine whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence. The test measures the closeness of the fraction of ones to ½, that is, the number of ones and zeroes in a sequence should be about the same. All subsequent tests depend on the passing of this test.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-value} = \mathit{erfc}\left(\frac{s_{obs}}{\sqrt{2}}\right) \qquad \textbf{......(1.1)}$$

where *erfc* is the complementary error, $S_{obs} = \frac{Sn}{\sqrt{n}}$, $S_n = X_1 + X_2 + \cdots + X_n$

, where $X_i = 2\varepsilon_i - 1$, ε : input sequence.

2. Frequency Test within a Block:

The focus of the test is the proportion of ones within M-bit blocks. The purpose of this test is to determine whether the frequency of ones in an M-bit block is approximately M/2, as would be expected under an assumption of randomness. For block size M=1, this test degenerates to test 1, the Frequency (Monobit) test.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-}value = \textbf{igamc}\,(N/2,\ \chi^2(obs)/2) \qquad \text{……….}\textbf{(1.2)}$$

where igamc is the incomplete gamma functions.

3. Runs Test

The focus of this test is the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits. The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such zeros and ones is too fast or too slow.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-}value = \textbf{\textit{erfc}}\left(\frac{|V_n(obs) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right). \qquad \text{…}\textbf{(1.3)}$$

4. Longest Run of Ones in a Block

The focus of the test is the longest run of ones within M-bit blocks. The purpose of this test is to determine whether the length of the longest run of ones within the tested sequence is consistent with the length of the longest run of ones that would be expected in a random sequence. Note that an irregularity in the expected length of the longest run of ones implies that there is also an irregularity in the expected length of the longest run of zeroes. Therefore, only a test for ones is necessary.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-}value = \textbf{igamc}\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right). \qquad \text{…..}\textbf{(1.4)}$$

5. Rank Test

The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The purpose of this test is to check for linear dependence among fixed length substrings of the original sequence.

The function of P-value of this test is as follows:

$$\text{Compute } P - value = e^{-\chi^2(obs)/2} \quad \text{.....}\mathbf{(1.5)}$$

6. Fast Fourier Test

The purpose of this test is to detect periodic features (repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-}value = erfc\left(\frac{|d|}{\sqrt{2}}\right). \quad \text{....}\mathbf{(1.6)}$$

7. Overlapping Template Matching Test

The focus of the Overlapping Template Matching test is the number of occurrences of pre-specified target strings. This test uses an m-bit window to search for a specific m-bit pattern, if the pattern is not found, the window slides one bit position.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-}value = \mathbf{igamc}\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right). \quad \mathbf{....(1.7)}$$

8. Serial Test

The focus of this test is the frequency of all possible overlapping m-bit patterns across the entire sequence. The purpose of this test is to determine whether the number of occurrences of the 2^m m-bit overlapping patterns is approximately the same as would be expected for a random sequence.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-}value = \textbf{\textit{erfc}}\left(\left|\frac{f_n - expectedValue(L)}{\sqrt{2}\sigma}\right|\right) \quad \textbf{....(1.8)}$$

9. Entropy Test

The focus of this test is the frequency of all possible overlapping m-bit patterns across the entire sequence. The purpose of the test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths (m and m+1) against the expected result for a random sequence.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-}value = \textbf{igamc}(2^{m-1}, \frac{\chi^2}{2}). \quad \textbf{....(1.9)}$$

10. Cumulative Sums Test

The focus of this test is the maximal excursion (from zero) of the random walk defined by the cumulative sum of adjusted (-1, +1) digits in the sequence. The purpose of the test is to determine whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences. This cumulative sum may be considered as a random walk. For a random sequence, the excursions of the random walk should be near zero.

The function of P-value of this test is as follows:

$$\text{Compute } P\text{-}value = 1 - \sum_{k=\left(\frac{-n}{z}+1\right)/4}^{\left(\frac{n}{z}-1\right)/4} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right] + \sum_{k=\left(\frac{-n}{z}-3\right)/4}^{\left(\frac{n}{z}-1\right)/4} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right)\right] \quad \textbf{...(1.10)}$$

B. Entropy

B.1 Entropy Concept [21]

This metric is used for computing the random characteristics. In (1865) this term was invented to describe a measure of the disorder of a thermodynamically system, and in 1948 it was modified by Shannon as a measure of the unpredictability of information content. It is difficult to realize the difference between entropy and randomness.

Possibly the simplest way to think about the variance is: (Entropy is the uncertainty of an output result after it happen.

While Randomness is the quality of uncertainty from a historical viewpoint. Usually, high entropy means, more secure. Note that:

i. X bits of Entropy equal to 2^{x} possible states.

ii. 7 bits less Entropy equals to ($\frac{1}{2^7}$ or $\frac{1}{128}$) less computational work for an enemy.

iii. Without enough Entropy, Exploitation becomes easy.

iv. More Entropy means More Security.

B.2 Entropy Test

It is a metric which is using to measure the unpredictability of the state, or to measure its average information content [22]. The technique of this metric takes into account the probability of observing a specific event, so the information encapsulates is information about the underlying probability distribution; not the meaning of the events themselves [23]. These processes contain entropy to provide computational randomness but may still contain large sections of bad statistical randomness. Therefore we need to provide statistical randomness as well as computational randomness. Data is statistically random if it passes statistical randomness tests (such as the NIST tests).

Computational randomness considers the amount of entropy in data, data is computationally random if it contains enough entropy and is therefore not predictable for an enemy [22].

B.3 Shannon Entropy [24]

It is defined as (the Shannon entropy) of a discrete random variable z with alphabet Z and probability mass function Pz(Z) = P (Z = z); such that (z∈Z) is defined as:

$$H_b(Z) = -\sum pz(Z)\ \log_b pz(Z) \qquad \textit{such that}\ z \in \mathrm{Z}\ldots\ (1.11)$$

If the base (b) of the logarithm equals 2, the entropy is measured in bits. Similarly, if it is 10 it is measured in digits. If (b) is not specified it is usually assumed to be 2, and the index (b) is dropped.

The entropy is often expressed in bits, which is a logarithmic scale: the entropy of "n bits" is entropy equal to 2^n.

C. Correlation

It is one of the most widely used and widely misunderstood statistical concepts. The term "correlation" refers to a mutual relationship or association between quantities. It is a statistical measure that describes the association between random variables[25]. It is a useful metric because it can help in predicting one quantity from another.

C.1 Correlation test

In statistics, this coefficient (mean square emergency coefficient), is symbolized by (**φ**) or (**$r_{φ}$**), used as a measure of correlation for two binary variables. It was discovered by (Karl Pearson). It is like Pearson's correlation coefficient (PCC) in its interpretation. The fact is that the (PCC) for two binary variables do provides the (phi coefficient). The square of the Phi coefficient is related to the chi-squared statistic for a (2×2) emergency table.

C.2 Interpretation

When the value of phi-Co is found, it is required to interpret what this value stands for. Phi coefficient value which is between (-1, 1), actually shows the linear dependence between variables.

The closer absolute value to 1 represents a strong correlation. This will be illustrated below [25]:

i. The first state: If the result falls within the range (-1.0, -0.7); it means that there is a strong-negative relationship.

ii. The second state: If the result falls within the range (-0.7, -0.3); it indicates weak-negative correlation.

iii. The third state: If the result falls within the range (-0.3, 0.3); it concludes very little or no correlation.

iv. The fourth state: If the result falls within the range (0.3, 0.7); it means that there is a weak-positive correlation.

v. The fifth state: If the result falls within the range (0.7, 1.0); it denotes a strong-positive correlation.

In the field of encryption and decryption, correlation is often roughly used as a measure of the uniqueness of a Key. Correlated process variation negatively affects the uniqueness [26].

C.3 Phi coefficient

It can be used whenever the variables having categories like success or failure, yes or no for example in educational as well as psychological testing;

It can be computationally defined as the (square root) of the ratio of (chi-square) to the sample size, phi coefficient is defined as [27]:

$$\phi = \sqrt{\frac{\chi^2}{\mathrm{n}}} \qquad \ldots (1.12)$$

Or

$$\phi = \frac{\chi^2}{n} \qquad \ldots (1.13)$$

This means that the size of sample (n) is removed, by dividing chi-square on (n) and calculating its square root.

Let us introduce emergency tables first. This table will be a table in two categories because each variable in it can allocate only two values. A (2×2) emergency table for two random variables (a and b) as given below:

Table (1.1) A (2×2) Emergency Table

a/b	1	0	Total
1	v	w	v +w
0	x	y	x +y
Total	v +x	w +y	N

Let the total v +w = o, x + y = p, v + x = q, w + y = s. Thus, the table becomes:

Table (1.2) The Value Of The Total

a/b	1	0	Total
1	v	w	O
0	x	y	P
Total	q	S	N

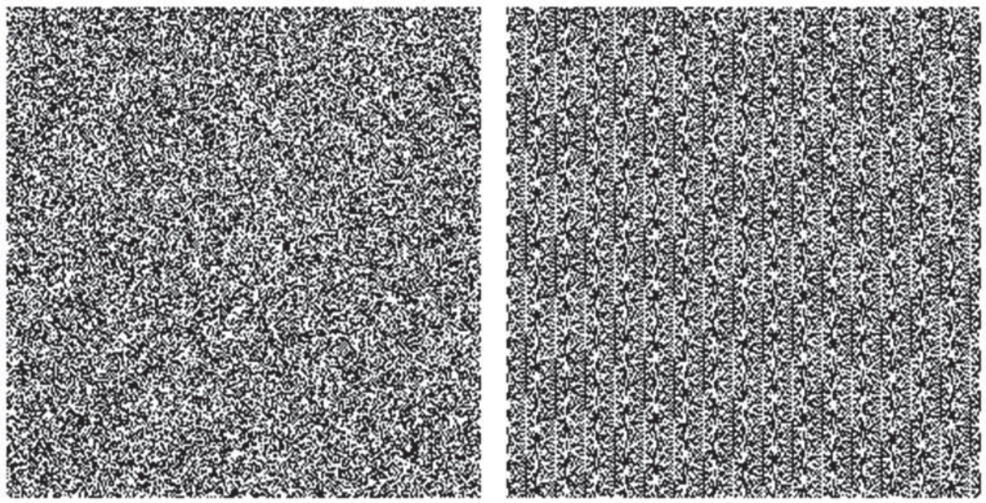
Then, the formula for phi coefficient is given by the following relation:

$$\phi = \frac{v.w - w.x}{\sqrt{o \times p \times q \times s}} \ldots (1.14)$$

D. Simple Visual Analysis [28]

Creating visualization to the outputs is one way to examine an RNG. Human's visions are excellent in spotting patterns, but cannot consider this approach as a proper analysis; it is an enjoyable and quick way which is used to get a rough results of a specific generator's act.

Figure (1.1) shows bitmaps that are parts of large bitmaps; which were created by Bo Allen on April (2008). He was created the bitmap on the left with RANDOM.ORG's bitmap, which is generated by a (TRNG), while on the right there is the bitmap with the RAND ( ) function (from PHP on Microsoft Windows), which is generated by (PRNG).



(a) Random.ORG **(b) PHP RAND ( ) On Microsoft Windows**

Figure (1.1) The Randomness of Two Bitmaps Images[28]

1.6 Image Quality Metrics

A. Mean Square Error (MSE)

These metrics are used to determine the quality of hiding. It represents the cumulative squared error between the stego and the original one. If it is low, the process will be accepted else it will be refused. It Can be calculated using equation (1.15) [29].

$$MSE = (\sum[\mathbf{I}2(i,j) - 2(i,j)]^2)/(P \times Q) \qquad \dots (1.15)$$

Where, Q and P represent the number of columns and rows in the input images, sequentially. $I_1$is the (original image) and I_2 is the corrupted image (stego image).

B. Peak Signal to Noise Ratio (PSNR)

It is one of the metrics that used to find out the degradation in the embedded image with respect to the host image. In PSNR, all values above

(36 dB) are means not to notice significant deterioration (degradation) by the human eye. This metrics can be calculated by equation (1.16) [29].

$$PSNR = 10\log_{10} \mathrm{A}^2 / \mathrm{MSE} \ldots. \quad (1.16)$$

Where A is the maximum difference in the input image data. The (PSNR) refers to peak error measurement. If the PSNR is high the hiding process will be accepted else refuse it.

C. Number of Pixel Change Rate Error (NPCR):

It is used to compute the number of pixels in the variance of two images. It can be computed as [30]:

$$NPCR = \sum_{\mathrm{i,j}} \frac{\mathrm{D(i,j)}}{\mathrm{W \times H}} \times (100\%) \quad \ldots (1.17)$$

Where (H) and (W) denote to the height and width of the image; (C1), and (C2) are source image and cover image; while D(i; j) is defined as follows:

i. If C1(i; j) = C2(i; j); then D(i; j) = 1.

ii. If C1(i; j) ≠ C2(i; j); then D(i; j) = 0.

D. Unified Average Changing Intensity (UACI):

One of the various measures that used in protection field to determine the power of the encryption process; it is calculate the image average intensity to compare the variances between the two images. It is calculated as follows [30]:

$$UACI = \frac{1}{H \times W} \sum_{\mathrm{i,j}} \frac{|\mathrm{C1(i,j)} - \mathrm{C2(i,j)}|}{255} \times 100 \qquad \ldots (1.18)$$

Where (H) and (W) are the height and width of the image, (C1) and (C2) are source image and cover image.

1.7 Problem Statement

The first difficulty of this work is how to produce random and unpredictable numbers with these attributes:

1- Created by (physical one-way functions).

2- More security with no stability system.

3- Achieved high S/W security and Low H/W complexity.

4- Low-cost to manufacture.

5- Hard to duplicate.

6- Not subject to any mathematical description.

7- No initial point input is needed.

8- create non-linear system to solve the problems of attacks.

The second difficulty is how to use HRN_s, which is created by the first challenge (difficulty), as Keys for the security system.

1.8 Aim of The Thesis

The objective of this work is to design and implement multilevel secure system consists of cryptographic system (RC4 system) to encrypt the secret message, and steganographic system (LSB system) in order to hide the existence of the encrypted secret message.

Both RC4 cipher system and LSB steganographic system are depended on HRN_s. These HRN_s were used in the RC4 algorithm where the weaknesses in the standard algorithm were resolved and also used in LSB algorithm to distribute the bits blindly (randomly). These HRNs are created through design and implement Ring Oscillator PUF (ROPUF). Microcontroller PUF with a chip (PIC32MX795F512L) is used to create ROPUF with outputs (RN_s), and without requiring any initial point inputs.

1.9 Thesis Outlines

This thesis is structured around Five chapters, including chapter one, it contains the following chapters:

- **Chapter Two**: **(Physical Unclonable Function)**

  This chapter explains in detail Physical Unclonable Function, Applications, Metrics, Classifications, and Microcontroller PIC32MX795F512L.

- **Chapter Three** :**(Design of The Proposed System)**

  In this chapter, the proposed algorithm design and the implementation steps are given.

- **Chapter Four: (Results and Evaluation)**

  This chapter is dedicated to showing the outputs and tests of the proposed system.

- **Chapter Five: (Discussion, Conclusions and Recommendations for Future Works)** some concluding remarks which are derived from the outputs of the conducted tests are given in this chapter; also some suggestions for future work are presented.