AudioMag: A Hybrid Audio Protection Scheme in Multilevel DWT-DCT-SVD Transformations for Data Hiding

Jingjin Yu[1], Chai Wen Chuah*[2], Rundan Zheng[3], Janaka Alawatugoda*[4]
Guangdong University of Science & Technology, Dongguang, Guangzhou, China[1,2,3]
Research & Innovation Centers Division, Rabdan Academy Abu Dhabi, UAE[4],
and Institute for Integrated and Intelligent Systems, Griffith University, Nathan, Queensland, Australia[4]

***Abstract*—Steganography is a technique used to hide data within an image or audio in order to maintain the secrecy of the message being communicated. There are several methods used in steganography to achieve this, but commonly, the data hiding is between the same stego-entity, such as an image with an image or audio with audio. One drawback of hiding data within the same entity is that once the security is compromised, one may be able to access the particular data. Therefore, this research proposes hiding audio within an image. The first step is to transform the audio into a hexadecimal value. Next, the hexadecimal value is hidden within the shortened uniform resource locators. The uniform resource locators are concatenated, shuffled, and converted into a quick response code. Finally, the quick response code is embedded into an image. The simulation results show the successful hiding of the audio message within the image, while maintaining the security and confidentiality of the hidden messages.**

***Keywords*—Steganography; image steganography; audio steganography; data hiding; stego-entity**

I. Introduction

As the rapid growth of the Internet and the bandwidth continues, cryptography is responsible for ensuring secure communication over insecure interconnected networks, guaranteeing transmission privacy and authentication [1], [2]. There are two common methods for sending secret messages: data encryption [3] and data hiding [4], [5]. In the case of an encrypted message, only the intended recipient can view the message by decrypting it with a secret key. Even if an attacker obtains the encrypted message, they will be unable to decipher the content. However, if the key is revealed to the public or stolen by the attacker, the message can be compromised [6]. Data hiding, or steganography, which falls under the cryptography umbrella, is another technique that is widely used to hide secret messages within harmless messages in a way that prevents attackers from detecting the existence of the secret message.

Generally, data hiding includes the process of generating a stego-image (embedding payload into the image), while extraction is the process of viewing the payload from the stego-image [5]. The security level of the stego-image is based on its similarity to the original image, making it difficult for unintended observers to be aware of the existence of the hidden secret message. There are two common types of steganography: audio steganography and image steganography.

Image steganography [8], [9], [10] refers to the process of hiding data within an image file. The image selected for this purpose is known as a cover image, and the image obtained after steganography is called a stego-image. The least significant bit (LSB) technique is one of the simplest approaches by replacing the LSB of each pixel in the cover image with a piece of the hidden data [11], [12]. Since it only involves one bit change in a pixel, the stego-image appears identical to the original image. Hence, these changes will be hard to detect by the human eye, which is not sensitive.

Audio steganography is a technique of inserting hidden messages within sound files [13], [14], [15]. Users can insert a secret message into the audio by manipulating the binary sequence of the file, like adjusting its length or making subtle changes to its structure. This is undetectable to the human senses, as human are not sensitive to the small changes. Common audio steganography sound file formats to date include waveform audio file format (WAV), audio units (AU), and MPEG audio layer 3 (MP3) [16]. The process of embedding secret messages in digital sound is typically more complex than embedding messages in other forms of media, such as digital images.

However, information shared online by individuals often faces various risks, such as being maliciously intercepted by third parties or misused in terms of ownership. Steganography offers effective solutions to address these issues. The objective of this research is to develop and evaluate audio steganography, which conceals audio within images, to enrich people's comprehension beyond the traditional methods of steganography. The significance of utilizing both image and audio steganography lies in taking advantage of the lack of sensitivity of the human ear and eye to subtle changes, which makes it easier to hide communication information and enhances security [17], [18]. This will aid in protecting communication privacy and preventing unauthorized access to the information.

The remainder of this paper is organized as follows: Section II presents the literature review. Section III shows the proposed AudioMag. The AudioMag simulation is shown in section IV. Section V contains the result discussions. Finally, Section VI concludes the paper.

II. Literature Review

Steganography is a technique of hiding secret information in a host, such as an image, audio, or video. The secret information is known as payload which could be a message, image, or audio. Steganography ensures that this hidden information is not noticeable to anyone who is not specifically looking for it. Therefore, individuals are able to transmit confidential data without raising suspicion through steganography. There are three common techniques for steganography: image steganography and audio steganography.

A. Image Steganography

Vleeschouwer, et al. proposed a method for embedding the payload into the stego-image based on patchwork theory, which has certain robustness against JPEG loosely compression [7]. Loosely compression meant that even though the image being compressed, there will be loss of some information, but with this proposed method, they still are able to retrieve back the payload. The image is divided into two zones. The histogram of each zone is mapped to a circle where it is the place to allocate the payload. That is, each bit of the payload is associated with a group of pixels, for example a block in an image, and then the payload is map into the associate zones. The embedding process hides the payload by changing the whole pixel value of the image. This algorithm suffers from the salt-and-pepper noise. To overcome this problem, this algorithm is suitable for small size of payload with multi-tone image and not suitable for halftone images.

Honsinger's group patent the data hiding technique used for fragile authentication [8]. The method that they carried out is based on adding the payload to original image, pixel by pixel using modulo 256 additions to form the stego-image. This method is comfortable only for those payloads with the capacity in category from 1k to 2k bits. Overall, the process of embedding the payload can be described in two mathematics equation, the first one as $I'(x,y) = I(x,y) + \alpha M(x,y) * C(x,y)$, this formula is used to find the location of payload in original image, where and $I'(x,y)$ is the location of payload in the image, $\alpha M(x,y) * C(x,y)$ denotes as payload coordinate and $I(x,y)$ as location of original image. While the second equation is $Iw = (I + W)$ mod 256, Iw denotes marked image, I original image, W is the payload comes from the hash function of the original image and the modulo 256 addition ensures that the modified image values are always be in the same range as the original image values. By using these two equations, with first equation they find the location in original image then second equation responsible in avoiding over or underflow happens in stego-image. This technique changes the entire pixel value of the original image, hence, stega-image suffers salt-and-pepper noise.

Chandran and Khoushik compared the efficacy of LSB, discrete cosine transform (DCT), and discrete wavelet transform (DWT) techniques in the context of steganography [9]. The findings indicated that the DCT technique demonstrated superior performance when compared to both LSB and DWT methods. The evaluation was centered on the quality of the stego-image and the original cover image. Limitations of each approach were highlighted, with the LSB method showing weaknesses in terms of invisibility and robustness, the DCT method lacking robustness, and the DWT method exhibiting lower PSNR values and higher MSE.

Moon and Kawitkar proposed data hiding by using four LSB (4 LSB) substitution method and password for advance security [10]. With the protection layer by using password, the user only can view the payload with correct password. Without the password, the user only can get back the cipher text which is the junk characters. And this technique can embed large size of payload. But, stego-image is suffer with noise as well if the original image is halftone image, thus the attackers easily notice that particular image is the stego-image then he can break the password to get the payload.

There are many researchers [11], [12] tend to use LSB method for concealing data by utilizing the least significant bit of the cover image, making it imperceptible to the naked eye. However, this technique is deficient in terms of limited payload capacity.

B. Audio Steganography

Biswajita Datta et al. [13] presented an innovative method that employs LSB encoding across multiple layers, allowing for the simultaneous embedding of two data bits into the cover media to increase stego-audio capacity. The extraction procedure entails bitwise operations, adding complexity to interpreting the data without understanding these operations. Furthermore, techniques such as bit adjustment and flag setting are utilized to maintain the perceptual transparency of the stego-audio.

Sayed et al. [14] investigates the integration of two distinct steganography methods in a multi-level steganography framework. The initial method entails concealing a message in an audio cover through a modified LSB technique, whereas the second method involves concealing a second message in the output of the first level using a phase coding approach. The stego-audio file that results contains two audio covers with concealed messages.

Nugraha explores the utilization of steganography in audio data through the direct sequence spread spectrum technique which transmitting hidden messages via radio waves, where the message is carried by a noise-like wave [15]. This technique can be adapted for concealing messages within audio data, where the embedded information will manifest as noise. The proposed method requires a key to embed messages within the noise, generating a pseudo-noise waveform. Prior to embedding, the information to be hidden must first be modulated using this pseudo-noise signal.

C. Image Processing Methods

1) Discrete Wavelet Transform: Discrete wavelet transform (DWT) decomposes an image into a sequence of images with different spatial resolutions. Fig. 1 shows how the image is decomposed into the first level, resulting in 'LL', 'LH', 'HL' and 'HH'. The second level is then applied to the low frequency 'LL' only, where 'L' represents low frequency bands and 'H' represents high frequency bands. Threshold is applied to the wavelet coefficients in the high frequency levels, as the noise present in the low frequency wavelet coefficients will be averaged out. This calculation can effectively reduce noise without significantly distorting the underlying signal [19].

2) Discrete Cosine Transform: Discrete cosine transform (DCT) converts a series of pixels in an image into a series of frequency domain coefficients [20]. Eq. (1) computes the (i, j)

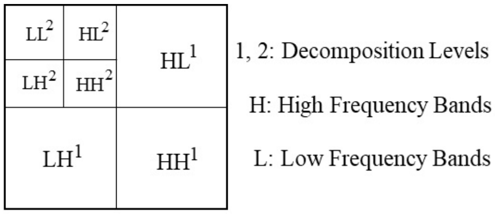


Fig. 1. 2D-DWT with 2-Level decomposition.

entry of the DCT of an image, where i and j are the coordinates of the transformed image. The $p(x, y)$ is the x and y^{th} element of the image, and N is the size of the block calculated by the DCT. Eq. (2) represents a normalization constant.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

$$Y(i,j) = \frac{1}{\sqrt{2K}} C(i)C(j) \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} p(x,y) * \cos\left[\frac{(2x+1)i\pi}{2K}\right] \cos\left[\frac{(2y+1)i\pi}{2K}\right] \tag{1}$$

$$C(b) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if b = 0} \\ 1, & \text{if b} > 0 \end{cases} \tag{2}$$

The equation computes the value of one specific entry (i, j) in the transformed image by using the pixel values from the original image matrix. For example, in the case of a 4 x 4 block image, the value of N is 4 and the range of x and y is 0 - 3. Therefore, $Y(i, j)$ is determined based on Eq. (3).

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

$$Y(i,j) = \frac{1}{\sqrt{8}} C(i)C(j) \sum_{x=0}^{3} \sum_{y=0}^{3} p(x,y) * \cos\left[\frac{(2x+1)i\pi}{8}\right] \cos\left[\frac{(2y+1)i\pi}{8}\right] \tag{3}$$

3) Singular Value Decomposition: Singular Value Decomposition (SVD) manipulates the original image into three different matrices as shown in Eq. (4) [21]. A is original M X N matrix, U is the M X R matrix with orthonormal columns matrix, σ is the R X R diagonal matrix of singular values, and V^T is the transpose orthonormal columns matrix with dimension of R X N. It should be noted that, M, N, and R are the dimensions of the original matrix A.

$$A = U\sigma V^T \tag{4}$$

D. Caesar Cipher

Caesar cipher is one of the simplest and most widely known encoding and decoding techniques. The encoding [see Eq. (5)] and decoding [see Eq. (6)] techniques are based on a substitution method where the letters of plaintexts are shifted backwards (or forwards) by a fixed number (d) to produce encoding letters [22].

$$c = p + d \; mod \; n, \tag{5}$$

$$p = c - d \; mod \; n, \tag{6}$$

For example, when the letters of plaintext are based on ASCII table, then the n value is 128. Let the "d" be 15 and the plaintext is "Pe@C3", the encoding process is shown in Table I.

TABLE I. CAESAR CIPHER - ENCODING PLAINTEXT OF "PE@C3"

Plaintext	80	101	64	67	51
$c = p + d$ **mod** 128	75	96	59	62	46
Ciphertext	K	‘	;	>	.

The decoding process is the reverse of the encoding process. If the encoding of "d" is 15, then the value of "d" for decoding is -15. Let's decode the ciphertext of "K‘;¿.", the complete process is shown in Table II.

TABLE II. CAESAR CIPHER - DECODING CIPHERTEXT OF "K‘;¿."

Ciphertext	75	96	59	62	46
$p = c - d$ **mod** 128	80	101	64	67	51
Plaintext	P	e	@	C	3

III. Proposed AudioMag Steganography

The proposed AudioMag steganography data hiding technique involves hiding audio data within an image. This process is illustrated in Fig. 2. Audio consists of time and amplitude, modification must be made to the audio in order to embed it into the image, which consists of the x-axis and y-axis. The explanation is provided in 1. The algorithm outlines the specific modifications and calculations that need to be performed on the audio data in order to align it with the x-axis and y-axis of the image.

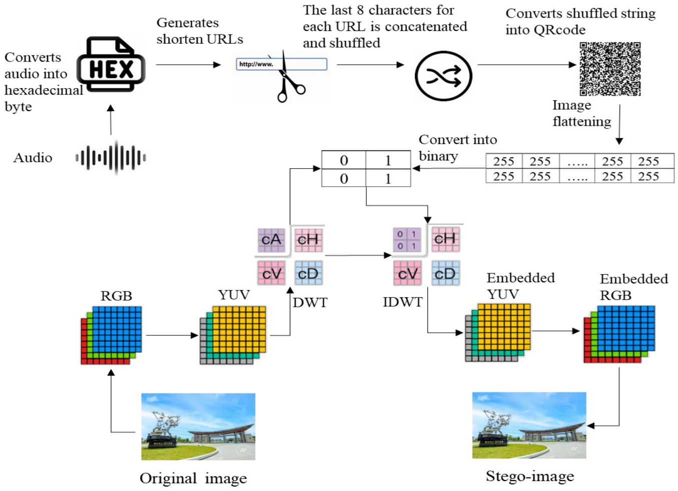


Fig. 2. AudioMag steganography - audio hiding.

The proposed AudioMag steganography audio extraction technique involves extracting the QR code from the stego-image, reading the string of the QR code, manipulating the string into hexadecimal value, and finally converting it into audio. This process is illustrated in Fig. 3. The explanation is provided in Algorithm 2.

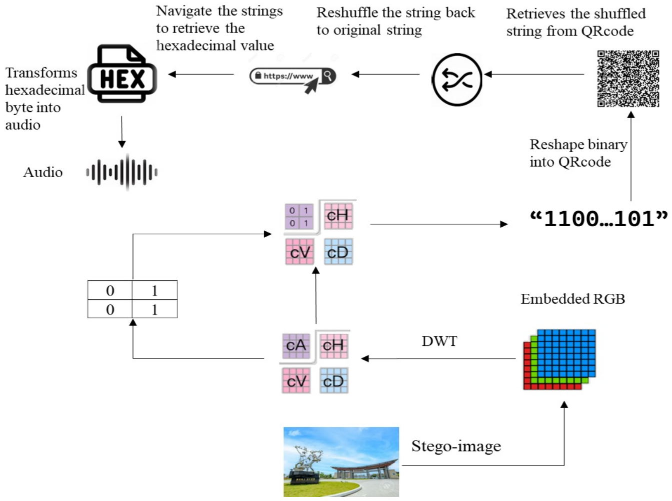
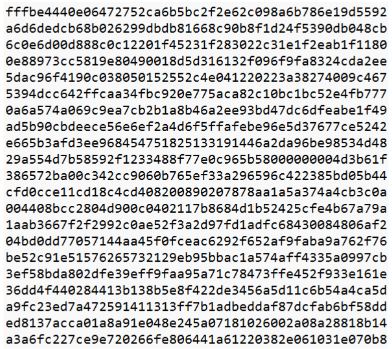

Navigate the strings to retrieve the hexadecimal value
Reshuffle the string back to original string
Retrieves the shuffled string from QRcode
Transforms hexadecimal byte into audio
HEX
https://www
Audio
Reshape binary into QRcode
0 1
0 1
cH
cV cD
“1100…101”
0 1
0 1
Embedded RGB
cA cH
cV cD
DWT
Stego-image

Fig. 3. AudioMag steganography - audio extraction.

IV. AudioMag Simulation

An audio recording is made using Realme UI 3.0. The spoken text is “Cryptography” and lasts approximately 1 second. The audio file is 41.5 kilobytes (kb) in MP3 format. The audio is then converted into a hexadecimal value, as illustrated in Fig. 4, resulting in a file size of 82.5kb. The hexadecimal value is then divided into blocks, resulting in three blocks and consequently, three shortened URLs.



```
fffbe4440e06472752ca6b5bc2f2e62c098a6b786e19d5592
a6d6dedcb68b026299dbdb81668c90b8f1d24f5390db048cb
6c0e6d00d888c0c12201f45231f283022c31e1f2eab1f1180
0e88973cc5819e80490018d5d316132f096f9fa8324cda2ee
5dac96f4190c038050152552c4e041220223a38274009c467
5394dcc642ffcaa34fbc920e775aca82c10bc1bc52e4fb777
0a6a574a069c9ea7cb2b1a8b46a2ee93bd47dc6dfeabe1f49
ad5b90cbdeece56e6ef2a4d6f5ffafebe96e5d37677ce5242
e665b3afd3ee968454751825133191446a2da96be98534d48
29a554d7b58592f1233488f77e0c965b58000000004d3b61f
386572ba00c342cc9060b765ef33a296596c422385bd05b44
cfd0cce11cd18c4cd408200890207878aa1a5a374a4cb3c0a
004408bcc2804d900c0402117b8684d1b52425cfe4b67a79a
1aab3667f2f2992c0ae52f3a2d97fd1adfc68430084806af2
04bd0dd77057144aa45f0fceac6292f652af9faba9a762f76
be52c91e51576265732129eb95bbac1a574aff4335a0997cb
3ef58bda802dfe39eff9faa95a71c78473ffe452f933e161e
36dd4f440284413b138b5e8f422de3456a5d11c6b54a4ca5d
a9fc23ed7a472591411313ff7b1adbeddaf87dcfab6bf58dd
ed8137acca01a8a91e048e245a07181026002a08a28818b14
a3a6fc227ce9e720266fe806441a61220382e061031e070b8
```

Fig. 4. Fraction hexadecimal of the audio.

Each block hexadecimal value is prefixed with “http” at front and suffixed with “.com” at back. A tinyurl website is used to generate the shorten URL. The last eight characters of each shortened URL are used and concatenated. The entire string is “sn3mjvze5n6un8s2mwwefvc6”.

The ASCII is a character set with 128 characters, each represented by seven bits. It includes the numbers 0-9, both upper and lower case English letters from A to Z, and a selection of special characters. The string “sn3mjvze5n6un8s2mwwefvc6” is consider input plaintext which is shuffled using the Caesar cipher. The key for the Caesar cipher is “-5” and modulus 128, resulting in the output ciphertext “ni.hequ‘0i1pi3n-hrr‘aq^1”. This output ciphertext is used to generate the QR code. The generated QR code is in “.png” format, with dimensions of 414 x 414, and a file size of 1.29kb bytes, as shown in Fig. 5.

Fig. 5. QR code consists shuffled URL string.

Next, the generated QR code is embedded into an image (see Fig. 6a) and results the embedded image (see Fig. 6b). The size of the original image and embedded image is in “.jpg” format, with dimensions of 5109 x 3400 and a file size of 1.71mb. With the naked eye, it is hard to notice the difference between the original image and the stego-image, as only a single bit is changed per pixel, as shown in Fig. 6.

To retrieve the audio one first needs to extract the QR code from the embedded image. The extracted QR code, shown in

Algorithm 1 AudioMag steganography - data hiding

```
Require: Input: Audio, Original image.
Audio is converted into hexadecimal, the size of audio hexadecimal is denoted as $l_a$.
The audio hexadecimal $l_a$ is divided into block, each block hexadecimal is 32 kilo byte (kb). The last block can be less than 32kb.
Each block's hexadecimal value is prefixed with "http" at front and suffixed with ".com" at back. The entire string is then used to generate a shortened uniform resource locator (URL).
The last eight characters of each shortened URL are used, concatenated, and shuffled. The shuffling step is based on the Caesar cipher, with the modulus size based on the total number of ASCII characters, which is 128.
The shuffled string is converted into a quick response code (QR code). The QR code is in two dimensions, the size is $M$ x $N$, where the values of $M$ x $N$ are fixed as 414 x 414. Maximum string embedded into the QR code is 140 bytes.
Flatten the QR code. Converts the flatten QR code value into binary, we denoted it as $qr$.
Retrieves the original image. Note that the original image must be at least eight times larger in dimensions than the QR code.
Obtains red, green and blue (RGB) of original image.
Transforms RGB of original image into brightness and color (YUV).
Decomposes the YUV original image into four sub-images via discrete wavelet transform (DWT). The sub-images are known as approximation (cA), horizontal (cH), vertical (cV), and diagonal (cD) respectively. The size for each sub-image is $\frac{M}{2}$ x $\frac{N}{2}$.
The cA is further divided into 4 x 4 block via discrete cosine transform (DCT), we denote the number of block as $b$.
for $i \leftarrow 1$ to $b$ do
    The block of cA, $b_i$ is decomposed via singular value decomposition (SVD), the return value ($S$) contains singular values. One bit data of $qr_i$ is embed into the first singular value.
end for
Reconstructs cA via inverse discrete cosine transform (IDCT).
Converts the embedded cA together with cH, cV and cD into YUV data via inverse wavelet transform (IDWT).
Transforms YUV data into RGB.
Output: Stego-image.
```

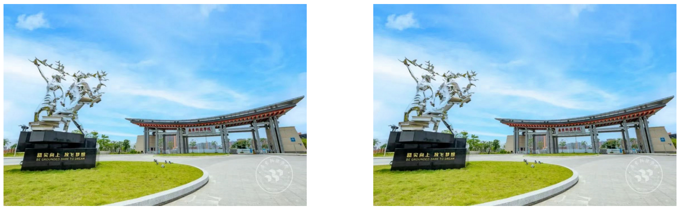

(a) Original image. (b) Stego-image.

Fig. 6. Original image and stego-image.

Fig. 7, is in ".png" format with dimensions of 414 x 414 and a file size of 83.6kb. One may observe noise, but the QR code content remains.

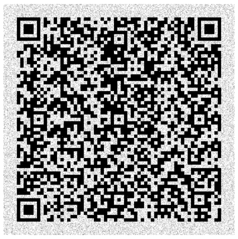


Fig. 7. Extracted QR code from stego-image.

The ciphertext "ni.hequ‘0i1pi3n-hrr‘aqˆ1" is decoded using Caesar cipher with decoded key of"+5" and modulus 128, resulting in the string "sn3mjvze5n6un8s2mwwefvc6". This string is divided into three sub-strings, with each sub-string containing eight characters. Each sub-string is prefixed with "https://tinyurl.com/" to complete as URL. Place the URL in a web browser and navigate to it; it will display a long URL string. Remove the prefixed "http" and suffixed ".com", and you will get the hexadecimal value. Repeat this process for each sub-string. Finally, concatenate all the hexadecimal values to recover the audio.

A. *Robustness of Stego-image*

This section examines the robustness of stego-images, as shown in Table III. The rationale behind this experiment is to determine if stego-images can be sabotaged, yet still be able to extract the embedded QR code. To assess this risk, two potential scenarios were investigated: 1) when the stego-image is masked, and 2) when the stego-image is cropped. The findings show that the QR code can be extracted from manipulated stego-images, even when subjected to higher levels of noise compared to the results depicted in Fig. 7. It is noted that the information in the QR code remains readable. This means the proposed method maintains the integrity of the embedded data indirectly.

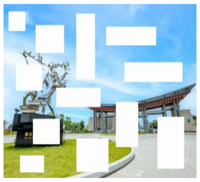
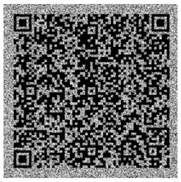
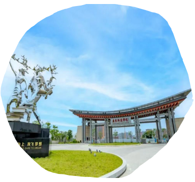
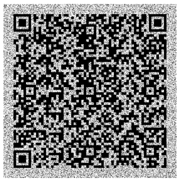
V. Result and Discussion

The proposed method for concealing secret audio in an image begins by converting the audio into hexadecimal values, which are then split into fixed 32kb strings. The strings are hidden using shortened URLs, with the last eight characters of each URL concatenated. This means that each shortened URL's size is 8 bytes. The concatenated string is shuffled using a Caesar cipher and the resulting shuffled string is used to create a 414 x 414 QR code. Lastly, the QR code is embedded into the image. Note that the image must be at least eight times larger than the QR code; therefore, the minimum size of the image is 3319 x 3319.

Algorithm 2 AudioMag Steganography - Data Extraction

```
Require: Input: Stego-image.
Obtains RGB of stego-image.
Decomposes the RGB stego-image into four sub-images via DWT. The sub-images cA, cH, cV, and cD respectively. The size for each sub-image is $\frac{M}{2}$ x $\frac{N}{2}$.
The cA is further divided into 4 x 4 blocks via DWT, we denoted the block as $b$.
for $i \leftarrow 1$ to $b$ do
    The block of cA, $b_i$ is decomposed via DCT.
    Flatten the output of DCT into one-dimensional array and then indexed it, such that $D_j$, where $j \in 0, 1, 2, 3$.
    The indexed array is rearranged back into a 4*4 shape.
    Performs SVD on the rearranged array.
    Obtains first singular value ($S_0$) by extracting the first value of $S_0$.
    Calculates binary data extraction.
    for $j \leftarrow 1$ to 3 do
        Obtains first singular value ($S_j$) by extracting the first value of $S_j$.
        Calculates binary data extraction and merge with $S_{j-1}$.
    end for
    Calculate the average extracted bit information.
end for
Each mean bit block is multiplied by 255 to convert the normalized pixel value into the actual pixel representation so that it can correctly represent color or brightness in the image.
Retrieves the QR code.
Read the string in the QR code.
Decodes the string using Caesar cipher.
The decipher string is divided into eight characters sub-strings. We denoted the number of sub-strings as $ls$. Let's an empty string is $string_{sub}$.
for $i \leftarrow 1$ to $ls$ do
    Each sub-string is prefixed with "https://tinyurl.com/" and navigated to in a web browser.
    The web browser will show the long URL string, removes the prefixed "http" and suffixed ".com". We denoted it as $URL'$.
    Concatenate the $URL'$ into $string_{sub}$, such that $string_{sub}$ = $string_{sub}||URL'$.
end for
Convert the $string_{sub}$ into audio.
Output: Audio.
```

TABLE III. ROBUSTNESS OF STEGO-IMAGE

Case	Stego-image	QR code
1		
2		

The 414 x 414 QR code can only contain a maximum of 140 bytes of a string. According to the simulation, the audio file size is approximately 41kb per second. Once the audio is converted into hexadecimal values, the total size of the hexadecimal values is 82kb, resulting in a total of 24 bytes for three different URLs. This implies that the maximum amount of secret audio that can be embedded in a single image of minimum size 3319 x 3319 is roughly six seconds worth of secret audio. The six seconds secret audio consists of 492kb hexadecimal values, resulting in a total of 128 bytes for three different URLs.

The proposed method embeds the QR code into the an image via DWT, DCT, and SVD, based on the simulation, the original image (see Fig. 6a) appears similar to the stego-image (see Fig. 6b). This ensures that the hidden QR code (also known as converted secret audio) is invisible to the naked eye.

However, a flaw occurs when extracting the QR code from the stego-image (see Fig. 7); noise appears in the extracted QR code due to its high robustness. High robustness in this case means that even if the stego-image is corrupted, we can still retrieve the QR code as being discussed in Section IV-A. The noise, however, does not affect the efforts to retrieve the shuffled string stored in the QR code. Therefore, we can successfully decode the shuffled string and ultimately recover the secret audio.

VI. CONCLUSION AND FUTURE WORK

Audio steganography is a covert communication technique that involves embedding secret information within an audio signal, making it undetectable to the human ear. Image steganography is a technique used to hide secret messages within an ordinary images. Our innovative approach, AudioMag, converts audio data into hexadecimal format and then encodes it into a QR code. This QR code is subsequently inserted into an image, offering a beyond than traditional secure way to transmit hidden information, thereby preventing potential attackers from detecting the concealed audio message.

The effectiveness of this method lies in the fact that the proposed algorithm has been successful in converted audio into the QR code while preserving the similarity between the stego-image and the original image. This added layer of security helps to ensures that the hidden audio message remains undetectable by eavesdroppers. Hence, preserving the confidentiality of the transmitted data over digital channels.

However, one limitation of our design is that it only allows for embedding audio up to a maximum of six seconds. Therefore, as future work, we plan to design the QR code dimensions in a more flexible manner, allowing for the embedding of longer strings. This means that we will be able to record arbitrary lengths of audio to embed into the image.

Acknowledgment

The authors would like to thank Guangdong University of Science & Technology, China and Rabdan Academy, United Arab Emirates.

References

[1] K. Sutradhar, B. G. Pillai, R. Amin and D. L. Narayan, *A survey on privacy-preserving authentication protocols for secure vehicular communication*, Computer Communications, 2024.

[2] K. Moldamurat, Y. Seitkulov, S. Atanov, M. Bakyt and B. Yergaliyeva, *Enhancing cryptographic protection, authentication, and authorization in cellular networks: a comprehensive research study*, International Journal of Electrical and Computer Engineering (2088-8708), 14(1), 2024.

[3] J. Zhang, *The Application of Data Encryption Technology in Computer Network Security*, Transactions on Computer Science and Technology, 11(1), 2024.

[4] S. Pramanik, *A new method for locating data hiding in image steganography*, Multimedia Tools and Applications, 83(12), pp. 34323-34349, 2024.

[5] B. Song, P. Wei, S. Wu, Y. Lin and W. Zhou, *A survey on Deep-Learning-based image steganography*, Expert Systems with Applications, pp. 124390, 2024.

[6] H. J. Asghar, B. Z. H. Zhao, M. Ikram, G. Nguyen, D. Kaafar, S. Lamont and D. Coscia, *Use of cryptography in malware obfuscation*, Journal of Computer Virology and Hacking Techniques, 20(1), pp 135-152, 2024.

[7] C. D. Vleeschouwer, J. F. Delaigle and B. Macq,*Circular interpretation of bijective transformations in lossless watermarking for media asset management*, IEEE Trans. Multimedia , 5(1), pp. 97 - 105, 2003.

[8] C. W. Honsinger, P. Jones, M. Rabbani and J. C. Stoffel *Lossless recovery of an original image containing embedded data*, US Patent: 6,278,791, 2004.

[9] S. Chandran and B. Khoushik, *Performance Analysis of LSB DCT and DWT for Digital Watermarking Application using Steganography*, IEEE Int. Conf. Electr. Electron. Signals Commun. Optim, 15(978-1-4799-7678-2), pp. 2-6, 2015.

[10] S. K. Moon and R. S. Kawitkar, *Data security using data hiding*, In International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), 4 pp. 247-251, 2007.

[11] C. Kim, L. Cavazos Quero, K. H. Jung and L. Leng, *Advanced Dual Reversible Data Hiding: A Focus on Modification Direction and Enhanced Least Significant Bit (LSB) Approaches*, Applied Sciences, 14(6), pp. 2437, 2024.

[12] P. Naveen and R. Jayaraghavi, *Image Steganography Method for Securing Multiple Images using LSB–GA*, Wireless Personal Communications, 135(1), pp. 1-19, 2024.

[13] B. Datta, P. Pal and S. K. Bandyopadhyay, *Robust multi layer audio steganography*, In 2015 Annual IEEE India Conference (INDICON), IEEE, 1-6, 2015.

[14] M. H. Sayed and T. M. Wahbi, *Information Security for Audio Steganography Using a Phase Coding Method*, European Journal of Theoretical and Applied Sciences, 2(1), pp. 634-647, 2024.

[15] R. M. Nugraha, *Implementation of direct sequence spread spectrum steganography on audio data*, In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, IEEE, pp.1-6, 2011.

[16] K. Brandenburg and H. Popp, *MPEG layer-3*, MEBU Technical review, pp.1-15, 2000.

[17] I. Haverkamp and D. K. Sarmah, *valuating the merits and constraints of cryptography-steganography fusion: a systematic analysis*, International Journal of Information Security, pp.1-29, 2024.

[18] A. M. Khalaf and K. Lakhtaria, *A review of steganography techniques*, In AIP Conference Proceedings, 3051(1), 2024.

[19] M. Kimlyk and S. Umnyashkin, *Image denoising using discrete wavelet transform and edge information*, In 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp. 1823-1825, 2018.

[20] J. Patel, D. Tailor, K. Panchal, S. Patel, R. Gupta and M. Shah, *All phase discrete cosine biorthogonal transform versus discrete cosine transform in digital watermarking*, Multimedia Tools and Applications, 83(6), pp. 16121-16138, 2024.

[21] B. Mahaboob, D. Leela, B. Kothuru, G. B. Prakashand, A. Raheem and S. Nandakishore, *A note on singular value decomposition*, AIP Conference Proceedings, 3231(1), 2024.

[22] W. Stallings, *Cryptography and network security:principles and practices*, Pearson Education India, 2006.