Received: June 16, 2021. Revised: August 5, 2021. 550

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Mirroring-Based Data Hiding in Audio

Yoga Samudra[1] Tohari Ahmad[1]*

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

Abstract: Information technology has multiplied for some decades. It has affected the way how the public interacts with their environment. Specifically, this technology helps us to work more efficiently than before. Nevertheless, some problems arise due to the exploitation of its weaknesses, like capturing confidential data. Some schemes have been introduced, such as embedding the secret in a multimedia file. However, the quality of the file after the embedding process may be different from its original form significantly. Moreover, there is a limitation on the secret size that can be hidden. In this research, we propose to improve the existing method by taking audio as the carrier. Specifically, we explore the mirroring technique after the audio samples are interpolated. It is to reflect the samples to the specified mirroring point, purposing to minimize the difference between before and after the payload is embedded. The more similar the stego to the cover file, the higher the quality; besides, the embedding space has been enlarged. The experimental result shows that this proposed method can increase the quality of the stego file by about 20 dB for various payload sizes.

Keywords: Data protection, Data hiding, Network infrastructure, Information security.

1. Introduction

Nowadays, the advancement of technology has an essential role in developing data transmission and accessibility, which have become faster through any media. Nevertheless, this development also brings security problems, where data can be easily stolen or manipulated [1, 2]. This situation has made data security schemes necessary.

Information hiding is one of the techniques to secure the data, which includes watermarking and steganography. Even though they have similar processes, their goal is different from each other [2]. Watermarking is a technique to embed data into another data purposed ownership, authentication, Digital Ownership Management (DRM) [3, 4], or file tamper detection [5]. Meanwhile, steganography is used to embed confidential data into other data to protect them from illegal access. However, protecting data may raise several problems, such as the degree of similarity between stego-data and cover-data, and also the size of data embedded into the stego-data[6-8].

There has been some research about steganography using various cover media, such as image, audio, and video. Practically, image is popular and often implemented to carry the private data. Furthermore, its performance has improved[9, 10]. Wang et al. [7] proposed a reversible method by exploring Most Significant Bit (MSB) encoding, letting confidential data be veiled into the carrier image. It employed Huffman coding to manage a hiding stage. They claimed that the results are satisfying. Sahu et al. [11] also proposed a new image-based scheme using a different method called Pixel Modification Value (PVD) and modulus function. One advantage of using PVD is that the resulting stego-image has a higher embedding capacity, but there will be some samples from each segment that could not be used in the embedding process, making the quality of the stego-image relatively lower.

Most steganographic research is based on an image because it is the most commonly used medium whose implementation is relatively simple. Nevertheless, the downside of steganography using

images is its difficulty producing such a stego-image with a high embedding capacity. Therefore, research on steganography based on other media is also considered, which audio is one of them. Andra et al. [12] proposed new audio-based steganography to protect medical data records. This proposed method combines Generalized Difference Expansion (GDE) and Reduced Difference Expansion (RDE). Bobeica et al. [13] proposed a new thresholding point for Prediction Error Expansion (PEE) to improve quality. Then, Ahmad and Samudra [14] proposed new steganography audio based on linear interpolation sampling, which produced the stego-audio with a total sample twice as much as the cover audio.

In this research, we propose a new method on steganographic audio that focuses on linear interpolation sampling. By using this sampling approach, the embedding capacity will be higher. We try to maintain the total sample of stego-audio as many as the cover audio. This approach, however, will produce stego-audio in lower quality with a huge gap compared to the embedding capacity. Therefore, we combine this sampling approach with mirror embedding that works by reflecting the new sample value with mirroring point, so that the final sample value can be as close as possible to the real sample value of the cover audio.

The following sections are presented in the next parts of this paper. First, the literature study is provided in Section 2. Then, the proposed method is presented in Section 3, while Section 4 provides the results and analysis. To wrap up the paper, the conclusion is given in Section 5.

2. Related works

Our proposed work refers to some commonly used data hiding techniques, such as interpolation-based steganography and reduced difference expansion. We enhance their capability in protecting messages.

In [15], Benhfid et al. take the linear box spline-based interpolation in the three directional mesh to extend the space, different from previous research. In that study, they implement the hiding method in a 512×512 image. The embedding process itself is performed by using the difference (error) between the image generated by the interpolation and the corresponding initial image, called Message Adaptive Error (MAE); and the Objective of Applying (OPAP). Various existing interpolation methods are tested and analyzed to find the best. These algorithms include Interpolation by Neighboring Pixels (INP), Neighbor Mean Interpolation (NMI), and High-Capacity Reversible Steganography (CRS), which are often used in extending the embedding space.

Overall, they obtain a relatively good performance of all combinations. It is also found that different interpolation algorithms may not much affect the performance in terms of both capacity and quality. Furthermore, a deeper analysis should still be done to find the factors causing those different results. The possibility of implementing this method in other media is not discussed.

Differently, without using interpolation algorithms, Wahab et al. [16] implement some existing methods to hide the data. In their system, cryptography, compression, and data hiding methods are taken. Firstly, the payload is encrypted by using the RSA (Rivest-Shamir-Adleman) algorithm. Next, it is compressed to have a smaller file size before being hidden in the cover medium. For the compression, they consider the Discrete Wavelet Transform (DWT) and Huffman Code, while an LSB-based method is taken for the embedding.

It is shown that their implementation has a relatively good performance. Compressing the payload reduces its size, minimizing the difference between stego and cover files, whatever the methods to implement. Therefore, the stego quality is maintained. Nevertheless, we believe that it is more on implementing existing methods than finding a new one. Although, this concept applies to any medium.

Then, a new audio data hiding based on interpolation with smoothing is proposed [14]. The interpolation technique used in this method is for sampling the audio. This method introduced the segment as a parameter to determine the interpolation values generated as embedding space. The resulting interpolation values will be more varied using segment, depending on the segment value being used. This method also implements the compression on hidden data so that the bit value is as little as possible when being embedded. As a result, this method has a relatively high embedding capacity and reasonable value of quality in stego-audio. However, the generated stego-audio still results in a total sample of twice as much as the original cover audio.

Another common data hiding method, Reduced Difference Expansion (RDE), is also referenced in our proposed method. This technique performed an embedding process that does not need any additional information for the extraction process.

On the other hand, Andra et al. [12] implemented a new data hiding method based on audio which combined Reduced Difference Expansion [17] with Generalized Difference Expansion, which is then optimized in [18] using video. The main focus of this method is to improve the embedding capacity and

enhance the security level of data hiding because it is being used as medical data protection. This method generates a multi-dimensional array to store bit value, namely an array of bigits. This array of bigits is used in hidden data so that they can be embedded into the audio with 16-bit depth. As for the security aspect of this method is that the order of bigits assigned when embedding the data itself is done in a specific order so that it would not be easy to guess the hidden data.

While Bobeica et al. [13] introduce a new method to determine a threshold value within Prediction Error Expansion, the embedding method itself stays the same as explained in [19] using noncausal prediction of alterable orders. The new threshold value that is determined by this method gives a slight improvement in the quality of stego-audio produced, while the embedding capacity is similar compared to its referenced method.

3. Mirroring technique

The limitations, such as a wide gap between quality and embedding capacity of stego-data, need to be removed or reduced significantly to optimize such scheme, which is why this research is proposed. Previous research using the linear interpolation method to determine samples that could be embedded has proven to give high embedding capacity [8]. Nevertheless, the produced stego-audio has a total sample of twice as much as the original cover audio, whose quality cannot be compared. We proposed this new method of mirror embedding with modification based on linear interpolation to give high embedding capacity on stego-audio while maintaining its quality as high as possible. Also, we try to make the produced stego-audio has a total sample as much as the original cover audio to be compared in terms of quality.

This method consists of two primary processes: embedding and extraction, as shown in Fig. 1 and Fig. 2, respectively. The embedding process is divided further into two main processes: distribution sampling and mirror embedding. While the extraction process is also divided further into two primary processes: predict sampling and reverse extraction. Some significant changes from our proposed method are the embedding process that utilizes the geometry concept of reflection, translation, and modifying the linear interpolation formula to adjust the sample value.

3.1 Embedding

3.1.1. Distribution sampling

Audio samples obtained have values ranged from -32768 to 32767. In order to simplify the calculation of the embedding process, a normalization value is needed to shift the domain value to only a positive integer by adding each sample value with 32768, as shown in Eq. (1). After the normalization is done, the interpolation value for each sample is calculated. The calculation here uses standard linear interpolation as shown in Eq. (2), with S is the sample value, and I is the interpolation value, while n determines their index value.

$$S_n = S_n + 32768 \tag{1}$$

$$I_n = \left\lfloor \frac{S_n + S_{n+1}}{2} \right\rfloor \tag{2}$$

These interpolation values will determine how much embedding space for each sample can hold. This embedding space (E) can be found by calculating the difference between the interpolation value with the sample value after it, as shown in Eq. (3). Then, we find how much bit (b) can be embedded into such embedding space, as shown in Eq. (4).

$$E_n = S_{n+1} - I_n \tag{3}$$

$$b_n = \lfloor \log_2 E_n \rfloor \tag{4}$$

After this embedding space is determined for each sample, the next step is to distribute the payload data fairly to every sample available. We can do this by finding two values: the average bit per sample containing cover audio denoted as $BPS_{payload}$, and the average bit per sample contained in cover audio, denoted as BPS_{cover}. $BPS_{payload}$ can be determined by calculating the average between the total payload bits with the total samples available for embedding later, as shown in Eq. (5). Meanwhile, BPS_{cover} can be determined by calculating the average between embedding space in bits with total samples available, as shown in Eq. (6).

$$BPS_{payload} = \left\lceil \frac{total\_bits\_payload}{total\_sample} \right\rceil \tag{5}$$

$$BPS_{cover} = \left\lfloor \frac{\sum_{i=1}^{n} b_i}{total\_sample} \right\rfloor \tag{6}$$

These values are compared as in Eq. (7) to determine the maximum bit data for each sample to be embedded, which we called the bit threshold (bt). If the value bt is 0, the payload will be considered unable to be embedded. Therefore, the embedding process fails.

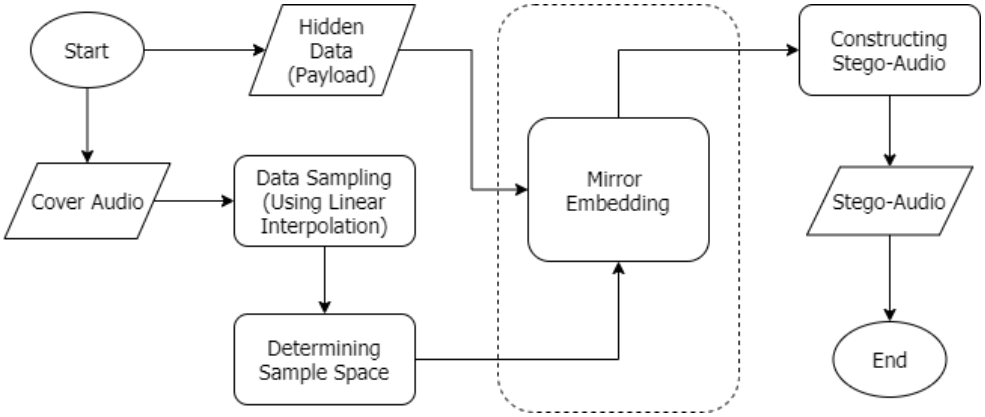

Figure. 1 General embedding process

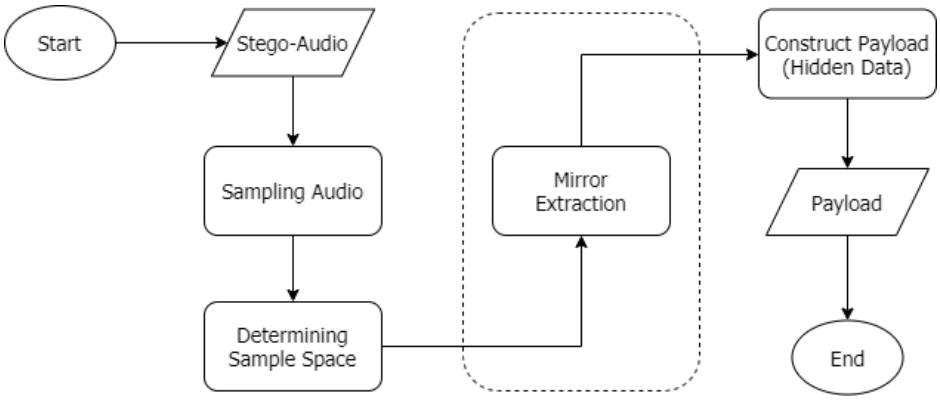

Figure. 2 General extraction process

$$bt = \begin{cases} BPS_{payload}, & if\ BPS_{payload} \leq BPS_{cover} \\ 0, & otherwise \end{cases} \tag{7}$$

3.1.2. Mirror embedding

After the bit threshold value is determined, the next step is to perform the embedding process. The cycle of this embedding process, as shown in Fig. 3, consists of: finding embedding space, calculating the new interpolation value, determining the first mirror point, embedding payload, and mirror embedding value. This cycle is implemented for each sample one by one until all payloads have been embedded. It is still considering the maximum payload size for the corresponding sample, as described in the previous step.

3.1.2.1. Find embedding space

The embedding space of each sample (in bits) is compared with the bit threshold (bt) value as in Eq. (8), where n is the index in the cover samples. If the available embedding space is higher than bt value, then bt value is used as embedding space (es). However, if the available embedding space is lower than bt value, then the embedding space itself will be used as it is.

$$es_n = \begin{cases} bt, & if\ b_n \geq bt \\ b_n, & if\ b_n < bt \end{cases} \tag{8}$$

3.1.2.2. Calculating new interpolation value

From Distribution Sampling, we already found every interpolation value in cover samples only to estimate how much embedding capacity can be held in the samples. Meanwhile, here, we will re-determine the new interpolation value by modifying the linear interpolation formula to adapt to embedding space (es) as in Eq. (9), then Eq. (10). The higher the value of embedding space will likely cause the new interpolation value to getting closer to the original sample value on the same index.

$$I'_n = \frac{(2^{es_n}-1)S_n + S_{n+1}}{2^{es_n}} \tag{9}$$

$$I'_n = \begin{cases} \lfloor I'_n \rfloor, & if\ S_n \leq S_{n+1} \\ \lceil I'_n \rceil, & if\ S_n > S_{n+1} \end{cases} \tag{10}$$

This new interpolation value is designed to be as close as possible to the cover sample value before

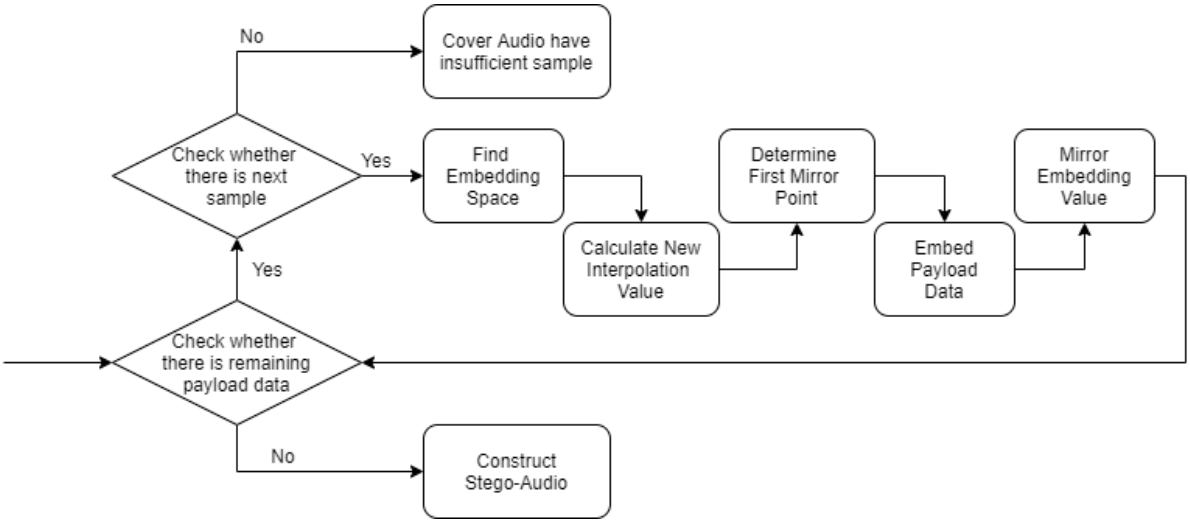

Figure. 3 Specific flow of mirror embedding

them, so that the bit(s) of payload data that will be embedded can go towards the cover sample after it, and so on.

3.1.2.3. Determining first mirroring point

A mirroring point is a value in a range of sample values that will be used as a reflection point for some value of one side to have its reflection value on the other side of this point. Before determining the first mirroring point of each sample, we should first determine the last mirroring point ($MP_{n,l}$) and how many mirroring points each sample has, as it will be represented in Eq. (11) and Eq. (12), where l symbolizes a mirror point as the last mirror point of each sample.

$$MP_{n,l} = (I'_n \,.\, 2^{es_n}) - \left(S_n \,.\, (2^{es_n} - 1)\right) \quad (11)$$

$$Total\ Mirror\ Point\ (TMP_n) = \left\lceil \frac{|MP_{n,l} - I'_n|}{2^{es_n}} \right\rceil \quad (12)$$

From here, we can determine the value of the first mirroring point ($MP_{n,1}$) of each sample as in Eq. (13). It is possible for the last mirroring point also to be the first mirroring point which makes the total mirroring point on such sample only one.

$$MP_{n,1} = \begin{cases} MP_{n,l} - \left((TMP_n - 1) \,.\, 2^{es_n}\right), & if\ S_n \le S_{n+1} \\ MP_{n,l} + \left((TMP_n - 1) \,.\, 2^{es_n}\right), & if\ S_n > S_{n+1} \end{cases} \quad (13)$$

3.1.2.4. Embedding payload

The available sample is then ready to be embedded with the bit(s) of the payload. The bit(s) value is determined based on how much embedding space (es) of each sample as in the image below. Payload data are represented as a string of bit(s) values. These bit(s) values will take some digits depending on the embedding space value for each sample.

$$Payload = 0010110010011$$
$$es = \{3,1,2,1,2,2,1,1\}$$
$$Payload\ for\ each\ sample\ (Ps) = \{001,0,11,0,01,00,1,1\}$$

Later, this bit(s) value should be converted into decimal value (Ds) as in Eq. (14). Then, it can be embedded from interpolation value (I') towards its next cover sample value in the same space sample as in Eq. (15), which give the resulting new interpolation value that contains the hidden value (R).

$$Ds_n = decimal(Ps_n)_2 \quad (14)$$

$$R_n = \begin{cases} I'_n + Ds_n, & if\ S_n \le S_{n+1} \\ I'_n - Ds_n, & if\ S_n > S_{n+1} \end{cases} \quad (15)$$

However, this R-value should be checked whether it is getting past through the first mirroring point or not. If the R-value is indeed getting past through it, we need to translate the R-value to another value to avoid getting past beyond the first mirroring point. This check is done as in Eq. (16).

$$R_n = \begin{cases} MP_{n,1} - \left(2^{es_n} - \left|R_n - MP_{n,1}\right|\right), \\ \quad if\ S_n \le S_{n+1}\ and\ R_n > MP_{n,1} \\ MP_{n,1} + \left(2^{es_n} - \left|R_n - MP_{n,1}\right|\right), \\ \quad if\ S_n > S_{n+1}\ and\ R_n < MP_{n,1} \\ R_n,\ otherwise \end{cases} \quad (16)$$

3.1.2.5. Mirror embedding value

After we get the R-value containing the confidential data, the final thing to do in this cycle is to move the value as close as possible towards the next cover sample value. This is done by reflecting the R-value through every mirroring point available in a sample, which will give the final value (R') that serves as stego-audio sample value as in (17).

$$R'_n = \begin{cases} MP_{n,1} + G_n,\ \ if\ S_n \le S_{n+1} \\ MP_{n,1} - G_n,\ \ if\ S_n > S_{n+1} \\ MP_{n,l},\ \ if\ R_n = MP_{n,1} \end{cases} \quad (17)$$

$$G_n = \begin{cases} (TMP_n - 1)\ .\ 2^{es_n} + \left|R_n - MP_{n,1}\right|, \\ \quad if\ TMP_n\ is\ odd \\ (TMP_n - 1)\ .\ 2^{es_n} + \left(2^{es_n} - \left|R_n - MP_{n,1}\right|\right), \\ \quad if\ TMP_n\ is\ even \end{cases} \quad (18)$$

Gap value (G) is used to ease the calculation process of moving R-value to focus on finding the R' value from the first mirroring point perspective and the difference in direction between a cover sample value and its next sample value. It depends on how many mirroring points each sample has, as in Eq. (18), to determine the G value. Then after we find $the\ R'$ value, the cycle repeats for the embedding process on the following interpolation sample.

If, by any chance, every available sample has been embedded, but the payload still has not fully being embedded, then we increase the bit threshold value (bt) by 1. The next step is to repeat the embedding cycle from the beginning of the available cover sample.

3.2 Extraction

Before the extraction process begins, some information is needed for extraction to be done, which is bit threshold value (bt) and last index of the stego-audio sample that contains hidden data. Meanwhile, to make it reversible so that the original audio file can also be recovered, another information is needed, which will be explained later.

The extraction cycle process is similar to the embedding cycle process explained above. Nevertheless, the main purpose is to get confidential data from each sample containing them. As shown in Fig. 4, this cycle consists of finding embedding space, determining interpolation value, determining mirroring points and mirror extraction. Furthermore, similar to the embedding process, this extraction cycle is also done per sample. However, different from it, the processed sample order is reversed from the last stego-audio sample, comprising the hidden data into the beginning of the stego-audio sample.

3.2.1. Finding embedding space

First, we need to determine the embedding space of each sample in a bit, which is similar to the one in the embedding cycle. This value then will be compared to the bit threshold value (bt)to determine how much bit data is hidden in the sample. The process to find these embedding space follows as in Eq. (1) until Eq. (4) consecutively, with S being a set of stego-audio sample values. After that, the result (b) is compared to the bit threshold value (bt) as in Eq. (8) to find the embedding space (es) used in the sample. However, we do not need to do this process for every sample as in the embedding process. We only need to do this cycle from the last sample containing hidden data towards the beginning of the stego-audio sample.

3.2.2. Determining interpolation value

Determining interpolation values in this cycle is needed to determine how many mirroring points are being used and to extract hidden decimal value within a sample. As for determining these values, it is also similar to the one in the embedding process mentioned in Eq. (9) and Eq. (10), consecutively.

3.2.3. Determining mirror points

This step is also similar to the one in the embedding process, starting from determining the last mirroring point, which has the mirror point value closest to the next stego-audio sample value from the checked sample. Formula Eq. (11) shows how to determine the last mirroring point of a sample, then followed by Eq. (12) to find how many mirroring points the checked sample has. After that, also followed by Eq. (13) to determine its first mirroring point, or in this process, the mirror point value closest to the interpolation value of the checked sample.

3.2.4. Mirror extraction

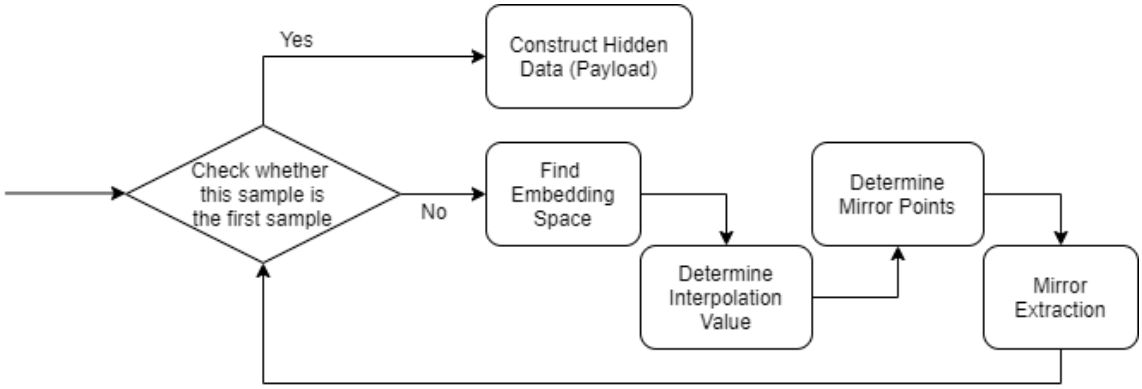

Figure. 4 Specific flow of mirror extraction

Unlike those three previous processes, which is the same as the embedding process, mirror extraction is like the concept in the mirror embedding. Meanwhile, its purpose is to find out how many decimal values a sample hides. To do this, we need to reflect the value of its following stego-audio sample from the checked sample through all mirroring points.

$$R'_n = \begin{cases} MP_{n,l} - G_n, & if\ S_n \le S_{n+1} \\ MP_{n,l} + G_n, & if\ S_n > S_{n+1} \\ MP_{n,1}, & if\ R_n = MP_{n,l} \end{cases} \quad (19)$$

$$G_n = \begin{cases} (TMP_n - 1)\ .\ 2^{es_n} + \left|S_{n+1} - MP_{n,l}\right|, \\ \qquad if\ TMP_n\ is\ odd\ number \\ (TMP_n - 1)\ .\ 2^{es_n} + \left(2^{es_n} - \left|S_{n+1} - MP_{n,l}\right|\right), \\ \qquad if\ TMP_n\ is\ even\ number \end{cases} \quad (20)$$

The formula structure is similar to Eq. (17) and Eq. (18), respectively, but the variables used are slightly different. Gap value (G) used in Eq. (20) has a similar purpose with Eq. (18), which is to ease the calculation process of Eq. (19) so that we can quickly find the value which contains the hidden data (R'). Then, to extract the hidden data (DP) from the R' value, we only need to find the difference between $the\ R'$ value with the interpolation value of the checked sample (I') as in Eq. (21).

$$DP_n = \begin{cases} \left|R'_n - I'_n\right|, & if\ S_n < S_{n+1}\ and\ R'_n \ge I'_n \\ \left|R'_n - I'_n\right|, & if\ S_n > S_{n+1}\ and\ R'_n \le I'_n \\ \left|\left(R'_n + 2^{es_n}\right) - I'_n\right|, & if\ S_n < S_{n+1}\ and\ R'_n < I'_n \\ \left|\left(R'_n - 2^{es_n}\right) - I'_n\right|, & if\ S_n > S_{n+1}\ and\ R'_n > I'_n \end{cases} \quad (21)$$

Finally, we convert this decimal value to its bit representation with the bit length following embedding space value (es) to get the bit data. This one cycle is done through a sample, repeated over every sample from the last index to the beginning of those samples.

4. Experimental results

In this section, we will explain the experiment and its result. This experiment takes two types of data: audio files taken from [20] used as a cover and various payload files containing the hidden message. Specification of the audio file used in this experiment are as follows:

- Audio file is in the format of .wav (Wave Audio Format) file
- Mono channel
- Bit rate of 44100 Hz
- Approximate duration of 3 seconds
- 16 bit-depth

Meanwhile, there are no specific requirements on the payload file, and any form of file is acceptable. In this experiment, we use a basic text file as our payload file. These payload files vary in file size from 1 kb to 100 kb. The scenario of this experiment is to test whether our proposed method has successfully maintained the quality of stego-audio or not by comparing it with other methods: those are the research in [12, 13, 14, 17] as shown in Fig. 5. Here, the hiding step in [14] is without smoothing because it increases the complexity.

Generally, the experimental results by using the proposed method show that increasing the payload size causes decreasing the quality of the stego files, represented by a PSNR (Peak Signal to Noise Ratio) value. Rising the payload size from 1 kb to 10 kb is relatively significant in reducing this quality. It is shown in Fig. 5 that the other hiding techniques implementing interpolation have a significant drop, too. On the contrary, the stego files generated by both [12] and [13], which do not take interpolation, are more stable. This trend, however, may change when the maximum payload capacity is reached.

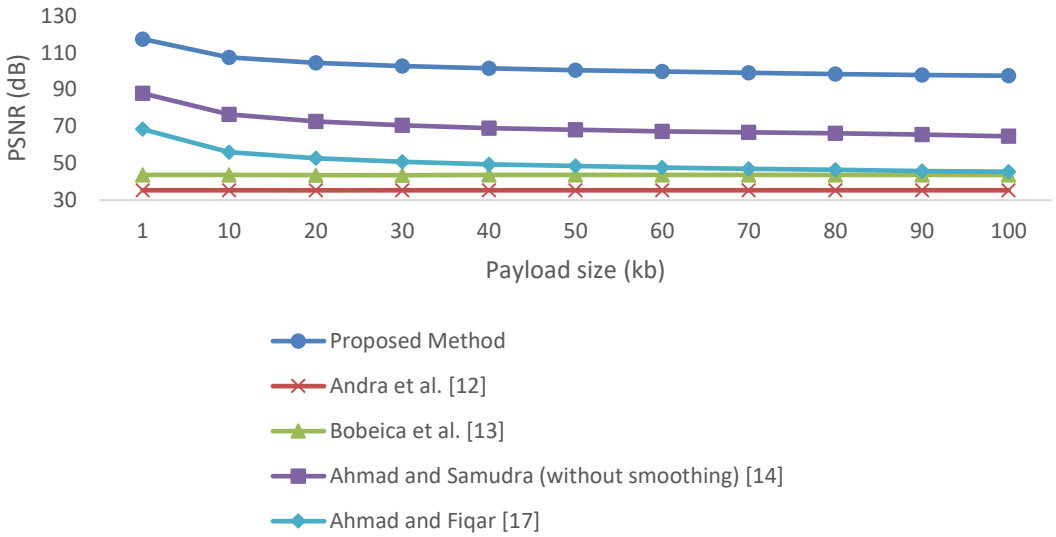

Figure. 5 Comparison of PSNR values between previous and the proposed data hiding methods

From 10 kb to 100 kb, the proposed method produces a gradual PSNR decrease. As in other research, it is common because there is a trade-off between the capacity and the quality. Therefore, in the application level, we should focus on either one of those two factors, depending on the environment and the purpose of the method. Finally, by taking the largest embedded payload size, this technique obtains around 98 dB.

That slight drop pattern is predicted to happen along with the growth of the payload size. Therefore, further expanding the number of bit payloads to more than 100 kb is still possible. Moreover, it is still much higher than those of the other research.

This proposed method has a significantly higher PSNR value because the payload data to be embedded are distributed evenly among every sample available for embedding. Furthermore, we also reduce the difference between the embedded sample value with original sample value by using mirror embedding so that the embedded sample has its value as close as possible to the original sample value.

5. Conclusion

This paper proposes a new steganography method based on the Linear Interpolation method without any additional samples after the embedding process. We also improve the embedding method, which is based on the Reduced Difference Expansion method, into a new Mirror Embedding method to maintain the quality of stego-audio based on its PSNR value. Furthermore, to enhance our Mirror Embedding, we also tried to distribute the payload data to be embedded evenly among every sample available for the embedding process.

The result of our proposed method is satisfactory. A 100 kilobit of payload can be embedded with 97.5 dB, which is pretty good compared to the other method. The Mirror embedding method is proven to be useful to maintain the quality of our stego-audio by keeping the stego sample value to be as close as possible to the original sample value so that the resulting PSNR value can be maintained as well. Also, distributing the payload data evenly among every sample made the embedding capacity seem dynamic to hold more payload data. Furthermore, even though our proposed method is based on Linear Interpolation sampling, we do not increase its total sample so that the resulting stego-audio has the same total sample as the original one.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

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

References

[1] P. Rajba and W. Mazurczyk, “Information Hiding Using Minification”, *IEEE Access*, Vol. 9, pp. 66436–66449, 2021, doi:

10.1109/ACCESS.2021.3077197.

[2] S. Jiao, C. Zhou, Y. Shi, W. Zou, and X. Li, "Review on optical image hiding and watermarking techniques", *Opt. Laser Technol.*, Vol. 109, No. January 2018, pp. 370–380, 2019, doi: 10.1016/j.optlastec.2018.08.011.

[3] L. Rakhmawati, S. Suwadi, and W. Wirawan, "Blind robust and self-embedding fragile image watermarking for image authentication and copyright protection with recovery capability", *Int. J. Intell. Eng. Syst.*, Vol. 13, No. 5, pp. 197–210, 2020, doi: 10.22266/ijies2020.1031.18.

[4] R. Schmitz, "Use of SHDM in commutative watermarking encryption", *Eurasip J. Inf. Secur.*, Vol. 2021, No. 1, pp. 1–12, 2021, doi: 10.1186/s13635-020-00115-w.

[5] L. Rakhmawati, T. Suryani, W. Wirawan, S. Suwadi, and E. Endroyono, "Exploiting self-embedding fragile watermarking method for image tamper detection and recovery", *Int. J. Intell. Eng. Syst.*, Vol. 12, No. 4, pp. 62–70, 2019, doi: 10.22266/ijies2019.0831.07.

[6] C. Lee, J. Shen, Y. Wu, and S. Agrawal, "PVO-Based Reversible Data Hiding Exploiting Two-Layer Embedding for Enhancing Image Fidelity", *Symmetry (Basel).*, Vol. 12, No. 7, p. 1164, Jul. 2020, doi: 10.3390/sym12071164.

[7] X. Wang, C. C. Chang, and C. C. Lin, "Reversible data hiding in encrypted images with block-based adaptive MSB encoding", *Inf. Sci. (Ny).*, Vol. 567, pp. 375–394, 2021, doi: 10.1016/j.ins.2021.02.079.

[8] H. Yao, F. Mao, C. Qin, and Z. Tang, "Dual-JPEG-image reversible data hiding", *Inf. Sci. (Ny).*, Vol. 563, pp. 130–149, 2021, doi: 10.1016/j.ins.2021.02.015.

[9] C. C. Chen, C. C. Chang, and K. Chen, "High-capacity reversible data hiding in encrypted image based on Huffman coding and differences of high nibbles of pixels", *J. Vis. Commun. Image Represent.*, Vol. 76, No. January 2020, p. 103060, 2021, doi: 10.1016/j.jvcir.2021.103060.

[10] D. Xu and S. Su, "Reversible data hiding in encrypted images with separability and high embedding capacity", *Signal Process. Image Commun.*, Vol. 95, No. May 2020, p. 116274, 2021, doi: 10.1016/j.image.2021.116274.

[11] A. K. Sahu, G. Swain, M. Sahu, and J. Hemalatha, "Multi-directional block based PVD and modulus function image steganography to avoid FOBP and IEP", *J. Inf. Secur. Appl.*, Vol. 58, No. March, p. 102808, 2021, doi: 10.1016/j.jisa.2021.102808.

[12] M. B. Andra, T. Ahmad, and T. Usagawa, "Medical Record Protection with Improved GRDE Data Hiding Method on Audio Files", *Eng. Lett.*, Vol. 25, No. 2, 2017.

[13] A. 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 Int. Conf. Syst. Theory, Control Comput*, pp. 810–815, 2018, doi: 10.1109/ICSTCC.2018.8540672.

[14] T. Ahmad and Y. Samudra, "Reversible data hiding with segmented secrets and smoothed samples in various audio genres", *J. Big Data*, Vol. 7, No. 1, 2020, doi: 10.1186/s40537-020-00360-3.

[15] A. Benhfid, E. B. Ameur, and Y. Taouil, "Reversible steganographic method based on interpolation by bivariate linear box-spline on the three directional mesh", *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 32, No. 7, pp. 850–859, 2020, doi: 10.1016/j.jksuci.2018.09.016.

[16] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein, and H. F. A. Hamed, "Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques", *IEEE Access*, Vol. 9, pp. 31805–31815, 2021, doi: 10.1109/ACCESS.2021.3060317.

[17] T. Ahmad and T. P. Fiqar, "Enhancing the performance of audio data hiding method by smoothing interpolated samples", *Int. J. Innov. Comput. Inf. Control*, Vol. 14, No. 3, pp. 767–779, 2018.

[18] B. Peng and J. Yang, "An optimized algorithm based on generalized difference expansion method used for HEVC reversible video information hiding", In: *Proc. of IEEE 17th International Conference on Communication Technology (ICCT)*, Chengdu, China, 2017, doi: 10.1109/ICCT.2017.8359914.

[19] S. Xiang and Z. Li, "Reversible audio data hiding algorithm using noncausal prediction of alterable orders", *Eurasip J. Audio, Speech, Music Process.*, Vol. 2017, No. 1, 2017, doi: 10.1186/s13636-017-0101-9.

[20] "IRMAS: a data set for instrument recognition in musical audio signals." https://www.upf.edu/web/mtg/irmas/.(Accessed August 2017). [Online].