Modern Applied Science; Vol. 13, No. 1; 2019
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

One-Dimensional Audio Scrambling based on Cellular Automata

Abdel Latif Abu Dalhoum[1]

[1] King Abdulla II School for Information Technology, The University of Jordan, Amman, Jordan

Correspondence: Abdel Latif Abu Dalhoum, King Abdulla II School for Information Technology, The University of Jordan, Amman, Jordan. E-mail: a.latif@ju.edu.jo

Received: Oct. 27, 2018 Accepted: Nov. 5, 2018 Online Published: December 15, 2018

doi:10.5539/mas.v13n1p136 URL: https://doi.org/10.5539/mas.v13n1p136

Abstract

Digital audio scrambling is a process used in audio security applications. Scrambling of audio files breaks the correlation between adjacent samples in order to convert the original audio to an unintelligible format. Scrambling is used to protect the audio against wiretapping and illegal surveillance, in addition to being a step in security algorithms, such as watermarking and encryption algorithms. Cellular automata are models that are discrete in nature and depend on simple and local rules to achieve an interesting overall behavior. Two-dimensional cellular automata were previously proposed as a key generation mechanism to scramble audio files. The mechanism was built upon be researchers in the multimedia security field. This paper explores the use of one-dimensional cellular automata in audio scrambling, which simplifies the process as deploying two-dimensional cellular automata requires changing the dimension of the audio file and the one-dimensional cellular automata does not, additionally, elementary one-dimensional cellular automata requires less parameters to configure. The scrambling degree is used to evaluate the model effectiveness in breaking the correlation of adjacent samples. In the experiments, different parameters are taken into account including the cellular automata class, the iterations needed and the method used to calculate the cells at the boundary. Experiments show that the one-dimensional cellular automata are capable of scrambling the audio file without any dimensional change and the chaotic rules tested give the highest scrambling degree.

Keywords: One-dimensional audio scrambling, Chaotic rules and scrambling, Multimedia security, Audio encryption

1. Introduction

Scrambling techniques were used to hide analog audio in transmission, for example, scrambling techniques were used in cable TV broadcast to protect copyrights and in transferring images from satellites to ground stations, in addition to military communications (Yan, Fu, & Kankanhalli, Progressive Audio Scrambling in Compressed Domain, 2008). In other words, scrambling was used to ensure data confidentiality just like encryption techniques today.

Nevertheless, the applications of audio scrambling have changed because of the fast development of technology and the possible serious consequences of security and privacy breaches (Alqatawna, Madain, Al-Zoubi, & Al-Sayyed, 2017). Scrambling is now used in digital and analog applications. Scrambling can be used alone, in applications such as copyright protection (Yan, Fu, & Kankanhalli, Progressive Audio Scrambling in Compressed Domain, 2008) (Fu, Yan, & Kankanhalli, 2005) and to protect from illegal Surveillance and wiretapping (Augustine, George, & Deepthi, Sparse representation based audio scrambling using cellular automata, 2014). But mostly scrambling is used as part of some security algorithm, since if used alone it provides lower security. For example, the scrambling can be used as a pre-process to audio watermarking algorithms (Li, Qin, & Shao, Audio Watermarking Pre-process Algorithm, 2009) (Hiary, Abu Dalhoum, Madain, Ortega, & Alfonseca, 2016).

In general, scrambling is similar to modern encryption in that it converts a file to an unintelligible format and usually both the scrambling and the encryption requires a secrete key to recover the original data. In some rare cases, scrambling does not require a key, such as the ITST algorithm used in audio scrambling in (Chen & Hu, 2010). Unlike encryption, the file after scrambling can still be viewed using the same software even after the scrambling process.

Cellular Automata (CA) are models of computation that are parallel and discrete in their nature. In addition, their ability to provide a chaotic or even a complex behavior from applying simple rules was proven. This is the reason

why CA are widely used in simulating and modeling complex systems, since it is possible that the simplest rules can be the main cause of different forms of randomness and complexity. Elementary cellular automata (ECA) is a special type of CA. ECA are by far the simplest CA and with significantly less varieties. There are only 256 possible rules for ECA, which made it possible to extensively study this type of CA.

All elementary CA's are one dimensional but not all one-dimensional CA's are elementary. Since One-dimensional CA's can have different neighborhood possibilities with any radius whereas elementary CA's consider only the nearest neighbors of each cell, namely, the one to its right and the one to its left.

In literature, CA variants were used in cryptography such as the game of Scintillae (Di Stefano & Navarra, 2014) and complemented CAs (Mukhopadhyay & Roychowdhury, 2007). Two-dimensional CAs were used to scramble image and audio files, for example, 2D CA with chaotic behavior was used in image scrambling (Ye & Li, 2008), and two-dimensional CA with complex behavior was used in image and audio scrambling in (Abu Dalhoum, et al., 2012) and (Madain A. , Abu Dalhoum, Hiary, Ortega, & Alfonseca, 2014), respectively. An interesting outcome of these papers depending on two-dimensional CA is the fact that, two-dimensional CAs with complex behavior presented by the game of life are better than the rules known for their chaotic behavior in terms of scrambling. Despite the simplicity of the game of life rules its evolution is essentially unpredictable (Aleksic, 2000).

This paper proposes the use of Elementary one-dimensional Cellular Automata in Audio Scrambling (ECAAS). The proposed scheme is simple and does not require any dimensional change. In addition, it is not limited to a certain audio encoding. Compared to other CA based schemes, the new scheme achieves better results with chaotic behavior. The paper studies the CA parameters such as the boundary type and the number of generations and the effect of repeating the scrambling process to the overall scrambling degree. The evaluation is made based on the scrambling degree, which can be used to measure the effectiveness of breaking the correlation between adjacent audio samples.

The organization of this paper is as follows: Section 2 presents related work; Section 3 gives an overview of cellular automata; then in section 4 the scrambling degree used in the audio case is described along with the equations used; Section 5 clarifies the steps of scrambling and gives the scrambling algorithm; Section 6 gives the experimental results along with the analysis of the results, and finally, in Section 7, the work done is concluded and future work is given.

2. Related Work

Nowadays, there are many approaches to digital scrambling. The research we discuss in this section varies in the multimedia type used (audio, image, and video) and varies in the method used for scrambling. Image scrambling techniques are studied extensively, unlike audio scrambling techniques where much less research is available. Some of the methods available in image scrambling are based on 2D Sudoku associated bijections (Wu, Zhou, Agaian, & Noonan, 2016), Arnold transform (Liu, et al., 2012), and cellular automata (Abu Dalhoum, Madain, & Hiary, 2016), (Abu Dalhoum, et al., 2012), and (Ye & Li, 2008).

Digital scramblers in general can be used to protect against wiretapping and to protect copyrights. Nonetheless, scramblers are usually used as a step in an algorithm, for example, image scramblers are used as a phase or even before and/or after some security related algorithm, such as data hiding (Parah, Sheikh, Hafiz, & Bhat, 2014) , watermarking (Wang & Li, 2015) and encryption algorithms (Li, et al., 2013), (Liu & Sheridan, 2013), (Wu, Guo, Liang, & Zhou, 2014), and (Zhong, Chang, Shan, & Hao, 2012).

In (Abu Dalhoum, et al., 2012) and (Ye & Li, 2008) two-dimensional CA were used in image scrambling, the techniques are quite similar but differs in the rules used and the behavior of these rules, the comparison given in (Abu Dalhoum, et al., 2012) shows that complex two-dimensional CA can achieve higher scrambling degree than two-dimensional CAs of chaotic behavior.

Audio files are different from image files in their content and structures and therefore they are different in the way scrambling is done and measured. There are multiple ways to compare scrambling algorithms. In general, the choice of the algorithm is based on the application and resources available. The algorithms differ in the key size, dimension, the length of the resulting audio, robustness, the audio type and the algorithms complexity.

In (Chen & Hu, 2010) the authors propose two algorithms, namely, CDST and ITST, in addition to a combination between them. All algorithms proposed in (Chen & Hu, 2010) does not use any padding and the output audio duration is equal to the input audio duration, and all algorithms does not require any dimensional change. ITST, CDST, and the combination require no, one or two integers to descramble, respectively, which is quite vulnerable to different attacks, but again if it is used as a part of an algorithm that has a proper key size, the use of these algorithms might be appropriate.

Many audio scrambling algorithms require changing the dimension of the original file and/or the scrambling key either because there is a need to map values from one dimension to the other, or the algorithm depends mainly on the dimension change for scrambling as in the work done in (Li & Qin, Audio Scrambling Algorithm Based on Variable Dimension Space, 2009) (Li, Qin, & Shao, Audio Watermarking Pre-process Algorithm, 2009).

In (Li & Qin, Audio Scrambling Algorithm Based on Variable Dimension Space, 2009) the output might need padding, and the whole algorithm is dependent on the idea that changing the dimension will result in better scrambling, but in the algorithm proposed here we show that there is no or little benefit from changing the dimension in schemes dependent on cellular automata since the simplest CA are capable of scrambling the audio effectively.

Two-dimensional CAs were also used in scrambling audio files. In audio scrambling algorithms introduced in (Madain A. , Abu Dalhoum, Hiary, Ortega, & Alfonseca, 2014), (Augustine, George, & Deepthi, Sparse representation based audio scrambling using cellular automata, 2014), (Hato, 2015), and (George, Augustine, & Pattathil, 2015), the dimension was changed since the CA used is two-dimensional and the audio file is one-dimensional. A general diagram of processes needed to convert a one-dimensional audio into a two-dimensional matrix is given in Figure 1.

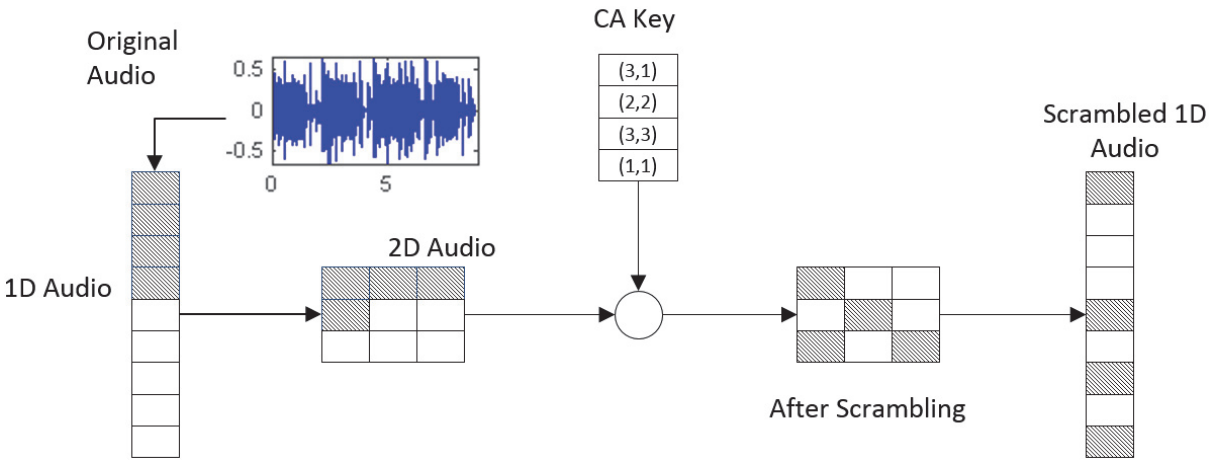


Figure 1. Scrambling Using 2D Cellular Automata

CAs are general models and might be used in any security application for any reason other than scrambling. There is a special benefit to those systems that depend on CA to apply scrambling using CA. CA design is open and flexible since any grid, cells shape, neighborhood, simple rules, boundary, finite set of states, are acceptable. CAs are capable of producing stable, periodic, chaotic and complex behavioral dynamics based on simple rules. Additionally, CAs are parallel models, which increase the performance of applications relying on it.

The proposed scheme employs a special type of cellular automata, namely, the elementary type that adds to the benefits of using cellular automata in audio scrambling, as follows:

(1) Elementary CAs are one dimensional just like the audio files, which makes the mapping simpler and more straight forward than other types of CA. ECA can scramble audio files without changing the audio or the key dimension.

(2) Elementary CAs are extensively studied since there are only 256 possible rules of ECA. It is also feasible to extensively study this type of CA in the context of audio scrambling.

(3) Elementary CAs have less parameters to configure. Work done in two-dimensions has the complications of the neighborhood and possible set of rules. The use of elementary CA eliminated such complications.

Different number of generations, boundary types, and rules were tested using the scrambling degree measure which was proposed in (Madain A. , Abu Dalhoum, Hiary, Ortega, & Alfonseca, 2014). The concept of the scrambling degree is inspired by the one used in measuring images scrambling effect proposed in (Ye & Li, 2008). Experimental results show that using ECA gives different results from those given by 2D CA regarding the rules class that can scramble audio files better.

3. Cellular Automata

CA were originally proposed as formal models of self-reproducing organisms by John von Neumann in the 1940s (Sarkar, 2000), and they are used in modeling the central dogma of molecular biology (Madain, Abu Dalhoum, & Sleit, Computational Modeling of Proteins based on Cellular Automata, 2016), (Madain, Abu Dalhoum, & Sleit, Protein Folding in the Two-dimensional Hydrophobic Polar Model based on Cellular Automata and Local Rules, 2016), (Madain, Abu Dalhoum, & Sleit, Application of local rules and cellular automata in representing protein translation and enhancing protein folding approximation, 2018) and (Madain, Abu Dalhoum, & Sleit, Computational Modeling of Proteins based on Cellular Automata: A Method of HP Folding Approximation, 2018). CA models have many advantages such as being parallel but the most interesting advantage of using these models is their simplicity and global behavior where local interactions lead to global dynamics.

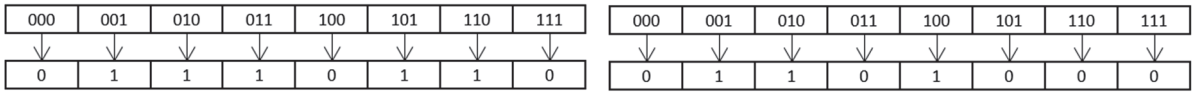
CA can be thought of as a group of cells where each cell has a state. The state should be one of a finite number of states predefined to the CA cells. Cells states may change over time, at each iteration or a CA generation, a transition function is applied, where the output of the transition function determines the new state of the cell. The transition function uses the state of the cell neighbors (and in some cases its own state) as input in order to decide the next state of the cell in the next generation.

As the size of the CA lattice is limited, there are multiple methods to calculate the next state of the cells at the boundary. One method is to use a null boundary condition, where the neighboring cells of the boundary cell are assumed to be in the state of zero. Another method is to use a periodic boundary, where it is assumed that the boundary cells at opposite sides are adjacent to one another (Shin & Yoo, 2009).

Some CA rules are capable of producing a certain global behavior, based on Wolfram (Wolfram, A New Kind of Science, 2002) the behavior of the CA can be one of four classes, the first and second classes are ordered and periodic behavior whereas the third and fourth are the chaotic and complex behavior, respectively.

Wolfram (Wolfram, Statistical mechanics of cellular automata, 1983) proposed a numbering scheme that is convenient in referring to elementary CA's. In elementary CA, the CA has the set of possible states set to {0,1} and the lattice is one-dimensional and the neighborhood considered consists of the three nearest neighbors (right, left, and the current cell state), then the number of possible neighborhood configurations is $(2^3) = 8$, and the number of possible transition functions is $(2^8) = 256$.

The rule number proposed be Wolfram in case of elementary CA is the decimal number equivalent to the binary number resulting from concatenating the output of all possible combinations of the three cells (core cell and left and right neighbors of the core cell), as shown in Figure 2.



(a) Rule 110 (b) Rule 22

Figure 2. Wolfram Numbering Scheme

4. Scrambling Degree in Audio Files

Scrambling can be defined as the disordering of a semantic piece of media in a sufficient manner (Yan & Weir, Fundamentals of Media Security, 2010). The effectiveness of the scrambling describes to what extent the scrambling is able to break the correlation of a cell and its neighborhood. The scrambling degree equation used here is the one proposed in (Madain A. , Abu Dalhoum, Hiary, Ortega, & Alfonseca, 2014) and employed in (Augustine, George, & Deepthi, Compressive Sensing Based Audio Scrambling Using Arnold Transform, 2014) (Augustine, George, & Deepthi, Sparse representation based audio scrambling using cellular automata, 2014) and (George, Augustine, & Pattathil, 2015).

The audio file amplitude and the difference between images and audio files are taken into consideration in the calculation. Assuming that each audio file sample has a value of P(i) and the length of the entire audio file is N. In order to calculate the scrambling degree, the difference at each cell is calculated first as in equation 3, so cell (i) considers the samples around it, at four neighboring positions, namely, (i-1), (i-2), (i+1), and (i+2), which are referred to in the equation as (i').

$$D(i) = \frac{1}{4}[p(i) - p(i')]^2 \quad (3)$$

The output of equation 3 is used to calculate the mean difference for the entire audio file as follows:

$$M = \frac{\sum_{i=3}^{N-2} D(i)}{N-4} \quad (4)$$

In order to calculate the scrambling degree, two values are needed. First, the original audio mean difference is calculated according to equation 4. Secondly, the audio mean difference after scrambling is calculated. Equation 5 defines the scrambling degree, where M is the mean difference of the original audio file and M' is the mean difference of the scrambled audio file. This equation results in a value ranging from -1 to 1, and higher values are considered better scrambling.

$$SD = \frac{M\prime - M}{M\prime + M} \quad (5)$$

5. Scrambling Algorithm

This section introduces the proposed algorithm, with the pseudo code and a diagram to make it easier to understand. The algorithm is easy to implement.

The input is simply the original audio file and after the scrambling process, the output will be the scrambled audio file with the key needed to regenerate the original audio file.

The algorithm begins in calculating the original audio file length. The algorithm then generates a random initial state with the same length as the length of the original audio file. ECA then starts with this initial state for a number of generations (10 generations are used in the experiments). At each generation, the indices of the resulting ones are added to the key and the indices of zero's are ignored.

The length of the resulting key is not necessarily the same as that of the audio file, but the array used to generate the key has the same length of the audio file. If the indices generated by the CA does not cover the audio file samples, the remaining samples are inserted in available positions one by one. The effect of this approach is not negative as some positions are already occupied which makes the remaining samples scattered.

The key produced by the one-dimensional CA is directly applied to the audio file without any dimensional change and the repetition of the scrambling process depends on the requirements and the needs of the application and can be considered optional as the algorithm results in a good scrambling degree even without repetition.

As scrambled files have the same format as their corresponding files, the file resulting from the scrambling algorithm is written in the same format after the scrambling process. Also, the file has the same number of bits per sample and the same sample rate of the original file. Figure 3, shows an example of an audio file and an example of a key produced by a one-dimensional CA, where the key holds new positions.

The proposed algorithm (ECAAS) can be described in the following steps:

Input: Original Audio File (X)

Output: Scrambled Audio File (R) and Key

Step 1: Read the input audio and calculate its length (N)

Step 2: Use the CA to generate the scrambling key, as follows:

1. Initialize a length N cellular automaton C_0 (start with a random configuration).
2. Initialize a length N array A (start with an array of zeros).
3. Run the transition function for a number of generations starting from C_0, where subseqent configurations are C_1, C_2, ..., C_{NOG}
4. For C= 1,2, ..., NOG, if C_C [i]=1 and A[i]=0 then add i to the key and let A[i]=1.

Step 3: Scramble the audio based on the key, where the first audio sample is moved to the first position specified by the key.

Step 4: insert the remaining samples in order, if any.

Step 5: Repeat steps three and four when needed.

Step 6: Write the scrambled audio in the same format, sample rate and number of bits per sample.

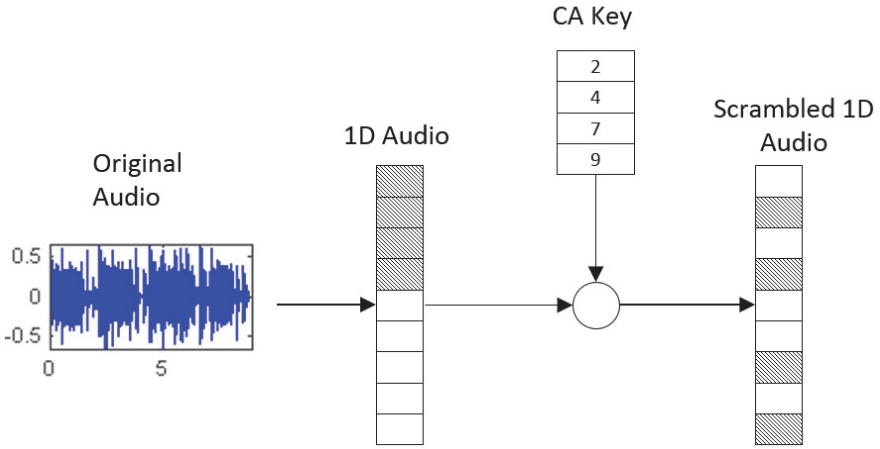


Figure 3. Scrambling Using 1D Cellular Automata.

6. Results and Analysis

There are many parameters to consider while dealing with CA, for example, the lattice characteristics, the number of generations or iterations, the rule characteristics, the boundary, the neighborhood considered and so on. There are two parameters determined beforehand in the algorithm proposed as it depends on elementary CAs. First, elementary CAs use a one-dimensional array. Secondly, elementary CAs use the simplest neighborhood of one neighbor to the left of the core cell and one neighbor to the right of the core cell.

Twenty public domain audio files were used in the experiments, some of them are speeches and the others are music. Although the repetition enhances the scrambling degree, the repetition is set to zero in all the experiments except the ones testing the repetition effect. Subsection 6.1 discusses results of deploying different CA parameters; Subsection 6.2 discusses repetition and subsection 6.3 discusses the proposed one-dimensional scheme properties.

6.1 CA Parameters

The number of generations (NOG) experiment uses the periodic boundary and rule 22. The same initial configuration is used per audio, and as mentioned before no repetition was used. The algorithm was tested for 1, 5, 10, and 15 generations and the detailed results are given in

Table 1.

Table 1. Scrambling Degree with Changing Number of Generations.

Audio/ Generations	1	5	10	15	Audio/ Generations	1	5	10	15
1.wav	0.931	0.939	0.939	0.939	11.wav	0.992	0.996	0.996	0.996
2.wav	0.811	0.874	0.874	0.874	12.wav	0.87	0.917	0.918	0.918
3.wav	0.907	0.947	0.948	0.948	13.wav	0.998	0.999	0.999	0.999
4.wav	0.913	0.928	0.928	0.928	14.wav	0.898	0.94	0.94	0.94
5.wav	0.791	0.84	0.841	0.841	15.wav	0.925	0.947	0.947	0.947
6.wav	0.853	0.894	0.894	0.894	16.wav	0.958	0.975	0.976	0.976
7.wav	0.836	0.887	0.887	0.887	17.wav	0.885	0.933	0.934	0.935
8.wav	0.916	0.945	0.945	0.945	18.wav	0.91	0.951	0.952	0.952
9.wav	0.888	0.932	0.932	0.932	19.wav	0.94	0.967	0.968	0.968
10.wav	0.86	0.91	0.91	0.91	20.wav	0.887	0.932	0.934	0.934

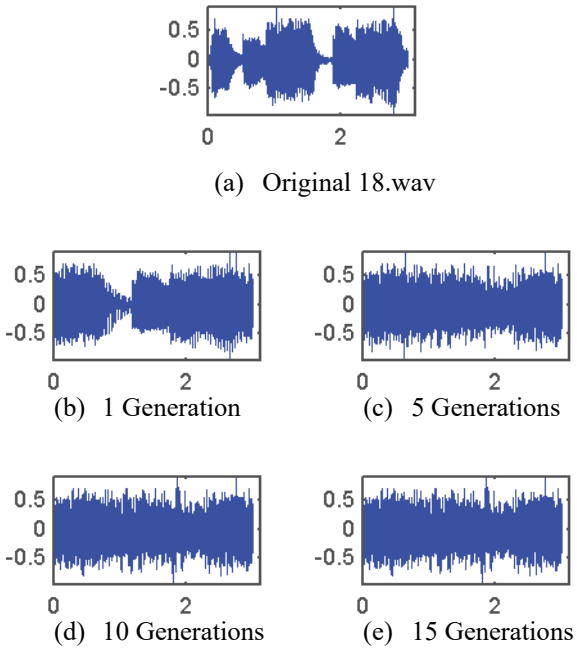
The results indicate that better scrambling is achieved when the algorithm runs more iterations in the key generation process as the more iterations the more indices are specified and the more scattered the audio samples are in the scrambled audio. If the key indices are much less than the audio length, then chances are that the remaining audio samples are inserted in neighboring positions which weakens the correlation breakage and lowers the scrambling degree.

Nevertheless, the improvement on the scrambling degree will stop when all audio samples are covered by the key. Additionally, when the key size becomes large enough, less values are inserted in order and the influence of adding iterations becomes less. In fact, it can be seen from

Table 1 that the degree stabilizes after NOG=10.

The scrambling degree ranges from 0.791 to 0.999. Figure 4 shows the audio file 18.wav and its plot after scrambling for different generations. The figure shows that scrambling for 10 and 15 generations is very similar which can be considered a reflection of the results shown in

Table 1.



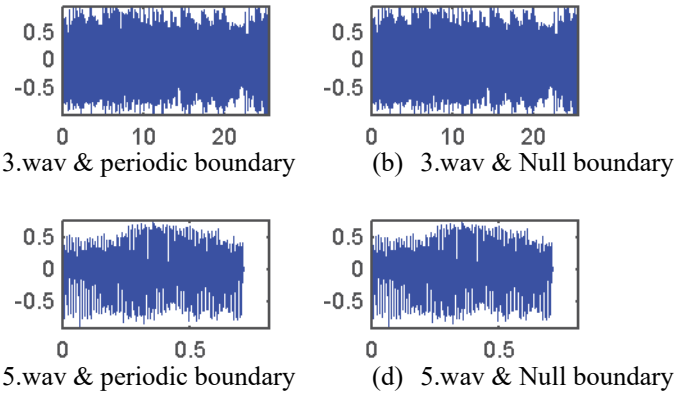
(a) Original 18.wav

(b) 1 Generation (c) 5 Generations

(d) 10 Generations (e) 15 Generations

Figure 4. 18.wav audio scrambling using ECAAS with different number of generations

The second parameter considered is the boundary. In a one-dimensional array the boundary represents two cells, one at the extreme left and the other at the extreme right. There are many options to choose from when it comes to dealing with the boundary. Two common methods are considered here, namely, the null and the periodic. After scrambling with null boundaries and periodic boundaries, the results show that the output for both is the same, taken into account the precision of three. Other parameters are fixed including the key and the number of generations (10 in this case), in order to compare the results. Figure 5 shows the result of scrambling with different boundary conditions, it can be seen from the audio that the results are very similar.



(a) 3.wav & periodic boundary (b) 3.wav & Null boundary

(c) 5.wav & periodic boundary (d) 5.wav & Null boundary

Figure 5. Scrambling with different Boundary Types

Complex rule (110) is known for its complex behavior and capability of universal computation, in studying the

effect of the rule behavior, the complex rule 110 is compared to chaotic rules (22, 30, 126, 150, 182). In (Abu Dalhoum, et al., 2012) the complex behavior of two-dimensional CA showed better results than the CA with chaotic behavior, but tests of elementary CA shows the opposite.

This result is justifiable as some chaotic CAs are studied and used in generating random numbers. The more patterns in the scrambling key, the more it is predictable. In security applications, having a scrambler that generates random numbers makes the scrambling more effective.

Table 2 shows the scrambling degree when all parameters are fixed and the rule used changes. As in the boundary experiments, the NOG=10 and the same key was used for each audio. All the rules in the table are chaotic except the complex rule 110.

Table 2. Scrambling Degree when Different Rules are Applied

Audio	22	30	126	150	182	110
1.wav	0.939	0.934	0.856	0.933	0.914	0.912
2.wav	0.874	0.856	0.705	0.855	0.808	0.818
3.wav	0.948	0.942	0.901	0.942	0.924	0.935
4.wav	0.928	0.918	0.82	0.918	0.891	0.885
5.wav	0.841	0.822	0.647	0.825	0.769	0.751
6.wav	0.894	0.889	0.826	0.886	0.836	0.858
7.wav	0.887	0.88	0.808	0.875	0.815	0.843
8.wav	0.945	0.938	0.898	0.938	0.911	0.922
9.wav	0.932	0.926	0.802	0.923	0.893	0.895
10.wav	0.91	0.908	0.843	0.907	0.88	0.879
11.wav	0.996	0.995	0.991	0.995	0.993	0.994
12.wav	0.918	0.916	0.861	0.906	0.873	0.889
13.wav	0.999	0.999	0.998	0.999	0.998	0.998
14.wav	0.94	0.938	0.897	0.937	0.912	0.922
15.wav	0.947	0.942	0.883	0.942	0.925	0.933
16.wav	0.976	0.974	0.954	0.974	0.964	0.97
17.wav	0.934	0.929	0.879	0.927	0.909	0.921
18.wav	0.952	0.946	0.915	0.946	0.932	0.942
19.wav	0.968	0.964	0.941	0.963	0.952	0.96
20.wav	0.934	0.928	0.885	0.93	0.902	0.918

Not all the chaotic rules result in good scrambling, in fact, the complex rule achieves more effective scrambling in some cases. Rule 22 gives the best scrambling degree, Figure 6 shows the results of scrambling 18.wav. The original wave plot of 18.wav is shown in Figure 4 a.

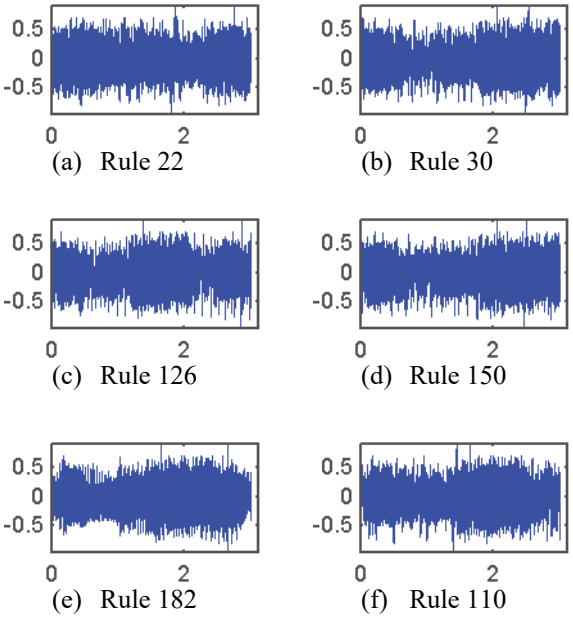
6.2 Repetition Effect

Other than the CA parameters studied so far, one other parameter that effects the scrambling degree is the repetition. The algorithm repeats the scrambling using the same key generated at the beginning so no need to rerun the CA. The repetition is not always required as it is application dependent. It can be considered optional as the scrambling degree without repetition is suitable for many applications. In addition, repeating too many times might be for little or no gain.

.

shows the repetition for K times and its effect on scrambling. K is set to 0 when there is no repetition and 1 when repeated once and 2 when repeated twice. The effectiveness of the scrambling increases when scrambling is repeated once and it is enhanced more when the scrambling is repeated twice.

Figure 7 shows scrambling of three audio files when k equals 0, 1, and 2. Although all the scrambled audio waves do not indicate the original audio, it can be seen from the figure that the scrambling with one or two repetitions gives better results in terms of covering the details of the original audio.



(a) Rule 22 (b) Rule 30

(c) Rule 126 (d) Rule 150

(e) Rule 182 (f) Rule 110

Figure 6. Scrambling based on different rules

Table 3. Scrambling Degree with Repetition

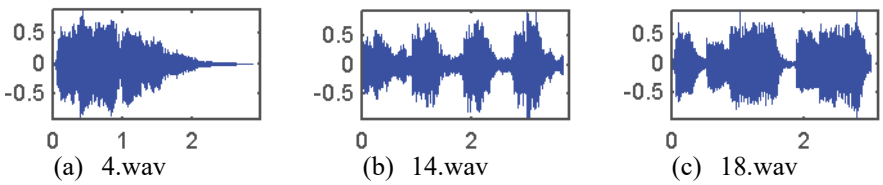
Audio	0	1	2	Audio	0	1	2
1.wav	0.939	0.958	0.962	11.wav	0.996	0.996	0.997
2.wav	0.874	0.903	0.908	12.wav	0.918	0.936	0.939
3.wav	0.948	0.958	0.96	13.wav	0.999	0.999	0.999
4.wav	0.928	0.951	0.955	14.wav	0.94	0.953	0.955
5.wav	0.841	0.885	0.89	15.wav	0.947	0.962	0.965
6.wav	0.894	0.917	0.92	16.wav	0.976	0.981	0.982
7.wav	0.887	0.909	0.913	17.wav	0.934	0.947	0.95
8.wav	0.945	0.957	0.958	18.wav	0.952	0.961	0.962
9.wav	0.932	0.946	0.95	19.wav	0.968	0.974	0.975
10.wav	0.91	0.931	0.935	20.wav	0.934	0.947	0.949

6.3 Discussion

The scrambling algorithms differ in many aspects. The scrambling algorithms are usually used as part of other security related algorithms, so sometimes some special features are required such as scrambling with no key, or a more secure scrambling with a large enough key. The following are some of the properties of the proposed scheme:

(1) The key size depends on the original audio size which is beneficial from a security point of view.
(2) No dimensional change is required.
(3) The proposed scheme is not limited to a certain audio encoding
(4) based on experiments the algorithm stabilizes after a small number of generations (5-10) and the repetition effect enhances the scrambling degree without rerunning CA.

The elementary CA scheme proposed here is simpler than the two-dimensional cellular automata scheme and doesn't require any changes in the dimension of the file or the key.



(a) 4.wav (b) 14.wav (c) 18.wav

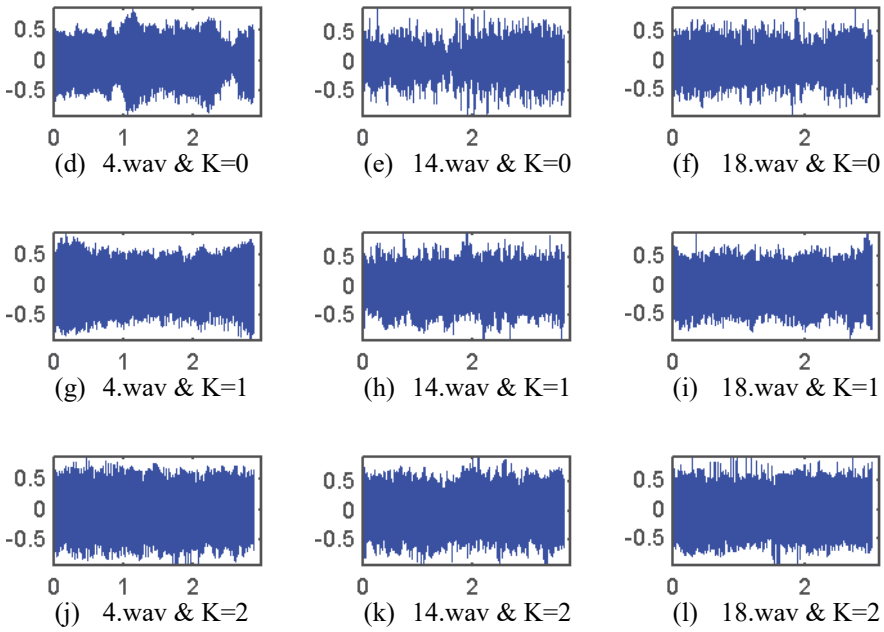


Figure 7. Repetition Effect on Scrambling

7. Conclusions and Future Work

A new cellular automata approach to audio scrambling was proposed. The approach depends on one-dimensional cellular automata which can scramble audio files effectively without any dimensional change and with the simplest neighborhood possible. Audio scrambling proposed in literature requires changing the dimension twice and dealing with more neighbors for each cell. In addition to being more efficient, scrambling using one-dimensional cellular automata is effective too as it achieved high scrambling degree. As there are many possible cellular automata parameters, the tests include a number of iterations, different methods of dealing with the boundary cells and different transition functions. Also, the effect of repetition on the scrambling degree was tested.

There are many practical applications of the scrambling approach proposed. Applying the work to different security applications is left for future work. Another interesting expansion of the work proposed is to test all possible transition functions in order to determine which rule is the best in terms of scrambling effectiveness, this is possible as the cellular automata proposed is one dimensional with the simplest neighborhood.

References

Abu Dalhoum, A. L., Madain, A., & Hiary, H. (2016, 12 01). Digital image scrambling based on elementary cellular automata. *Multimedia Tools and Applications, 75*, 17019-17034. https://doi.org/10.1007/s11042-015-2972-z

Abu Dalhoum, A., Mahafzah, B., Awwad, A., Al-Dhamari, I., Ortega, A., & Alfonseca, M. (2012, 10). Digital Image Scrambling Using 2D Cellular Automata. *IEEE MultiMedia, 19*, 28-36. https://doi.org/10.1109/MMUL.2011.54

Aleksic, Z. (2000). Artificial life: growing complex systems. In T. R. Bossomaier, & D. G. Green (Eds.), *Complex Systems* (pp. 91-126). Cambridge University Press.

Alqatawna, J., Madain, A., Al-Zoubi, A. M., & Al-Sayyed, R. (2017). Online Social Networks Security: Threats, Attacks, and Future Directions. In N. Taha, R. Al-Sayyed, J. Alqatawna, & A. Rodan (Eds.), *Social Media Shaping e-Publishing and Academia* (pp. 121-132). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-55354-2_10

Augustine, N., George, S. N., & Deepthi, P. P. (2014). Compressive Sensing Based Audio Scrambling Using Arnold Transform. (G. Martínez Pérez, S. M. Thampi, R. Ko, & L. Shu, Eds.) 172-183. https://doi.org/10.1007/978-3-642-54525-2_15

Augustine, N., George, S. N., & Deepthi, P. P. (2014, 1). Sparse representation based audio scrambling using cellular automata. *Electronics, Computing and Communication Technologies (IEEE CONECCT), 2014 IEEE International Conference on*, (pp. 1-5). https://doi.org/10.1109/CONECCT.2014.6740186

Chen, G., & Hu, Q. (2010, 7). An audio scrambling method based on combination strategy. *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, *5*, pp. 62-66. https://doi.org/10.1109/ICCSIT.2010.5564989

Di Stefano, G., & Navarra, A. (2014). The Game of Scintillae: From Cellular Automata to Computing and Cryptography Systems. *Journal of Cellular Automata, 9*, 167-181.

Fu, W.-G., Yan, W.-Q., & Kankanhalli, M. S. (2005, 5). Progressive scrambling for MP3 audio. *2005 IEEE International Symposium on Circuits and Systems*, (pp. 5525-5528 Vol. 6). https://doi.org/10.1109/ISCAS.2005.1465888

George, S., Augustine, N., & Pattathil, D. (2015). Audio security through compressive sampling and cellular automata. *Multimedia Tools and Applications, 74*, 10393-10417. https://doi.org/10.1007/s11042-014-2172-2

Hato, E. (2015, 6). Cellular Automata and Chaotic Maps for Speech Signal Encryption. *International Journal of Applied Information Systems, 9*, 9-16. doi:10.5120/ijais15-451377

Hiary, H., Abu Dalhoum, A. L., Madain, A., Ortega, A., & Alfonseca, M. (2016). Blind Audio Watermarking Technique Based on Two Dimensional Cellular Automata. *International Journal of Security and Its Applications, 10*, 175-184. doi:10.14257/ijsia.2016.10.9.18

Li, H., & Qin, Z. (2009, 4). Audio Scrambling Algorithm Based on Variable Dimension Space. *Industrial and Information Systems, 2009. IIS '09. International Conference on*, (pp. 316-319). https://doi.org/10.1109/IIS.2009.105

Li, H., Qin, Z., & Shao, L. (2009, 10). Audio Watermarking Pre-process Algorithm. *e-Business Engineering, 2009. ICEBE '09. IEEE International Conference on*, (pp. 165-170). https://doi.org/10.1109/ICEBE.2009.30

Li, H., Wang, Y., Yan, H., Li, L., Li, Q., & Zhao, X. (2013). Double-image encryption by using chaos-based local pixel scrambling technique and gyrator transform. *Optics and Lasers in Engineering, 51*, 1327-1331. https://doi.org/10.1016/j.optlaseng.2013.05.011

Liu, S., & Sheridan, J. T. (2013). Optical encryption by combining image scrambling techniques in fractional Fourier domains. *Optics Communications, 287*, 73-80. https://doi.org/10.1016/j.optcom.2012.09.033

Liu, Z., Gong, M., Dou, Y., Liu, F., Lin, S., Ahmad, M. A., … Liu, S. (2012). Double image encryption by using Arnold transform and discrete fractional angular transform. *Optics and Lasers in Engineering, 50*, 248-255. https://doi.org/10.1016/j.optlaseng.2011.08.006

Madain, A., Abu Dalhoum, A. L., & Sleit, A. (2016). Computational Modeling of Proteins based on Cellular Automata. *International Journal of Advanced Computer Science and Applications (IJACSA), 7*, 491-498. https://doi.org/10.14569/IJACSA.2016.070768

Madain, A., Abu Dalhoum, A. L., & Sleit, A. (2016). Protein Folding in the Two-dimensional Hydrophobic Polar Model based on Cellular Automata and Local Rules. *International Journal of Computer Science and Network Security (IJCSNS), 16*, 48-54.

Madain, A., Abu Dalhoum, A. L., & Sleit, A. (2018, 3 30). Application of local rules and cellular automata in representing protein translation and enhancing protein folding approximation. *Progress in Artificial Intelligence*. https://doi.org/10.1007/s13748-018-0146-8

Madain, A., Abu Dalhoum, A. L., & Sleit, A. (2018, 5 25). Computational Modeling of Proteins based on Cellular Automata: A Method of HP Folding Approximation. *The Protein Journal*. https://doi.org/10.1007/s10930-018-9771-0

Madain, A., Abu Dalhoum, A., Hiary, H., Ortega, A., & Alfonseca, M. (2014). Audio scrambling technique based on cellular automata. *Multimedia Tools and Applications, 71*, 1803-1822. https://doi.org/10.1007/s11042-012-1306-7

Mukhopadhyay, D., & Roychowdhury, D. (2007). Theory of a Class of Complemented Group Cellular Automata and its Application to Cryptography. *Journal of Cellular Automata, 2*, 243-271.

Parah, S. A., Sheikh, J. A., Hafiz, A. M., & Bhat, G. M. (2014). Data hiding in scrambled images: A new double layer security data hiding technique. *Computers & Electrical Engineering, 40*, 70-82. https://doi.org/10.1016/j.compeleceng.2013.11.006

Sarkar, P. (2000, 3). A Brief History of Cellular Automata. *ACM Comput. Surv., 32*, 80-107. 10.1145/349194.349202

Shin, S.-H., & Yoo, K.-Y. (2009, 8). Analysis of 2-State, 3-Neighborhood Cellular Automata Rules for Cryptographic Pseudorandom Number Generation. *Computational Science and Engineering, 2009. CSE '09. International Conference on*, *1*, pp. 399-404. https://doi.org/10.1109/CSE.2009.299

Wang, T., & Li, H. (2015). A novel scrambling digital image watermarking algorithm based on contourlet transform. *Wuhan University Journal of Natural Sciences, 19*, 315-322. https://doi.org/10.1007/s11859-014-1019-z

Wolfram, S. (1983, 7). Statistical mechanics of cellular automata. *Rev. Mod. Phys., 55*(3), 601-644. https://doi.org/10.1103/RevModPhys.55.601

Wolfram, S. (2002). *A New Kind of Science.* Champaign, Ilinois, US, United States: Wolfram Media Inc.

Wu, J., Guo, F., Liang, Y., & Zhou, N. (2014). Triple color images encryption algorithm based on scrambling and the reality-preserving fractional discrete cosine transform. *Optik - International Journal for Light and Electron Optics, 125*, 4474-4479. https://doi.org/10.1016/j.ijleo.2014.02.026

Wu, Y., Zhou, Y., Agaian, S., & Noonan, J. P. (2016). 2D Sudoku associated bijections for image scrambling. *Information Sciences, 327*, 91-109. https://doi.org/10.1016/j.ins.2015.08.013

Yan, W. Q., & Weir, J. (2010). *Fundamentals of Media Security.* Denmark: Ventus Publishing ApS.

Yan, W. Q., Fu, W. G., & Kankanhalli, M. S. (2008, 10). Progressive Audio Scrambling in Compressed Domain. *IEEE Transactions on Multimedia, 10*, 960-968. 10.1109/TMM.2008.2001373

Ye, R., & Li, H. (2008, 8). A Novel Image Scrambling and Watermarking Scheme Based on Cellular Automata. *Electronic Commerce and Security, 2008 International Symposium on*, (pp. 938-941). https://doi.org/10.1109/ISECS.2008.138

Zhong, Z., Chang, J., Shan, M., & Hao, B. (2012). Double image encryption using double pixel scrambling and random phase encoding. *Optics Communications, 285*, 584-588. https://doi.org/10.1016/j.optcom.2011.11.025