

Int. J. Advance Soft Compu. Appl, Vol. 15, No. 3, November 2023
Print ISSN: 2710-1274, Online ISSN: 2074-8523
Copyright © Al-Zaytoonah University of Jordan (ZUJ)

Improve Steganography Encrypted Audio Message in Video Frame

Esraa H. Abdul Ameer¹, Zainab Hussain Mohammed², and Alyaa Mohsin Dhayea³

¹Department of Computer Sciences, College of Education for Girls, Kufa
University, Iraq.
e-mail: israah.alzubaidy@uokufa.edu.iq

²Department of Computer Sciences, College of Computer Science and Mathematics,
Kufa University, Iraq.
e-mail: zainabh.almahdawie@uokufa.edu.iq

³Department of Computer Sciences, College of Archaeology, Kufa University, Iraq
e-mail: aalyam.zayi@uokufa.edu.iq

Abstract

Information security has become the topic of most interest nowadays. Because of significant technological development, transferring information from one person to another has become very easy. This information may be exposed to penetration by unauthorized persons, so the security interest in encryption and steganography has increased. In this paper presents a novel approach by integrating the methodologies of steganography and encryption, resulting in an algorithm that exhibits enhanced resilience against external attacks. The key is first initialized using Grigorchuk's group, which gives an unstructured permutation of data bits. Followed by the stage of encrypting the audio file using RC4, which relies on random generation of the final key in which it is encrypted. In order to increase security and protect data from hacking, the encrypted audio file is hidden in video frames using a non-sequential method based on LSB. The proposed method proved effective results in data preservation as it provided the Peak Signal-to-Noise Ratio scale value of 88, which makes high imputation capacity.

Keywords: *Grigorchuk's group, RC4, Audio encryption, Stego video, image frame, video frame.*

1 Introduction

The current century is the age of technology. Recent years have witnessed rapid and comprehensive progress in communication technology, and many means of communication are used in this field. Paying attention to the information exchanged between the parties has become essential. Robust security methods must be available to face the challenges facing these means from unauthorized entry. The biggest current problems are related to the security of information sent over the Internet. Security is available in two ways. A secure communication channel can send explicit information without attack [1], [2]. The data is encrypted and transmitted through an insecure channel so that it becomes understandable [1], [2]. Encryption and concealment are the most well-known methods used to protect information, each with its characteristics. Encryption is a

technique used to change the data format so that it becomes incomprehensible and unclear only to the intended person. Encryption uses secure algorithms of three types, block or stream, or traditional encryption methods. These algorithms can be reasonably secure, and it is possible to combine more than one method to produce a flexible and data-sensitive technique [3][4]. The second method of maintaining data security is the embedding technique, which enjoys sending the secret message in a cover where it looks like a normal cover that has nothing inside it. Several methods are used in this field, including directly embedding in the least essential bit, special programs for Windows systems, hiding the message by adding it to the end of an executable file, and others [5]. The Least Significant Bit method is commonly used because it is effective and does not degrade the image, as it may preserve its quality. After realizing the change, it also enjoys the embedded image and can be included in one, two, three, or four bits without a noticeable difference.

It is characterized by a large capacity for confidential data [6]. In the proposed work, the two security technologies, embedding and encryption, are combined to provide a method for maintaining the confidentiality of transmitting ed data. Multiple ways of encoding data in various fields, including Non-commutative encryption. It is a field of cryptography consisting of a set of protocols and groups. The non-reciprocal encryption methods, systems, and fundamentals are based on the algebraic structures represented in semi-groups, groups, and rings. Non-reciprocal encryption solves the problems related to encryption in terms of key exchange, authentication, and decryption [7]. This paper used Grigorchuk's Group as one type of Non-Commutative Cryptography. The Grigorchuk group G was invented by the great mathematician Rostislav Grigorchuk in 1980. It provides a wide general construction to convert anyone ' It depends on four generators that are based on encryption and includes a , b , c , and d , each of which has its law. It depends on binary data zero and one instead of decimal numeric values explained in the proposed method [8].

2 Related Work

Many researchers who have worked within this field used video frames to include confidential information, whether audio, image, or video. The next part explains some of the works within this field.

Ola [1] presented two image steganography techniques depending on Chaotic-Address Steganography. The first strategy is based on the well-known Least Significant Bit technique, while the second technique is based on a more recent approach that searches for the identical bits between the cover image and the hidden message. Both methods seek the least significant bits. Logistics is applied here in these techniques to extract the shuffled addresses bits.

Balaji et al. [9] introduced a video-based masking method, as the change in the video is not noticeable. The technique depends on indexing the transmitted data secretly and then embedding this indexing in the video frames. This indexing exists at the receiving party, which helps it extract confidential information from the frames in which it was included. This indexing approved by the sender and recipient reduces the chance of the attacker knowing it. Thus, the data is not sequentially included in the video.

Dasgupta et al. [10] introduced a method that relies on the well-known LSB algorithm but relies on the hashing method. It determines the hash function to hide the position of the least significant bit of the pixel. Since each image in the video consists of three RGB layers, 8 bits portions of the hidden message are cloaked within the image's pixel structure. Three

bits are contained within the red layer, three within the green layer, and two within the blue layer. Then the layers are combined to form the video after hiding.

Bhole et al. [11] also provided a hiding technology inside the video. Still, it depends on the byte that does not contain the information in the video and does not include the least significant bit sequentially. The video is read, and the frames are completely separated. Then a byte that does not contain information is determined in the video, and the following message that represents a text is taken and converted to binary and included in the bytes that do not contain information. This method is more random in embedding because after the empty byte of the secret data is filled, it is difficult to detect where it is included and in what layer.

In this paper presents a novel approach by integrating the methodologies of steganography and encryption. The key is first initialized using Grigorchuk's group, which gives an unstructured permutation of data bits. Followed by the stage of encrypting the audio file using RC4, which relies on random generation of the final key in which it is encrypted to increase security and protect data from hacking, the encrypted audio file is hidden in video frames using a non-sequential method based on LSB. The proposed method gives effective results in data preservation, which makes high imputation capacity.

3 Methodology

This section deals with the topics that have been used to encrypt the data:.

3.1 Grigorchuk's group

Grigorchuk's group is non-commutative as it represents a set of automatic shapes resulting from the binary tree, the characteristics of this tree being regular and infinite, denoted by the symbol T_2 . Infinite binary tree T_2 is realized as the set X^* , where $(*)$ is alphabet and $X = \{0, 1\}$ the set of all finite strings from $\{0, 1\}$ with root ϕ (empty). T_2 consists of two branches: a left turn denoted by (0) and a right turn denoted by (1). The Grigorchuk group G is then defined as the subgroup of $Aut(T_2)$ generated by four specific elements of $Aut(T_2)$ represented by (a, b, c, d). $Aut(T_2)$ represents the group of automorphisms of T_2 [12]:

$$G = \langle a, b, c \rangle \leq Aut(T_2)$$

If $S \in \{0, 1\}$, define \bar{s} by $\bar{s} = 1 - s$, in another expression ($\bar{0} = 1$ and $\bar{1} = 0$), then the four automorphisms with k bits will be:

$$\begin{aligned} a(s_1.s_2 \dots \dots \dots sk) &= (\bar{s}_1.s_2 \dots \dots \dots sk) \\ b(0.s_2.s_3 \dots \dots \dots sk) &= (0.\bar{s}_2.s_3 \dots \dots \dots sk) \text{ if } s_1 = 0 \\ b(0.s_2.s_3 \dots \dots \dots sk) &= (0.\bar{s}_2.s_3 \dots \dots \dots sk) \text{ if } s_1 = 1 \\ c(0.s_2.s_3 \dots \dots \dots sk) &= (0.\bar{s}_2.s_3 \dots \dots \dots sk) \text{ if } s_1 = 0 \\ c(1.s_2.s_3 \dots \dots \dots sk) &= (1.d(s_2.s_3 \dots \dots \dots sk) \text{ if } s_1 = 1 \\ d(0.s_2.s_3 \dots \dots \dots sk) &= (0.\bar{s}_2.s_3 \dots \dots \dots sk) \text{ if } s_1 = 0 \\ d(1.s_2.s_3 \dots \dots \dots sk) &= (1.b(s_2.s_3 \dots \dots \dots sk) \text{ if } s_1 = 1 \end{aligned}$$

3.2 Rivest Cipher 4 (RC4)

RC4 algorithm is a symmetric key for encryption that splits the plaintext into many streams and encrypts them separately. It safeguards the transmission of sensitive information over the Internet by encrypting the data stored on the system. The RC4 encryption algorithm is made up of two sub-algorithms: "KSA," which stands for "Key scheduling algorithm," and PRGA,

which stands for "Pseudo-random generation algorithm," which makes up the RC4 encryption technique. Fig. 1 explains RC4, The RC4 algorithm can construct the stream cipher with the assistance of these two additional algorithms [13], [14]. Algorithm 1 The algorithm below shows the steps of RC4 in detail.

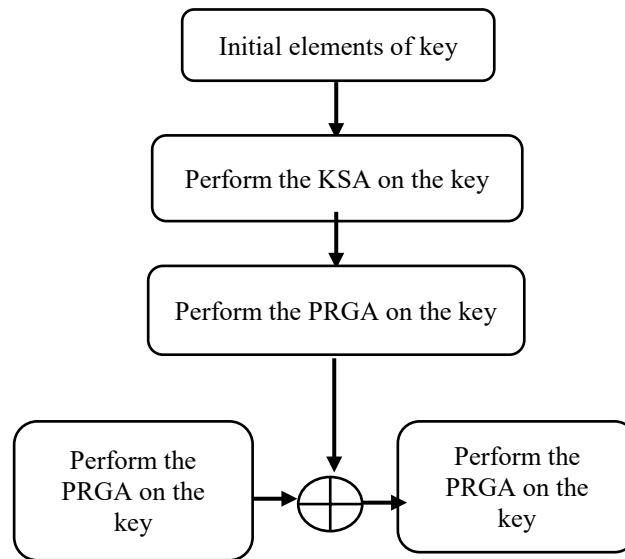


Fig.1: flowchart explains RC4.

Algorithm 1: RC4 algorithm.

Input: initial key (in_key) and stream of plain text.

Output: Cipher text.

Begin:

1. Generation of the key depends on the initial key:

For loop $i = 0$ to 255

$G_key[i] = i$

$S = \text{Mod}(i, \text{In_key-length of key})$

$T_key[i] = \text{In_key}(S)$

2. Perform KSA based on step one:

$i=0$ $j=0$

while loop where $i < 256$:

$R = j + G_key[i] + T_key[i]$

$j = \text{mod}(R, 256)$

$var = 0$

$var = G_key[i]$

$G_key[i] = G_key[j]$

$G_key[j] = var$

$i=i+1$

3. Perform PRGA based on step two:

$i=0$ $j=0$

while loop until all the plain text is a cipher:

$i = \text{mod}(i + 1, 256)$

$j = \text{mod}(j + G_key[i], 256)$

$var = 0$

$var = G_key[i]$

$G_key[i] = G_key[j]$

$G_key[j] = var$

$Y = \text{mod}(G_key[i] + G_key[j], 256)$

$\text{Cipher_key} = G_key[Y]$

$\text{Cipher_text} = \text{XOR}(\text{plain_text}, \text{Cipher_key})$

End

Confidential Sent Data is an audio file with a secret message to the recipient. The audio file is encoded using the RC4 algorithm, and the initial key is generated using Grigorchuk's group.

3.3 Proposed Method

Numerous studies have been conducted in this area to ensure the safety of data that is being transferred (such as text messages or photographs). Still, all of these studies rely on the overarching premise of the complete obscuring of images or video frames. Additionally, the usage of audio files to communicate with one another is not addressed because this method is more open to being discovered and having unlawful interference. This study describes an enhanced approach for embedding audio files into video media. The suggested way intends to reduce the inaccuracy between the original video's frames and the carrier video of the audio messages. In addition to this, it plans to incorporate big audio files that are scattered at random throughout the pixels of the frames rather than in a sequential manner, as is the case with the approaches that have been traditionally used.

The proposed system consists of two parts: the first is the encoding of audio messages, and the second is the inclusion of the encrypted message within the video frames. We depend on Grigorchuk's group to generate the initial values of the key. The hiding process involves separating the video frames from each other and selecting specific frames according to a particular seed. The sender determines the value of this seed in agreement with the receiving party. Since each frame is a color image consisting of three layers, only the blue layer is included. The embedding is not done sequentially in the pixels but instead in an even or odd way, depending on the value of the seed. If the seed value is odd, then the embedding is done in the pixels with odd locations, and if it is even, then the embedding is in the even locations. The embedding is only in the blue layer because it has the least effect on the human eye. Algorithms 2, 3, and 4 show the encoding, embedding, and de-embedding processes.

Algorithm 2: Encryption audio message.

<p>Input: Audio (plain text). Output: Cipher text. Begin: 1. $X := \text{values between } (0,255):$ For $i = 1$ to 255 $X[i] = i$ 2. $X := \text{Binary}(X)$. 3. $Ini_{Key} = \text{Grigorchuks}_G(X)$. 4. $Y := \text{Audio message}$. 5. $Y := \text{Binary}(X)$. 6. $Result_{enc1} := Ini_{Key} \oplus Y$. 7. $Key_{RC4} := \text{algorithm}(1)$. 8. $Result_{enc2} := Key_{RC4} \oplus Result_{enc1}$. End</p>

Algorithm 3: Encryption audio message.

put: Audio (plain text).

Output: Cipher text.

Begin:

1. $V := \text{Select any video as a cover.}$
2. *For* $i = 1$ *to* *num. of frames in video*
 $K[i] := \text{read}(V(i))$
3. $V_{Stego} := V$
For $i = 1$ *to* *num. of frames in video*
 $Seed := \text{random value}$
 $F := K[i]$ *based on seed*
 $F := \text{LSB}(\text{Result}_{enc2}, F(n, m, 3))$
 $V_{Stego} := F(n, m, l)$

Where n and m rows and Colum of image, l is layer of image.

End

Algorithm 4: Extraction message and decryption.

Input: Stego_frames.

Output: Plain text (audio).

Begin:

1. $V := \text{read } V_{Stego}.$
2. *For* $i = 1$ *to* *num. of frames in video*
 $Seed := \text{random value}$
 $F := K[i]$ *based on seed*
 $Audio_{enc} := \text{De_LSB}(F(n, m, 3))$
 $Aduio := \text{algorithm2}(Audio_{enc})$

End

A diagram can illustrate these three processes as in Fig. 2 and 3. Fig. 2 shows the process of embedding the encrypted message in the video, and Fig. 3 shows the process of de-embedding and extracting the original message.

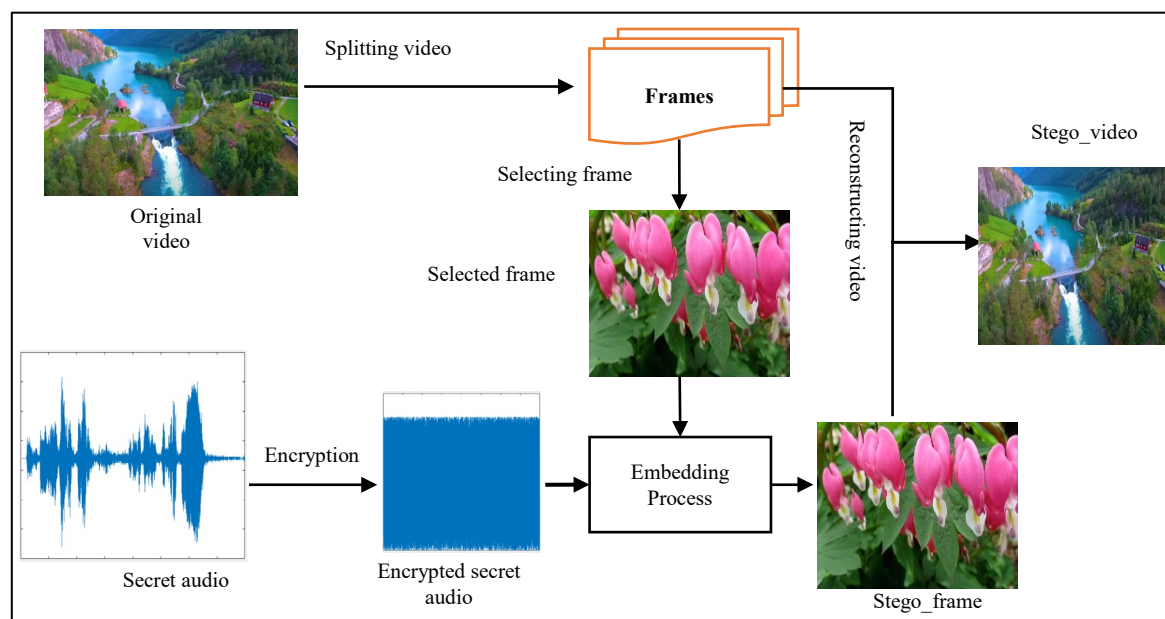


Fig. 2: Explains the hidden message.

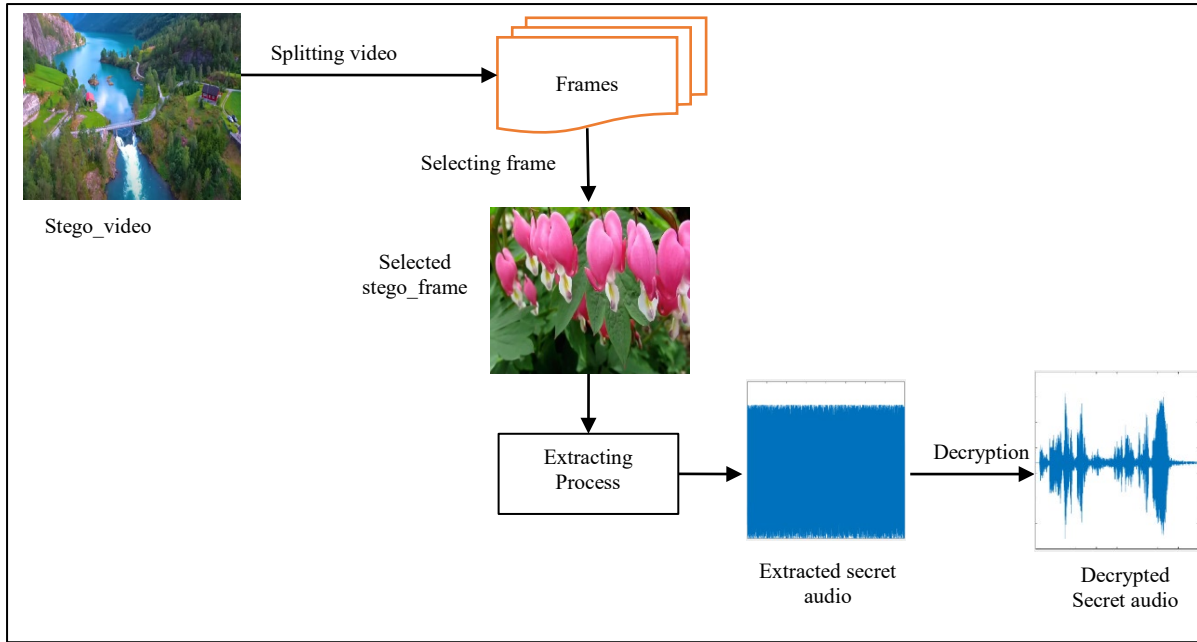


Fig.3: Explains the extraction message

4 Results and Discussions

The proposed method was simulated using MATLAB 2021. The suggested system has two components, and the first section contains encryption using the algorithm described in section (3.3). Multiple audio files of varying sizes were recorded, and internet-downloaded audio was encoded to evaluate the efficacy of the suggested approach. Attackers frequently make use of differential methods of attack. The attacker discovers the relationship between the plaintext and encryption text and can crack the method by analyzing the modification between them. If a minor alteration in the plaintext can significantly change the cipher text, the encryption technique may withstand a differential attack. Because of this, calculating NSCR and UACI is a common practice for determining whether or not an algorithm can withstand differential attacks. Using equation (1) and equation (2), we can calculate the average values for NSCR and UACI (2). UACI is one of the differential analyses used to evaluate the strength of audio encryption, where the contrast between the ciphered and the origin audio is estimated. The highest value of the UACI (Approximately 33.46%) implies that the proposed procedure is safe against differential assaults [15].

$$NSCR = \sum_{i=1}^n (D(i, j) / N) * 100 \quad (1)$$

$$UACI = \sum_{i=1}^n ((p(i, j) - c(i, j)) / (2^k - 1)) * 100 \quad (2)$$

In this context, N denotes the whole duration of the audio, and "k" refers to the minimum number of bits necessary to show the audio. C and P denote the cipher audio that results when the original audio is changed by one value. If C(i) is not equal to P(i), then D(i) is equal to 1; otherwise, D(i) is equal to 0. The NSCR should ideally be at 100%. Other measures can also be used, such as in eq (3) and (4), measuring the error amount between the original and encoded audio.

$$MSE = \sum_{i=1}^n \sum_{j=1}^n (I(i, j) - c(i, j))^2 / M * N \quad (3)$$

$$PSNR = 10 \log_{10}(255^2 / MSE) \quad (4)$$

Where I the original audio and C audio after hidden, M and N size of audio. We note in Table 1 the values of the metrics that reflect the efficiency of the proposed encryption method, the majority of which were very near to the ideal values for each metric. This is because the suggested approach encodes data in a way that minimizes the amount of space it takes up. There is a collection of audio samples taken, each of which is a different length. When it comes to audio files, the NSCR scale always has a value of 100, which indicates that there is absolutely no resemblance between the audio before and after encryption. On the other side, the UACI scale is getting close to 33, and the MSE scale shows that an increase in the difference results in an increase in the amount of error. PSNR does not reflect the difference ratio between two files; rather, it indicates the similarity ratio between two files. Since its value is near to zero, this suggests that there is no resemblance between the two files. Whenever the PSNR value gets close to 100 or infinity, it indicates that the two files are identical to one another [16] [17].

The strength of this method lies in the use of Grigorchuk's group which is the seed of key generation. Grigorchuk's group relies on binary data so the number of possibilities used for this seed is too many to guess. In addition to guessing each number from Grigorchuk's group, the attacker would have to guess twice the cases in the primary key generation. As soon as one bit differs, it will change the key, and changing one bit in the key will lead to the inability to break the key.

Table 1: Performance values for encoding audio files of different sizes.

Size of audio	NSCR	UACI	MSE	PSNR
2kb	100	33.00	120	5
10kb	100	33.3	150	7
15kb	100	33.3	110	3
20kb	100	33.3	200	2
30kb	100	33.5	190	3

The results show that the more color the video and more objects, the embedding process is very effective because this large change in the frames makes it impossible to detect there is a change in the pixels of the images contained in the video. In other words, color video is more efficient for embedding than gray video, and the more frames it is, and hiding is not sequential, i.e. random, the better because randomization increases the sensitivity of the method and makes it difficult to detect hiding. The results show that the more color the video and more objects, the embedding process is very effective because this large change in the frames makes it impossible to detect there is a change in the pixels of the images contained in the video. In other words, color video is more efficient for embedding than gray video, and the more frames it is, and hiding is not sequential, i.e. random, the better because randomization increases the sensitivity of the method and makes it difficult to detect hiding. Contain encrypted audio files hidden within them. It was decided to collect

a number of videos, some of which had been recorded by the researcher while others had been obtained from the internet. We can see that the appearance of the video frames has not been altered in any way after they have been hidden by referring to the figure that can be found below. Even when we take each frame separately, we see PSNR values of up to 80, which is a respectable result considering the quantity of data that was included in it. Because of the distributed nature of the concealment, it is extremely difficult to find these files. When it comes to the usage of videos for covert purposes, it is best to record your own versions of the videos rather than relying on those that can be found on the internet. The fact that the original video is not readily available is the reason for this, and the reason for this is that it makes it harder for the attacker to check whether or not the video has been modified.

After hiding the data inside the video, which needed four images of the video of fig. 4, shows the difference between video images before hidden and after hidden. The first field represents the sequence (id) of the images (video frames) and the second field represents the images before hidden (before stego) and the third field represents the image of itself, but after hidden (after stego) and the fourth field represents the value of PSNR to measure the similarity between the image itself before and after the hidden. We conclude two things through figure 4 the first that all images with a sequence (1,2,3,4) were not affected by the amount of hidden data in them, so the images appear as they are before and after hidden despite the use of the LSB algorithm to hide the secret data in it, because just a simple change in the image indicates that the proposed method is wrong and causes problems, and it will reveal the existence of a hidden message inside the image. The best, which in our proposed method, is that the image is before the hiding process and after the hiding process, there is no difference between them, and they must be similar the encryption, there must be a difference between images, and our proposed method is not encryption, but rather concealment. The main goal of hiding process, which we used, is that the image before hiding process and after hiding process be similar in appearance and not in content, so they are similar.

The images in the whole figure have hidden an amount of data the image is supposed to be VISUALLY is equal and similar so as not to arouse suspicion or the attention of the attack The second matter is a different PSNR value for the images, and this depends on the extent of changing the least importance in the image, for example, with a sequence of 1, the value of PSNR is equal to 80, while the image has a sequence 2 and the value increased because a change did not happen in all pixels, so there were pixels match to hidden data. Image with a sequence 4 is the highest value for the PSNR and the reason because it is a small part of the secrecy data that has been hidden the most included in the(1,2and3).









Frames	Before stego	After stego	PSNR
1			80
2			82
3			81
4			84

Fig.4: PSNR for some stegano audio in video.



Fig. 5: some screenshot of video1 and video2.

Table 2: Values of PSNR and MSE.

Test video1 (308×540)	
Ave. PSNR	83.56
Ave. MSE	0.083
Test video2 (433×740)	
Ave. PSNR	85.87
Ave. MSE	0.091

Table 3: Selection step of Animation AVI video for 4-time selection using key 10, 32, 68, 100.

Video	Frame number	Message size	PSNR	MSE
V1	520	2 kb	80	0.07
	420	4 kb	79	0.06
	190	10 kb	77	0.06
	300	20 kb	75	0.05
V2	520	2 kb	82	0.04
	420	4 kb	79	0.081
	190	10 kb	75	0.089
	300	20 kb	70	0.9

We notice through the second table, which indicates the total rate of similarity and difference between the original frames and the frames after masking. The value of the two measures rises to the optimum, for example PSNR. This value exceeds 80, which expresses the amount of similarity between the before and after frames. Although the amount of hidden data for the audio file is large. As well as the values of the error rate that measures the differences were close to zero less than one.

According to Fig. 5, two samples of videos of different sizes were used, and we performed the hiding process for multiple audio files. We notice from Table 2, which

shows the error rate and similarity, that the PSNR values for the second videos are higher than the first video. As for the amount of error, there is very little difference between the two, which increases with the increase in the amount of hidden data. Table 3 shows the relationship between the number of frames and the size of the hidden message against each measure of error and similarity. Table 3. Different key values were taken, and the audio file of different sizes was hidden in two videos according to Fig. 5. The output of the table shows the extent to which masking is affected by the size of the audio file. Does not reflect distortions in the appearance of the image. This can be illustrated by charts 6 and 7 showing the difference in terms of scale values.

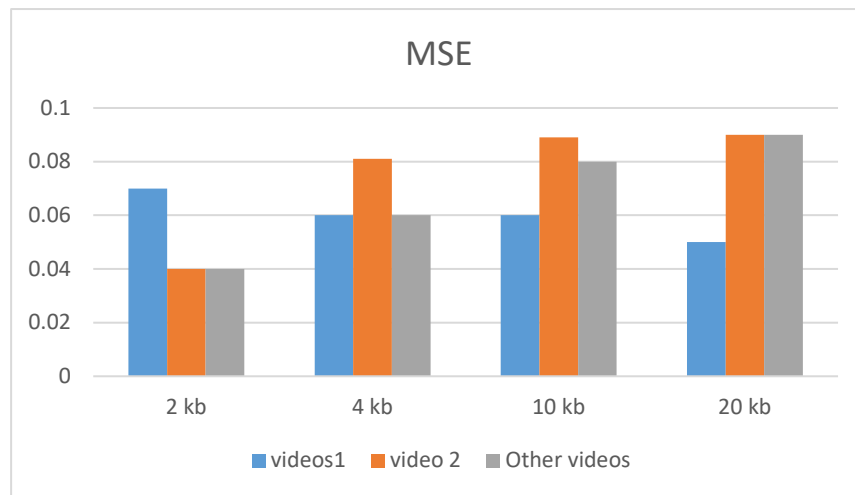


Fig. 6: The relationship between message size and MSE.

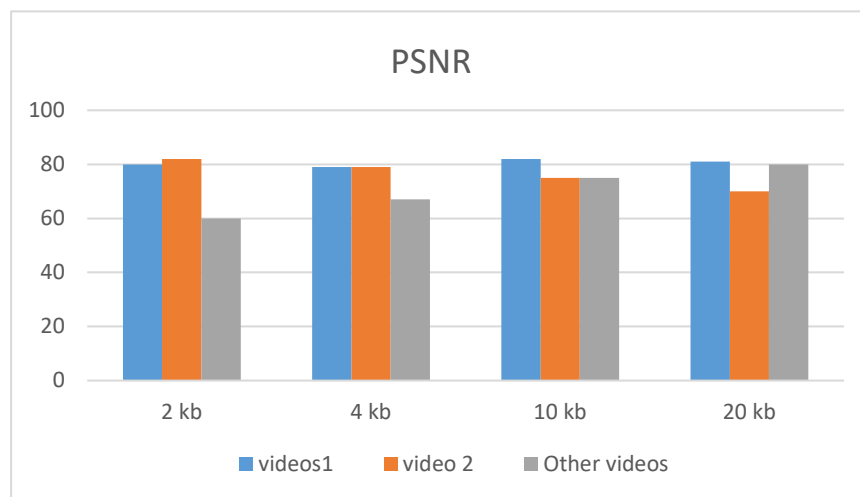


Fig.7: The relationship between message size and PSNR.

Table 4: Compare our method and other work.

Ref.	MSE	PSNR
[18]	—	60.440
[19]	0.02	—
[20]	—	56

6 Conclusion

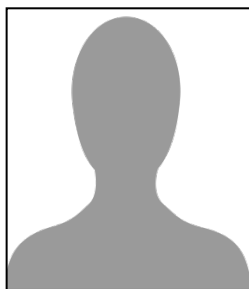
The proposed method provides an enhanced system for safeguarding sensitive audio data during transmission over unprotected channels. Utilize two distinct safeguards, the first of which is the application of a non-reciprocal aggregates-based form of encryption. When the key is connected using the RC4 method, it generates a multiple probability that is difficult to predict due to the manner in which it is formed; therefore, it is used to generate the key. The encrypted data is then appended to the video frames in a manner distinct from the preceding phase. In order to reduce distortions and disparities between the original frames, which were concealed within them, only a portion of the frame is utilized. The MSE and PSNR values obtained from the measurements indicate a low presence of difference in the hidden files, suggesting that the steganography procedure has resulted in the maintain of the original data frame. Key sensitivity study demonstrates that even a small modification to the secret key results in unsuccessful decryption, and key space analysis demonstrates the required level of protection against brute-force assaults. The findings of the cryptographic study lead us to the conclusion that the proposed technique provides sufficient cryptographic security for the encryption of audio data.

References

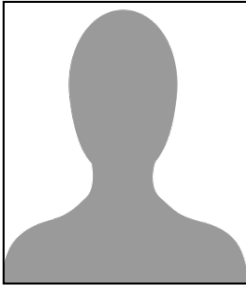
- [1] Talal, M.L., Hassan, I.A., Zaidan, F.K., & Badr I.M. (2023). Steganographic Data Hiding Using Quantum Behaved Particle Swarm Optimization and An Enhanced Aes Algorithm. *International Journal on Technical and Physical Problems of Engineering (IJTPE)*, 15(1),102-109.
- [2] Bellat, A., Tyass, I., Mansouri, Kh., & Raihani, A. (2021). Optimization of Wind Farms by The Particle Swarm Algorithm Considering Gaussian Wake Model. *International Journal on Technical and Physical Problems of Engineering (IJTPE)*, 13(3), 48-54.
- [3] Forouzan, B. A., & Mukhopadhyay, D. (2015). Cryptography and network security. *New York, NY, USA: Mc Graw Hill Education (India) Private Limited*, 12.
- [4] Pal, S., & Bandyopadhyay, S. K. (2016). Various Methods of Video Steganography. *International Journal of Information Research and Review*, 3(6), 2569-2573.
- [5] Luo, W., Huang, F., & Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on information forensics and security*, 5(2), 201-214.
- [6] Kaur, K., & Kaur, B. (2018). DWT-LSB approach for video steganography using artificial neural network. *International Advanced Research Journal in Science, Engineering and Technology (IARJSET)*, 5(7), 20-25.
- [7] Skuratovskii, R., & Osadchyy, V. (2020, July). An application of Miller Moreno groups to establishment protocol Non commutative cryptography. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)* (pp. 126-130). IEEE.
- [8] Hemeida, F., Alexan, W., & Mamdouh, S. (2021). A comparative study of audio steganography schemes. *International Journal of Computing and Digital Systems*, 10, 555-562.

- [9] Balaji, R., & Naveen, G. (2011, May). Secure data transmission using video Steganography. In *2011 IEEE International Conference on Electro/Information Technology* (pp. 1-5). IEEE.
- [10] Sadkhan, S. B., Mahdi, A. A., & Mohammed, R. S. (2019, December). Recent Audio Steganography Trails and its Quality Measures. In *2019 First International Conference of Computer and Applied Sciences (CAS)* (pp. 238-243). IEEE.
- [11] Bhole, A. T., & Patel, R. (2012, December). Steganography over video file using Random Byte Hiding and LSB technique. In *2012 IEEE international conference on computational intelligence and computing research* (pp. 1-6). IEEE.
- [12] Horan, K., & Kahrobaei, D. (2018). The hidden subgroup problem and post-quantum group-based cryptography. In *Mathematical Software–ICMS 2018: 6th International Conference, South Bend, IN, USA, July 24-27, 2018, Proceedings 6* (pp. 218-226). Springer International Publishing.
- [13] Sumartono, I., Siahaan, A. P. U., & Mayasari, N. (2016). An overview of the RC4 algorithm. *IOSR J. Comput. Eng.*, 18(6), 67-73.
- [14] Bricout, R., Murphy, S., Paterson, K. G., & Van der Merwe, T. (2018). Analysing and exploiting the Mantin biases in RC4. *Designs, Codes and Cryptography*, 86, 743-770.
- [15] Parthasarathi, M., Shreekala, T., Nadu, T., & Nadu, T. (2017). Secured data hiding in audio files using audio steganography algorithm. *Int. J. Pure Appl. Math.*, 114(7), 743-753.
- [16] Farsana, F. J., & Gopakumar, K. (2020). Speech encryption algorithm based on nonorthogonal quantum state with hyperchaotic keystreams. *Advances in Mathematical Physics*, 2020, 1-12.
- [17] Al-Mashhadi, H. M., & Abduljaleel, I. Q. (2017, April). Color image encryption using chaotic maps, triangular scrambling, with DNA sequences. In *2017 International Conference on Current Research in Computer Science and Information Technology (ICCRIT)* (pp. 93-98). IEEE.
- [18] Mstafa, R. J., Younis, Y. M., Hussein, H. I., & Atto, M. (2020). A new video steganography scheme based on Shi-Tomasi corner detector. *IEEE Access*, 8, 161825-161837.
- [19] Ramalingam, M., & Isa, N. A. M. (2015). A steganography approach over video images to improve security. *Indian Journal of Science and Technology*, 8(1), 79.
- [20] Mstafa, R. J., & Elleithy, K. M. (2015, April). An efficient video steganography algorithm based on BCH codes. Asee.

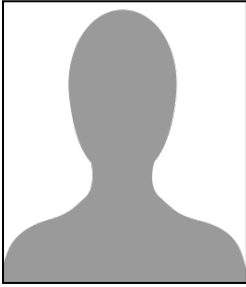
Notes on contributors



Esraa H. Abdul Ameer is Lecturer at the Department of Computer Science, Kufa University, Iraq. Her research interests are image processing, Classify the data, Data analysis, Data encryption. She has published several research articles in international journals of Computer Science.



Zainab Hussain Mohammed is Lecturer at the Department of Computer Science, Kufa University, Iraq. Her research interests are image processing image processing and Data encryption. She has published several research articles in international journals of m Computer Science.



Alyaa Mohsin Dhayea is Lecturer at the Department of Computer Science, Kufa University, Iraq. Her research interests are image processing image processing, and computer vision. She has published several research articles in international journals of Computer Science.