Hiding Encrypted Images in Audios Based on Cellular Automatas and Discrete Fourier Transform

Jose Alva Cornejo, Esdras D. Vasquez, Jose Calizaya Quispe, Roxana Flores-Quispe, Yuber Velazco-Paredes
Universidad Nacional de San Agustín de Arequipa, Arequipa, Perú

***Abstract*—With the increasing need for secure long-distance communication, protecting sensitive information such as images during transmission remains a significant challenge. This paper proposes a new method for hiding encrypted images inside audio files by integrating Cellular Automata (CA) and the Discrete Fourier Transform (DFT). The primary aim is to enable secure transmission of large encrypted images without altering the audio's perceptual quality. The scheme leverages the cryptographic properties of CA to generate encrypted images, which are then embedded into inaudible frequencies of audio using DFT. Results show that this method successfully hides and recovers images of considerable size, maintaining bit-level integrity of the original images while preserving audio quality. However, the scheme lacks resilience to signal processing attacks, such as compression or filtering, the resulting size of the audio is also bigger. Despite this limitations, the method provides a competitive advantage in payload capacity and efficiency, making it suitable for applications where the transmission of large, sensitive data is necessary but not subject to aggressive signal attacks.**

***Keywords*—Cellular automaton; Fourier Transform; cryptography; synchronization; stenography; embedding**

I. Introduction

Nowadays with the development of information and communication technologies, access to information has become easier and establishing communication in a secure way has become a necessary requirement [1]. As a result, people can easily exchange information and distance is no longer a barrier to communication. However, the safety and security of long-distance communication remains an issue [2], because in many cases, the Internet is being affected by hackers.

For that reason, attempts have been made to provide a cyber security environment to protect the assets of institutions, organizations, and individuals such as encryption systems, watermarking, steganography, fingerprinting, hybrid systems [3].

It is therefore essential to investigate more secure and efficient methods for safeguarding sensitive data, such as images, during transmission over open channels.

In the case of steganographic techniques, many were proposals to provide secure data exchange through an open communication channel. These approaches are mainly hosted under three domains: In spatial domain techniques [4] [5], the data is hidden, and replacement is directly applied to the pixels of the image; Transform Domain Methods hide the messages in significant areas of the cover image to produce more efficient stego-images. It manipulates the image indirectly by various transformation techniques; the most popular of these techniques are: Discrete Cosine Transformation (DCT) and Discrete Wavelet Transformation (DWT); and the third domain considers hybrid domain techniques; which is a type of steganography where spatial and transform domains may be combined. The hybrid approaches also provide some security and capacity enhancements but still are in their beginnings and need more research. [6].

In other cases, Cellular Automata (CA) models have been used for their good cryptographic properties that provide security against attacks and better confusion and diffusion properties [7]. CA models also give a secret key for the encryption which cannot be predicted since it evolves into a chaotic and complex system starting from an initial state [8].

So the question arises How can encryption methods be integrated with a CA cellular automata to hide encrypted images in audio files to improve transmission security? With all that has been seen, it was asked, how can encryption methods be integrated with a well-encryption system like CA with steganography to improve the security of encrypted image transmission?

For that reason, this paper proposes a novel method to increase security in image transmission: A hybrid method to encrypt high quality images using CA and hide them inside audio files using the Discrete Fourier Transform (DFT) based on the methods proposed by Alvarez et al. [9] and Hwai-Tsu and Tung-Tsu [10] for improved security in hiding encrypted images. This method could be of great use in industries that require the protection of sensitive information, such as the financial, military or healthcare sectors, where data integrity and confidentiality are paramount.

The rest of the paper is organized as follows: The review of previous works is presented in Section II, the proposed scheme is described in Section III, where a description of fundamental concepts is made and then the detailed description of each step in encryption and decryption is presented, in Section IV the results are shown along with their respective analysis, the discussion is presented in Section V and it ends with the conclusion in Section VI.

II. Related Work

In recent years, significant efforts have been made to solve the problem of information security in data transmission, such as the work done by Alvarez et al. [9], who have proposed a scheme based on the bidimensional reversible CA with memory. These schemes are cryptographic procedures to share a secret among a set of participants in such a way that only some qualified subsets of these participants can recover the secret. Also, the security of the scheme is studied and it is proved that the protocol is ideal and perfect and also resists the most important statistical attacks. To validate the

protection of the original information, the number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI) randomness test were used, with scores of 99 and 33 respectively, which would indicate a high level of change in the encrypted image compared to the original. This suggests a robust encryption that is very sensitive to changes in the original image; in other words, a minimal modification in the original image will cause a noticeable difference in the encrypted image, which is positive for the security of the encryption.

Similarly, Hwai-Tsu and Tung-Tsu [10] have proposed a study to introduce an innovative phase modulation (PM) scheme based on the fast Fourier transform (FFT) that facilitates efficient and effective blind audio watermarking. The results reflected the robustness of phase modulation against a variety of common signal processing attacks, and comprehensive and rigorous tests confirmed the PM's robustness against a variety of common signal processing attacks, including resampling, requantization, and low pass filtering. But the FFT–PM was less resistant to attacks that caused severe phase perturbations.

Likewise, Eslami et al. [11] proposed a model using CA and a double authentication mechanism to propose a new threshold image sharing scheme with steganographic properties. That proposed scheme uses 2 bits in each pixel of cover images for embedding data and so a better visual quality for the produced stego images was achieved. Consequently, this study got a Peak Signal-to-Noise Ratio (PSNR) value of 48, showing that the difference between the original image and the compressed or encrypted image is minimal and that the loss of quality is practically imperceptible to the human eye.

Similarly, Hernández et al. [12] proposed a new graphic symmetrical cryptosystem in order to encrypt a colored image defined by pixels and by any number of colors. This cryptosystem is based on a reversible bidimensional CA and uses a pseudo-random bit generator where the session key is the seed used to generate the pseudorandom bit sequence. In consequence, the decrypted image is identical to the original, i.e., no loss of resolution occurs.

On the other hand, Tanwar & Bisla [13] understood that the goal of audio steganographic technique is to embed data in audio cover files that must be robust and resistant to malicious attacks. That paper presents various audio steganographic methods like LSB, echo hiding, spread spectrum etc. Also, merits and demerits of each method are described. Finally, they showed that the low bit coding (LSB) method has a low robustness, echo masking, and phase coding have a low capacity for data embedding; in relation to the spectrum, it has a higher robustness but is vulnerable with respect to the modification of the time scale.

Abdirashid, Solak & Saku [14] argued that image steganography techniques provide better data embedding capability. So, they proposed secure data hiding algorithms based on frequency domain in image steganography. The methods were evaluated according to the criteria of imperceptibility, payload capacity and robustness, where they obtained good results of PSNR of 50 dB and Structural Similarity Index Measure (SSIM), which represents a successful restoration for the same image.

Also noteworthy is the comparison of different types of encryption realized by Louis [15], which tests the DCT and DFT encryption techniques in order to compare which of the two is better where it is seen that even though the images are embedded using bytes complete directly into the Fast Fourier Transform (FFT) transformed dimension, the byte difference is distributed somewhat evenly over the entire image, with small differences of the pixel values. That method is thus harder to detect than embedding methods that directly use the spatial domain.

An interesting paper was realized by Najiya and Renjith [16] who describe a method of compressing the image using wavelet compression and converting it into a bit sequence in order to embed it in the modified cover audio using a secret key, then the audio is encoded with an error correction code to improve its robustness of this technique. Where at the receiver section, the original secret image is reconstructed successfully. The PSNR value of the image in the system is 32.84. This means that the difference between the original image and the altered image is small.

The review of related works shows that many of these methods focus on information protection using different techniques in the spatial domain, transform domain and hybrid domain, this encouraged further research and improvement in this field.

III. Proposed Method

Fig. 1 shows the representation of the proposed method hiding encrypted images by CA in audios based on DFT. And, Fig. 2 shows the process to decrypt the information.

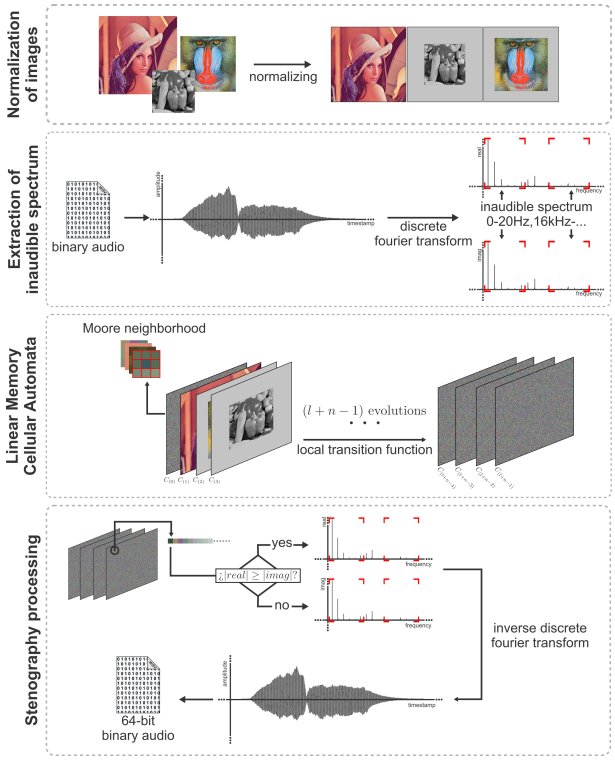

Fig. 1. Encryption of the proposed scheme.

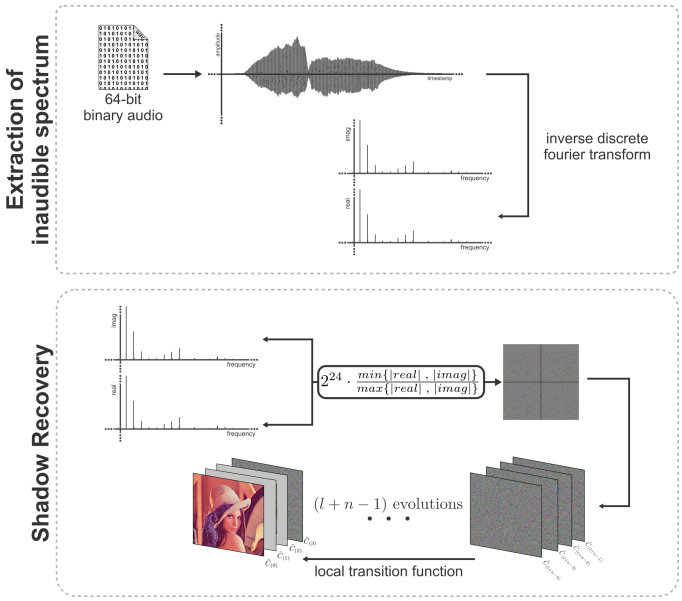

Fig. 2. Decryption of the proposed scheme.

A. Normalization of Images

This first stage is considered essential, because the operation of the CA depends on its environment, i.e. "neighborhood", based on the number of cells given by the number of rows and columns, which is simply feasible for any case. However, the proposal is to hide images by means of these evolutions, that means that the space operated by the CA must be the same in all cases of evolutions, by adding white pixels around each image if necessary.

Therefore, from a set of RGB images of different sizes, they are normalized by adding a padding of white pixels so that they have the same dimensions, Fig. 3 shows an example, not including the borders.

(a) (b) (c) (d)

Fig. 3. Normalized images.

B. Extraction of Inaudible Spectrum

Although the CA gives the necessary images to be able to reverse the process, these images could have essentially noise, which can generate suspicion, therefore, a higher stage of encryption is proposed, which means hiding it inside an audio.

The method supports the reading of different types of audio in wav format $(16-bit\ PCM,\ IEEE\ float)$, internally all processing is done in $64-bit$ float format, so any conversion from an audio with a smaller data format does not result in any alteration of the original audio [17]. Audio recorded at sampling rates greater than or equal to $44100Hz$ are used, because this is the standard used in music CDs and provides good frequency precision [18]. The output audio will be a $64-bit\ IEEE\ float\ .wav$ file.

On the other hand, of all the forms of audio steganography that exist, the decision has been to use the algorithm called "Fast Fourier Transform" or FFT, to hide data in the less perceptible frequencies: from 0 to $20Hz$ and from $16kHz$ [19].

Where, for each audio channel, it will be divided into M segments of length L each one. Using the DFT, each segment frequency is obtained using the Eq. 1.

$$X^{(m)}(k) = \sum_{n=0}^{L-1} x^{(m)}(n) \mathrm{e}^{-\mathrm{i}2\pi k \frac{n}{L}} \tag{1}$$

where $m = \{0 \dots M\}$
$k = \{0, \dots, 20, 16000, 16001, \dots \frac{L}{2}\}$
* Illustrative quantity, take into account the frequency resolution.

However, the nature of the FFT algorithm will return $N/2$ samples from $N = 2^n, n \in \mathbb{R}^+$ samples. Therefore it is set that $L = 65536 = 2^{16}$ to ensure that a wide range of frequencies is obtained. That results by the concept called frequency resolution, which is shown in Eq. 2.

$$\Delta f = \frac{f_s}{N} \tag{2}$$

f_s : Frequency sample rate
N : Number of samples

$$e.g: \ \Delta f = \frac{44100}{65536} \approx 0.67$$
$$\Delta f = \frac{44100}{32768} \approx 1.34$$
$$\Delta f = \frac{44100}{16384} \approx 2.69$$

$\therefore$ More sample gets more resolution

Then the "frequency resolution" is used to put the encrypted data into an array, in this way it can approximate a 1:1 relationship between the indices of an array and frequencies it want to access.

Additionally, a function is used to access a frequency by indices of an array, which is shown in Eq. 3:

$$freqToIndex(freq) = \lfloor \frac{freq}{2^{16}} \rfloor, \quad freq \geq 0 \tag{3}$$

C. Linear Memory Cellular Automata

CA is a computational model that simulates dynamic systems and processes within a concrete space composed of cells. These cells have different states that vary in relation to time due to predefined rules. Each cell of a CA is composed of a dynamic state in relation to its neighboring cells by means of specific rules, which define the state configuration, simulating the evolution of a system over multiple continuous time steps [9].

According to [9], the following elements define a CA:

- State It is the set S of possible values that a cell will have. This paper makes use of RGB images leads to a $S = \mathbb{Z}_{2^{24}}$, it is the minimum value containing all possible values of an RGB pixel.
- Local transition function Function that gives a new state to the cell (i, j) from its neighborhood V_{ij}, using a number ω of 9 bits defined as a rule. It is defined in the following way:
$$s_{ij}^{(t+1)} = F(V_{ij}^{(t)}, ..., V_{ij}^{(t-n)})$$
- Neighborhood It is defined as the set of neighbors of a cell. Therefore, in this paper the Moore neighborhood is used.

Instead of working on a one-dimensional line, two-dimensional CA are organized in a grid or matrix of cells, where each cell can have a particular state, and offers a way to model and understand complexity starting from simple local rules. Their versatility and ability to represent a wide variety of events make them a valuable tool [20].

The change of the evolutions depends on the states previous to these since are based on the $k - th$ order, and each of them defines a new change of states of the CA based on the k previous evolutions. For example, if k takes the value of 3, then the new state in $(t + 1)$, depends on the states of (t), $(t - 1)$ and $(t - 2)$.

Now, the security of the evolutions to be performed in the Linear Memory Cellular Automata (LMCA) depends on an image composed of random pixels, indicating dimensions and other characteristics, see Fig. 4. This will be considered as configuration 0, C_0 while the images to be encrypted will be the images shown in Fig. 3: C_1 C_2 C_3 C_4 respectively. Based on the fact that this has $n = 4$ initial images, along with an initial configuration, the order of the cellular automaton would be 5; which implies that the 5 previous ones will always be used to generate the next one (times), and so on until the number of evolutions [9][21].

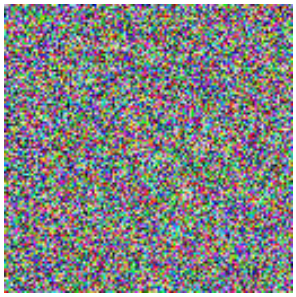

Fig. 4. Configuration C_0.

For each time it have, w is a random number. According to the model of an LMCA, a random number l is necessary, which determines the number of evolutions defined as $\#evo = n + l - 1$, for the present investigation the value of l will be equal to 10 ($l = 10$), the results of which are shown in Fig. 5.

At the moment of making the evolutions, it can be observed how images of what is apparently noise (shadows) are created,

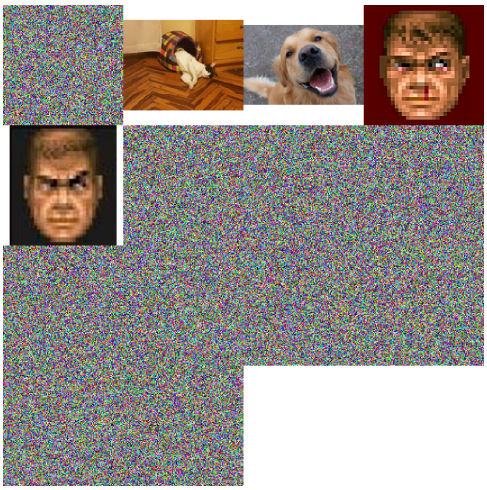

Fig. 5. Evolutions with a value of l=10.

and according to [11], the correlation between shadows is very small, which guarantees that it is not possible to extract relevant information from the original images by having only a fraction of the shadows. The LMCA has a set of times, because it used is a k to k symmetric scheme, the keys for the recovery will be composed by the same number of initial configurations as it had. In the example, the last four noise images would be presented together with the one preceding them. The latter would serve as a public key, while the other four are secret.

D. Stenography Process

It is widely acknowledged that the human auditory system exhibits relatively low sensitivity to variations in phase [22]. For this reason, the key to imperceptible audio primarily involves manipulating the FFT phase. Since each FFT coefficient comprises a real and an imaginary component, this scheme for numerical embedding involves the manipulation of the ratio between the magnitudes of these two components. The component with the largest magnitude was identified as the baseline unit. Consequently, the extent of the other component is modulated based on the intended numeric value, such as a pixel value extracted from a color image. Now, the Eq. 4 is derived from the equation proposed by [10]. This equation allows encrypting an entire RGB color.

$$\begin{aligned}
&\text{if } |\text{Re}\{X^{(m)}(k)\}| \geq |\text{Im}\{X^{(m)}(k)\}| \\
&\quad \hat{X}^{(m)}(k) = \text{Re}\{X^{(m)}(k)\} \\
&\qquad + i{\cdot}sgn(\text{Im}\{X^{(m)}(k)\}) \cdot |\text{Re}\{X^{(m)}(k)\}| \cdot \frac{v(\Delta + k)}{2^{24}} \\
&\text{else} \\
&\quad \hat{X}^{(m)}(k) = sgn(\text{Re}\{X^{(m)}(k)\}) \cdot |\text{Im}\{X^{(m)}(k)\}| \cdot \frac{v(\Delta + k)}{2^{24}} \\
&\qquad + i \cdot \text{Im}\{X^{(m)}(k)\}
\end{aligned} \tag{4}$$

where $v(x)$ refers to the value of the ID array that contains the pixels of an image.
$\Delta = m(L - 16020)$ storage capacity in a block m.

Once the values of each pixel of an image have been positioned inside the inaudible frequencies of an audio, the Inverse Fourier Transform (see Eq. 5) is operated to obtain an audio that will be saved in a 64-bit .wav file in order to safeguard the precision of the mathematical operations and the values of this new audio.

$$\hat{x}^{(m)}(k) = \frac{1}{L}\sum_{n=0}^{L-1} \hat{X}^{(m)}(n)\mathrm{e}^{\mathrm{i}2\pi k\frac{n}{L}} \tag{5}$$

This whole process is shown in Algorithm 1.

Algorithm 1 Encryption

Require: D is a vector of data to encrypt
Require: A is a $m \times n$ matrix where

$$m = \begin{cases} 1 & \text{if audio channel is mono} \\ 2 & \text{if audio channel is stereo} \end{cases}$$

and $n = Length\ of\ audio$

Ensure: fft(...) returns an array in CCs format
  $res \leftarrow A\ m \times n\ matrix\ with\ encrypted\ data$
  $samples \leftarrow 2^{16}$
  $encrypted \leftarrow 0$ ▷ Progress
  for all $signal$ **in** A **do**
    $segment \leftarrow 0$
    while $segment \leq |signal|$ **and** $encrypted \neq |D|$ **do**
      $out \leftarrow$ fft$(signal[segment{:}])$ ▷ out is a vector of $\frac{samples+1}{2}D$
      $out \leftarrow \frac{out}{samples}$ ▷ Normalize
      for $i \leftarrow 1$ **to** $|out| - 1$ **and** $encrypted \neq |D|$ **do**
        if $i = freqToIndex(20.0)$ **then** ▷ See equation 3
          $i \leftarrow freqToIndex(16000.0)$
          continue
        end if
        $x \leftarrow out[i]$
        if $|real(x)| \geq |imag(x)|$ **then** ▷ See equation 4
          $imag(x) \leftarrow sgn(imag(x)) * |real(x)| * \frac{D[encrypted]}{2^{24}}$
        else
          $real(x) \leftarrow sgn(real(x)) * |imag(x)| * \frac{D[encrypted]}{2^{24}}$
        end if
        $encrypted \leftarrow encrypted + 1$
      end for
      $out\_inv \leftarrow$ ifft(out)
      $copy\ out\_inv\ into\ res$
    end while
    $copy\ signal[segment{:}]\ to\ res[signal]$ ▷ Remainder
  end for
  return res

E. Shadow Recovery

Similar to the previous step, the recovery is based on separating the audio stego into segments of 2^{15} audio samples, where in each one the FFT is applied to recuperate the values of the frequencies where the data is saved.

$\tilde{X}^{(m)}(k)$ is define as the sequence of values obtained by using the FFT in a block; the same variables defined in the previous step are shown by Eq. 6.

$$v(\Delta + k) = 2^{24} \cdot \frac{min\{|\mathrm{Re}\{\tilde{X}^{(m)}(k)\}|\ ,\ |\mathrm{Im}\{\tilde{X}^{(m)}(k)\}|\}}{max\{|\mathrm{Re}\{\tilde{X}^{(m)}(k)\}|\ ,\ |\mathrm{Im}\{\tilde{X}^{(m)}(k)\}|\}} \tag{6}$$

After getting the pixel array, they are saved for later use in the CA.

Continuing, the images have been recovered from the audio files and a process identical to the first evolution is developed where the public key will be the configuration 0, and the following shadows are added in the order they were generated, by effecting the number of evolutions with the same value l, the original images are obtained (with their respective padding. See Fig. 6 and 7.

This whole process is shown in Algorithm 2.

Algorithm 2 Decryption

Require: A is a $m \times n$ matrix where

$$m = \begin{cases} 1 & \text{if audio channel is mono} \\ 2 & \text{if audio channel is stereo} \end{cases}$$

and $n = Length\ of\ audio$

Require: $size$: The desired size of the decrypted data
Ensure: fft(...) returns an array in CCs format
  $res \leftarrow$ Array of decrypted data
  $decrypted \leftarrow 0$ ▷ Progress
  for all $signal$ **in** A **do**
    $segment \leftarrow 0$
    while $segment + 2^{16} < |signal|$ **and** $decrypted \neq size$ **do**
      $out \leftarrow$ fft$(signal[segment{:}])$
      $out \leftarrow out/samples$ ▷ Normalize
      for $i \leftarrow 1$ **to** $|out| - 1$ **do**
        if $decrypted = size$ **then**
        end if
        if $i = freqToIndex(20.0) + 1$ **then**
          $i \leftarrow freqToIndex(16000.0)$ **continue**
        end if
        $x \leftarrow out[i]$
        $v \leftarrow \lfloor \frac{min\{|real(x)|,|imag(x)|\}}{max\{|real(x)|,|imag(x)|\}} * 2^{24} \rfloor$ ▷ See Eq. 6
        $res.append(v)$ ▷ Store decrypted value
        $decrypted \leftarrow decrypted + 1$
      end for
    end while
  end for
  return res

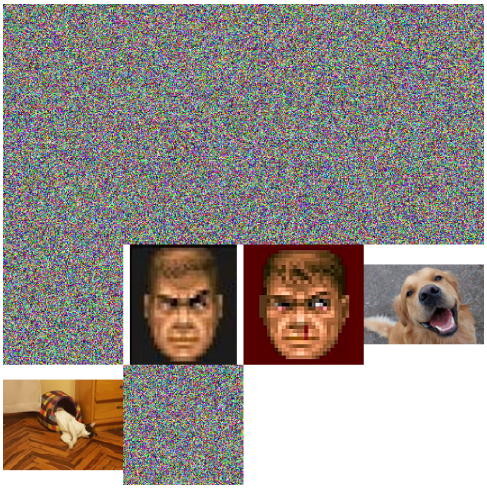

Fig. 6. Inverse evolution.

Fig. 7. Images recovered.

IV. Results

A. Random Noise NIST Test Results

To ensure that the secret images are indeed noise, the NIST statistical test suite is used, it consists 15 different tests that measure the randomness of a sequence of bits. For each test a p-value is calculated, which varies from 0 to 1, where 0 indicates that the bits are non random and 1 perfect randomness [23].

To test the first step of the scheme, a large image(4000x3000 pixels) is used to generate the appropriate noise images to hide. For this example, nine evolutions are created; and then, for each images, a binary file composed entirely of the pixel RGB bytes is made.

Each of the nine files passed to the NIST suite contain 288 million bits and are of size 36MB. The results in Table I give them an average p-value of a little under 0.5, which shows good randomness. There were only a couple of failed tests in the earlier evolutions, however, it can be concluded that the secret images to be hidden in an audio file will be very close to random noise.

TABLE I. Results of Randomness Test

Evolution	Average p-value	Passrate
1	0.467008	99.47%
2	0.495432	100.00%
3	0.481725	98.94%
4	0.471555	99.47%
5	0.478481	100.00%
6	0.502455	100.00%
7	0.457972	100.00%
8	0.474596	100.00%
9	0.499604	100.00%

B. Size and Capacity of Audio Files

Because the process of hiding images in audios using the FFT internally uses $64 - bit$ floating numbers, it cannot save the frequency values in audio formats such as $16 - bit$ fixed-point and $32 - bit$ floating-point without losing data in the truncation process, which forces to save $64 - bit$ IEEE wav audios. See Table II to view the comparison of the created audios.

TABLE II. Size Analysis of Created Audios

	Audio1	**Audio2**	**Song1**	**Song2**
Duration	4s	7s	6:23 min	2:49 min
Orig Size	812.6 KiB	1.5 MB	64.5 MB	28.5 MB
64bit Size	3.2 MB	5.8 MB	257.9 MB	114.1 MB
Increase	293.798%	286.667%	299.845%	300.351%
Max Capacity (NxN)	256	330	2153	1434

It can be observed that the created audios are about four times larger, but it allows them to save an RGB image of considerable size, or in the case of smaller images, a set of these.

C. Spectrograms

By performing a spectrogram analysis of both the original audio (see Fig. 8) and the one containing their payload (see Fig. 9), it can be observed how it proceeds according to the proposed method, where only the less audible frequencies have been covered by what amounts to noise.

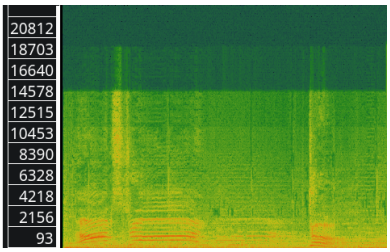

Fig. 8. Spectrogram of original audio.

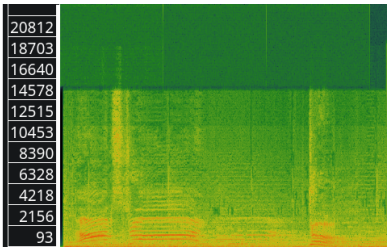

Fig. 9. Spectrogram of stego-audio.

In a second sample only a 128×128 pixel image has been inserted, it can be observed that only a section of the high frequencies is modified, this due to the fragmentation of the audio for the realization of the process, see Fig. 10 and 11.

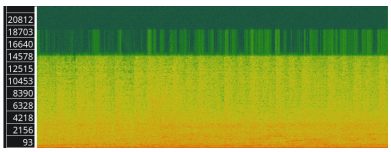

Fig. 10. Spectrogram of second original audio.

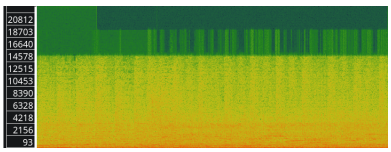

Fig. 11. Spectrogram of second stego-audio.

D. Measuring Similarity to Original Audio

1) Dynamic Time Warping: As a way of measuring how similar the payloaded audio is to the original one, the algorithm

Dynamic Time Warping (DTW) has been used. DTW is a really powerful tool with uses way beyond just measuring differences in audio files, such as speech and sign language recognition, computer vision, animation, data mining, music and signal processing [24]. DTW compares two signals that may or may not be of the same length to find an optimal alignment with minimal cost, to do this a cost matrix is found, by comparing every value of both signals, were the higher the difference between values, the higher the cost. After obtaining the matrix, the optimal warping path is the one with the lowest accumulated cost from traversing from bottom left to top right [25]. Because the signals to be compared are practically of the same length, the alignment path will always be a straight line from bottom to top.

To find a measurement related to the similarity of both audio signals, two comparisons have been decided upon. The first comparison uses DTW on the original audio and the stego audio. The second comparison is performed between the original signal and a noisy signal, which is generated by adding almost imperceptible random noise to the original. This approach provides a baseline for assessing how different the stego signal is.

Fig. 12 and 13 show the cost matrix of both comparisons, where orange tones show a greater cost. The shortest path is always the direct one, as a better way of showing the differences of the comparison, Fig. 14 and 15 shows a slice of the cost matrices, but plotted as a 3d bar chart, the first comparison presents lower values overall and by calculating the distance cost of the optimal warping path. The values obtained are 1528.1893 for the stego audio and 82826.3947 for the noisy signal. This indicates that the stego audio signal is easier to adapt and correct, and thus very similar to the original audio.

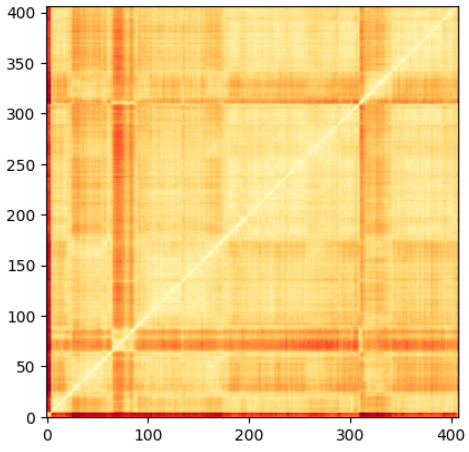

Fig. 12. DTW cost matrix of stego-audio comparison.

2) Error Testing: An evaluation will be made with the Mean Square Error (MSE) and Signal-to-Noise Ratio (SNR) coefficients to see the efficiency of the proposed method, by the use of the Fourier transform to audio.

In the MSE, the error signal $e_i = x_i - y_i$ represents the difference between the original and distorted signals [26], which will be evaluated with respect to audio quality, after performing data hiding within the audio during the transform. See Eq. 7.

$$MSE = \frac{\sum(O - E)^2}{N} \tag{7}$$

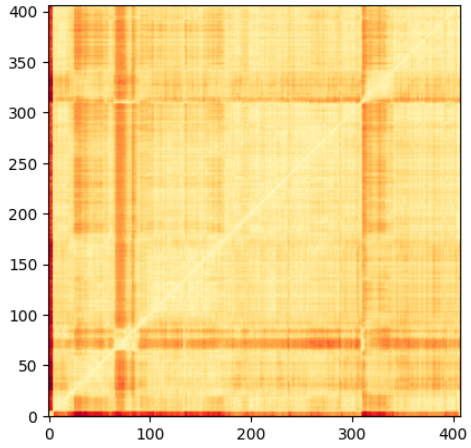

Fig. 13. DTW cost matrix of noisy audio comparison.

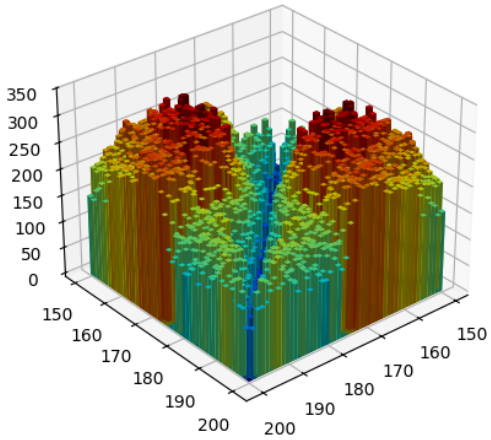

Fig. 14. 3D bar chart of DTW cost matrix of stego-audio comparison.

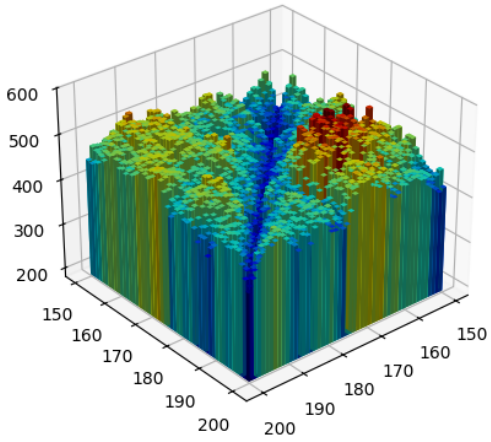

Fig. 15. 3D bar chart of DTW cost matrix of noisy-audio comparison.

On the other hand, SNR reflects the signal-to-noise relation of an audio, usually written as S/N, is an estimate of the robustness of the source signal in relation to the possible noise (unwanted signal) [27], see Eq. 8

$$SNR = 10 * \log_{10}(\frac{O^2}{MSE}) \tag{8}$$

After several comparisons of encrypted and decrypted audios, The resulting values are shown in the Table III.

TABLE III. ANALYSIS OF ORIGINAL AND ENCRYPTED AUDIO

Sample Indices	Original (O)	Encrypted (E)	(O^2)	(O - E)^2
0	0	-0.0048	0.0048000	2.28 $\times 10^{-5}$
6500	0.2461	0.2510	0.060500	2.44 $\times 10^{-5}$
13000	0.0080	0.0113	0.0000634	1.13 $\times 10^{-5}$
19500	-0.1737	-0.1757	0.0302000	4.11 $\times 10^{-6}$
26000	0.1949	0.2014	0.0380000	4.26 $\times 10^{-5}$
32500	0.1617	0.1589	0.0261000	7.75 $\times 10^{-6}$
39000	0.0972	0.0927	0.0094300	2.09 $\times 10^{-5}$
45500	0.0008	-0.0098	0.0000007	1.12 $\times 10^{-4}$
52000	-0.2294	-0.2190	0.0526000	1.08 $\times 10^{-4}$
58500	0.0896	0.0811	0.0080300	7.30 $\times 10^{-5}$
65000	-0.0730	-0.0728	0.0053300	7.22 $\times 10^{-8}$
71500	0.1217	0.1218	0.0148000	2.02 $\times 10^{-8}$
78000	-0.1572	-0.1565	0.0247000	5.69 $\times 10^{-7}$
84500	-0.0857	-0.0888	0.0073400	9.72 $\times 10^{-6}$
91000	0.0306	0.0295	0.0009340	1.08 $\times 10^{-6}$
97500	-0.0554	-0.0560	0.0030700	3.33 $\times 10^{-7}$
104000	-0.0363	-0.0365	0.0013200	3.67 $\times 10^{-8}$
110500	-0.0547	-0.0512	0.0029900	1.23 $\times 10^{-5}$
117000	-0.0070	-0.0077	0.0000497	4.86 $\times 10^{-7}$
123500	0.0403	0.0422	0.0016300	3.59 $\times 10^{-6}$
130000	0.0410	0.0404	0.0016800	4.13 $\times 10^{-7}$
136500	-0.0800	-0.0824	0.0064100	5.68 $\times 10^{-6}$
143000	0.0247	0.0249	0.0006100	3.68 $\times 10^{-8}$
149500	0.0616	0.0613	0.0038000	8.06 $\times 10^{-8}$
156000	0.0523	0.0520	0.0027300	1.02 $\times 10^{-7}$
162500	-0.2395	-0.2397	0.0573000	3.63 $\times 10^{-8}$
169000	0.1144	0.1134	0.0131000	1.03 $\times 10^{-6}$
175500	0.0418	0.0435	0.0017500	2.90 $\times 10^{-6}$
182000	0.0649	0.0637	0.0042200	1.63 $\times 10^{-6}$
188500	0.0071	0.0074	0.0000501	8.14 $\times 10^{-8}$
195000	-0.0119	-0.0114	0.0001410	2.42 $\times 10^{-7}$
201500	0.0072	0.0072	0.0000523	4.9 $\times 10^{-14}$
208000	0.0054	0.0054	0.0000295	2.8 $\times 10^{-14}$
Total			**0.37908**	**0.00047**

The difference of the amplitude of the original audio minus the encrypted audio is solved for each row of the table and the result elevated to 2.

All the results in column 5 of Table III are summed, and the value of the variable N is equal to the number of samples that exist in the table, in this case 33. Therefore using the Eq. 7

$$MSE = \frac{\sum(O-E)^2}{N}$$
$$MSE = \frac{0.00047}{33}$$
$$MSE = 0.0000142$$

This MSE result affirms that there is a minimum change between the original audio and the encrypted audio because the audio is close to 0, and as explained above, the closer the MSE is to zero, the smaller the average difference will be; therefore, the better the quality of the audio reproduction or processing, so it is reaffirmed that the audio quality is maintained after applying the FFT algorithm to the audio.

The summation of O^2 is in column 4 of Table III, and the value of the MSE is the result obtained previously, therefore using the Eq. 8:

$$SNR = 10 * \log_{10}(\frac{0.37908}{0.0000142})$$
$$SNR = 10 * \log_{10}(26695.77)$$
$$SNR = 10 * 4.43$$
$$SNR = 44.26dB$$

This result of 44.26 dB, reaffirms that the quality of the signal (sound) is preserved, after doing the Transform algorithm, so it deduced that the sound emitted by the audio remains clear and neat after encryption, and that any evident change between the values of the original audio with the encrypted audio is not perceptible or is minimally perceptible.

E. Recovered Images Similarity

By way of a more detailed analysis of the decrypted images in order to ensure pixel equality, MSE and SSIM measurements are used in the comparison of the original image (with some padding) and the one resulting from the encryption and decryption process. See the Table IV.

TABLE IV. IMAGES SIMILARITY MEASUREMENTS

	MSE	SSIM
Cat	0.00	1.00
Dog	0.00	1.00
Doom1	0.00	1.00
Doom2	0.00	1.00

MSE is used, because of its ample use and its simplicity to compare two images [28], every single image recovered by the scheme got a perfect score of 0, which indicates that the squared sum of the errors always equal 0, therefore, every single pixel value of both the recovered images and the original ones are the same.

The use of SSIM helps to evaluate the similarity between original and distorted images after applications in the evaluation of image quality, image recovery, image encryption and data hiding [29]. In effect, a value of 1 was obtained as result, this is evidence that the images still maintain the exact same image quality before and after encryption.

V. DISCUSSION

The use of Cellular Automata in the scheme ensures the security of the sensitive images to be hidden. These images are converted to effectively noise, any attacker that manages to extract the data hidden in the very low and high frequencies would find nothing but useless data.

In this study, an SNR of 44.26 has been obtained using the DFT together with CA. This result is notably superior to that obtained by another research on the embedding of color images in audio signals using residual networks [10]. In that work, the SNR ranged from 18.75 to 25.004, depending on the embedding scheme and the range of FFT indices employed.

An SNR of 44.26 indicates that the audio signal has a significantly higher SNR than the watermarking schemes employed in [10]. This suggests that the method using DFT and CA can insert information into the audio signal with less degradation of the perceived quality, which is crucial in applications where maintaining the fidelity of the original signal is a priority.

The presented scheme shows good payload capacity for the resulting file sizes, a 4 seconds audio file is capable of hiding an image of at most 256x256 dimensions, with a final size of 3.2MB. When comparing it to the scheme proposed by [10], a similar size audio file (4.23MB) of 24.15 seconds of duration, is able to hide an image of 64x64 dimensions. This shows a considerable increase in capacity per MB, albeit with a very reduced resistance to signal processing attacks and compression techniques because of the need to preserve every bit of the secret CA images to be able to decrypt the original images. Because of this, the proposed scheme should probably be used on relatively small audio files, so that the increase in size is not as dramatic and any action that affects the audio file data, such as applying mp3 compression, should be considered destructive to the encrypted information.

VI. Conclusion

Based on the challenge of secure and private information sharing, a new encryption method using CA and Fourier Transform based steganography has been proposed in this research. This method demonstrates the ability to preserve the full integrity of the encrypted images while maintaining an extremely similar quality to the original audio, where the hidden data is no different to noise, supported SNR, MSE and a magnitudes lower optimal warping path length compared to an audio containing an almost imperceptible noise.

Because of the combination of the produced audios file format and the range of frequencies that are embedded with data, the proposed scheme has good payload capacity, albeit with the limitation of the resulting file size and the vulnerability to small changes in the audio file.

Therefore, as possible improvements for future work, different approaches could be explored, such as implementing support for lower and higher color depth images at $24 - bits$, optimizing the Fast Fourier Transform process to use less precision, which would allow us to hide encrypted images in smaller floating point audio formats such as 32 and 24 bits, reducing the audio file size at the expense of data storage capacity.

References

[1] S. Karakus and E. Avci, "A new image steganography method with optimum pixel similarity for data hiding in medical images," *Medical Hypotheses*, vol. 139, p. 109691, 2020.

[2] N. Hamid, A. Yahya, R. B. Ahmad, D. Najim, and L. Kanaan, "Steganography in image files: A survey," *Australian Journal of Basic and Applied Sciences*, vol. 7, pp. 35–55, 2013.

[3] O. Tayan, "Concepts and tools for protecting sensitive data in the it industry: A review of trends, challenges and mechanisms for data-protection," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 2, 2017.

[4] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. Ho, and K.-H. Jung, "Image steganography in spatial domain: A survey," *Signal Processing: Image Communication*, vol. 65, pp. 46–66, 2018.

[5] S. Solak and U. Altinişik, "Image steganography-based gui design to hide agricultural data," *Gazi University Journal of Science*, vol. 34, no. 3, pp. 748–763, sep 2021.

[6] J. W. M. C. P. W. B. C. 2, "A survey on digital image steganography," *Journal of Information Hiding and Privacy Protection*, vol. 1, no. 2, pp. 87–93, 2019. [Online]. Available: http://www.techscience.com/jihpp/v1n2/29000

[7] F. E. Ziani, A. Sadak, C. Hanin, B. Echandouri, and F. Omary, "Ca-pcs: A cellular automata based partition ciphering system," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, 2020. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2020.0110376

[8] S. BOUCHKAREN and S. LAZAAR, "A fast cryptosystem using reversible cellular automata," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 5, 2014. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2014.050531

[9] G. Alvarez, L. Hernández Encinas, and A. Martín del Rey, "A multisecret sharing scheme for color images based on cellular automata," *Information Sciences*, vol. 178, no. 22, pp. 4382–4395, 2008.

[10] H.-T. Hu and T.-T. Lee, "Hiding full-color images into audio with visual enhancement via residual networks," *Cryptography*, vol. 7, no. 4, 2023.

[11] Z. Eslami, S. Razzaghi, and J. Zarepour, "Secret image sharing based on cellular automata and steganography," *Pattern Recognition*, vol. 43, pp. 397–404, 01 2010.

[12] G. A. M. L. H. E. A. H. E. A. M. del Rey and G. R. Sanchez, "Graphic cryptography with pseudorandom bit generators and cellular automata," *Digital.CSIC*, 2002.

[13] R. Tanwar and M. Bisla, "Audio steganography," pp. 322–325, 2014.

[14] A. Mohamed Abdirashid, S. Solak, and A. K. Sahu, "Data hiding based on frequency domain image steganography," no. 42, p. 71–76, 2022.

[15] F. I. Louis, "Image steganography in the frequency domain," 2023.

[16] N. T. E1 and R. V. Ravi, "An effective technique for hiding image in audio," *International Journal of Science and Research*, vol. 4, 2015.

[17] I. D. A. Focus and T. W. Groups, "Recommended practices for enhancing digital audio compatibility in multimedia systems," *DATWG Recommendation*, 1992.

[18] D. J. Katz and R. Gentile, "Chapter 5 - basics of embedded audio processing," in *Embedded Media Processing*, ser. Embedded Technology, D. J. Katz and R. Gentile, Eds. Burlington: Newnes, 2006, pp. 149–187. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780750679121500081

[19] G. Plenge, H. Jakubowski, and P. Schöne, "Which bandwidth is necessary for optimal sound transmission?" *Journal of the Audio Engineering Society*, vol. 28, no. 3, pp. 114–119, 1980.

[20] P. Lazzari and N. Seriani, "Two-dimensional cellular automata—deterministic models of growth," *Chaos, Solitons & Fractals*, vol. 185, p. 114997, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077924005496

[21] D. Anani and K. M. Faraoun, "Designing robust lmca-based threshold secret sharing scheme for digital images using multiple configurations assignment," *Journal of Communications Software and Systems (JCOMSS)*, vol. 11, no. 2, JUNE 2015.

[22] F. E. Toole, "Sound reproduction loudspeakers and rooms," *ELSEVIER Ltd*, 2008.

[23] L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, S. Leigh, M. Levenson, M. Vangel, N. Heckert, and D. Banks, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010-09-16 2010. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762

[24] P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, no. 1-23, p. 40, 2008.

[25] *Dynamic Time Warping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 69–84. [Online]. Available: https://doi.org/10.1007/978-3-540-74048-3_4

[26] W. Zhou and A. Bovick, "Mean squared error, love it or leave it?" *IEEE SIGNAL PROCESSING MAGAZINE*, 2009.

[27] M. R. Orora Tasnim, Selim Hossain, "Audio steganography with intensified security and hiding capacity," *Eur. Chem Bull Section A-Research paper*, 2023.

[28] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through fsim, ssim, mse and psnr—a comparative study," *Journal of Computer and Communications*, vol. 07, no. 03, p. 8–18, 2019.

[29] H. R. S. Zhou Wang, Alan Conrad Bovik and E. P. Simoncelli, "Image quality assesment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.