**ISSN: 0067-2904**

# Enhanced RC5 Key Schedule Using One-Dimensional Cellular Automata for Audio File Encryption

**Yussra Majid Hameed*, Nada Hussien M. Ali**
Department of Computer Science, College of science, University of Baghdad, Baghdad, Iraq

**Abstract**

Audio security is an important aspect in various areas of communication. This paper deals with audio encryption as many of the data communication depends on audio data. In this paper, a new proposal of audio encryption system has been introduced. The system can be divided into two phases, the first phase focuses on generating a high-quality Pseudo Random Number generator (PRNGs) using elementary, periodic and hybrid rules of cellular automata (CA). The system suggests a new combination of CA rules in an endeavor to provide high randomness and to improve the strength of the proposed cryptosystem. Whereas the second phase produces the Enhanced Rivest Cipher 5 (ERC5) algorithm which employs the generated Random Number Sequence (RNS) in an effort to strengthen the security and randomness of the original Rivest Cipher 5 (RC5) algorithm.

The results show that the proposed PRNGs based on CA can generates RNS with a high period which can reach to more than 100,000 keys without repetition or string duplication. Moreover, the tests demonstrates that the proposed ERC5 improves the security of the original RC5 algorithm. The proposed cryptosystem is evaluated in terms of Shannon theory of information entropy, randomness tests, computation time and key space analysis. The results verify that the suggested audio crypto-system increases the growth of the security level of original RC5 encryption algorithm with high degree of randomness and confidentiality.

**Keywords**: Encryption; Cellular Automaton; RC5; Audio Encryption.

# تحسين جدولة مفاتيح RC5 بإستخدام الخلويات الاوتوماتية ذات البعد الواحد لتشفير الملفات الصوتية

**يسرى ماجد حميد*، ندى حسين محمد علي**
قسم علوم الحاسوب، كلية العلوم، جامعة بغداد، بغداد، العراق

**الخلاصة**

أمنية الملفات الصوتية هو جانب مهم في مختلف مجالات الاتصال. يتناول هذه البحث تشفير الصوت لأن العديد من اتصالات البيانات تعتمد على البيانات الصوتية في هذا البحث ، تم تقديم اقتراح جديد لنظام تشفير الصوت. يمكن تقسيم النظام إلى مرحلتين ، تركّز المرحلة الأولى على توليد مولد Pseudo Random Numberعالي الجودة (PRNGs) باستخدام القواعد الأولية والدورية والهجينة للأوتوماتية الخلوية(CA) . يقترح النظام مجموعة جديدة من قواعد CA في محاولة لتوفير درجة عالية من العشوائية وتحسين قوة نظام التشفير المقترح. في حين أن المرحلة الثانية تنتج خوارزمية RC5 محسنة (ERC5) التي تستخدم تتابع الأرقام

*Email: yussra.majid@gmail.com

العشوائية المولدة (RNS) في محاولة لتقوية الأمان و العشوائية لخوارزمية Rivest Cipher 5 (RC5) الأصلية.

تظهر النتائج أن PRNGs المرتكزة على CA يمكن أن تولد RNS مع فترة عالية والتي يمكن أن تصل إلى أكثر من 100000 مفتاح دون تكرار. علاوة على ذلك ، توضح الاختبارات أن ERC5 المقترحة تعمل على تحسين أمان خوارزمية RC5 الأصلية. تم تقييم نظام التشفير المقترح من خلال نظرية شانون للإنتروبيا المعلوماتية ، اختبارات العشوائية ، وقت الحساب وتحليل الفضاء الرئيسي. تتحقق النتائج من أن نظام التشفير الصوتي المقترح يزيد من نمو مستوى الأمان لخوارزمية التشفير RC5 الأصلية مع درجة عالية من العشوائية والسرية.

## 1. Introduction

With the growth of the internet, the security of information is gaining more and more interest. The encryption techniques can efficiently safeguard people's information transmitted over public channels. However, the classical encryption systems have restrictions in encrypting such as low efficiency, bulky data, and the high correlation between samples and so on [1].

Recently, cellular automata have attained a great interest to deal with the problematic of highly and fast secure cryptosystems. Cellular Automata (CAs) are highly distributed and parallel systems that can pretend sophisticated behaviors [2]. The wide range of CA rules facilitates many ways to generate sequences; moreover, cellular automata work out by only easy and simple logic computations with complex and pseudorandom behaviors. Generating good-quality Random Number Sequence (RNS) is not an easy function, Pseudo Random Number generator (PRNGs) based CA attracts many researchers because they are easy to implement in both hardware and software [3]. A cellular automation is like a machine, which means that the input is placed, the CA will use it to produce an output.

The benefits of cellular automata in cryptography can be summarized as follows [2]:

- ❖ Large growth of rules space.
- ❖ Cellular automata contain only logic operations or integer arithmetic, so these characteristics lead to simplifying the computation.
- ❖ Cellular automata have parallelism and show complex behaviors.

Also, RC5 is a famous block cipher distinguished for its speed, simplicity, suitability for software and hardware execution and a low memory requirement. Moreover, RC5 is a parameterized algorithm and iterative in its design. This gives the prospect for an unlimited flexibility in both the level of security and performance characteristics [4]. In this paper, an enhancement method of RC5 encryption algorithm had been suggested by strengthen its original weak keys through generating good-quality random number sequences by one elementary, periodic and hybrid CA.

The rest of the paper is organized as follows: related works are summarized in Section 2. Section 3 gives essential theoretical definitions of cellular automata while Section 4 produces an explanation of RC5 encryption algorithm. Section 5 gives a detailed description of the proposed system whereas Section 6 discusses the security analysis and results of the proposed algorithms. Finally, the main conclusion is summarized in Section 7.

## 2. Related Works

1. In 2009: Ho et al. [5] have attempted to find out the ideal combination of CA rules and logical operations for choosing von Neumann neighbors. The authors propose a various number of CA-rules for excellent impact and also concluded that the periodic boundary is better than null boundary. The authors provided fail/pass rate between periodic and null boundary conditions. Additionally, they deduced that the non-uniform CA has better influence than uniform CA.
2. In 2011: Osama [4] enhanced the RC5 block cipher algorithm based on chaos for achieving higher security and better image encryption. This was achieved by combining the cryptographic primitive operations and the chaotic skew tent map to develop a new structure for the key schedule, as well as the heavy use of data-dependent rotations increased the diffusion achieved per round. The system provides fast block cipher besides high security level.
3. In 2014: Dogaru Radu et al. [6] considered many solutions for building a good Pseudo Random Number Generation (PRNG) for a cryptographic system. The system is based on Hybrid Cellular Automata (HCA). The first solution was based on making chains of HCA, i.e. the non-linear map which changed dynamically were controlled by another HCA map within a chain (to ensure maximum

throughput). The second solution was based on a single HCA output, which was down-sampled by a factor D. The suggested scheme provides a good PRNG with low complexity fulfillments. Moreover, the system expected to have a high immunity to different types of attacks.

## 3. Cellular Automata

The CAs are type of dynamical systems that have been effectively and broadly used to construct strong cryptosystems by taking the advantages of their randomness and dynamical properties, with the capability of reveal unpredictable and complex behavior [7]. A cellular automation encompasses of a lattice (grid) of identical cells within a Boolean value for each cell, mentioned as current state of cell. The state of each cell is updated at discrete time step according to a local update rule. The equivalent decimal of the 8 outputs is called 'rule' [8]. Cellular automaton is composed by the four following pieces of information: An alphabet ($S$): The limited set of all acceptable states. A lattice ($\mathcal{L}$): an ordered grid, typically $\mathbb{Z}^d$, with d-dimensional lattice and $d \in \mathbb{Z}_+$. A neighborhood ($N$): a finite well-organized sub-set of $\mathcal{L}$. A local transition function or rule ($f$): The next states of every cells are decided by its rule [9].

CA lattice can be composed by two main functions, local and global. In the local transition function: $S^{|N|} \to S$ , $S^{|N|}$ represents the set of all possible states that the neighborhood can be in, with each of the values is a tuple of states $(S_0, S_1, S_2, \dots, S_{|N|-1})$, with $S_i \in S$ . For example, if S= {0, 1} and |N| = 3, $S^{|N|}$ can be represented by the set {(0,0,0), (0,0,1),..., (1,1,1)}. The local transition function (rule) determines how the state of each cell is changed from an instant to the next. This decision is usually based on the cell's own current state and of its neighbors. In the global function**,** If $c$ is considered as the current configuration of the automata with $c \in \mathbb{Z}^d$. The CA's next configuration is given by $\Phi(c)$,where $\Phi: \sum \mathbb{Z}^d \to \mathbb{Z}^d$ . $\Phi$ Called global map or global function. The CA's temporal evolution is then:

$$\mathbf{c} \to \boldsymbol{\Phi}(\boldsymbol{c}) \to \boldsymbol{\Phi}^{\mathbf{2}}(\boldsymbol{c}) \to \cdots \quad \textbf{..... (1)}$$

The name the sequence c , $\Phi(c), \Phi^2(c), \dots$ is the orbit or population of c.

The cellular neighborhood of a cell consists of itself and of the surrounding (adjacent) cells [10]. There are two basic types of CA dimensions [8]: One-dimensional (1D) CA where each cell has a two possible states, and a cell's neighbors are the adjacent cells on each side of it. Figure-1 demonstrates the 1D CA. 2D CA has some of the similar features as do 1D CA. There are two essential types of neighborhood that are mostly deliberated. The first type is the Von Neumann neighborhood which consists of the 4 or 5 cell array based on whether or not the central cell is counted. The second type is the Moore neighborhood which consists of the 8 or 9 cell array based on whether or not the central cell is counted. Figure-2 demonstrates the Von Neumann Neighborhood and Moore neighborhood [11].

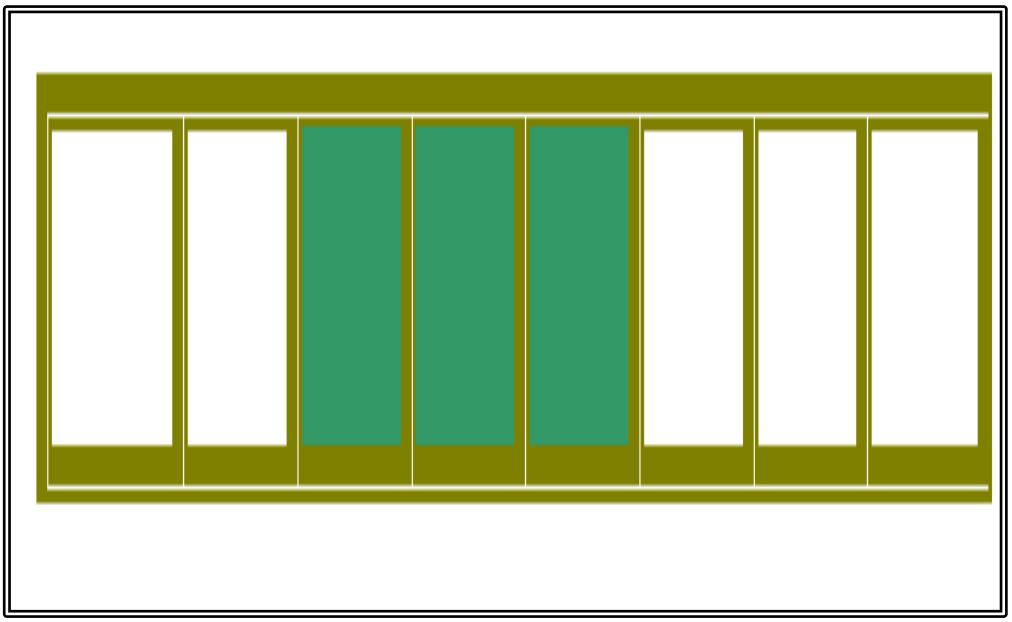
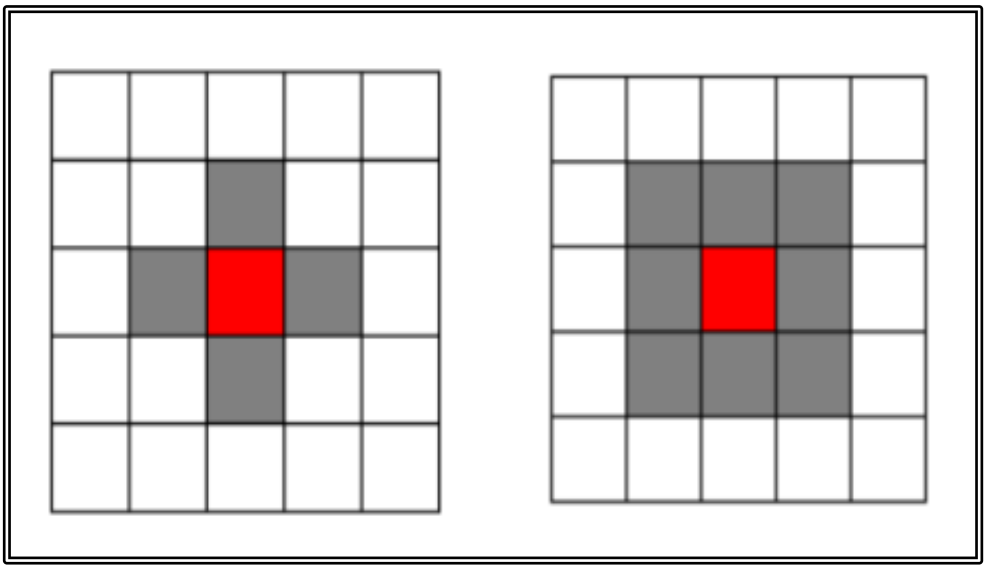


**Figure 1**-One dimensional Cellular Automata [11].



(a) (b)

**Figure 2**-Two dimensional Cellular Automata [11]: (a) Von Neumann neighborhood, (b) Moore neighborhood.

### 3.1 Elementary Cellular Automata

A neighborhood of a cell x with radius r is the set of the r cells both to the left and right of x, including cell x. An ECA is any cellular automata with r=1 and a binary state set S = {0, 1}. The equation of ECA can be as follows:

$$x_i^{t+1} = f(x_{i-1}^t, x_i^t, x_{i+1}^t) \quad \text{..... (2)}$$

Where $x_i^t$ is the cells state, $x_i^{t+1}$ is the cells next state, $x_{i-1}^t$ is the state of the cells left neighbor, $x_{i+1}^t$ is the state of cells right neighbor and $f$ is the rule function. 1D ECA is special class of discrete dynamical systems formed by a finite 1D array of $N$ cells . The total number of rules for radius r neighborhood is $2^n$ where n=$2^{2r+1}$ [12]. Therefore, ECA has $2^{2^3} = 256$ possible rules [10].

### 3.2 Boundaries of Finite Lattices

Infinite CA have no boundaries accounts associated with them. When we are dealing with CA with a finite $\mathcal{L}$, the neighborhood used by the local transition function exceeds the lattice boundaries. There are primary solutions for this problem: Null boundary CA**,** in a null boundary where the CA contains $n$ - cells as $X1, X2 .., Xn,$ the leftmost neighbor of $X1$ and the rightmost neighbor of $Xn$ are considered as zeros for each of them. Also, if $Xn$ is taken as leftmost neighbor of $X1$ and $X1$ is taken as the rightmost neighbor of $Xn$ then it is called cyclic or periodic boundary CA [8].

### 3.3 Types of CA Suitable for Cryptography

There are many types of CA suitable for cryptography, if a cellular automation utilizes some control signals, it is known as Programmable Cellular Automata (PCA) [12]. If CA always return to its initial state then it will be called Reversible Cellular Automata [13]. And if the same rules determine the next bit cells of CA, then it is called uniform CA otherwise it is called non-uniform CA or Hybrid Cellular Automata (HCA) [14]. The HCA generates more complex patterns than uniform cellular automaton. This property is very useful in the case of cryptography, as they will generate more complex ciphers [15].

## 4. Rivest Cipher 5 (RC5) Algorithm

RC5 is a block, symmetric key encryption algorithm; its simple design makes it appropriate for implementation in software and hardware. RC5 has a high degree of flexibility in terms of security and performance due to its flexible options such as variable key size which works on 0 up to 2040 bits, variable block size (32, 64 or 128 bits block of data) and variable number of rounds (0 to 255) [4].

The parameters of RC5 are as follows: w refers to the block size, r refers to the number of rounds and b refers the key length in bytes, simply written as RC5-w/r/b. Ronald L. Ronald L. Rivest indicates that the greater the security/encryption level when the higher the number of rounds. [16].

The overall scheme of RC5 is divided into 3 stages: key expansion, encryption stage and decryption stage. The key operations in these three stages are Bit-wise XOR of words, addition of words modulo 2w and shift in two directions left (<<) and right (>>).

## 5. The Proposed System

The proposed system can be recapitulated into two phases, the first phase is designed a pseudo random number generator based on cellular automata phase, whereas the second phase is composes of enhancement of RC5 encryption algorithm phase.

### 5.1 Phase One: Design a Pseudo Random Number Generator Based on Cellular Automata

In this phase, a symmetric key cryptographic technique using CA has been implemented through Pseudo Random Number Generator Based on Cellular Automata (PRNGs-CA). The algorithm uses elementary, periodic, hybrid and programmable CA with rules set namely 30, 90 and 150 CA to form a sequence of 128-bit random numbers. The PRNGs algorithm can be divided into three parts as bellow:

#### 5.1.1 Generating Initial Seed

The Initial Random Seed (IRS) of the system is defined by generating a random string composed of 1024 characters. Then, the random string passes to the MD5 algorithm in order to generate the IRS which consist of 128 bits (16 bytes) that represented as output of MD5". The purpose of passing the string to the MD5 algorithm is to ensure that it is infeasible to produce two messages having the same hash value, or to produce any IRS having a given pre-defined target message digest, thus increasing the strength of the IRS to makes the brute-force attack more difficult.

#### 5.1.2 Rule Scheduling Procedure

In order to make the CA as a PCA, this part has been applied to increase the strength of the algorithm and make the cryptanalysis more difficult. The rule scheduling algorithm involves choosing which rules set of CA would be applied to the blocks of generated RNS (currently eight rules sets have been included in this study).

Random Integer Number (RIN) is represented a random integer with a value between 1 and 8. Then the rules set will be chosen based on this RIN. For each sub-key (described in details by the next sub section), the value of RIN will be changed and thus the rules set would be different with each sub keys. The main goal of the rule scheduling algorithm is to make cryptanalysis infeasible, and consequently increasing the system robustness against attacks. And also to increase the strength of the system. Table-1 shows which rules set will be applied based on RIN value. This combination of rules is used in the third part which consists of generating symmetric keys.

**Table 1**-RIN value and its corresponding rules set.

| RIN value | Rules set |
|---|---|
| **1** | 90, 150, 30, 30, 150, 150, 90, 30 |
| **2** | 90, 30, 90, 150, 90, 30, 90, 150 |
| **3** | 150, 30, 30, 90,150, 30, 150, 90 |
| **4** | 150, 90, 150, 30, 90, 30, 30, 90 |
| **5** | 90,150,30,30,150,30,90,90 |
| **6** | 90, 30, 150, 30, 90, 150, 150, 90 |
| **7** | 150, 90, 30, 90, 150, 30, 90, 30 |
| **8** | 90, 30, 150, 90, 30, 30, 90, 150 |

#### 5.1.3 Key Propagation Mechanism

The fundamental intentions of the proposed key-propagation mechanism part are:

- The sub keys should be a cryptographic pseudo random and collision resistant.
- Ease, effortlessness and simplicity of implementation.

In the suggested system, the derivative of the sub-keys are done through IRS (128 bits) which is considered as an initial pattern (seed) using a hybrid 1D periodic class of CA. The purpose of using 1D CA is to ensure simplicity and high speed of execution time.

- **Key propagation steps:**

**Step 1**: Consider IRS as pattern #0.

**Step 2**: Divide pattern into blocks of 8 bits (1-byte), i.e. the pattern will be converted into 16 blocks, 8 bits for each (8bit *16 block=128-bit total pattern length).

**Step 3**: Input current pattern to the hybrid CA machine. The hybrid CA machine involves 16 rounds of same rule scheduling for each block, i.e. generating pattern of 128 bits. Each bit of current pattern represents cell's current state $x_i^t$ of CA. for each bit, one of three transition rules, 30, 90 and 150 is applied based on the decision of rule scheduling procedure. Hybrid CA machine updates the cell's current state $x_i^t$ to the next state $x_i^{t+1}$ by using the cell's current state $x_i^t$ and two close adjacent neighbors (left neighbor cell $x_{i-1}^t$ and right neighbor cell $x_{i+1}^t$.
**Step 4**: The output of the hybrid CA machine defines a new pattern.
**Step 5:** Consider the new pattern as a new sub-key.
**Step 6**: repeat steps (2, 3 4 and 5) until all the sub-keys generated.

This mechanism of key expansion can generate sub keys with a large period and high population. Figures-(3, 4 and 5) show key propagation mechanism and sub-keys blocks schedule respectively.

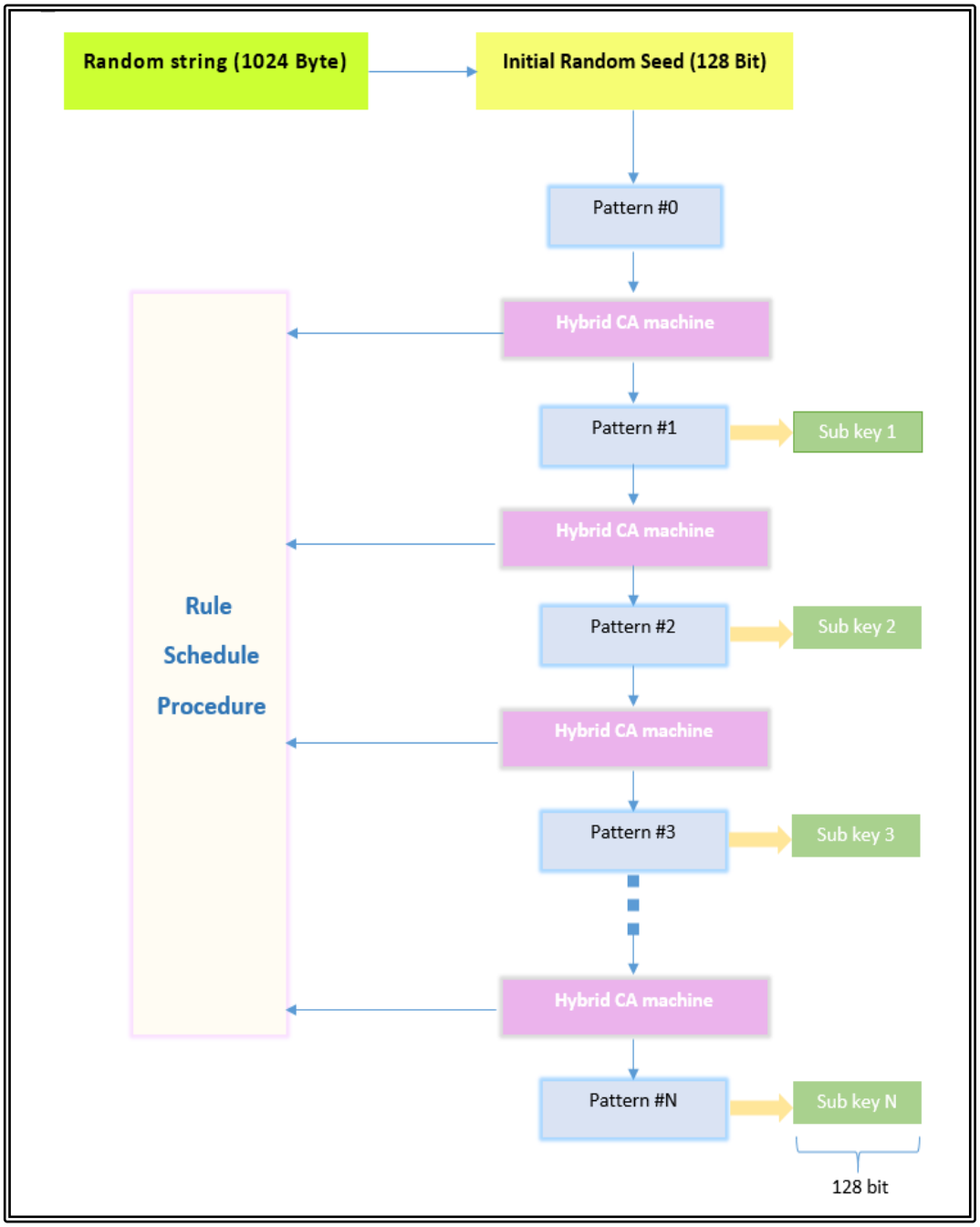



**Figure 4**-Key propagation mechanism.

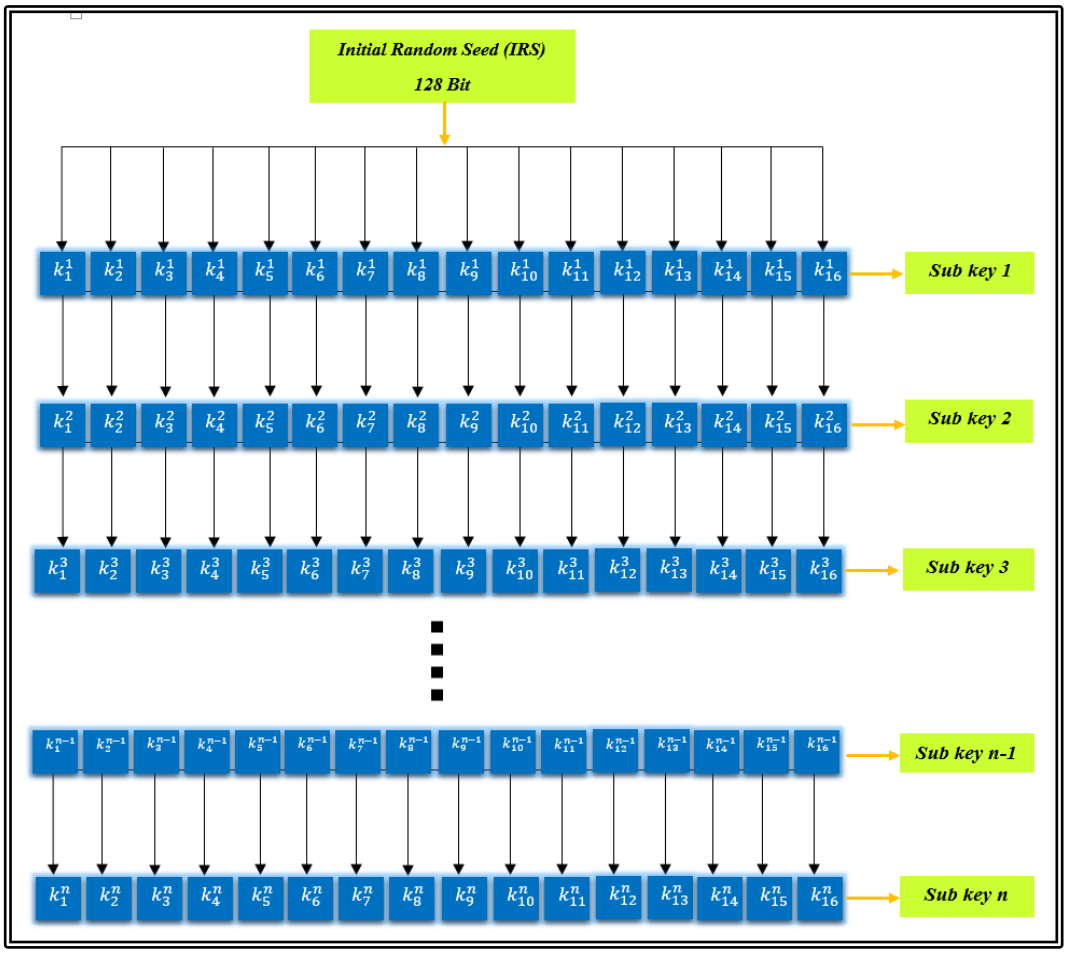



**Figure 5**- sub-keys blocks Schedule.

### 5.2Phase Two: Enhancement of RC5 Encryption Algorithm

The original RC5 has a weak keys and in order to replace it with strong keys, a suggested Enhancement of RC5 encryption algorithm has been implemented as follows:

#### 5.2.1 Enhanced RC5 Key Schedule

The algorithm of Enhanced RC5 Key Schedule (ERC5KS) can be divided into 3 parts: the first part is initializing S array, the second part is initializing L array and the last is the mixing part, respectively, as shown in Figure-6.

1- **Initializing S array:** S array is initialized by the same way of the original RC5 key schedule.

2- **Initializing L array:** L array of the suggested scheme is initialized with each block in the encryption process through different sub-keys which derived from CA as discussed in the previous section. The sub-keys size which generates L array is 128-bit (16 bytes) for each. Initializing of L array is illustrated as the following pseudo code:

| **Algorithm: Initialize L array** |
|---|
| **Input:**<br>sub-keys: sub-keys generated based on CA |
| **Output:**<br>L array |
| **1: Parameter Setting**<br>b = key length; // 16 bytes<br>u = 4; // word size in bytes<br>c = b / u;<br>L = L array (word of c length)<br>**2: Initialize L array**<br>**for** i = 0 to sub-key length **do** |

**Begin**
Key[i] = convert sub-key[i] from string to byte
**End For**
**for** i = b – 1 to 0 **do**
**Begin**
L[i / u] = Left shift L[i / u], 8) + key[i];
**End For**

**3- Mixing S and L arrays:**

The two arrays S and L are mixing by the same as the original RC5 mix function, the only difference is that the L array is initialized with different sub-key for each RC5 encryption and decryption block. Note that, as original RC5 algorithm, because of the probability of getting different sizes of S and L arrays, the larger array is processed three times whereas the smaller array may be processed more times. The following pseudo code demonstrates the mixing procedure.

**Algorithm : Mixing S and L arrays**

**Input**:
S array
L array

**Output**:
A: first word of encryption block
B: Second word of encryption block

**1: Parameter setting**
A=B=0;
i,j=0;
t =S array length
c = array length
word v = 3 *Max (t, c);
**2: Mixing S and L arrays**
**for** counter = 0 to v **do**
**Begin**
A = s[ii] = Left shift ((s[i] + A + B), 3)
B = L[j] = Left shift ((L[j] + A + B), (A + B))
i = (i + 1) mod t
j = (j + 1) mod c
**End For**

### 5.2.2 Enhanced RC5 Algorithm

The block size of the Enhanced RC5 (ERC5) is composed of two words (equivalence to eight bytes), S array is initialized only at the beginning of the encryption process. Whereas L array initialized for each block of wave audio data by CA sub-keys as mentioned in earlier sections, the rest of the encryption algorithm is the same as the original RC5. The idea of encrypting each block of data with different sub-key is to strengthen the original RC5 weak keys so that making cryptanalysis more difficult. The following steps determine the ERC5 encryption algorithm.

**Algorithm 3.10: ERC5**

**Input:**
Input : audio wave samples

**Output:**
Cipher array

**1: Parameter Setting**
Integer i=0
Byte round= value between 0 to 255
Length= wave samples length
**2: ERC5**

```
While Length >0 do // block loop
Begin
Initialize L array();
Mixing S and L arrays();
A=convert input[i] from byte to word // A and B size=4 bytes
B=convert input[i] from byte to word // A and B size=4 bytes
i=i+4; //   word=4 bytes
A = A + s[0]
B = B + s[1]
For i= 0 to round do
Begin
A = Left shift (A XOR B, B) + s[2 * i]
B = Left shift (B XOR A, A) + s[2 * i + 1]
End For
I=i+4 // word=4 bytes
Length=Length-8 // block size
End while
```

Note that if the wave data size does not equal to 64-byte or its multiples (block size is equal to 2 word, 32 byte for each=64 byte), a necessary padding is added to the wave data array.

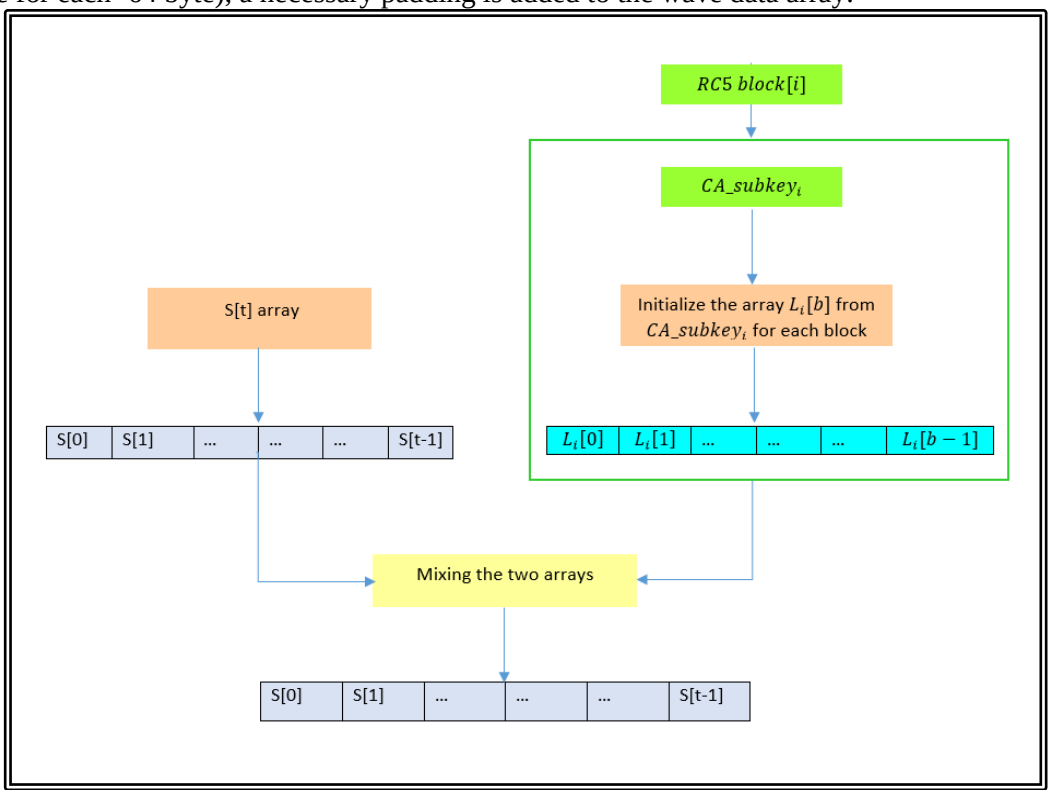



**Figure 6**-ERC5KS scheme.

## 6. Evaluation of the Proposed System

The evaluation of the proposed system have been run under windows 10 Pro. Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20 GHz, 8 GB Random Access Memory (RAM), the system type is 64-bit Operating System, 64x-based processor. C# programming language is utilized to develop the suggested system (Visual Studio 2013). Also, the proposed encryption system used the following data characteristics in an attempt to evaluate the system performance: WAVE (Waveform Audio File Format) 8-bit mono have been used in this research, this means that the number of the wave channels is one, and the sample size (bit depth) of these file is 8 bit. Each audio file evaluated in the proposed system has different sizes and sampling attributes. The tested results can be shown as bellow:

### 6.1 Finding the Rules

In order to find the efficient rules that generates high-quality PRNS to the proposed PRNGs, the system tested the following rules: 30, 45, 51, 60, 90, 102, 105, 150, 153, 165 and 195. In some situations, a uniform lattice was tested, while other simulations evaluate non-uniform (hybrid) CAs.
All the aforementioned rules were tested as a uniform CAs taking into consideration the three most important NIST tests. The system also examined many non-uniform rules sets such as 30, 90,105 with 150 and 30, 45 with 150 and many others. The results show that, generally, non-uniform CAs are a better RNGs than uniform ones, since some of these combinations pass some of NIST tests and could generate RNS with a large periodically.

Finally, producing a RNS with good quality and perfect period was the challenge of this paper, thus the scheme detected that in over 100 experiments, and three rules which gave the best results are tended to govern the concluding developed lattices: 30, 90, and 150. In spite of the low impact of rule 90 on uniform CAs, it has a very good influence on non-uniform CAs.

### 6.2 Results of Period and Statistical Tests

The eight CAs rules sets of the proposed scheme show approximately the equivalent performance level. The tests were applied for different generations, i.e. number of generated sub keys (10,000, 32,000, 50,000, 75,000 and 100,000 generations were tested), and used the same initial value to all of the tested rules set and amount of generation. They all did pass the three NIST statistical tests. Also, they all generates sub keys without any duplication, which means that the proposed system discovered a very large periodicity with *p*-values. This also means that their generation of random sequences give something to be looked-for, desired and worthy. Therefore; the results show that the high quality of the generated RNS is sufficient for many applications. Table-2 summarizes the average p-values of the rules sets applied with different amount of generations (note that if the p-value is greater than 0.01, it considered successful). The table also shows the execution time for these generations. Thus, the applied elementary, non-reversible, hybrid, periodic and programmable CAs provide good and fast RNGs.

**Table 2**-Average *p*-value and execution time of the proposed system

| No. of generated sub keys | Execution time (Sec) | Average *P*-value | | | Status |
|---|---|---|---|---|---|
| | | Frequency | Block | Run | |
| **10,000** | 0.075 | 0.7479 | 0.5811 | 0.5796 | **Successful** |
| **32,000** | 0.239 | 0.5420 | 0.3996 | 0.4851 | **Successful** |
| **50,000** | 0.375 | 0.4831 | 0.4646 | 0.5174 | **Successful** |
| **75,000** | 0.560 | 0.5662 | 0.6181 | 0.4265 | **Successful** |
| **100,000** | 0.748 | 0.6572 | 0.4459 | 0.3116 | **Successful** |

### 6.3 Evaluation of the Enhanced RC5 Algorithm

In order to make RC5 algorithm stronger, it has to weed out its weak keys and replaces them with stronger keys. In an attempt to carry out this task, ERC5 algorithm has been introduced.
The submitted scheme encrypts every block of RC5 with a different key which was generated based on CA as discussed earlier. This algorithm was adopted to progress the weaknesses of RC5 keys and to guarantee advanced security in addition to introduce a better audio encryption. In an effort to check the strength of the proposed work, some experiments have been performed as follows:

**1. Information Entropy Analysis**

Table-3 shows the entropies of the original wave file, RC5 and ERC5.

**Table 3-**1st and 2nd entropy of original wave files, RC5 and ERC5

| Name | 1st Entropy | | | 2nd Entropy | | |
|---|---|---|---|---|---|---|
| | Original Audio | Original RC5 | ERC5 | Original Audio | Original RC5 | ERC5 |
| **Audio 7** | 6.1100 | 7.8440 | **7.9851** | 12.2032 | 15.6757 | **15.9662** |
| **Audio 8** | 7.0400 | 7.9933 | **7.9958** | 14.0678 | 15.9804 | **15.9890** |
| **Audio 9** | 4.9278 | 7.5664 | **7.9959** | 9.8500 | 15.1307 | **15.9891** |
| **Audio 10** | 7.6814 | 7.9957 | **7.9964** | 15.3592 | 15.9890 | **15.9906** |
| **Audio 11** | 5.3865 | 7.9973 | **7.9977** | 10.7175 | 15.9923 | **15.9935** |
| **Audio12** | 2.9016 | 7.2883 | **7.9980** | 5.8009 | 14.5644 | **15.9945** |
| **Audio 13** | 6.3162 | 7.9982 | **7.9982** | 12.6156 | 15.9951 | **15.9953** |
| **Audio14** | 6.5813 | 7.9985 | **7.9987** | 13.0806 | 15.9962 | **15.9965** |
| **Audio 15** | 6.8635 | 7.9911 | **7.9986** | 13.7233 | 15.9807 | **15.9966** |

From the above Table, the successful of the ERC5 is clear through the enhanced results of entropy analysis. When the audio files are enciphered, their 1st entropy should perfectly be 8 and the 2nd entropy should be 16. If the output of such a cipher produces ciphers with 1st entropy less than 8 and/or 2nd entropy less than 16, there exists a confident degree of expectedness, which threats its security. The higher the rate of entropy of enciphered audio, the more improved the security. Table-3 clarifies that the results attained from the proposed system are very close to the theoretical value of entropy compared to the original RC5. This means that the information leakage in the encryption procedure is infeasible and the encryption system is secure against the entropy attack.

**2. Randomness Tests**

In order to ensure the quality of the proposed ERC5 over original RC5, NIST tests were made and the results are shown in Table-4. Also, Figure-7 specifies the success rate of each, the proposed ERC5 always wins.

**Table 4**-NIST tests results of RC5 vs. ERC5.

| Name | Frequency | | Block | | Run | |
|---|---|---|---|---|---|---|
| | Original RC5 | ERC5 | Original RC5 | ERC5 | Original RC5 | ERC5 |
| **Audio 7** | 0.0029 | **0.7732** | **0.8810** | 0.8361 | 0.0434 | **0.2168** |
| **Audio 8** | 0.0830 | **0.9971** | 0.1502 | **0.4969** | **0.7583** | 0.6603 |
| **Audio 9** | 0 | **0.6314** | 0 | **0.4069** | **0.3230** | 0.2911 |
| **Audio 10** | **0.3212** | 0.0816 | **0.3719** | 0.3596 | **0.9565** | 0.5432 |
| **Audio 11** | 0.6776 | **0.9333** | 0.0056 | **0.3021** | 0.0391 | **0.9634** |
| **Audio12** | 0 | **0.3880** | **0.8780** | 0.4393 | 0 | **0.6971** |
| **Audio 13** | 0.0736 | **0.3774** | 0.3571 | **0.2390** | 0.6038 | **0.8048** |
| **Audio14** | 0.3080 | **0.5287** | 0.2287 | **0.959** | 0.5538 | **0.9564** |
| **Audio 15** | 0.0693 | **0.1284** | **0.99999** | 0.1481 | 0.000003 | **0.9712** |
| **Success Rate** | 1/9 | **8/9** | 4/9 | **5/9** | 3/9 | **6/9** |

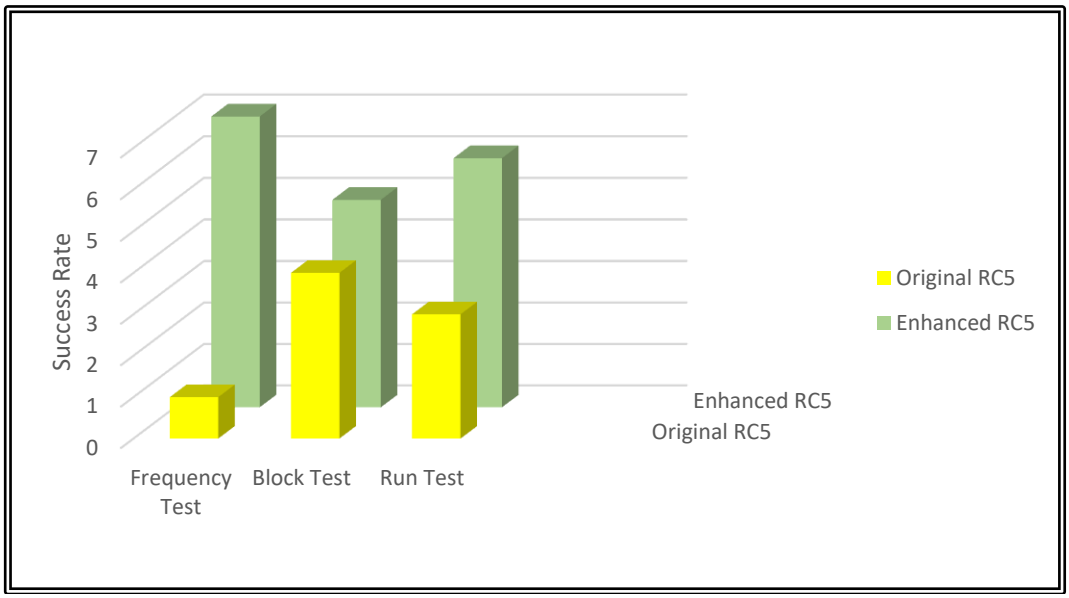



**Figure 7**- Success rate of frequency, block and run tests for ERC5 vs. original RC5.

### 6.4 Execution Time

The following Table shows that the proposed system is fast enough and can be used as an encryption system.

| Name | ERC5 Execution Time (Sec.) |
|---|---|
| **Audio 7** | 0.012 |
| **Audio 8** | 0.032 |
| **Audio 9** | 0.040 |
| **Audio 10** | 0.045 |
| **Audio 11** | 0.060 |
| **Audio12** | 0.070 |
| **Audio 13** | 0.071 |
| **Audio14** | 0.087 |
| **Audio 15** | 0.095 |

### 6.5 Key Space Analysis

A good audio encryption algorithm should be sensitive to the cipher keys, and the key space should be large enough to make brute-force attacks infeasible. For the proposed encryption system, the key space analysis has been carefully studied and can be summarized as follows:

#### 6.5.1 Key Sensitivity Analysis

A major feature for a good crypto-system is an extremely key sensitivity to ensure the security of the crypto-system across the brute-force attack in a measure. Key sensitivity of any crypto-system can be detected in two various ways: Firstly, the cipher audio generated by any crypto-system should be sensitive to the key, for example, if two a bit different secret keys have been used to encrypt the same original plain-audio, then the two generated cipher-audio produced ought to be completely disconnected to each other. Secondly, the cipher audio cannot be decrypted correctly even if there is just a bit variation among encryption and decryption secret keys. In the proposed encryption algorithm, cipher audio depended on each bit of the key, this dependency was achieved by the enhanced RC5 phase, where each block of data depends on the CA sub keys and each sub key of the CA phase depends on the previous key, consequentially this dependency results in making the system's key sensitivity as shown in Figures-(8 and 9). Figure 8 shows different cipher audio of same plain audio encrypted with two keys differs from each other in only one bit whereas Figure-9 shows that if the decryption key differ in only one bit from the encryption key, then the resulting decrypted audio is different from the original plain-audio.

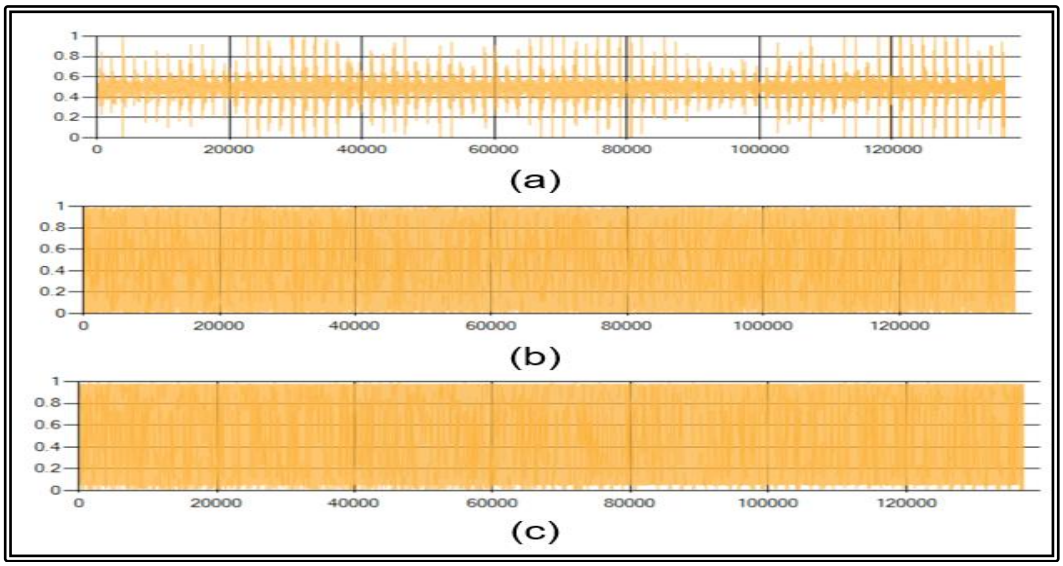



**Figure 8**-Different encryption process for same file audio with two keys differ from each other in only one bit, (a) shows original wave audio file before encryption and (b,c) shows the same wave file encrypted with the two keys.

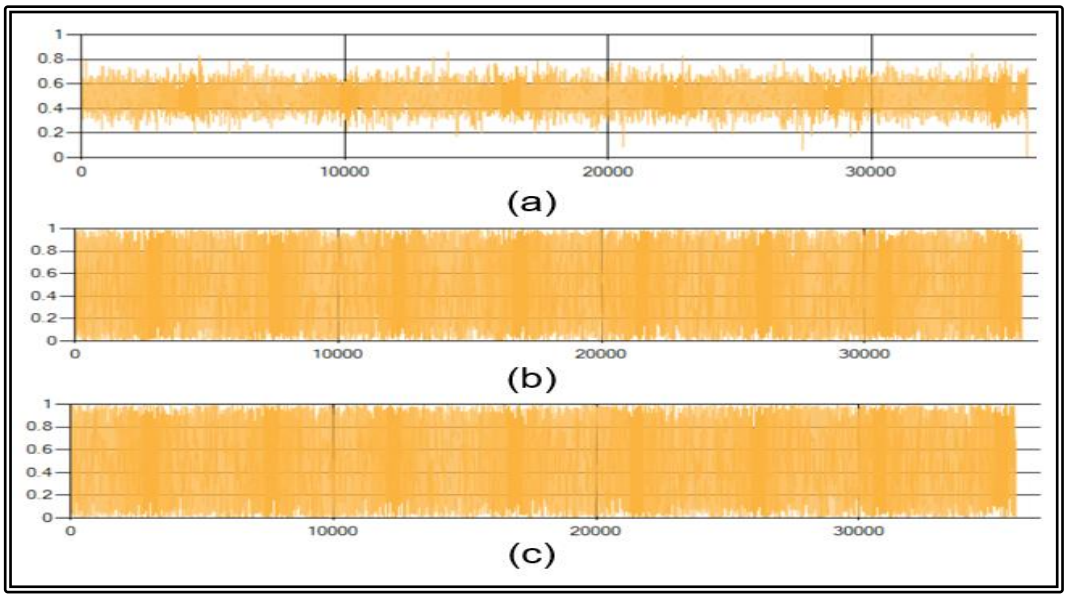



**Figure 9**-Plot of waveform decrypted with a key differ in only one bit from the original key. (a) Original waveform plot. (b) Plot of waveform decrypted with a key differ in only one bit from the original key. (c) Plot of waveform decrypted with a key differ in only one bit from the original key on another position from (b).

### 6.5.2 Exhaustive Key Search

The brute-force attack has the capability of attacking against existing types of encryption, with variant levels of success. In this kind of attack, the attackers have compromised the secret key and cipher audio as well, and they try to check each distinct secret key automatically with a help of a computer which is quick to search for the exact key faster. The brute-force attack basically starts with one digit secret key, and then goes to two-digit secret key going on until the end of secret key. In order to withstand against this kind of attacks, the secret key space should be totally large. The key space of the proposed system is $2^{128}$ for each ERC5 block of data. Thus the proposed encryption system has a large enough key space to withstand against all expected kinds of brute-force attacks.

## 7. Conclusions

This paper proposes an audio encryption system based on cellular automata with enhanced RC5 encryption algorithm. The main conclusions of this paper can be summarized as follows: The

proposed system found a new combination of CA rules in order to generate a strong RNS with a high population and large period, in addition to a high randomness. Thus, suggested PRNGs increases the security of the algorithm. Also, Enhancement of RC5 algorithm comes to strengthen the original RC5 weak keys and to harden the original RC5 in term of security and randomness by taking advantage of the proposed PRNGs based on CA, this PRNGs is used as a onetime pad key for each block of data. The proposed system is key sensitive. This sensitivity is achieved by the dependency of each block of ERC5 phase on the generated PRNGs-CA phase sub-keys. Also, the reported results of the proposed system had confirmed the positive influence of harden the power of original RC5 in both terms of the entropy and randomness tests.

**References**


1. Rhouma, R., Meherzi, S. and Belghith, S. **2009**. OCML-based colour image encryption. *Chaos, Solitons & Fractals*, **40**(1): 309–18.
2. Wang, X. and Luan, D. **2013**. A novel image encryption algorithm using chaos and reversible cellular automata. *Communications in Nonlinear Science and Numerical Simulation*, **18**(11): 3075-3085.
3. Saqib, N. A., Zia, M., Mahmood, H. and Khan, M. A. **2017**. On Generating High-Quality Random Numbers. *Journal of Circuits, Systems and Computers*, **26**(2): 1750019.
4. Faragallah, O. S. **2012.** An enhanced chaotic key-based RC5 block cipher adapted to image encryption. *International Journal of Electronics*, **99**: 925–943.
5. Shin, S. H and Yoo, K. Y. **2009**. Analysis of 2-state, 3-neighborhood cellular automata rules for cryptographic pseudorandom number generation. Computational Science and Engineering, 2009. CSE'09. International Conference on. IEEE, **1**:399-404.
6. Dogaru, R. and Dogaru, I. **2014**. Efficient and cryptographically secure pseudorandom number generators based on chains of hybrid cellular automata maps. Communications (COMM), 2014 10th International Conference on. IEEE: 1–4.
7. Mohamed, F. K. **2014**. A parallel block-based encryption schema for digital images using reversible cellular automata. Engineering Science and Technology, an International, **17**(2): 85-94.
8. Sethi, B. and Das, S. **2016**. On the Use of Asynchronous Cellular Automata in Symmetric-Key Cryptography. International Symposium on Security in Computing and Communication, Springer, Singapore, **625**: 30-41.
9. Vigneshwaran, N. **2014**. 2 Dimensional Cellular Automata Based Design of Private Key Encryption Algorithm. IJEDR| Conference Proceeding.
10. Tiago, C. **2014**. Cellular automata and cryptography. M.Sc. Thesis, University of Porto, Computer Science, Portugal.
11. Schiff, J. L. **2011**. Cellular Automata: A discrete View of the World. John Wiley & Sons.
12. Zhang, X., Lu, R., Zhang, H. and Xu, C. **2014**. A New Public Key Encryption Scheme based on Layered Cellular Automata. KSII Transactions on Internet & information Systems, **8**(10).
13. Das, D. and Ray, A., **2010**. A Parallel Encryption Algorithm for Block Ciphers Based on Reversible Programmable Cellular Automata. Journal of Computer Science and Engineering, **1**(1).
14. Seredynski, M. and Bouvry, P. **2005**. Block Cipher Based On Reversible Cellular Automata. New Generation Computing, Springer, **23**(3):245-258.
15. Mehta, R. K. and Rani, R. **2016**. Pattern Generation and Symmetric Key Block Ciphering Using Cellular Automata. Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on. IEEE: 2692-2695.
16. Singh, H. G **2016**. Selection of Parameter 'r' in RC5 Algorithm on the basis of Prime Number. Engineering and Computational Sciences (RAECS), 2014 Recent Advances in: 1-4.