Indonesian Journal of Electrical Engineering and Computer Science
Vol. 13, No. 1, January 2019, pp. 147~154
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v13.i1.pp147-154

Developing audio data hiding scheme using random sample bits with logical operators

Mohammed Hatem Ali Al-Hooti[1], Tohari Ahmad[2], Supeno Djanali[3]
[1,2,3]Department of Informatics, Institut Teknologi Sepuluh Nopember, East Java, Indonesia
[1]Faculty of Computer Science and Information Technology, Sana'a University, Sana'a, Yemen

Article Info

Article history:

Received Jul 6, 2018
Revised Nov 2, 2018
Accepted Nov 19, 2018

Keywords:

Data hiding
Data protection
Information security
Secret data
Steganography

ABSTRACT

Sharp development progress of information technology has affected many aspects including data security. This is because classified data are often transferred between systems. In this case, data hiding exists to protect such data. Some methods which have been proposed, however, are not yet optimal concerning the amount of the secret and the quality of the resulted stego data. In this paper, we explore an audio file as the medium to carry the secret data which has been extracted into binary. Before the process begins, the cover is converted to binary and each sample's bits are divided into two groups, one is used as the location of the embedded 4 bits whereas the second part locates the two bits that are randomly selected as the key. The experimental results have validated that the capacity is high and there is no much impact on the quality. Moreover, compared to the current LSB methods the security is exceedingly enhanced.

Corresponding Author:

Mohammed Hatem Ali Al-Hooti,
Department of Informatics,
Institut Teknologi Sepuluh Nopember,
Surabaya, Sukolilo, 60111, East Java, Indonesia.
Email: moh_hat84@yahoo.com

1. INTRODUCTION

Presently, multimedia network and communication is sharply progressing. Digital communication has become so significant for many organizations. It supports even daily life activities and communications [1, 2]. For instance, we transfer medical images, digital audios, music, videos, digital books, etc. This transmission occurs on public networks such as the internet which means, anyone can easily access them. As a result, those files might be simply forged or manipulated. To minimize this threat, information security has come to protect data against some issues, such as piracy, data forgery, etc [3, 4]. In this field, data hiding acts as a recent and well-known research topic which is specified into the following topics such as steganography that is concerning much to protect the communications privacy [5, 6]. Steganography aims to hide a secret message using a harmless cover, so no other party can detect this secured message. In the past, data hiding started in conventional techniques such as invisible inks, spread-spectrum communications, and covert channels [3, 7]. Currently, this can be carried out by hiding the secret inside another cover file [8]. This medium might be an image, audio, video, or text [9-13]. Watermarking [14-16] conceals secret information such as signatures within any media without having a perceptual distortion [17, 18].

Audio data hiding schemes are classified into two types: lossless and lossy. The lossless algorithm considers retrieving both the secret message and the original cover file. Lossy method cares only on recovering the secret information [19]. There are many parameters which are used to evaluate the performance of audio steganogrphy schemes. Not all of them should be met, since each implementation may

have different purposes. We can summarize some significant ones in Figure 1, which can be described as follows [11, 20]:

a) Capacity [21]: this represents the amount of the secret message that can be carried by the cover file.
b) Imperceptibility (stego file quality): it considers the quality of the stego audio file after being merged with the secret message.
c) Secrecy (Temper confrontation): this means that the concealed message is secured and it is difficult for unauthorized parties to access it. As an example, this can be a secret key that is shared between the sender and the receiver.
d) Computational complexity: it concerns the processing time taken by embedding and extraction operations, which should not be long. Especially, this applies to constrained devices, such as smartphone.
e) Robustness: this enables us for recovering the secret data from the stego audio file even if it is being attacked.

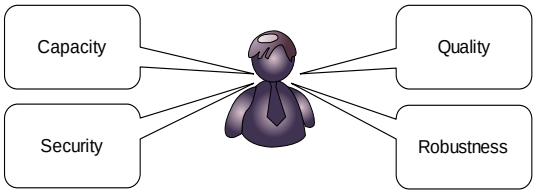

Figure 1. Steganography measurement parameters

In audio data hiding systems, it is essentially required to have high payload while maintaining the quality, and at the same time the existence of the message should not be easily detected. Thus, many research papers are still looking forward to accomplish these issues. This is because HAS (Human Auditory System) is an obstacle which stands against data hiding algorithms since it has a wide dynamic and differential range [22].

Audio data hiding techniques using Least Significant Bit (LSB) substitution techniques [5] have much of simplicity and popularity. They can have a great performance in terms of the embedding capacity and computational complexity. On the other hand, their security level against hackers is low and not comparable. The secret message can be detected and even obtained easily. Some researchers have solved this problem by combining LSB steganography with a cryptographic scheme in order to increase the security level. However, this is not a satisfactory solution due to the wasted computational complexity and lost bandwidth storage.

In this paper, we propose a new scheme that explores digital operators (e.g. XOR & NOT gates) merged by an LSB substitution technique. This research is applied in the temporal domain, the audio cover is extracted into samples and are converted into binary. In this research, each sample is represented by 16 bits. In the embedding process, the positions of two random bits are selected within the interval (5, 16) as a key. According to that position, two bits from each sample are selected to be used as a guide that helps recovering the secret message. The embedding is performed by applying XOR gate on the secret bits and the selected sample's bits. The four XOR outputs are compared with their corresponding sample's least significant bits. If they are not matched then the LSBs are changed by using an inverter (NOT) logical operator. Otherwise, there is no change. Normally, this helps increasing the difficulty of detecting the secret message.

The rest of this paper is structured as the following. In Section 2, we describe some existing audio steganography schemes, and determine their strengths and weaknesses. The detail of the proposed method is provided in Section 3. Section 4 presents the experimental results. The conclusions are drawn in Section 5.

2. RELATED RESEARCH

This section provides the basic theory about some audio data hiding methodologies as follows. Audio data hiding research has been popular due to its strength, for example its simplicity. Generally, audio amplitude values are represented in real numbers, so during the embedding the secret bits does not cause much noise [23]. It is normal that each scheme has its own pros and cons. All are measured and evaluated using the aforementioned data hiding criterias.

Binny et al. [24] has proposed a method that uses LSB of each sample to hide the secret information while maintaining the stego file imperceptibility. Nevertheless, they fail to get the secrecy and the robustness

and could not fully recover the original audio file. In the next research, Al-Hooti et al. [20] design an algorithm based on modulus function and location maps. This affects the quality and the reversibility, which are successfully obtained. In contrast, this work is fragile to stand against hackers' detection.

Chowdhury et al. [25] has worked on a method based on LSB and it is combined with a cryptographic scheme called pattern matching algorithm. This is done in order to encrypt the secret text so its secrecy is guaranteed. By using a similar idea, in [26] the authors propose an algorithm that implements LSB-based data hiding in an android environment. This scheme starts encrypting the secret message, and put it in MP3/WAV audio. Meanwhile, the cover audio is divided into regions. After that, the secret message is embedded by modifying these regions. However, the embedding consumes much time to process.

Authors of [21] has combined Generalized Difference Expansion and Reduced Difference Expansion to produce a new scheme that is able to hide more payload and maintain the quality. However, this work does not pay attention to the secrecy. This method has increased the capacity, but their imperceptibility may not be high enough.

The method described in [27] has proposed a reversible steganographic scheme using probabilistic DNA-XOR secret sharing-based color image. Similar to other research, each DNA-XOR truth table value is evaluated by using PSNR. The input that has the highest PSNR value is considered for the secret sharing. These inputs are hidden in the cover image which are used as the key. This work divides the secret shares into three groups: R, G, B where the secret message is embedded.

As it is mentioned, we focus on the temporal domain and LSB audio data hiding represents high percentage from the currently published methods in this domain. Moreover, LSB-based methods simply increase or decrease one value from the LSB value of each sample [28]. However, these schemes are not able to fulfill the other steganographic requirements such as secrecy and robustness. Also, the secret message might be lost easily if the stego file being attacked or compressed [23].

3. RESEARCH MENTHOD

This section explicitly describes the proposed algorithm of this research. Commonly, LSB-based embedding methods have been proposed in many research papers [24]. However, the researchers do not pay much attention to the secrecy (Temper confrontation) feature. As a result, those methods are less useful in the current practical data hiding applications. This is because of the secret data extraction simplicity and detectability [24]. Currently, secured LSB schemes are only the ones which are combined with one of the encryption works [25]. This costs much in the computational complexity. In accordance, we mainly focus on increasing the security parameter.

Normally, combining data hiding with crypto system [26] can increase the security level but it requires much computational complexity. According to this condition, we design an LSB-based audio data hiding method by considering the payload, quality, secrecy, and computational complexity aspects. This work does not pay attention to the robustness. Different from the previous research [22, 24], we propose a new technique that uses two logical operators (XOR and NOT) based on LSB substitution. This algorithm mainly hides four bits in each sample which is previously converted into 16 bits. These bits are divided into two parts as indicated in Figure 2. The first part is used to carry the outputs of each XOR operation, and the random two selected bits are chosen in the second part that contains 12 bits.

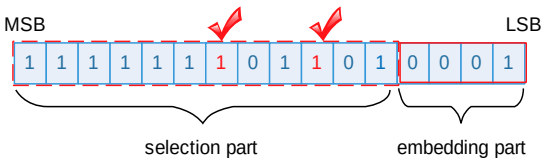

Figure 2. The use of audio sample represented in bits

3.1. The Procedure of Embedding

The embedding operation is illustrated as shown in Figure 3 and as summarized in the next five steps as follows:

Step 1: read the secret message and consider each four bits ($b_{i,1}$, $b_{i,2}, b_{i,3}, b_{i,4}$) to be embedded within each sample.

Step 2: read the samples of audio file (excluding the first 40 samples) where each sample is in the interval [0, 65535], and convert each sample S_i into 16 bits. Every sample is divided into two parts. The first

one contains 4 bits which are used to be changed. The second part contains 12 bits that two of them $A_{i,j}$ and $B_{i,w}$ are selected randomly to be processed with the first and the third secret bits using XOR gate (denoted in this paper by ⊕). It is worth to note that the position of these two chosen bits are sent to the destination as a key.

Step 3: consider each four secret bits and apply the XOR logical operator as shown in (1). Here, $O_{i,1}$ is the output of the first XOR-ing that is applied between the first secret bit and the first selected sample binary digit where i is an index that represents the number of the used samples. Both the two values j and w are in the range of 5 and 16 which denotes the position of the random selected bits $A_{i,j}$ and $B_{i,w}$, respectively. It is compulsorily that both random position values should not be the same. The parameter $O_{i,2}$ carries the output value of XOR-ing the second secret bit $b_{i,2}$ and the first output $O_{i,1}$. To increase the complexity of compromising the method, the second random sample bit $B_{i,w}$ is XOR-ed with the third secret bit $b_{i,3}$. After that, we implement XOR to the output $O_{i,3}$ with fourth secret bit $b_{i,4}$.

$$\begin{aligned} O_{i,1} &= b_{i,1} \oplus A_{i,j} \\ O_{i,2} &= b_{i,2} \oplus O_{i,1} \\ O_{i,3} &= b_{i,3} \oplus B_{i,w} \\ O_{i,4} &= b_{i,4} \oplus O_{i,3} \end{aligned} \tag{1}$$

Step 4: embed all the outputs $O_{i,1}, O_{i,2}, O_{i,3}$, and $O_{i,4}$ into every sample that is located in the first part. Based on (2), the outputs are compared with each first part of sample bit $T_{i,1}, T_{i,2}, T_{i,3}, T_{i,4}$. If they are not equal, then each sample bit is flipped using NOT logical operator. Otherwise, there is no change.

$$\left.\begin{aligned} T_{i,1} &= \begin{cases} T_{i,1} = NOT(T_{i,1}) & ,if\ T_{i,1} = O_{i,1} \\ T_{i,1} = T_{i,1} & ,if\ otherwise \end{cases} \\ T_{i,2} &= \begin{cases} T_{i,2} = NOT(T_{i,2}) & ,if\ T_{i,2} = O_{i,2} \\ T_{i,2} = T_{i,2} & ,if\ otherwise \end{cases} \\ T_{i,3} &= \begin{cases} T_{i,3} = NOT(T_{i,3}) & ,if\ T_{i,3} = O_{i,3} \\ T_{i,3} = T_{i,3} & ,if\ otherwise \end{cases} \\ T_{i,4} &= \begin{cases} T_{i,4} = NOT(T_{i,4}) & ,if\ T_{i,4} = O_{i,4} \\ T_{i,4} = T_{i,4} & ,if\ otherwise \end{cases} \end{aligned}\right\} \tag{2}$$

Step 5: build the stego audio file based on the new modified sample values. Then it is sent to the receiver via any public network.

As an example for this embedding procedure, assume we have 6 audio samples which are converted into binary as presented in Figure 4. These two samples are subdivided into two parts, namely the embedding part which is represented by four least significant bits, and the second part that is used to generate the two randomly chosen bits. Suppose the sender chooses the bit number 5 as first position and bit number 10 as the second position. Then, based on (1) and (2) the obtained results are presented in the Table 1 whereas successfully the stego four LSB bits are replaced by the four outputs (O_1, O_2, O_3, O_4) of the Exclusive OR operation.

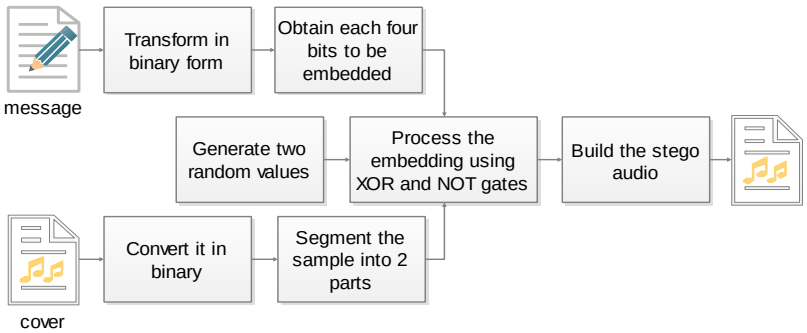

Figure 3. The embedding process

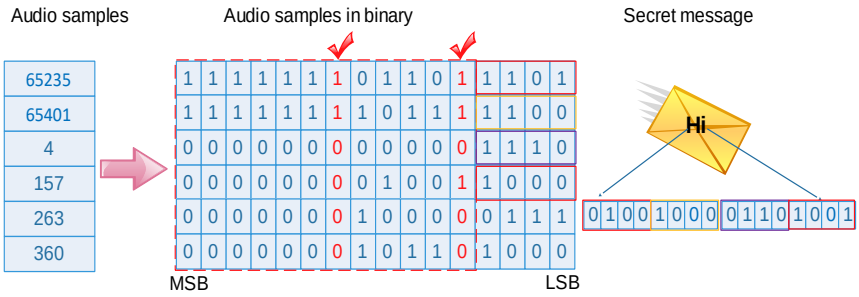

Figure 4. Example of the embedding process

Table 1. The Embedding Example Outputs

A	b_1	O_1	b_2	O_2	b_3	B	O_3	b_4	O_4
1	0	1	1	0	0	1	1	0	1
1	1	0	0	0	0	1	1	0	1
0	0	0	1	1	1	0	1	0	1
1	1	0	0	0	0	0	0	1	1

3.2. The Extraction Procedure

It is common that the extraction process is just inversing the embedding operation. The receiver takes the delivered stego file and two chosen key values that indicates the positions of the two selected sample bits. After that, this procedure is performed as shown in Figure 5 and the following steps:

Step 1: extract the stego audio file into samples which are converted into binary. Correspond to the embedding process, the first 40 samples are ignored.

Step 2: insert the two key values j, w into the extraction system to determine the positions of the randomly selected sample two bits $A'_{i,j}$, $B'_{i,w}$.

Step 3: apply XOR operator between the first selected sample bit $A'_{i,j}$ and the first LSB of the stego sample. As a results, the first secret bit $b'_{i,1}$ is obtained. This first secret bit is also XOR-ed with the second LSB of the stego sample in order to obtain the second secret bit $b'_{i,2}$. After that, the third secret bit $b'_{i,3}$ is obtained by XOR-ing the third LSB of the same sample $T'_{i,3}$ and the second selected random binary digit $B'_{i,w}$. Finally, getting the fourth secret binary digit $b'_{i,4}$ XOR is applied to the obtained third secret digit $b'_{i,3}$ and the sample fourth LSB $T'_{i,4}$ as in Equation (3).

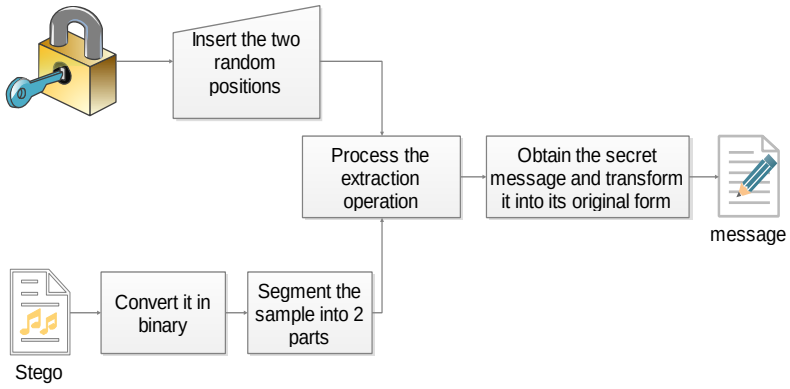

Figure 5. The extraction operation

$$
\begin{aligned}
b'_{i,1} &= T'_{i,1} \oplus A'_{i,j} \\
b'_{i,2} &= b'_{i,1} \oplus T'_{i,2} \\
b'_{i,3} &= T'_{i,3} \oplus B'_{i,w} \\
b'_{i,4} &= b'_{i,3} \oplus T'_{i,4}
\end{aligned}
\tag{3}
$$

Step 4: finally, combine all of the obtained secret bits $b'_{i,1}, b'_{i,2}, b'_{i,3}$, and $b'_{i,4}$ which are obtained from each sample in a vector and convert that vector into its original form that might be text, audio, or image.

4. RESULTS AND ANALYSIS

In this research, the secret message is generated using Randi function in (Matlab-2017a). For the audio cover, we use 10 audio *.wav files. Each cover file has its own duration and size, however, all are sampled in 16 bit depth [29-31]. It is intended that the files are chosen in different cases e.g. speech, music. Some are in high level and others has low sound level. The number of samples (excluding the 40 ones which are ignored since they can be impacted) is varied. This simulation is done in a computer laptop Asus that has core i3 processor, 6 GB RAM, and 500 GB HD. Four scenarios are conducted to evaluate four features as follows:

First, capacity or embedding rate that is represented as shown in Table 2. This method is able to hide secret bits 4 times the number of the cover samples which means each sample is able to carry 4 bits wherein the existing works [20, 24] each sample can only carry one bit. However, the more the inserted data causes slight impact on the quality of the stego file as indicated in both Table 3 and Table 4.

Second, imperceptibility which is measured by using Signal to Noise Ratio (SNR) and Peak Signal to Noise Ratio (PSNR) as in Equation (4) and (5), respectively [8]. The variable t denotes the number of samples, and C means the cover audio where the stego file is presented by S while μ represents the maximum value of audio sample. The higher the value of SNR and PSNR, the greater the similarity between the cover and the stego files.

$$SNR = \frac{\sum_{i=1}^{t} C_i^2}{\sum_{i=1}^{t} (C_i - S_i)^2} \quad (4)$$

$$PSNR = \frac{\mu^2}{\sum_{i=1}^{t} (C_i - S_i)^2} \quad (5)$$

Third, the security parameter is improved in this paper. This is because in most existing modulus and LSB works such as [20, 24], the secret message recovery is simply obtained by taking the LSB of each stego sample. In contrast, this work hides the 4 outputs that are obtained from XOR-ing gates in each sample. These outputs are not the exact secret bits. Therefore, suppose the attacker knows our methodology, he/she needs to conduct 16 trials for each sample in order to obtain the exact 4 secret bits. In this paper, we designed an equation that is presented in (6). It is used to measure the amount of trials that are required for obtaining the secret message. Here, P is the number of times needed to illegally obtain the secret bits, β represents the four selected bits, q is the possibilities of each secret bit (since it is in binary so q is always 2), and C is symbolized as the number of the stego samples that are embedded. As a result, hackers need 16 trials for each sample in order to obtain the exact four secret bits. This is multiplied by the number of the embedded samples. Therefore, the more the amount of hidden secret bits the higher the difficulty for hackers to detect it. Based on Equation (6), let the embedded stego samples amount is 661500 and it is known in each sample 4^2 trails are needed in order to obtain the 4 secret bits, so the possibility of illegally gaining the whole secret message is $(4^2) * 661500 = 10584000$ times, whereas in the existing works [20, 24] allow anyone to directly extract the secret message by obtaining the LSB of each sample.

$$P = (\beta^2) * C \quad (6)$$

Table 2. The Embedding Rate based on Four Scenarios

No	File name	No. Samples	100% payload	75% payload	50% payload	25% payload
1	001_01	752068	3008272	2256204	1504136	752068
2	001_02	815520	3262080	2446560	1631040	815520
3	AM2_35U_5	5783377	23133508	17350131	11566754	5783377
4	Acomic	661500	2646000	1984500	1323000	661500
5	001_03	997119	3988476	2991357	1994238	997119
6	Blues	661500	2646000	1984500	1323000	661500
7	Classical	661500	2646000	1984500	1323000	661500
8	02.School Boy-9	1763999	7055996	5291997	3527998	1763999
9	Jazz	661500	2646000	1984500	1323000	661500
10	Rock	661500	2646000	1984500	1323000	661500

Table 3. The Stego Audio Quailty based on Signal to Noise Ratio

No	File name	SNR values in dB			
		In100%	In75%	In50%	In25%
1	001_01	77. 02	78.27	80.03	83.02
2	001_02	76.97	78.25	80.02	83.03
3	AM2_35U_5	76.62	77.86	79.61	82.58
4	Acomic	74.46	75.71	77.47	80.48
5	001_03	77.02	78.28	80.05	83.07
6	Blues	74.05	75.30	77.06	80.08
7	Classical	74.92	76.17	77.92	80.94
8	02.School Boy-9	76.48	77.72	79.48	82.49
9	Jazz	75.94	77.19	78.96	81.97
10	Rock	76.10	77.35	79.11	82.12

Table 4. The Stego Audio Quailty based on Peak Signal to Noise Ratio

No	File name	PSNR in dB			
		In100%	In75%	In50%	In25%
1	001_01	80.04	81.29	83.04	86.04
2	001_02	79.98	81.26	83.03	86.04
3	AM2_35U_5	80.02	81.27	83.01	85.99
4	Acomic	80.05	81.30	83.06	86.07
5	001_03	80.02	81.28	83.05	86.07
6	Blues	80.04	81.29	83.04	86.06
7	Classical	80.05	81.30	83.06	86.08
8	02.School Boy-9	80.05	81.29	83.05	86.06
9	Jazz	80.05	81.30	83.06	86.08
10	Rock	80.06	81.31	83.06	86.07

Fourth, computational complexity, this method costs in average nearly 30.20 seconds to be implemented. This fast computational process allows us to install the proposed system even in smartphones. Based on the correlation method [20] the full recovery of the secret message is 100% guaranteed.

In brief, this work has successfully obtained great capacity and maintained stego file quality. The secrecy is also high in a fast computational complexity performance. In different words, this algorithm can enhance the capacity, secrecy and complexity while maintaining the quality. On the other hand, this work is not robust to stand against any attacks e.g. WAV to MP3 compression. Therefore, any attack on it can cause losing the secret message which can be constructive in some cases.

5. CONCLUSION

Audio data hiding has its security significance in many fields such as banking, military, and airlines systems. In this paper, we present a scheme that focuses on improving the capacity, complexity, and the secrecy criteria of an LSB substitution algorithm without being combined with a cryptography scheme. The probability of detection is minimized by using XOR gates whereas the computational complexity performance is achieved. In the worst case, this scheme consumes around 30.20 seconds wherein cryptographic schemes can consume much of computational complexity and waste much of storage. As comparison between our method and normal LBS works we are able to achieve more embedding rate while maintaining the quality. Moreover, this method can yield a great secrecy compared to the existing related schemes. In the future, we are planning to increase the embedding rate so each sample can carry one byte. We believe this will not only increases payload but also it will further complicate the secret message detection.

ACKNOWLEDGEMENTS

The authors would like to express their warmest gratitutes to Institut Teknologi Sepuluh Nopember ITS, Surabaya, Indonesia for the support that has been given to this research.

REFERENCES

[1] Bhuiyan SSN, et al. An Improved Image Steganography Algorithm based on PVD. Indonesian Journal of Electrical Engineering and Computer Science. 2018;10:569~77.

[2] Din R, et al. A Comparative Review on Data Hiding Schemes. Indonesian Journal of Electrical Engineering and Computer Science. 2018;11:768~74.

[3] Al-Dmour H, Al-Ani A. A steganography embedding method based on edge identification and XOR coding. Expert systems with Applications. 2016;46:293-306.
[4] Ahmad T, Fiqar TP. Enhancing the Performance of Audio Data Hiding Method by Smoothing Interpolated Samples. International Journal of Innovative Computing, Information and Control. 2018;14(3):757-79.
[5] Begum MB, Venkataramani Y. LSB based audio steganography based on text compression. Procedia Engineering. 2012;30:703-10.
[6] Hussain HS, et al. The Embedding Performance of StegSVM Model in Image Steganography. Indonesian Journal of Electrical Engineering and Computer Science. 2018;12:233~8.
[7] Al Huti MHA, et al., editors. Increasing the capacity of the secret data using DEpixels blocks and adjusted RDE-based on grayscale images. Information & Communication Technology and Systems (ICTS), 2015 International Conference on; 2015: IEEE.
[8] Mansor FZ, et al. An Antonym Substitution-based Model on Linguistic Steganography Method. Indonesian Journal of Electrical Engineering and Computer Science. 2018;12:225~32.
[9] Devi RR, Pugazhenthi D. Ideal Sampling Rate to Reduce Distortion in Audio Steganography. Procedia Computer Science. 2016;85:418-24.
[10] Rahmani P, Dastghaibyfard G. An efficient histogram-based index mapping mechanism for reversible data hiding in VQ-compressed images. Information Sciences. 2017.
[11] Shahadi HI, et al. Concurrent hardware architecture for dual-mode audio steganography processor-based FPGA. Computers & Electrical Engineering. 2016;49:95-116.
[12] Sloan T, Hernandez-Castro J. Forensic analysis of video steganography tools. PeerJ Computer Science. 2015;1:e7.
[13] Yan F, et al. Flexible representation and manipulation of audio signals on quantum computers. Theoretical Computer Science. 2017.
[14] Hu H-T, et al. Synchronous blind audio watermarking via shape configuration of sorted LWT coefficient magnitudes. Signal Processing. 2018;147:190-202.
[15] Hua G, et al. Twenty years of digital audio watermarking—a comprehensive review. Signal Processing. 2016;128:222-42.
[16] Lei BY, et al. Blind and robust audio watermarking scheme based on SVD–DCT. Signal Processing. 2011;91(8):1973-84.
[17] Kar DC, Mulkey CJ. A multi-threshold based audio steganography scheme. Journal of information security and applications. 2015;23:54-67.
[18] Renza D, Lemus C. Authenticity verification of audio signals based on fragile watermarking for audio forensics. Expert Systems with Applications. 2018;91:211-22.
[19] Chang C-C. A reversible data hiding scheme using complementary embedding strategy. Information Sciences. 2010;180(16):3045-58.
[20] Al-Hooti A, et al. Audio Data Hiding Based on Sample Value Modification Using Modulus Function. Journal of Information Processing Systems. 2016;12(3).
[21] Andra MB, et al. Medical Record Protection with Improved GRDE Data Hiding Method on Audio Files. Engineering Letters. 2017;25(2).
[22] Li X, Yu HH, editors. Transparent and robust audio data hiding in subband domain. Information Technology: Coding and Computing, 2000 Proceedings International Conference on; 2000: IEEE.
[23] Datta B, et al., editors. Multi-bit Data Hiding in Randomly Chosen LSB Layers of an Audio. Information Technology (ICIT), 2016 International Conference on; 2016: IEEE.
[24] Binny A, Koilakuntla M, editors. Hiding secret information using LSB based audio steganography. Soft Computing and Machine Intelligence (ISCMI), 2014 International Conference on; 2014: IEEE.
[25] Chowdhury R, et al. A view on LSB based audio steganography. International Journal of Security and Its Applications. 2016;10(2):51-62.
[26] Siburian R, editor Steganography implementation on android smartphone using the LSB (least significant bit) to MP3 and WAV audio. Wireless and Telematics (ICWT), 2017 3rd International Conference on; 2017: IEEE.
[27] Tuncer T, Avci E. A reversible data hiding algorithm based on probabilistic DNA-XOR secret sharing scheme for color images. Displays. 2016;41:1-8.
[28] Basu PN, Bhowmik T, editors. On embedding of text in audio a case of steganography. Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on; 2010: IEEE.
[29] Tzanetakis G. Datasets - marsyas [online]. 2002.
[30] Urbana-Champaign SSTGUoIa. Illinois Speech and Language Engineering. University of Illinois at Urbana-Champaign; 2003.
[31] Bosch JJ, et al. P. A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals. IRMAS: a dataset for instrument recognition in musical audio signals.