Research article

Hiding data in audio files: A smoothing-based approach to improve the quality of the stego audio

Tohari Ahmad *, Muhammad Hanif Amrizal, Waskitho Wibisono, Royyana Muslim Ijthadie

Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia

ARTICLE INFO

Keywords:
Computer science
Data embedding
Data protection
Data hiding
Information security
Multimedia security

ABSTRACT

In this digital era, transmitting data through a computer network has become common. Moreover, some applications have also been developed to do it. Nevertheless, users may not be aware of the security aspect of this data transmission, which can lead to disclosing this private message. In a case when a sensitive message is the object to transmit, a security mechanism should be applied. Data hiding is one of the methods introduced to work for this issue. In this algorithm, the message is embedded into the cover, such as an audio file, before being transmitted; on the other side, the recipient extracts it. However, the size of the message and the quality of the resulted stego data are still challenging. In this paper, we focus on these two problems by considering some factors: embedding space, embedding process, reducing, and smoothing steps. Firstly, the audio signal is discretized to obtain audio samples. Next, these samples are interpolated to provide spaces for hiding the secret. Considering that the quality of the generated stego audio is likely to drop, reducing and smoothing steps are designed. The experimental results show that this approach can improve the quality of the stego based on the specified payload capacity. That is, there is an increase in PSNR value of at least 10 dB, depending on the payload size and the methods.

1. Introduction

For some decades, information and communication technology has been seeing significant growth. This development has made it easy for people to transmit data between devices anywhere and anytime. Moreover, at present, limited devices such as smartphones can process data much more quickly than before. Nevertheless, this advancement may attract illegitimate users who actively attack the computer system [1], for example, by intercepting the transmitted data. This activity causes a security problem for secret and private information such as financial and medical data.

Some methods have been proposed to overcome this security issue that can be grouped into a network and information security, which complement each other. While the former can be implemented by designing a suitable intrusion detection system (IDS), honeypot, or other security tools; security tools can be designed to conceal the secret. This concealment can be achieved in some ways, including encrypting [2] or embedding the secret in a medium; this is called a data hiding method [3]. In this research, the terms hiding and embedding have the same definition.

In its design, data-hiding may use either image [4], audio [5], video [6], or even text [7, 8] to carry the secret. As predicted, each medium, called the carrier or cover, has its own characteristics. For example, an image concerns visual quality, while audio focuses on the voice. In the data-hiding terminology, quality is defined as the similarity level of the carrier between, before, and after the secret is embedded, regardless of the carrier being used. In fact, research in data-hiding is mostly carried out by using the image [3, 9]; on the contrary, that exploring audio is only a few, and even less than video. Furthermore, the embedding process can be performed either in the spatial domain, where the secret is directly hidden in the cover, or in the frequency domain, wherein the cover must be transformed before the embedding process, for example, by employing Fast Fourier Transform (FFT) [10] or Wavelet Transform [11].

Spatial domain-based data-hiding methods have been popular to protect secret data. Despite its limitation in generating excellent stego quality, this method may accommodate more secrets than that of the transformed-based algorithm. Therefore, improving the visual or audio quality is still a challenge. Additionally, those two algorithms may be implemented in a different environment, depending on the purpose of the required system.

* Corresponding author.
E-mail address: tohari@if.its.ac.id (T. Ahmad).

https://doi.org/10.1016/j.heliyon.2020.e03464
Received 1 October 2019; Received in revised form 1 January 2020; Accepted 19 February 2020

2405-8440/© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
In general, the previous algorithms can be classified in some groups, including [12]: LSB modification, histogram shifting (HS), pixel value difference (PVD), and difference expansion (DE). Two primary factors are still a challenge in the data-hiding field: the capacity of the secret being embedded in the cover, and the quality of the resulted stego file [13]. In this paper, we work on those two problems by focusing on the spatial domain and taking audio as the cover. Here, the embedding is implemented in the generated samples. Moreover, the difference between the stego audio and its corresponding file (cover) is reduced.

The remaining sections of this paper are structured, as follows. Section 2 describes the previous works of data-hiding. Section 3 explains the proposed method, followed by its experimental results and analysis in Section 4. The comparison of the method is provided in Section 5. Finally; the conclusion is presented in Section 6.

2. Data-hiding methods

Overall, the data-hiding method can be illustrated in Fig. 1, where an image or audio can be taken as the cover. The hiding is done by the sender before the data transmission starts. On the other side, the recipient receives the stego and extracts it to obtain the secret. Slightly different from the spatial-based approach, in the frequency-based method, the cover must be first transformed before being embedded. In [14], Lei et al. make use of Quaternion Wavelet Transform to achieve robustness and an appropriate capacity of the secret. In their research, the transformation is combined with Particle Swarm Optimization (PSO) and chaotic signals to improve the performance further. In many situations, the recipient also reconstructs the original medium (cover). In this case, the reversibility of the cover is a crucial factor, and the corresponding algorithm is classified in Reversible Data Hiding (RDH). Therefore, both the secret and the cover can be obtained without losing information. Furthermore, RDH is one of the appropriate methods to use for providing privacy of the users, protecting copyright, and authenticating users.

Generally, a data hiding algorithm is independent of the types of cover. Nevertheless, an algorithm cannot be just implemented on different media. So, even though the proposed method explores audio as the cover, in this section, we review both image- and audio-based algorithms. Moreover, many embedding algorithms are initially developed by using an image. One of the popular data-hiding methods, Difference Expansion (DE) [15], which was initially introduced for image, has been further developed for audio [16]. This algorithm ([15]) was the primary reference for the following data-hiding research. In practical terms, DE is designed for 2D, which represents the characteristic of an image. In order to manipulate this construction, audio samples (which consist of 1D data) are formed as two rows called bigits. By using the same type of medium, Andra et al. [5] improved its capability. In their research, intelligent partitioning combined with multilayer embedding was applied to the embedding process. It is claimed that their result was better than the previous method.

In 2015, Joong and Yoo [17] proposed a method by exploring the use of interpolation and the Least Significant Bit (LSB) modification. In their research, which utilized an image, they found that the algorithm is better at hiding the secret. In other research, Zhang et al. [18] generated prediction errors for the histogram, which was then shifted. Here, both the cover and the secret can successfully be retrieved from the stego. Echo hiding is the topic that is investigated in [16]. In order to ensure that only legitimate users can recover the hidden data, the shared-secret algorithm is applied in [19]. In this research, the number of users who are able to obtain the data can be determined according to the specified parameters. This system may increase the security of the secret since the extraction should be carried out by some users concurrently.

By manipulating the difference between values (e.g., a pixel of an image, a sample of audio) and a modulus function, Manirho and Ahmad [20] embed the secret. In that research, they further explore the use of both positive and negative difference values, unlike previous research. By employing this characteristic, they can raise the quality of the stego data from the existing method. In the case video is utilized as the cover, both image and audio are often used at the same time [21]. It means that the method combines the image- and audio-based data-hiding. As a result, the capacity of the secret significantly goes up; however, the quality of the stego drops.

In [9], Bobeica et al. implemented the Prediction Error Expansion (PEE) on audio RDH, which was initially introduced for image [22]. Their research intends to find an optimal threshold value to specify which samples should be used to hold the secret, along with the prediction error value. Indirectly, the threshold also affects the value of the expanded samples. The predictor value itself may consider a relatively small or large number of neighboring samples, as in [23] and [24], respectively.

Jung and Yoo [25] propose to expand the cover before the embedding process starts. It is intended to accommodate more secrets than the original cover can hold. While this algorithm is implemented in an image, in the next research, Ahmad and Fiqar [26] extend this method to an audio file. Some additional steps have been updated to make it able to maintain the quality of the respective stego. For evaluation, various types of files have been applied; their results reflect that the proposed method is suitable to implement.

Variations of the methods have also been introduced by combining them with other methods, such as secret sharing [27], homomorphic encryption [28, 29], multiple predictors [30], and the encrypted-mesh model [31]. Furthermore, Hua et al. [32] analyze some possible attacks on the resulting stego data and explain other aspects of the existing algorithms, for example, commercialization and its patent application. Even though this is intended for watermarking, which has a purpose that is slightly different from data hiding, the overall analysis and
method are the same. Overall, those methods are to prevent either active or passive attacks [33].

3. Secret data protection

This proposed method is developed based on those previous works, especially [17, 25]. The algorithm consists of two main processes: embedding and extraction. In the embedding process, the sender puts the secret in the cover which is then extracted by the receiver to obtain both the original cover and the secret, by reversing the embedding step. The overall process of the embedding can be presented in Fig. 2. In general, the audio cover is sampled before being normalized and interpolated. At the same time, the secret is binarized and segmented before being embedded. Next, some finishing steps are applied to have the stego audio, which includes smoothing to reduce the possible noise.

3.1. Embedding process

Before being embedded, the cover which is a continuous signal must be discretized to have audio samples. For this purpose, the audio signal is converted into 16-bit audio samples (see Fig. 3 for the example). In this pre-embedding process, we consider these original samples, which are represented by \( s_n = (y_1, y_2, y_3, \ldots) \). Like [26, 25], this discrete signal is interpolated to make more spaces within the signal. In this research, linear interpolation is applied (see (1)), to obtain \( s_k = (\overline{y_1}, \overline{y_2}, \overline{y_3}, \ldots) \), whose illustration is provided in Fig. 4. Here, \( y_n \) and \( \overline{y_n} \) respectively represent the magnitude of the original and the corresponding samples generated by the interpolation process in a group comprising \( y_{n-1}, y_n \) and \( y_{n+1} \) samples, where \( n \) is the index. It is worth noting that the index (location) of the original and interpolating samples is consecutively odd and even. After the interpolation is applied, a vector of samples \( s_T = (\overline{y_1}, \overline{y_2}, y_3, \overline{y_4}, \ldots, y_{n-1}, y_n, y_{n+1}) \) is obtained as the combination of \( s_0 \) and \( s_k \). Supposed there are 3 samples as follows: \( y_1 = 80, y_2 = 86 \) and \( y_3 = 90 \). The generated vector of samples is \( s_T = (80, 83, 86, 88, 90) \).

\[
\overline{y_n} = \left[ y_{n-1} + \frac{(n - (n - 1))(y_{n+1} - y_{n-1})}{(n + 1) - (n - 1)} \right]
\]

The space of the embedding is allocated by grouping some consecutive original samples, called segment \( k \), which in this research it is set as either 2, 3, 4, or 5. From each segment, we find both the highest \( (y_{\text{max}}) \) and the lowest \( (y_{\text{min}}) \) values of \( j \)th segment by using (2) and (3), respectively. Here, \( y_j \) is the \( j \)th original sample in a segment. Next, the average \( (d_j) \) of these two values and the respective sample of \( j \)th segment is calculated by using (4).

\[
y_{\text{max}} = \max(|y_j|_{n+k}) \tag{2}
y_{\text{min}} = \min(|y_j|_{n+k}) \tag{3}
d_j = \left[ \frac{y_{j,\text{max}} + y_{j,\text{min}} + y_{j,n}}{3} \right] \tag{4}
\]

By taking \( k = 2 \), from the vector \( s_k \), it is found that:

\[
y_{1,\text{max}} = \max(80, 86) \Rightarrow y_{1,\text{max}} = 86
\]

\[
y_{1,\text{min}} = \min(80, 86) \Rightarrow y_{1,\text{min}} = 80
\]

\[
y_{2,\text{max}} = \max(86, 90) \Rightarrow y_{2,\text{max}} = 90
\]

\[
y_{2,\text{min}} = \min(86, 90) \Rightarrow y_{2,\text{min}} = 86
\]

Therefore, the average of the samples in each segment is:

\[
d_1 = \left[ \frac{86 + 80 + 86}{3} \right] \Rightarrow d_1 = 82
\]

\[
d_2 = \left[ \frac{90 + 86 + 86}{3} \right] \Rightarrow d_2 = 87
\]

There are some conditions which determine the space of embedding \( (d'_j) \) of segment \( j \). That is, it depends on the value of the corresponding sample and its neighbors. If the sample is the last one in the cover, then (5) is applied. Otherwise, the value of \( d'_j \) is determined by the comparison between \( y_n \) and \( y_{n+1} \). That is, (6), (7) or (8) is employed if \( y_n = y_{n+1} \), \( y_n < y_{n+1} \) or \( y_n > y_{n+1} \), respectively.

\[
d'_j = |d_j - y_{n-1}|, \text{ if } y_n = \text{EOF}(s_k) \tag{5}
\]

\[
d'_j = 1, \text{ if } y_n = y_{n+1} \tag{6}
\]

\[
d'_j = \begin{cases} 
|y_{n+1} - d_j|, & \text{if } d_j \leq \frac{y_n + y_{n+1}}{2} \\
|y_n - d_j|, & \text{otherwise}
\end{cases} \tag{7}
\]

In the embedding process, the secret is converted into 16-bit samples, and it is then extracted by the receiver to obtain both the original cover and the secret. The embedding process is shown in Fig. 4. Next, some finishing steps are applied to have the stego audio, which includes smoothing to reduce the possible noise.
The value of $d_j'$ is then used for determining the number of bits ($N_j$) which can be embedded to the corresponding sample by using \( (9) \).

\[
N_j = \lceil \log_2(d_j') \rceil \times 2
\]  

By using \( (5)-(9) \), we find that $d_1' = 6$ and $d_2' = 10$. Therefore, the total of bits to be embedded in $y_1$ and $y_2$ are $N_1 = 4$ and $N_2 = 6$, respectively. It can also be inferred that the number of bits per sample does not have to be the same. So, this value varies depending on the other sample values in the corresponding segment.

Suppose the secret of being protected is $\rho = 1010000111$. This payload is split according to the values of $N_1$ and $N_2$. Based on them, the first and second groups of the secret are respectively:

\[
p_1 = (101010)_2 \rightarrow p_1 = (1010)_{10}
p_2 = (0001111)_2 \rightarrow p_2 = (7)_{10}
\]

This secret group of bits ($p_j$) is reduced $i$th times to achieve a simpler form of the corresponding secret ($p_{j,i}$). To this purpose, \( (10) \) is implemented whose output relies on the value of the secret—whether it is even or odd. This process can be repeated until the output is either 0 or 1. The resultant value is then used in the extraction process by the receiver. For a simple example, here we specify that $i = 1$, so $p_{1,i} = 5$ and $p_{2,i} = 3$.

\[
p_{j,i} = \begin{cases} 
  p_{j,i-1} - 1, & \text{if } \text{mod}(p_{j,i-1}, 2) = 0 \\
  p_{j,i-1}, & \text{if } \text{mod}(p_{j,i-1}, 2) = 1
\end{cases}
\]  

(10)

Let $\overline{y}_j$ and $\overline{y}'_j$ be the $j$th interpolating sample before and after being embedded by the secret. The group of bits that has been allocated is added to that interpolating sample, as depicted in \( (11) \). It is shown that the original sample ($y_j$) does not hold the hidden data. Therefore, their values do not change at all.

\[
\overline{y}'_j = \overline{y}_j + p_{j,i}.
\]  

(11)

In this example, the generated values are:

\[
\overline{y}'_1 = 83 + 5 = 88
\]

For the next extraction process, the sender should send some information taken from \( (13) \) and \( (14) \) to the receiver. This information can be summarized in \( (15) \), which constructs a set of key $k_{j,i} = (k_{j,i,1}, k_{j,i,2}, k_{j,i,3})$ for the $i$th smoothing step of the $j$th sample.

\[
k_{j,i,1} = \text{mod}(t_{j,1}, 2)
k_{j,i,2} = \text{mod}(t_{j,2}, 2)
k_{j,i,3} = \text{mod}(S_j, 4)
\]  

(15)

By using this smoothing step, we have that $t_{1,1} = 166, t_{1,2} = 171$ for the first sample, and $t_{2,1} = 166, t_{2,2} = 171$ for the second sample. Correspondingly, the following values are obtained: $S_1 = 334$ and $S_2 = 353$. Therefore, after the first smoothing iteration, the embedded samples are $\overline{y}'_1 = 83$ and $\overline{y}'_2 = 88$. This illustration is presented in Figs. 5, 6 and 7. It is shown that an iteration has made the embedded samples smoother than before. From the example, we can summarize the audio samples as follows:

- Original samples $\rightarrow s_0 = (80, 86, 90)$
- Interpolating samples $\rightarrow s_1 = (83, 88)$
- Combined samples $\rightarrow s_4 = (80, 83, 86, 88, 90)$
- Embedded samples (non-smoothing) $\rightarrow \overline{y}'_1 = 83, \overline{y}'_2 = 88$
- Combined samples (non-smoothing) $\rightarrow s'_4 = (80, 83, 86, 91, 90)$
- Smoothing embedded samples $\rightarrow \overline{y}''_1 = 88, \overline{y}''_2 = 91$
- Stego samples $\rightarrow s''_4 = (80, 83, 86, 88, 90)$
- Smoothing key $\rightarrow k_{1,1} = (0, 1, 2), k_{2,1} = (0, 1, 1)$

Those samples can be further smoothed by carrying out this step several times, depending on the specified level. It can be inferred that doing more smoothing steps result in reducing the noise. Nevertheless,
Fig. 5. Original and interpolating audio samples.

![Original sample](image1)

![Interpolating sample](image2)

**Fig. 5.** Original and interpolating audio samples.

Fig. 6. Embedded sample before smoothing.

![Original sample](image3)

![Embedded sample (before smoothing)](image4)

**Fig. 6.** Embedded sample before smoothing.

Fig. 7. Embedded sample after smoothing.

![Original sample](image5)

![Embedded sample (after smoothing)](image6)

**Fig. 7.** Embedded sample after smoothing.

Doing this stage may take additional time, even though it is not significant. In the case that the smoothed signal has reached the optimal value, applying further smoothing does not change the result, as shown in this smoothing example where $y_1^S = 83$ and $y_2^S = 88$ cannot be further optimized.

In more detail, the comparison of the proposed and existing base methods ([25, 17, 26, 5, 9]) is presented in Table 1. It is shown that this method adds steps that previously did not exist, for example, the reducing step. Moreover, the embedding process is simpler than the one in [26].

3.2. Extraction process

In the extraction process, the cover medium and the secret data are taken from the stego. Principally, obtaining the secret is compulsory, while that of the cover is optional, depending on the reversibility of the method. Nevertheless, in this research, we work to extract those two types of data. This process is generally performed in the reverse order of the embedding process, as provided in Fig. 8. It is depicted that the normalization and smoothing are applied to the stego file, similar to the embedding step. Two parallel processes are designed to obtain both the original cover and the secret. This has made each step does not depend each other, which may speed up the processing time and reliability of both cover and secret data.

According to the stego audio resulting from the embedding step, the receiver has $s''_r = (80, 83, 86, 88, 90)$. Based on the fact that the odd-indexed sample is the original one, the receiver obtains $s_1 = (80, 86, 90)$ and $y_1^S = (83, 88)$. So, the original samples are interpolated by using (1) results in $\tilde{y}_1 = 83$ and $\tilde{y}_2 = 88$. These values are then used to obtain $\tilde{S}_1$ and $\tilde{y}_j$ by reversing the smoothing steps in (12)-(14). For this purpose, we have (16) to generate $\tilde{y}_j = 88$ and $\tilde{y}_2 = 91$. Therefore, those non-smoothing samples can be depicted as $s'_r = (80, 88, 86, 91, 90)$.

\[
\tilde{y}_j' = \frac{\left((\tilde{y}_j' \times 4) + \kappa_{j,1}\right)}{2} + \kappa_{j,2} - y_n. \tag{16}
\]

Next, by inverting (11) we have $p_{1,1} = 88 - 83 = 5$ and $p_{2,1} = 91 - 88 = 3$. Consecutively, these values are proceeded by (10). At this step, the payload $p_{1} = (10)_{10} = (1010)_2$ and $p_{2} = (7)_{10} = (000111)_2$ are held by considering the corresponding number of bits $N_1$ and $N_2$ which have been specified in (9). It is worth noting that the receiver is able to calculate these values. This means that the sender does not have to transmit them.

Finally, the whole payload is extracted to have $p = 010000111$. Correspondingly, the original audio sample is obtained by taking $s_1 = (80, 86, 90)$. Therefore, both the payload and the cover can be thoroughly reconstructed the same as the original values. It means that this method is reversible.

4. Experimental results

In order to evaluate the performance of the proposed method, we measure two values similar to other research such as in [26]. The first is the value of PSNR (peak signal to noise ratio) to find the similarity level of the cover between before and after the payload is embedded. Represented in dB (decibel), it is intended that this stego file is not much different from the original cover, which may attract the suspicion of the public. So, the better the quality, the more the similarity. The PSNR value itself is calculated by using (17), where MAX is the maximum value of the samples, and the corresponding MSE (mean-squared error) is defined in (18). Because the audio is a 1D array (vector), the value of $n$ is 1; while that of $n$ depends on the number of the total samples. The second value is the amount of payload (in bits) which can be embedded to the cover without much affecting the cover. Higher capacity means
Table 1
Comparison between [25], [17], [26], [5], [9] and the Proposed Method (inspired by [26]).

| Parameters                  | Method of [25] | Method of [17] | Method of [26] | Method of [5] | Method of [9] | Proposed Method |
|----------------------------|----------------|----------------|----------------|---------------|---------------|----------------|
| Original Medium            | Image          | Image          | Audio          | Audio         | Audio         | Audio          |
| Interpolation method       | Neighbor mean  | Image          | Audio          | Audio         | Audio         | Audio          |
| Interpolation (NMI)        | Static         | Static         | Static         | Dynamic       | Dynamic       | Dynamic        |
| Size of payload            | Dynamic        | Static         | Dynamic        | Dynamic       | Dynamic       | Dynamic        |
| Reducing                   | None           | None           | None           | None          | None          | None           |
| Smoothing                  | None           | None           | None           | None          | None          | None           |

![Diagram](image)

Fig. 8. Extraction process.

that more secret data can be protected. These two values, however, are often inversely proportional. There is a trade-off between them.

\[
\text{PSNR} = 10 \times \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}} \right) \quad (17)
\]

\[
\text{MSE} = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[ s(i,j) - s(i,j) \right] \quad (18)
\]

Similar to [26], for this evaluation, audio files are taken from [35] comprising 15 files, from Audio1 to Audio15 (see Table 2) are used. This data set consists of 3 genres (i.e., Country-Folk, Classical, Pop-Rock), each of which contains five instruments play three seconds (i.e., Cello, Acoustic Guitar, Piano, Saxophone, Voice). The payload is randomly generated whose size is varied, from 1 kb to 100 kb, as presented in Table 3. The experiment is carried out in some scenarios as follows.

4.1. Size of segment

In this experimental scenario, the size of segments is evaluated, from \( k = 2 \) to \( k = 5 \). We believe that this range represents the trend of possible values. It is found that in general, the bigger the size of segment results to a higher number of payload which can be held by the cover, as shown in Fig. 9. This graph shows that the pattern does not depend on the characteristics of the cover. Nevertheless, as predicted, the quality of the stego may decrease along with the increase of the corresponding capacity.

It is known that more significant segments enlarge the number of samples to explore. This means that many sample values within segments are available. Consequently, the number of spaces can be used goes up. Contrarily, a smaller segment reduces the possible space. This condition affects the selection of the maximum and minimum values that their average along with the respective sample in the segment relies on.
Moreover, a segment overlaps with the next segment minus 1 sample. So, a sample may be used for several other segments depending on the size of the previously specified segment. In other words, multiple-embedding can be implemented in each segment, which results in improving its capacity.

4.2 Level of reduced payload

The effect of the reducing level is provided in Fig. 10 by setting its value from $i = 0$ (not reduced) to $i = 6$, and $k = 2$. It is shown that the increase in PSNR value is proportional to that of reducing the level. Without applying the reduction before the embedding, the resulted PSNR is the lowest, and it rises according to the value of $i$. In general, there is an increase of about 5 dB in the PSNR value along with the increasing $i$.

This figure also explains that greater payloads stored in the cover results in a lower stego quality. As predicted, there is a trade-off between those two factors. On average, raising the payload size from 1 kb to 100 kb decreases about 25 dB of the stego quality.

It is also found that extending the size of the payload leads to reducing the gap of the PSNR of the respective payload size. For example, the PSNR difference resulted by $p = 1$ kb and $p = 10$ kb is higher than that of $p = 10$ kb and $p = 20$ kb, and so on. It means that this method has more influence on the smaller payload than on the bigger ones. Therefore, increasing the capacity of payload to a certain size may not greatly reduce the quality of the resulted stego. In other words, this proposed method has a better effect on bigger secret data. It is likely that the secret is distributed evenly among the samples. According to our previous research [36], this condition is better than when most secrets reside in a small group of samples.

4.3 Level of smoothing

Here, the effect of the smoothing step on improving the quality of the resulted stego file is to be evaluated. In this scenario, we vary the smoothing level from 1 to 6. An example of the experimental result is provided in Fig. 11, where Audio1 is employed as the cover to carry various sizes of payload. It is shown that a single smoothing step may increase about 15 dB in PSNR value. Nevertheless, there is a maximum number of how many times the smoothing step can be applied. As presented in Fig. 11, by setting $p = 1$ kb and applying the smoothing 6 times the PSNR value is infinite.

Furthermore, when this smoothing step is applied to Audio2 as the cover, the infinite value of PSNR is achieved in the 4th iteration for all sizes of payload (see Fig. 12), which is better than that of Audio1. It means that by using Audio2, a fewer number of smoothing step has been able to make the stego similar to the cover.

In more detail, it is also presented that smaller payload tends to achieve this condition with less smoothing steps than the bigger one, regardless of the characteristic of the cover. For example, different covers have been used in Fig. 11 and Fig. 13. It is shown that the payload $p = 1$ kb takes 6 and 3 times, respectively, lower than others. This is consistent with the results of the previous analysis.

4.4 Effect of genres

In this scenario, we evaluate the effect of the audio genre on the quality of the stego. For this purpose, we measure the PSNR of various sizes of the payload by setting the parameters as minimum as possible, i.e., $k = 2$, without payload reduction and single smoothing step. From the experimental results provided in Fig. 14, it is found that (Country-
Folk, Voice) is the highest, followed by (Classical, Cello), (Classical, Saxophone), and (Classical, Acoustic guitar). On the other hand, Pop-Rock combined with various instruments, produces low PSNR.

This PSNR measurement, in general, is inversely proportional to the capacity of the payload, as previously shown in Fig. 9. It is found that the level of differences between samples affects the PSNR value. This means that a lower difference maintains a higher PSNR level but a smaller size of the payload, and vice versa. Therefore, in the case that there are secret bits functioning as the main factors, Pop-Rock can be considered. Otherwise, Country-Folk or Classical is the best choice.

For the comparison, the methods of [25], [26], [5] and [9] are implemented whose results are provided in Fig. 15 and Fig. 16. They are evaluated by using the same experimental data. Respectively, those figures represent both [26] and the proposed method with single smoothing, while the others do not have it. As shown in Fig. 15, [25] has the best result. However, after being smoothed three times, the proposed method is superior in the most cover and payload size variations (see Fig. 16). At this stage, the stego quality of [26] is still relatively low, so that it is slightly higher than [5]. However, there is an increase in the quality level. On the contrary, the performances of [25], [5], and [9]
The effect of genre of audio cover on the PSNR of stego, where $k = 2, i = 0$.

Comparison between Jung and Yoo [25], Ahmad and Fiqar [26], Andra et al. [5], Bobeica et al. [9] and the proposed method, in terms of the quality with single smoothing.

Comparison between Jung and Yoo [25], Ahmad and Fiqar [26], Andra et al. [5], Bobeica et al. [9] and the proposed method, in terms of the quality with three times smoothing.

Comparison between Jung and Yoo [25], Ahmad and Fiqar [26], Andra et al. [5], Bobeica et al. [9] and the proposed method, in terms of the quality with four times smoothing.

According to these results, it can be inferred that the proposed method requires less smoothing iteration than [26] to achieve the same level of the stego quality.

5. Discussion

As previously described, the research in audio-based data hiding methods is less than that of image-based [3], despite its possible superiority. The use of interpolation has extended the number of samples in the respective stego being generated. Consequently, the sampling rate may change; otherwise, the playing time of the audio is longer than the original audio cover. The advantage is that it is able to carry more bits, as previously described. This has been proven in the experimental results provided in Table 1. Furthermore, different interpolation algorithms likely have similar results, so in this research, we take the linear method due to its simplicity.

Dynamic payload capacity per sample has been useful to keep the quality of the stego audio. It is because not all samples on the cover are stable because they have no smoothing step as previously described. Moreover, finding an appropriate embedding threshold value in [9] is still a challenge, which affects the performance.

When the smoothing is applied four times, the proposed method is the best in all covers and payloads as depicted in Fig. 17. There is also a rise in the performance of [26]; nevertheless, its improvement cannot exceed that of [25]. It is predicted that by running more smoothing steps, [26] will be better than [25]. According to these results, it can be inferred that the proposed method requires less smoothing iteration than [26] to achieve the same level of the stego quality.
have the same suitable number of bits to take. Some factors affect this characteristic, for example, the value of the previous and the next samples, as described in Section 3. Finding the details of these factors, along with their weight, is still challenging. That is, too many secret bits in a sample drop the quality, and too few bits cause the embedding process to fail.

To the best of our knowledge, only a few methods which design both reducing and smoothing steps in the embedding process. Indeed, these additional steps have made the process longer, but it is only in about a second which we may ignore it. Moreover, common devices are more advanced that the computation, in this case, is not the problem. In fact, reducing and smoothing have been able to increase the quality of the stego audio as provided in Section 4.

6. Conclusions

In this paper, audio-based data-hiding is proposed for protecting secret data. It is shown that without using a location map, this proposed method still works properly by taking the information about the sequence of the respective samples. Furthermore, the segmentation and smoothing steps designed for this scheme can increase the quality of the stego. It is also depicted that the genre of the audio has a significant effect on the performance or the secret size. According to the experimental results, this method is superior to existing ones. That is, it can obtain more than 90 dB, while others may not achieve the same performance. Moreover, the proposed method takes a smaller number of iteration than the previous research to have a certain stego quality level.

In the future, this method can be extended to raise the quality of the stego file further. This improvement may be made by more effectively specifying the size of the segments. That is, the size is dynamic according to some considered factors, including possible information should be stored. Additionally, the interpolation is explored to find the most appropriate one, considering the size and the number of the samples.

Declarations

Author contribution statement

Tohari Ahmad: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Muhammad Hanif Amriral: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Waskitho Wibisono, Royyana Muslim Jithiadic: Contributed reagents, materials, analysis tools or data.

Funding statement

This work was supported by Ministry of Research, Technology and Higher Education, the Republic of Indonesia under WCR research grant No. 718/PKS/ITS/2019.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] L. Hadlington, Human factors in cybersecurity; examining the link between Internet addiction, impulsivity, attitudes towards cybersecurity, and risky cybersecurity behaviours, Heliyon 3 (7) (2017).
[2] F.R. Shareef, A novel crypt technique based ciphertext shifting, Egypt. Inform. J. (2020), https://doi.org/10.1016/j.eij.2019.11.002, in press.
[3] Y. Shi, X. Li, X. Zhang, H. Wu, B. Ma, Reversible data hiding: advances in the past two decades, IEEE Access 4 (2016) 3210–3237.
[4] Y. Lin, C. Wang, W. Chen, F. Lin, W. Lin, A novel data hiding algorithm for high dynamic range images, IEEE Trans. Multimed. 19 (1) (2017) 196–211.
[5] M.B. Andra, T. Ahmad, T. Usagawa, Medical record protection with improved GRDE data hiding method on audio files, Eng. Lett. 25 (2) (2017) 112–124.
[6] M. Ramalins, N.A.M. Is, A data-hiding technique using scene-change detection for video steganography, Comput. Electr. Eng. 54 (2016) 423–434.
[7] A.A.Z. Gutub, K.A. Alaseri, Refining Arabic text stego-techniques for shares memorization of counting-based secret sharing, J. King Saud Univ. Comput. Inf. Sci. (2020), https://doi.org/10.1016/j.jksuci.2019.06.014, in press.
[8] S. Al-Nofae, A. Gutub, M. Al-Ghamdi, Enhancing Arabic text steganography for personal usage utilizing pseudo-squares, J. King Saud Univ. Comput. Inf. Sci. (2020), https://doi.org/10.1016/j.jksuci.2019.06.010, in press.
[9] A. Bobeica, I.C. Dragos, I. Lacuta, D. Cohnec, F. Albu, F. Yang, Capacity control for prediction error expansion based audio reversible data hiding, in: Proc. 22nd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 2018, pp. 810–815.
[10] T.T. Manunggal, D. Arifinato, Data protection using intercral quantified-phase steganography on stereo audio signals, in: Proc. IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 3817–3821.
[11] A. Miri, K. Faes, Adaptive image steganography based on transform domain via genetic algorithm, Optik 145 (2017) 158–168.
[12] M. Hussain, A.W.A. Wahab, Y.I.B. Idris, A.T.S. Ho, K. Jung, Image steganography in spatial domain: a survey, Signal Process. Image Commun. 65 (2018) 46–66.
[13] M. Hussain, A.W.A. Wahab, N.B. Anuar, R. Salleh, R.M. Noor, Pixel value difference steganography techniques: analysis and open challenge, in: Proc. IEEE International Conference on Consumer Electronics-Taiwan, Taipei, Taiwan, 2015, pp. 21–22.
[14] B. Lei, F. Zhou, E. Tan, D. Ni, H. Lei, S. Chen, T. Wang, Optimal and secure audio watermarking scheme based on self-adaptive particle swarm optimization and quaternion wavelet transform, Signal Process. 113 (2015) 80–94.
[15] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 890–896.
[16] K.-C. Choi, C-M. Pun, C.L. Chen, Application of a generalized difference expansion based reversible audio data hiding algorithm, Multimed. Tools Appl. 74 (6) (2015) 1961–1982.
[17] K. Jung, K. Yoo, Steganographic method based on Interpolation and LSB substitution of digital images, Multimed. Tools Appl. 74 (6) (2015) 2143–2155.
[18] W. Zhang, K. Ma, N. Yu, Reversibility improved data hiding in encrypted images, Signal Process. 94 (2014) 118–127.
[19] X. Wu, J. Weng, W. Yan, Adopting secret sharing for reversible data hiding in encrypted images, Signal Process. 143 (2018) 269–281.
[20] P. Maniriz, T. Ahmad, Information hiding scheme for digital images using difference expansion and modulus function, J. King Saud Univ, Comput. Inf. Sci. 31 (3) (2019) 335–347.
[21] M.M. Sadek, A.S. Khalifa, M.G. Mostafa, Video steganography: a comprehensive review, Multimed. Tools Appl. 74 (2015) 7063–7094.
[22] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible water-marking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.
[23] Y. Huo, S. Xiang, S. Liu, X. Luo, Z. Bai, Reversible audio watermarking algorithm using non-causal prediction, Wuhan Univ. J. Nat. Sci. 18 (5) (2013) 455–460.
[24] S. Xiang, Z. Li, Reversible audio data hiding algorithm using noncausal prediction of alterable orders, EURASIP J. Audio Speech Music Process. 2017 (1) (2017) 4.
[25] K. Jung, K. Yoo, Data hiding method using image interpolation, Comput. Stand. Interfaces 31 (2) (2009) 465–470.
[26] T. Ahmad, T.P. Figar, Enhancing the performance of audio data hiding method by smoothing interpolated samples, Int. J. Innov. Comput. Inf. Control 14 (3) (2018) 767–779.
[27] A. Gutub, N. Al-Juaid, E. Khan, Counting-based secret sharing technique for multimedia applications, Multimed. Tools Appl. 78 (5) (2019) 5591–5619.
[28] L. Xiong, D. Dong, Z. Xie, X. Chen, High-capacity reversible data hiding for encrypted multimedia data with some homomorph encryption, IEEE Access 6 (2018) 60635–60644.
[29] S. Xiang, X. Luo, Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group, IEEE Trans. Circuits Syst. Video Technol. 28 (11) (2018) 3099–3110.
[30] I.F. Jafar, K.A. Darabkh, R.T. Al-Zubi, R.A. Al Na’meh, Efficient reversible data hiding using multiple predictors, Comput. J. 59 (3) (2016) 423–438.
[31] R. Jiang, H. Zhou, W. Zhang, N. Yu, Reversible data hiding in encrypted three-dimensional mesh models, IEEE Trans. Multimed. 20 (1) (2018) 55–67.
[32] G. Hua, J. Huang, Y.Q. Shi, J. Goh, V.L.L. Thing, Twenty years of digital audio watermarking—a comprehensive review, Signal Process. 128 (2016) 222–242.
[33] L.K. Keshabatewse, A.M. Zunguru, M. Mangwala, J.M. Chuma, B. Sigwemi, Communication protocols for wireless sensor networks: a survey and comparison, Heliyon 5 (5) (2019).
[34] Newton’s divided difference interpolation formula. Available: http://mathworld.wolfram.com/. (Accessed August 2018).
[35] IRMAS: a dataset for instrument recognition in musical audio signals. Available: https://www.upf.edu/web/mtg/irmas/. (Accessed August 2017). [Online].

[36] T. Ahmad, J.N. Faruki, R.M. Ijthadie, W. Wibisono, Analyzing the effect of block size on the quality of the stego audio, in: Proc. the 5th International Conference on Science and Technology, Yogyakarta, Indonesia, 2019.