Received: June 12, 2020. Revised: July 17, 2020.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Utilizing Fuzzy Logic in Developing Reversible Data Hiding Method

Mohammad Muzayyin Amrulloh[1] **Tohari Ahmad[1]***

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia*
* Corresponding author's Email: tohari@if.its.ac.id

Abstract: The development of information technology has brought an avoidable consequence for data security. It is because data, specifically those are private or confidential, are vulnerable to illegal access. Therefore, a protection method, such as data hiding, is essential. This protection is done by inserting a secret message (payload) into a medium (cover). One of the media commonly used in this method is audio, where a payload is embedded in audio samples. Nevertheless, this common technique has several disadvantages, including the relatively small insertion space and the significant dissimilarity between stego-audio and its original audio cover. Moreover, there is a risk of payload that cannot be returned to the original file. To overcome these problems, we propose to employ the fuzzification stage in the fuzzy logic algorithm that the samples are grouped into five categories. The binary payload is evenly distributed to the available sample space, which previously was calculated by considering the total payload and the number of interpolated samples. In the case that space is not enough to carry the payload, it is dynamically increased. The results show an increase of about 48% in stego audio quality measured by the Peak Signal-to-Noise Ratio (PSNR). Moreover, the method is reversible, and the size of the sample space can be controlled to specific values.

Keywords: Data hiding, Audio steganography, Network security, Fuzzy logic, Cybercrime.

1. Introduction

The fast growth of information technology influences security that almost all data are vulnerable to public access, which happens explicitly to private or sensitive data [1, 2]. Consequently, improving security systems for transferring such data is crucial. Several alternatives have been introduced in securing confidential data, ones of which are cryptography and steganography, which is often called data hiding. In fact, both techniques have differences [3].

Cryptography works by changing the content and structure of information in such a way that it has a structure that is different from its original form [4, 5]. Unlike cryptography, steganography, such as [6], works by inserting a secret message into a medium, which can be in the form of audio, text, images, or video, as illustrated in Fig. 1. By using this approach, the message is sent to the recipient in the form of an ordinary file. Furthermore, the recipient simply extracts to get the original message. This concept is depicted in Fig. 2. This embedded file, called stego, has a structure that is similar to the original file. So, it is likely that the transmitting file does not cause suspicion of unwanted parties [7]. Also, many methods can be used in steganography techniques to minimize the risk of data being hacked.

Some problems start emerging, along with the implementation of steganography techniques. The first problem that often occurs is the low quality of the stego [8]. The second problem corresponds to the method, which is required to hold a big secret message. Some research has been performed to overcome those two problems, such as [9].

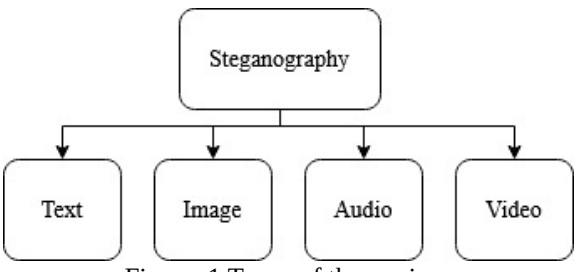

Figure. 1 Types of the carriers

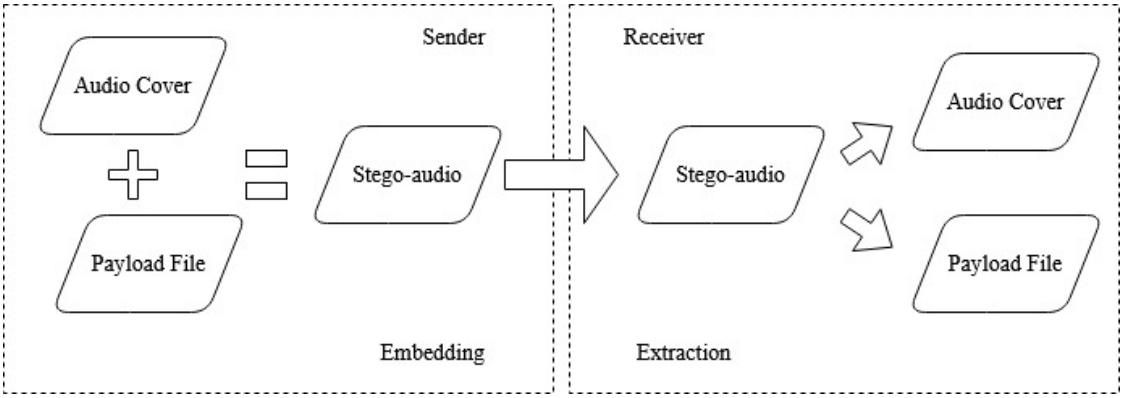

Figure. 2 Concept of data hiding

Some methods have been developed, for example, that is done by Jung and Yoo [10]. It proposes a method of concealing semi-reversible data using linear interpolation and LSB substitution. The stego that has been inserted by payload has different characteristics from the original sample. It is because the use of linear interpolation results in the number of samples being doubled from the original. The linear interpolation is not only to restore the payload in the extraction stage but also to obtain the original cover. In other research, [11] proposes a steganography technique using the fuzzy logic to embed secret messages in an image. Compared to the LSB method, it can produce better results in terms of the size of messages hidden. Next, Gutub et al. [12] combine it with secret-sharing, where specific information is required to construct the secret messages. Nevertheless, this previous research works only for a particular environment.

In this paper, our proposed method focuses on improving the quality of audio that has been inserted by payloads (stego-audio). This scheme calculates the possible embedding space in each sample that will be inserted by the payload. Different sizes of the payload may generate the different size of the embedding space. It evenly puts the payloads among the samples by combining the fuzzification method in fuzzy logic with the payload distribution. The advantage of this design is that the payload does not only reside in individual samples but all of them. Furthermore, the embedding process's success rate is 100% because the size of the embedding space has been estimated before the process starts.

The composition of this paper is explained coherently in each section. The first is the related works, which are provided in Section 2. It explains that previous research relates to this proposed method. Section 3 presents the proposed scheme, and the experimental results are provided in the following section. Then, conclusions are drawn and presented in Section 5.

2. Related works

As explained earlier, the proposed method uses audio as a carrier (cover) because an audio medium has a better insertion capacity than an image [13]. For the basis of this research, we review some related works as follows.

In general, the concept of hiding data can be seen in Fig. 2. It is divided into two categories, namely irreversible and reversible [14]. First, if only the payload can be reconstructed, then it is called irreversible. Second, if the payload and media can be returned, it is called reversible [8]. Each of them has advantages and disadvantages following their respective goals.

Data hiding algorithms have been introduced, including the LSB (Least Significant Bit) method [15]. It is relatively simple to implement, and it has been the basis of other research, especially in the image [16] and audio [17] environments. Dual stego file is proposed in [18], which makes it possible to increase the payload size, such that more bits can be hidden in the cover. However, each proposed method still has advantages and disadvantages of each following the research objectives. The original LSB-based method may suffer from an attack because of its simplicity, where a certain number of bits can be broken by the brute force attack.

In the research proposed by Jung and Yoo [10], LSB is combined with linear interpolation. The result allows the number of samples to be double the original one. So, the original frequency needs to be adjusted. Furthermore, the number of samples significantly increases, which may affect the size of the stego file.

In the research [9], interpolation accompanied by a smoothing sample method on audio media is proposed. The smoothing method allows stego-audio to have a better PSNR quality since it works by smoothing or minimizing the difference between stego-audio and the original audio. It is in contrast to

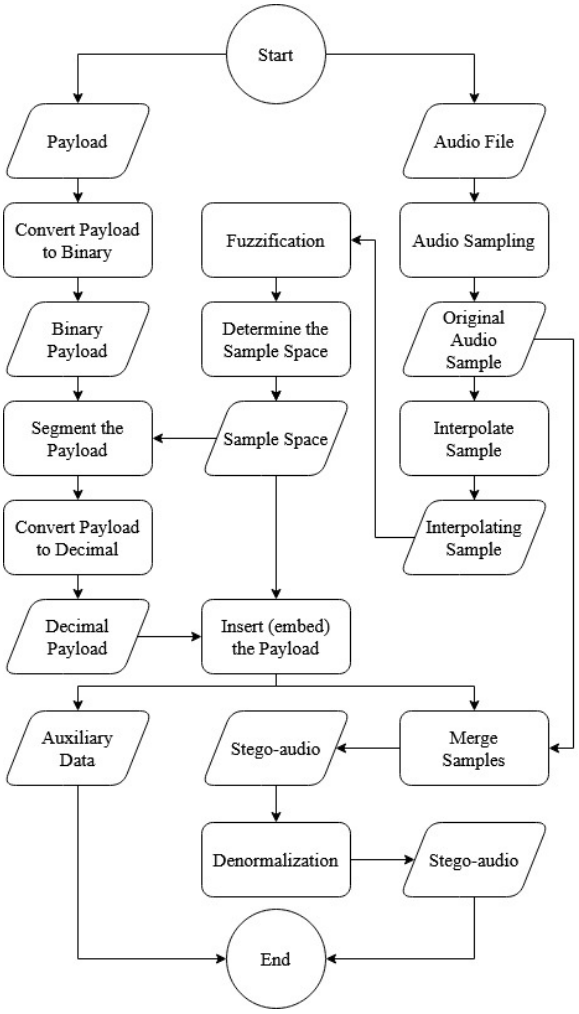

Figure. 3 The embedding process

the research conducted by Ahmad et al. [8], which they find that a payload distribution method that can produce better PSNR quality without having to do smoothing. This research does not estimate whether the whole generated space is sufficient to hold the payload or not. Therefore, it is possible that not all messages can be embedded, which leads the embedding process to fail.

Next, research [11] proposes an algorithm of hiding data using fuzzy logic. It produces better results compared to the LSB method in terms of the hidden message size. The results are shown by increasing PSNR and decreasing MSE obtained for the tested images. Nevertheless, its computational time and complexity should be further considered. In further research, Bobeica et al. [19] propose a reversible data hiding, which can control the embedding capacity. It is developed based on the prediction error expansion, relying on the specified threshold values. Nevertheless, it is sometimes not easy to find those values.

Generally, this existing research inspires us to develop a reversible method, which accomplishes their drawbacks. Moreover, some additional steps are designed to improve performance. The detail of the proposed method is provided in the next section.

3. Proposed method

In general, this proposed method consists of two essential processes: the insertion (embedding) process, and the extraction process (see Fig. 2).

3.1 Embedding

Insertion is the process of hiding a payload into an audio sample. In this process, the steps in inserting a payload are explained using the fuzzification method. The insertion process scheme can be seen in Fig. 3, whose steps are as follows.

1. Normalized audio input in the form of *.wav type is done by adding all samples with 32768 to remove negative values in the sample and place it in the range 0 to 32767.
2. The sample is interpolated. The interpolated sample is to determine a new point between each normalized sample by using Eq. (1).

$$S'_n = \left\lfloor \frac{S_n + S_{(n+1)}}{2} \right\rfloor \quad (1)$$

Notation S'_n is the result of interpolation at n index, whereas S_n is the sample value on the n index.

3. At this stage, the category of the sample is determined using the fuzzification method, according to Fig. 4 [20], by using Eq. (2), where a is the smallest domain of the smallest membership degree, b is the highest membership degree in the domain, and c is the highest domain of the smallest membership degree. The output $(\mu(x))$ of the normalized interpolated sample consists of 2 types, namely the upper and lower limits ranging from 0 to 1, where the two values will be used in the next step to determine the sample space.

$$\mu(x) = \begin{cases} 0, \text{if } x \le a \text{ or } x \ge c \\ \frac{x-a}{b-a}, \text{if } a \le x \le b \\ \frac{c-x}{c-b}, \text{if } b \le x \le c \end{cases} \quad (2)$$

4. At this stage, the embedding space is calculated to determine how many bits can be accommodated per interpolated sample. The method used is in the following Eqs. (3), (4), and (5).

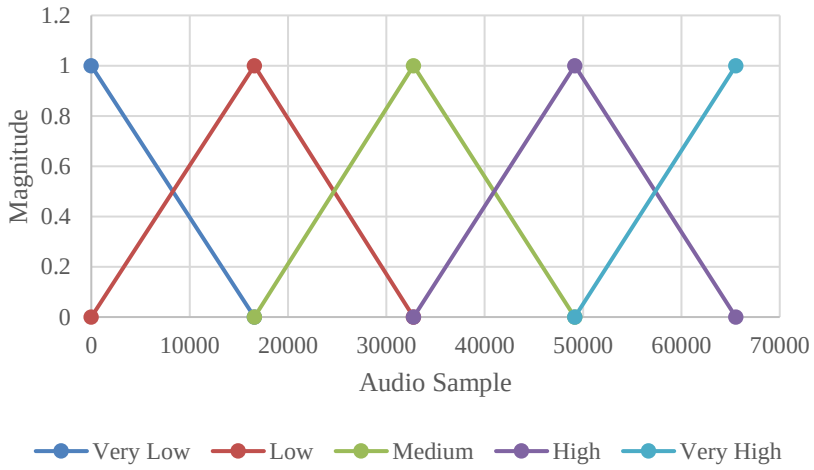

Figure. 4 Fuzzification graph

$$Xa_n = ba_n \lfloor S_n - S_{n+1} \rfloor \quad (3)$$

$$Xb_n = bb_n \lfloor S_n - S_{n-1} \rfloor \quad (4)$$

$$Total_n = \lfloor Xa_n + Xb_n \rfloor \quad (5)$$

Notation $Total_n$ shows sample space in decimal at n index. The symbol ba_n and bb_n show the fuzzification stage's output, namely the upper and lower limits at $the\ n$ index. Then S_n shows the audio sample at the n index. After finding the sample space in decimal, the value is converted into binary form. The method used is in Eq. (6) as follows.

$$N_n = \lfloor \log_2 Total_n \rfloor \quad (6)$$

The symbol N_n represents the number of bits that can be put into the interpolation sample at the n index. This value is to be input in the next stage.

5. The payload that has been converted into binary format is divided by considering the distribution of bits, as in Eq. (7). It is to determine how many bits will be shared to each interpolation sample. This number of bits adjusts the capacity that has been provided at the sample space determination as in Eq. (8).

$$Dp = \left\lceil \frac{Total\ payload\ bit}{Total\ interpolation\ sample} \right\rceil \quad (7)$$

$$Sp_n = \begin{cases} N_n, \text{if } N_n < Dp \\ Dp, \text{if } N_n \geq Dp \end{cases} \quad (8)$$

Notation Sp_n shows payload segmentation at n index, whereas notation of Dp shows the number of bits distributed to each interpolation sample. If there is a remaining payload, the value of Dp is incremented to 1, and then Eq. (8) is calculated until all the payloads have been successfully inserted.

6. The payload insertion into the interpolated audio sample corresponds to the number of bits that have been segmented at the previous stage. The method can be seen in Eq. (9).

$$S'_n = S_n - P_n \quad (9)$$

Notation S'_n shows the new sample value after the payload is inserted at the n index.

7. The merging between the embedded sample and the original audio sample is performed, whose method can be seen in Eq. (10).

$$S_i = \begin{cases} S_{i/2}, & \text{if } i \text{ is even} \\ S'_{t,(i-1)/2}, & \text{if } i \text{ is odd} \end{cases} \quad (10)$$

8. A stego-audio is made by denormalizing the sample, which is to place the sample in the original range: from -32768 to 32767 and reconstructing it into a *.wav format audio file.

3.2 Extraction

Extraction is the process of taking payload data from a stego-audio sample. In this process, the steps in retrieving payload data are explained using the fuzzification method. The scheme of the extraction process can be seen in Fig. 5. The stages of the process are as follows.

1. Stego-audio sample normalization is performed. This process is the same as the first step in the insertion process.
2. The stego-audio sample is split into two samples: odd index samples (samples with

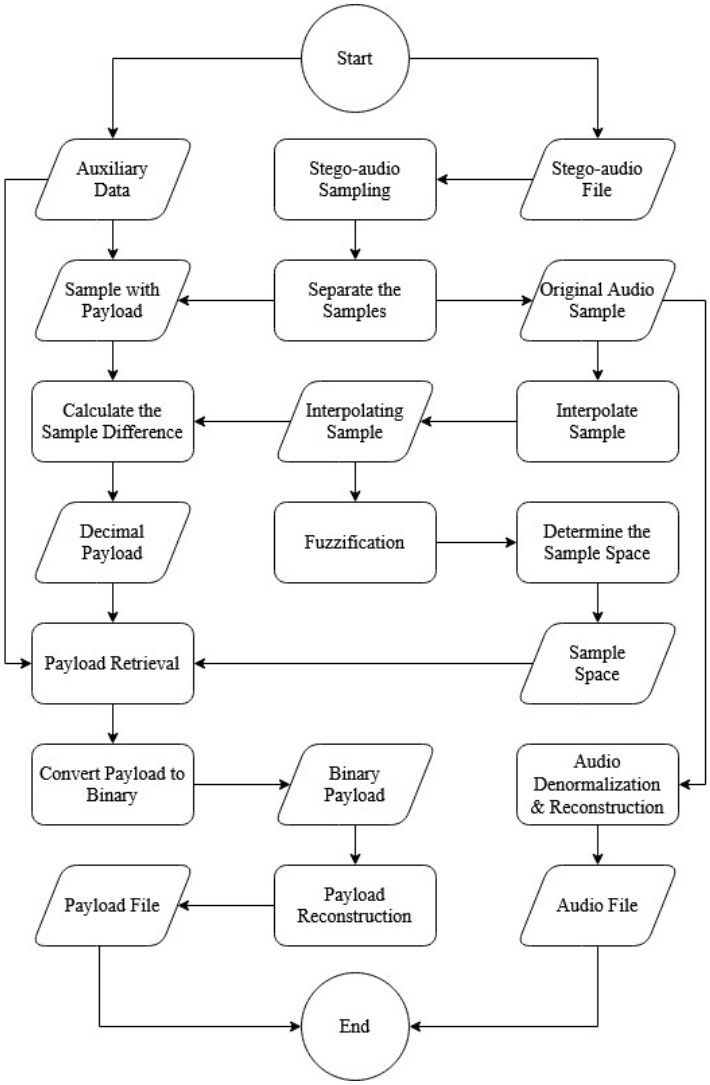

Figure. 5 The extraction process

payload) and even index samples (original audio samples). The method used can be seen in Eq. (11) and Eq. (12).

$$S_{n/2} = SSA_n \text{ for } n \text{ is even} \quad (11)$$

$$S''_{(n-1)/2} = SSA_n \text{ for } n \text{ is odd} \quad (12)$$

The symbol S_n denotes the original audio sample at the n index. Notation S''_n shows a sample with a payload at the n index. Then, notation SSA_n is the stego-audio sample at the n index.

3. Sample interpolation is carried out, which is to determine a new point between the original audio samples. The formula used is the same as the formula performed at the interpolation stage in stage 2 of the insertion process.
4. Fuzzification is carried out from the sample output values obtained from the interpolation stage.
5. Determining the sample space. The formula is equal to the Eq. (3), (4), and (5) of the embedding process. The sample used is the output from the fuzzification stage, which has two values, the upper and lower limit of the output from the fuzzification stage whose range is from 0 to 1.
6. The payload retrieval process is carried out. At first, the payload is taken in the decimal form using Eq. (13) as many as the total sample used in the embedding process. Then, it is converted into binary form according to the available sample space.

$$Pd_n = Si_n - Sp_n \quad (13)$$

Pd_n = Decimal payload at the n index.
Si_n = Interpolation sample at the n index
Sp_n = Samples with payload at the n index

Table 1. Audio dataset

Instrumen	Genre	Data	Name
Cello (cel)	Country-Folk (cou_fol)	[cel][jaz_blu]0011__2	Data 1
	Classical (cla)	[cel][cla]0007_1	Data 2
	Pop-Rock (pop-rock)	[cel][pop_roc]0060_2	Data 3
Acoustic guitar (gac)	Country-Folk (cou_fol)	[gac][jaz_blu]0549_2	Data 4
	Classical (cla)	[gac][cla]0530_1	Data 5
	Pop-Rock (pop-rock)	[gac][pop_roc]0560_3	Data 6
Piano (pia)	Country-Folk (cou_fol)	[pia][jaz_blu]1348_1	Data 7
	Classical (cla)	[pia][cla]1291_1	Data 8
	Pop-Rock (pop-rock)	[pia][pop_roc]1306_1	Data 9
Saxophone (sac)	Country-Folk (cou_fol)	[sax][jaz_blu]1605_2	Data 10
	Classical (cla)	[sax][cla]1598_1	Data 11
	Pop-Rock (pop-rock)	[sax][pop_roc]1588_1	Data 12
Human Singing Voice (voi)	Country-Folk (cou_fol)	[voi][jaz_blu]2358_1	Data 13
	Classical (cla)	[voi][nod][cou_fol]2442_1	Data 14
	Pop-Rock (pop-rock)	[pop_roc]2547_3	Data 15

7. The sample denormalization process is carried out to return the sample values in the right range, between -32768 and 32767. The process is also the same as the sample denormalization stage in the insertion. Then, it is reconstructed into an audio file in *.wav format.

4. Experimental result

The cover used in this research is an audio file in *.wav format, which is not compressed. So, no file is likely lost even if the file is inserted [21].

The audio file is a data set for Instrument Recognition in Musical Audio Signals (IRMAS) [22]. We take 15 audio files consisting of 5 instruments, and each instrument consists of 3 genres. The duration of each audio file used is 2 seconds. Audio data can be seen in Table 1.

The payload file is in *.txt, which is obtained from a text generator. Providers of text generators used to come from the page www.id.lipsum.com [23]. The payload is 11 data, with various sizes: 1 kb, 10 kb, 20 kb, 30 kb, 40 kb, 50 kb, 60 kb, 70 kb, 80 kb, 90 kb, and 100 kb.

For evaluating the quality, we measure some parameters as follows, in addition to the reversibility of the method.

1. Differences in stego-audio quality based on mean squared error (MSE) and peak signal to noise ratio (PSNR)
2. The difference in the amount of sample space for insertion

The first test scenario measurement compares the quality of stego-audio with the original audio file, whose method can be found in Eq. (14) and (15).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(S_i - S_i')^2 \qquad (14)$$

MSE (Mean Squared Error) calculates error values between a stego-audio sample and an original audio sample. The N notation depicts the number of samples in the audio; Si denotes a sample of the original audio, while S_i' is a sample of stego-audio.

$$PSNR = 10 \times \log_{10}\frac{(2^b-1)^2}{MSE} \qquad (15)$$

PSNR (Peak Signal to Noise Ratio) measures the quality comparison between the original audio and audio after payload insertion. Notation $2b$ is the maximum value of bit-depth where b is 16 bits. A comparison of PSNR is carried out with the method in [19], [10], and [9], which had been smoothed five times. It is worth noting that [9] is also taken for the comparison because no other method implements smoothing, as far as we know. The second scenario is to compare the method using the payload distribution with the smoothing method by calculating the total sample used in the payload insertion process.

4.1 The comparison result of MSE and PSNR

The evaluation result is provided in Fig. 6, which is compared to those taken from [9] [19] [10]. It shows the average of PSNR values of all audio cover that is embedded by each payload. Based on this experiment, we find that if the payload size is getting bigger, the stego-audio quality is lower. The more

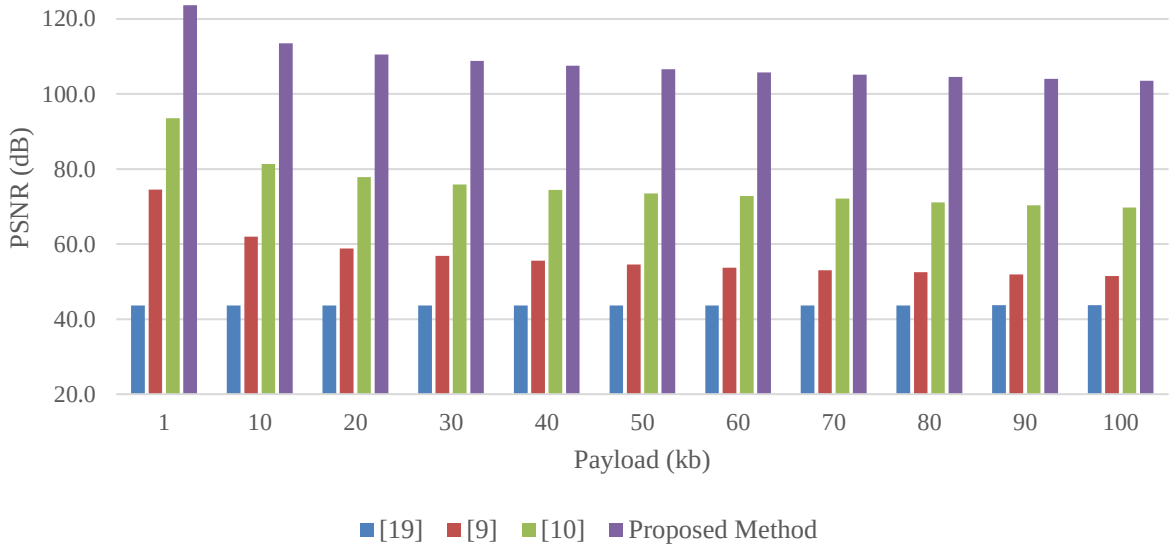

Figure. 6 The average of PSNR values taken from [19], [9], [10], and the proposed method

Table 2. Total sample used in the proposed method

Audio	Number of samples for each payload										
	1 kb	**10 kb**	**20 kb**	**30 kb**	**40 kb**	**50 kb**	**60 kb**	**70 kb**	**80 kb**	**90 kb**	**100 kb**
1	1008	10068	20122	30149	40200	50309	60381	70416	80491	90549	100581
2	1064	10787	21620	32231	42721	53705	65363	76314	87520	98685	109934
3	1001	10007	20011	30020	40029	50038	60046	70057	80065	90072	100073
4	1006	10060	20135	30207	40264	50324	60404	70477	80542	90646	100758
5	1018	10165	20378	30604	40929	51346	63229	73369	83669	94220	104763
6	1023	10172	20365	30489	40589	50730	60891	71059	81219	91297	101370
7	1024	10130	20288	30496	40740	51340	61397	71558	81645	91867	102185
8	1012	10168	20370	30521	40675	50821	61027	71175	81509	92013	102352
9	1004	10029	20067	30105	40128	50157	60186	70227	80253	90289	100332
10	1010	10079	20140	30188	40254	50338	60405	70461	80528	90615	100667
11	1058	10485	20923	31306	41725	52192	62606	72944	83207	93537	103903
12	1006	10035	20069	30142	40197	50244	60282	70339	80399	90477	100560
13	1325	10809	20980	31051	41105	51205	61437	71608	81792	91976	102061
14	1005	10023	20051	30086	40123	50164	60207	70248	80280	90299	100327
15	1005	10042	20094	30123	40140	50167	60187	70212	80237	90263	100285

payload data are inserted, the more different the stego-audio file is from the original audio.

Based on the experiment in [9], it is found that the best PSNR result is obtained by audio 13 with a payload size of 1 kb, which is 119 dB; and the worst PSNR results obtained in audio 3 with a payload size of 10 kb, which is 51.1 dB. Whereas in [19], results from various payload sizes are relatively similar. The variation between payloads is small since the number of embedded samples is not much different. It is also depicted that [10] has better quality than the previous research, even though it has no smoothing as in [9].

In the proposed method, the best PSNR quality is obtained in all audio with a payload size of 1 kb with the results of 123.67 dB and the worst PSNR obtained in all audio with a payload size of 100 kb, which is 103.56 dB. We find that the PSNR for all audio covers is stable for each payload. It means that it is not affected by the type of cover.

It is found that each audio has the same MSE value for each payload. It is because the payload is

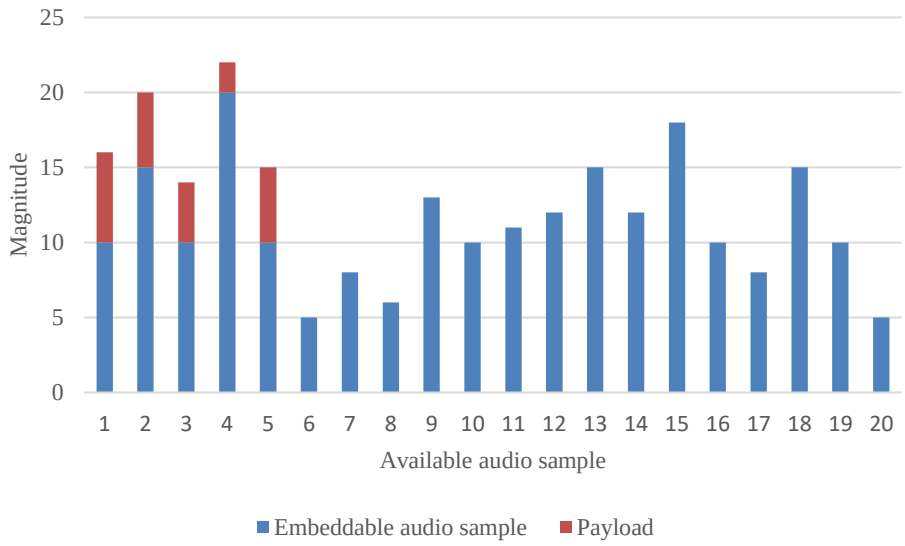

Figure. 7 Embedding without payload distribution

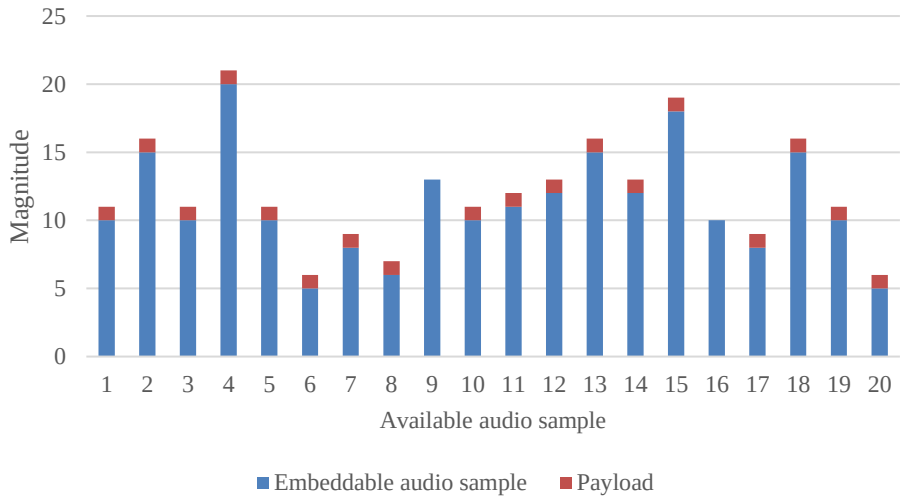

Figure. 8 Embedding with payload distribution

distributed among the supported samples. That is, each sample only holds 1 bit. It is likely that bit '1' and bit '0' cause the difference 1 and 0, respectively. Moreover, the number of available samples is more than the required (embedded) ones. So, all bits can be fully put in the audio covers. According to Eq. (15), this MSE values influence the corresponding PSNR.

Overall, the proposed method can produce better stego-audio quality than [9-10,19] without smoothing. It is indicated by an increase in PSNR results by about 48%. Furthermore, the triangular membership function (see Fig. 4) of the fuzzying step in the proposed method is useful in determining the sample space. It is by exploring the two output values for calculating the difference between the previous and the following *i*-th sample. Their total number is to find the number of bits that can be held in each sample. Therefore, without using this fuzzy scheme, the average number of samples space is likely to be smaller.

4.2 Total Samples

Although the MSE and PSNR values for all audio in each payload are the same, the number of evaluated samples for insertion is different, as provided in Table 2. The process of distributing the payload still refers to the available sample space.

Based on the experimental results, we also find that the proposed method can utilize more samples for the insertion process. As described in Section 3, the samples' real embedding space follows the total embedding space of the whole audio cover. So, we can be sure that all payloads can be inserted.

The method in [19] sometimes gets difficulties in finding an appropriate parameter value; therefore, the

embedding may not always be a success. It is because those thresholds should be found before the process begins. In our proposed method, we do not need those values; instead, we take the number of the samples' embedding space. Once the generated space is not enough, it is expanded. Consequently, the stego quality may fall.

The research in [9] works by maximizing the available sample space. Therefore, the insertion process only uses fewer samples. It results in swelling of the sample inserted with a large payload of data so that the smoothing stage is needed, as illustrated in Fig. 7. It is shown that the insertion process works by maximizing the available sample space. The sample used in the insertion is only those first five, while the other samples are not used. Consequently, the smoothing stage is needed to increase the PSNR value of the method. In [10], interpolation is also employed, which makes the capacity raising. Nevertheless, the balance between the capacity and the quality should be carefully found.

Fig. 8 is an example of the proposed payload distribution method. It is depicted that each sample has the same portion in holding the distributed payload bits except for samples that have a smaller sample space than the number of bits distributed, such as sample number 9 and number 16. These samples can not hold the bits distributed because their sample space is smaller than the number of bits. So, it is not fully distributed to each sample. The proposed distribution method still depends on the sample space that can be accommodated.

5. Conclusion

The experimental results show that the more payload size inserted, the more the quality of the stego-audio decreases. To address this problem, this research proposes the fuzzification method to determine the sample space and is supported by the payload distribution for the insertion method.

Based on the evaluation of MSE and PSNR results, we see that the proposed method can produce better stego-audio quality than the compared method without a smoothing stage. The results show an increase in PSNR results and a decrease in MSE results by almost 48%. From the evaluation of the total sample used, the proposed method can utilize a large number of samples for the insertion process to minimize the MSE value while the compared method overly maximizes the available sample space, so that swelling occurs. Consequently, the smoothing stage is needed.

In further research, it is suggested to develop insertion methods by considering each number of sample space in the audio cover. This design will make the method more adaptive to the closest samples.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, MMA and TA; methodology, MMA and TA; software, MMA; formal analysis, MMA and TA; investigation, MMA and TA; resources, MMA and TA; data curation, MMA and TA; writing—original draft preparation, MMA; writing—review and editing, TA; supervision, TA; administration, TA; funding, TA.

References

[1] L. Headlington, “Human factors in cybersecurity; examining the link between Internet addiction, impulsivity, attitudes towards cybersecurity, and risky cybersecurity behaviours”, *Heliyon*, Vol. 3, No. 7, 2017.

[2] L. K. Ketshabetswe, A. M. Zungeru, M. Mangwala, J. M. Chuma, and B. Sigweni, “Communication protocols for wireless sensor networks: A survey and comparison”, *Heliyon*, Vol. 5, No. 5, 2019.

[3] R. Mishra and P. Bhanodiya, “A Review on Steganography and cryptography”, In: *Proc. of International Conf. on Advances in Computer Engineering and Applications*, Ghaziabad, India, 2015.

[4] M. M. Amin, M. Salleh, S. Ibrahim, M. R. Katmin, and M. Z. I. Shamsuddin, “Information Hiding using Steganography”, In: *Proc. of 4th National Conf. on Telecommunication Technology*, Johor, Malaysia, 2003.

[5] Z. Kartit and M. E. Marraki, “Applying Encryption Algorithm to Enhance Data Security in Cloud Storage”, *Engineering Letters*, Vol. 23, No. 4, pp. 277-282, 2015.

[6] J. Mohajon, Z. Ahammed, and K. H. Talukder, “An Improved Approach in Audio Steganography Using Genetic Algorithm with K-Bit Symmetric Security Key”, In: *Proc. of International Conf. of Computer and Information Technology*, Dhaka, Bangladesh, 2018.

[7] H. Arif and H. Hajjdiab, “A Comparision between Steganography Software”, In: *Proc. of 16th International Conf. on Computer and Information Science*, Wuhan, China, 2017.

[8] T. Ahmad, J. N. Faruki, R. M. Ijtihadie, and W. Wibisono, "Analyzing the Effect of Block Size on the Quality", In: *Proc. of International Conf. on Science and Technology*, Yogyakarta, Indonesia, 2019.

[9] T. Ahmad and T. P. Fiqar, "Enhancing the Performance of Audio Data Hiding Method by Smoothing Interpolated Samples", *International Journal of Innovative Computing, Information and Control*, Vol. 14, No. 3, pp. 767-779, 2018.

[10] K. H. Jung and K. Y. Yoo, "Steganographic method based on interpolation and LSB substitution of digital images", *Multimedia Tools and Application*, Vol. 74, No. 6, pp. 2143-2155, 2015.

[11] A. A. Alghamdi, "Computerized Steganographic Technique using Fuzzy Logic", *International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 3, pp. 155-159, 2018.

[12] Gutub, N. Al Juaid, and E. Khan, "Counting-based secret sharing technique for multimedia applications", *Multimedia Tools and Applications*, Vol. 78, No. 5, pp. 5591-5619, 2019.

[13] A. H. Ali, M. R. Mokhtar, and L. E. George, "Research Article A Review on Audio Steganography Techniques", *Research Journal of Applied Sciences, Engineering and Technology*, Vol. 12, No. 2, pp. 154-162, 2016.

[14] T. Sarkar and S. Sanyal, "Reversible and Irreversible Data Hiding Technique", arXiv:1405.2684v2, 2014.

[15] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques", In: *Proc. of International Conf. on Image Processing*, Thessaloniki, Greece, 2001.

[16] A. Odat and M. A. Otair, "Image Steganography using Modified Least Significant Bit", *Indian Journal of Science and Technology*, Vol. 9, 2016.

[17] M. Asad, J. Gilani, and A. Khalid, "An Enhanced Least Significant Bit Modification Technique for Audio Steganography", In: *Proc. of International Conf. on Computer Networks and Information Technology*, Abbottabad, Pakistan, 2011.

[18] A. K. Sahu and G. Swain, "Dual Stego-imaging Based Reversible Data Hiding Using Improved LSB Matching", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 5, pp. 63-74, 2019.

[19] Bobeica, I. C. Dragoi, I. Caciula, D. Coltuc, F. Albu, and F. Yang, "Capacity Control for Prediction Error Expansion based Audio Reversible Data Hiding", In: *Proc. of 22nd International Conf. on System Theory, Control and Computing*, Sinaia, Romania, 2018.

[20] R. F. Cadiz, "Fuzzy logic in the arts: Applications in audiovisual composition and sound synthesis", In: *Proc. of Annual Conf. of the North American Fuzzy Information Processing Society*, 2005.

[21] K. Sharma and K. Gupta, "Lossless Data Compression Techniques and Their Performance", In: *Proc. of International Conference on Computing, Communication and Automation*, Greater Noida, India, 2017.

[22] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition In Musical Audio Signals", In: *Proc. of 13th International Society for Music Information Retrieval Conf.* Porto, Portugal, 2012.

[23] "Lorem Ipsum," [Online]. Available: http://www.id.lipsum.com/.