Journal of Information Hiding and Multimedia Signal Processing ©2020 ISSN 2073-4212
Ubiquitous International Volume **11**, Number **3**, September **2020**

Enhancement of LSB Audio Steganography Based on Carrier and Message Characteristics

Hussein A. Nassrullah, Wameedh N. Flayyih, Mohammed A. Nasrullah

Department of Computer Engineering
University of Baghdad
Baghdad, Iraq
h.nassrullah@coeng.uobaghdad.edu.iq; wam.nazar@coeng.uobaghdad.edu.iq;
mhmdnsrla@coeng.uobaghdad.edu.iq

Received June 2020; revised July 2020

Abstract. *Data steganography is a technique used to hide data, secret message, within another data, cover carrier. It is considered as a part of information security. Audio steganography is a type of data steganography, where the secret message is hidden in audio carrier. This paper proposes an efficient audio steganography method that uses LSB technique. The proposed method enhances steganography performance by exploiting all carrier samples and balancing between hiding capacity and distortion ratio. It suggests an adaptive number of hiding bits for each audio sample depending on the secret message size, the cover carrier size, and the signal to noise ratio (SNR). Comparison results show that the proposed method outperforms state of the art methods in terms of average segmental SNR, number of failing samples, and Czekanowski Distance (CZD). In addition, the proposed method shows the ability to operate with large message sizes (up to half of carrier size) with graceful degradation as opposed to the other methods which fail at large message size. So, the proposed method provides more flexibility in message and carrier sizes while preserving high efficiency.*
Keywords: Audio Steganography, Data Hiding, LSB.

1. **Introduction.** Steganography is the process of hiding secret data called message in another data called carrier [1]. In common steganography techniques, the carrier is normally chosen as a digital information of one of multimedia types such as text, audio, images, animations, or video. This is because the changes to the multimedia data by embedding a secret message may become undetectable by the human senses. Steganography techniques may be classified according to the type of carriers into audio, video, network, image, and text steganography. Also, it can be classified according to the embedding domains into time domain and frequency domain steganography.

In steganography, three main characteristics are desired to be enhanced, namely capacity, imperceptibility and robustness [2]. Capacity refers to the amount of data that can be embedded in the carrier. Imperceptibility indicates how transparent is the distortion caused by the embedded message. Robustness indicates the ability of hiding the message from the attackers. These three characteristics are inversely related; more embedded data results in higher distortion leading to higher perceptibility.

Watermarking is also a technique of embedding specific message in a carrier. The difference between steganography and watermarking is that the purpose of steganography is to hide the embedding message from eavesdroppers while the purpose of watermarking

is to prevent the eavesdropper from removing or replacing the watermark message in the carrier.

The popularity of audio files and the relatively unexplored audio steganalysis brought higher interest in audio steganography [3]. The amount of information that can be embedded while maintaining low level of distortion is highly restricted in audio steganography owing to the sensitivity of the human auditory system (HAS) [4]. Accordingly, many authors base their work on embedding the message bits in the least significant bit(s) (LSB) of the carrier samples [2, 4, 5].

This paper proposes a least significant bit (LSB) based audio steganography technique that uses time domain embedding algorithm. In LSB audio steganography, least significant bits of audio carrier are used to embed secret digital message. The main contribution of this paper is that it aims to maximize the carrier utilization by a two-step process. The first step estimates the capacity of the carrier and finds an embedding ratio. In the second step, each sample is embedded with a number of message bits in accordance with the embedding ratio and the sample value. The full carrier consideration allows the minimization of the distortion in each sample.

The remainder of this paper is organized as follows. Section 2 introduces the most relevant works. Section 3 presents the proposed audio steganography algorithm. section 4 gives the encoding algorithm. section 5 shows the decoding algorithm, Section 6 presents and discusses the results. The paper is concluded in Section 7.

2. **Related work.** Cvejic et al, [6] proposed a numerical method to enhance audio steganography by increasing hidden data channel capacity. It uses a three-step algorithm in order to embed an additional one bit of information to host audio. The main part of the method is applying Minimum Error Replacement method on standard LSB audio steganography.

Another audio steganography is presented by Gopalan, [2] to encrypt the secret message before the embedding operation. The presented algorithm encrypts the message by using simple XOR operation with a secret key of 256 bits and then embeds it in the LSB bit of the carrier sample.

Cvejic et al, [7] suggested an embedding method in audio steganography to increase the robustness. The suggested method uses the fourth LSB layer with 16-bit audio cover to maintain the secret message even when audio compression is applied. To enhance imperceptibility, the embedding method changes the value of the LSBs according to the value of embedding message bits and the value of fourth bit in the samples of host audio signal.

M. Lafta, [8] suggested third layer LSB steganography to enhance the robustness. The suggested method distributes the bits of secret message along the cover signal by hiding one bit of secret message in the third LSB bit of cover sample.

Some audio steganography techniques [9, 10] neglect the capacity and enhance the transparency and robustness by altering the number of samples in silent period of carrier according to the secret message instead of changing the amplitude of each sample.

Ahmed et al, [11] presented an encoding method to improve perceptual transparency in audio steganography with the same capacity. Due to the sensitivity of human auditory system to silent period, this algorithm improves the transparency by using noise gate software to avoid embedding in this period.

Asad et al, [4] presented an LSB audio steganography technique to increase security against intruder. AES encryption has been used to encrypt secret message to increase security. Special mapping technique that depends on the MSBs of the samples has been used to choose the embedding samples and to specify the location of embedded bit in

each selected sample. The main drawback of this method is that it neglects the effect of signal to noise ratio SNR which reduces the imperceptibility. Also, this method uses a well-known mapping algorithm that decreases 80 % of hiding capacity without noticeable increase of security parameter because the suggested mapping technique uses a predefined algorithm instead of secret key.

Divya et al, [12] aimed to enhance perceptual transparency by increasing signal to noise ratio (SNR). This method selects the number of embedding bits in each sample based on the value of the most significant bit in the sample.

Taruna et al, [5] proposed an adaptive LSB audio steganography method based on the value of the cover signal. It selects the number of secret message bits to be embedded in each sample depending on the value of the two most significant bits of that sample. This method increases the capacity of cover audio to hide more data without affecting the transparency of the audio signal.

Datta et al, [13] enhanced LSBs standard replacement method based on modulo operator by choosing the minimum between forward and backward difference. It enhances the performance of LSB audio steganography by increasing imperceptibility and decreasing distortion ratio. The signal to noise ratio (SNR) analysis showed that this method outperformed standard LSB technique.

Kar et al, [14] proposed a new encoding technique for audio steganography to enhance robustness against attacks. It preserves the statistical properties of carrier audio signal to avoid secret message detection by statistical attack.

Nasrullah, [15] enhanced imperceptibility and payload capacity by suggesting threshold SNR. It embeds a number of message bits in a carrier sample that ensures the SNR is greater than or equal to the threshold SNR.

As a continuation to previous works, this work aims to increase the embedding capacity without neglecting the perceptibility. Previous works applied their embedding decision on a per sample basis. On the contrary, this work takes the whole carrier and the message into consideration before applying the per sample algorithm. This supports the proposed algorithm with a global view of the message needs and the carrier capability allowing fair distribution of the message bits among the carrier samples.

3. **Proposed System.** The proposed audio steganography algorithm introduces a method to embed a variable size message in the carrier. To increase the hiding capacity and signal-to-noise ratio (SNR) as maximum as possible, the proposed method uses all the carrier samples to embed the hidden message. The number of message bits to be embedded in each sample depends on the message size, the value of the sample, and the values of all carrier samples. The message bits are allowed to be embedded in the eight lowest significant bits only, which keeps the highest significant eight bits intact. These most significant bits are used in the algorithm calculations at both the encoding and decoding parts. Each sample will take number of message bits depending on its most significant byte value as shown in Fig 1.

Fig 1 shows the scenario of hiding the message bits in the carrier samples. This method depends on the value of carrier samples. First step is to find the logarithm of each sample to base 2; the results represent the number of bits that can be embedded in each sample, the initial sample capacity. Then, the summation of these numbers represents the Initial Carrier Capacity (ICC). To distribute the message bits among the samples, the Embedding Ratio (ER) is calculated by dividing the Message Size (MS) in bits by ICC. The next step is multiplying each initial sample capacity by ER to find the actual number of bits to be embedded in each sample.

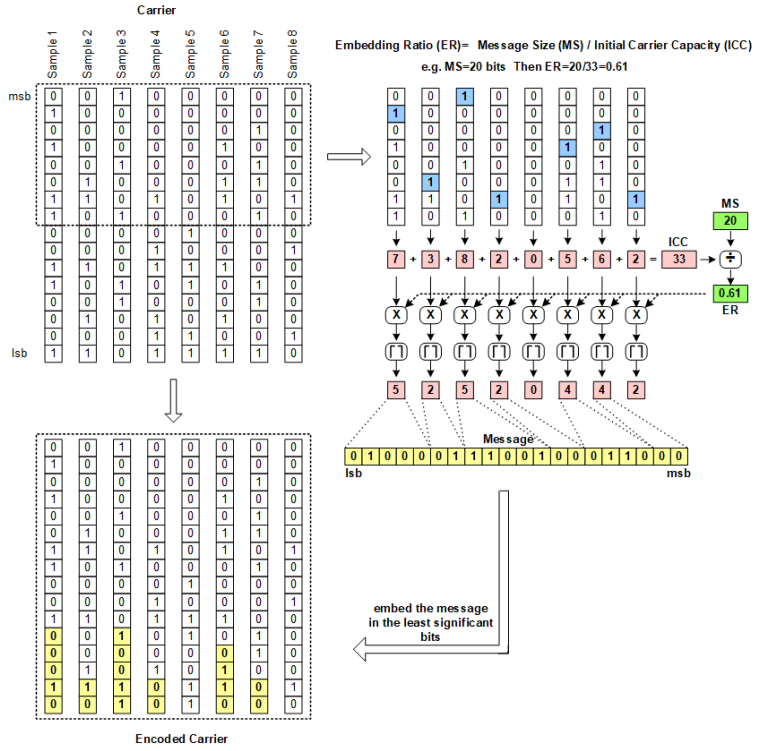

FIGURE 1. Number of message bits to be embedded in each carrier sample.

This scenario will operate with variable size of secret message and by distributing the message bits across the samples according to their values avoids unnecessary reduction in SNR.

It should be noted that there are some conditions for encoding and decoding the message bits. The first condition is that the number of embedded message bits (Bi) in each carrier sample should not exceed 8 bits. If the multiplication by the ER gives a number larger than 8, it will be considered 8. The second condition is that the number of bytes in the message should be no more than the number of carrier samples. When the message size, MS, is larger than ICC, the embedding ratio becomes greater than one. In this case, the number of embedding bits in some samples may be more than 8 as shown in Fig 2

To resolve this issue, the samples of the carrier are sorted in descending order according to the number of embedding bits . Then, the embedding ratio at each sample is computed from the remaining bits in message size and initial carrier capacity as follows:

Let carrier samples as Fig 1, then ICC = 33.

Let message size = 40

Then ER = 40 / 33 = 1.2

Now sort the samples descending as shown in Fig 3(a)

Then the number of embedding bits in each sample after sorting is found as follows:

1. 8×40/33=9.7 → 8 (number of embedding bits greater than 8),
   remaining MS = 40 8 = 32, remaining ICC = 33 8 = 25.
2. 7×32/25=8.7 → 8,
   remaining MS=32-8=24, remaining ICC=25-7=18.
3. 6×24/18=8,
   remaining MS=24-8=16, remaining ICC=18-6=12.
4. 5×16/12=6.7 → 7,
   remaining MS=16-7=9, remaining ICC=12-5=7.

5. 3×9/7=3.9 → 4,
   remaining MS=9-4=5, remaining ICC=7-3=4.
6. 2×5/4=2.5 → 3,
   remaining MS=5-3=2, remaining ICC=4-2=2.
7. 2×2/2=2,
   remaining MS=2-2=0, remaining ICC=2-2=0.

Then sort the samples according to their original order in time as in Fig 3

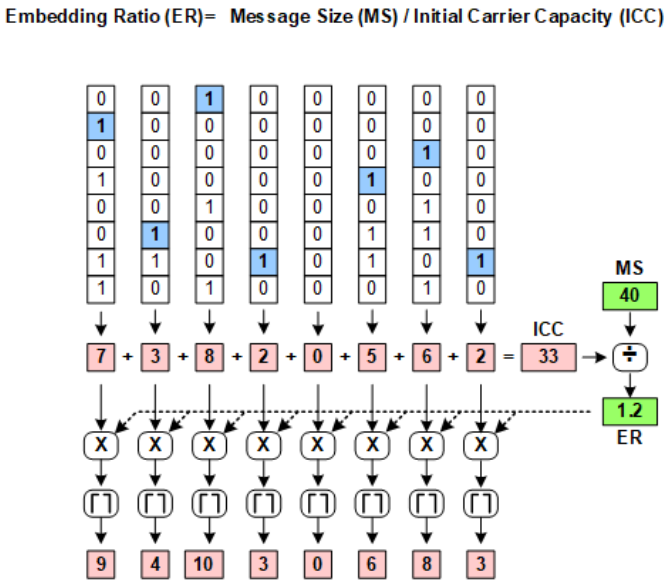

FIGURE 2. MS larger than ICC.

4. **Encoding Algorithm.** The encoding algorithm passes through four main phases:
   - Phase I Initialization and initial carrier capacity calculation.
   - Phase II Sorting, actual sample capacity calculation, and sorting back.
   - Phase III Message embedding.
   - Phase IV Distortion reduction.

4.1. **Phase I.** The first phase in the encoding algorithm starts by reading the message and the carrier. The size of message in bytes and the number of samples in carrier are compared. If the size of message is greater than the number of carrier samples, the algorithm exits as it violates the second condition. The initial number of message bits to be embedded in each sample, initial sample capacity, will be:

$$Sample\_bits(i) = \begin{cases} \lfloor log_2(Carrier(i)) \rfloor - 7 & ; Carrier(i) \geq 256 \\ 0 & ; Carrier(i) < 256 \end{cases} \quad (1)$$

Where Carrier(i) is the carrier sample. Then, the initial value of the carrier capacity (ICC) is given by:

$$ICC = \sum_{i=1}^{length(Carrier)} Sample\_bits(i) \quad (2)$$

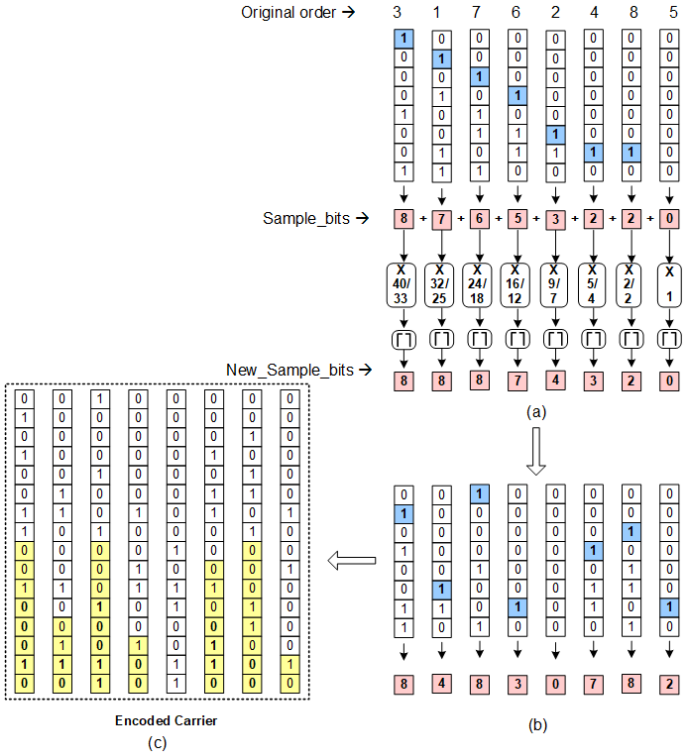

FIGURE 3. Number of embedding bits after correction (the number more than 8 is set to 8).

Algorithm 1 Phase I, initialization and carrier capacity calculation

```
read the message
read the carrier
if length(message) > length(carrier) then
    display "Out of range"
    break
for i=1:length(Carrier) do
    Sample_bits[i]= ⌊ log2(Carrier[i])⌋ -7
    if Sample_bits[i] < 0 then
        Sample_bits[i]=0
ICC=sum(Sample_bits)
```

4.2. **Phase II.** The second phase starts by sorting the carrier samples according to their value. To ensure identical sorting at both the encoding and decoding sides, the lower eight bits of the carrier samples are cleared, generating a temporary carrier (clear_carrier). The message is then converted to a stream of bits. Accordingly, the embedding ratio will be:

$$ER = \frac{MS}{ICC} \tag{3}$$

where ER is the embedding ratio and MS is the message size in bits. For each sample, the embedding ratio is calculated and then the new number of embedding bits in each carrier' sample (New_Sample_bits(i)) is found. Then, the New_Sample_bits are sorted back according to the original carrier. The calculation of sample amplitude and SNR is carried out on the samples by deleting the lower eight bits.

Algorithm 2 Phase II, sorting and actual sample capacity calculation

```
clear_carrier = BitAND(carrier, 'FF00 H')
Sort Sample_bits descending according to clear_carrier
CC = ICC
for i=1: length(Carrier) do
    if MS = 0 then
        Break
    ER=MS/CC
    New_Sample_bits(i) = ER * Sample_bits(i)
    if New_Sample_bits(i) > 8 then
        New_Sample_bits(i) = 8
    MS = MS - New_Sample_bits(i)
    CC = CC - Sample_bits(i)
Sort New_Sample_bits according to original order of carrier
```

4.3. **Phase III.** The third phase embeds the message bits in the carrier samples according to each sample capacity, represented by New_Sample_bits.

Algorithm 3 Phase III, message embedding

```
for i =1: length(Carrier) do
    if length(Message_Stream) <= 0 then
        Break
    MB=Message_Stream(1:New_Sample_bits(i))
    Shift right Message_Stream by New_Sample_bits(i)
    Mask = FFFF H  2^New_Sample_bits(i) + 1
    Carrier(i) = BitAND(Carrier(i), Mask)
    Carrier(i) = BitOR(Carrier(i), MB)
```

4.4. **Phase IV.** The last phase in the encoding process is intended to decrease distortion by adopting the method proposed in Datta et. al, [13].

Algorithm 4 Phase IV, distortion reduction

```
for i = 1 : length(Carrier) do
    E = abs( Carrier(i)  Old_Carrier(i) )
    if E > 2^(New_Sample_bits(i)-1) then
        if Carrier(i) > Old_Carrier(i) then
            Carrier(i)= Carrier(i)  2^(New_Sample_bits(i)-1)
            Carrier(i)= Carrier(i) + 2^(New_Sample_bits(i)-1)
```

5. **Decoding Algorithm.** Decoding algorithm passes through the following steps:
   - Read the carrier
   - Give the message length
   - Find the Sample_bits (section 4.1)
   - Find embedding ratio (section 4.2)
   - Find New_Sample_bits (section 4.2)
   - Find the message as algorithm 5

Algorithm 5 Find the message

1: **for** i = 1 : length(Carrier) **do**
2: MB = BitAND(Carrier(i), $2^{New\_Sample\_bits(i)}$ 1); the length of MB is New_sample_bits(i)
3: Message_Stream = Message_Stream + MB; (+) means concatenating MB with Message_Stream.
4: Make message file from Message_Stream

6. **Results and Discussion.** The proposed algorithm was compared with (Datta et. al, 2015) [13], (Taruna et. al, 2014) [5], (Divya et. al, 2012) [12] and (Nasrulla, 2018) [15]. Nasrullah's algorithm depends on a threshold value called threshold SNR, so it was tested in two cases (35 and 40 threshold SNR).

Three metrics are used to evaluate the tested algorithms: average segmental SNR, number of failing samples and CZD distance [16].

For each metric, an audio file with a sample rate of 44100 samples/second was considered with three carrier:

- Carrier_U: an unaltered carrier
- Carrier_L: the carrier with lower samples value (values divided by two)
- Carrier_H: the carrier with higher samples value (values multiplied by two)

Figures 4, 5 and 6, show the average segmental SNR as a function of the message size for the three carriers. The proposed algorithm achieves different SNR values for different message sizes. For Carrier_U, the average SNR falls from 84 dB at the smallest message size considered down to 36 dB at 950,000 bytes message size. The proposed algorithm clearly outperforms the other algorithms represented by its higher SNR for message sizes up to 600,000 bits. For message sizes 650,000 up to 750,000 (Datta et al, 2015) [13] achieves higher SNR than our algorithm, but the latter message size represents its maximum capacity. On the other hand, the proposed algorithm provides higher capacity than all other algorithms while maintaining SNR above 36 dB, thus achieving acceptable perceptual transparency with high capacity. Similar behavior can be noticed at the Carrier_H case with slight difference in values, where the proposed algorithm minimum SNR is 41 dB but at a high capacity of 1,000,000 bytes. Although Datta et al outperforms the proposed algorithm at message sizes 650,000 to 750,000 at Carrier_U and Carri_H cases, its performance falls behind the proposed algorithm and some other algorithms at the Carrier_L case. From the three figures, the proposed algorithm achieves the highest capacity among all the considered algorithms, outperforming Nasrullah_35 by 25% at Carrier_H. Thus it provides a wide range of capacity with graceful degradation of SNR.

From another perspective, Figures 7, 8, 9, show the number of samples that have an SNR below 40 dB for each message size. As in the average SNR results, the proposed algorithm achieves a good balance between the number of failing samples and the provided capacity. As opposed to the standard, Datta et. al., Divya et. al., and Taruna et. al., the proposed algorithm has zero failing samples at small message sizes and this number gracefully increases for large message sizes. The key factor contributing to this behavior is the fact that the proposed algorithm maximizes the utilization of the carrier by distributing the embedded message bits among all the carrier samples with consideration of each sample SNR. In contrast, the other algorithms do not take into account the carrier length and its embedding capacity.

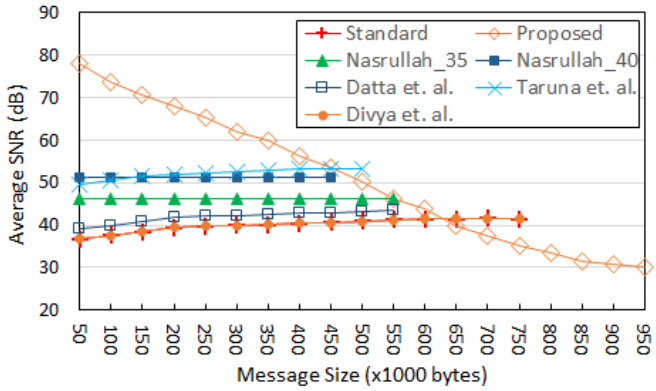

FIGURE 4. Average segmented SNR as a function of the message size in Carrier_L.

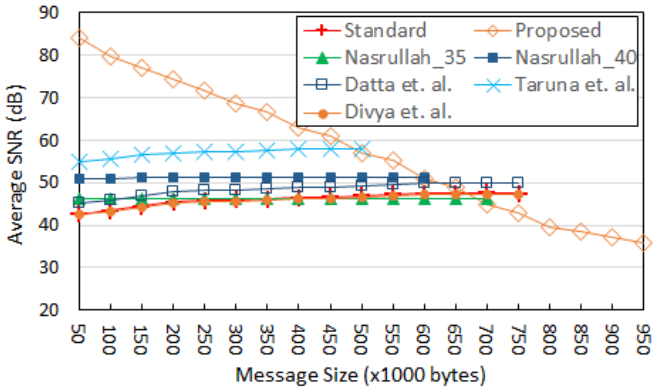

FIGURE 5. Average segmented SNR as a function of the message size in Carrier_U.

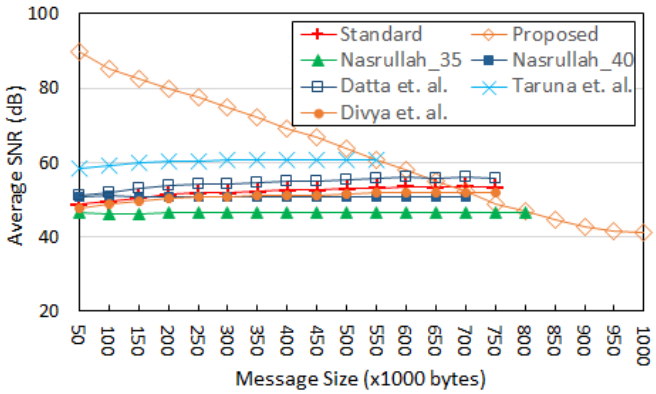

FIGURE 6. Average segmented SNR as a function of the message size in Carrier_H.

In addition to the previous two metrics which measure the distortion amount, we consider the Czekanowski Distance (CZD); a correlation based measure to evaluate the closeness degree between the carrier and the stego carrier. Figures 10, 11, 12, reveal a clear correlation obtained by the proposed algorithm represented by the low CZD results for all message sizes at the three carriers. Although Datta et. al. and Nasrullah_40 achieve close CZD results to the proposed algorithm, they lack behind the proposed algorithm

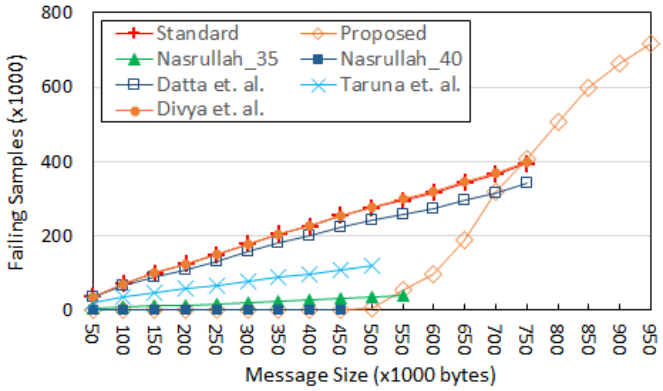

FIGURE 7. Number of samples that have an SNR below 40 dB for each message size in Carrier_L.

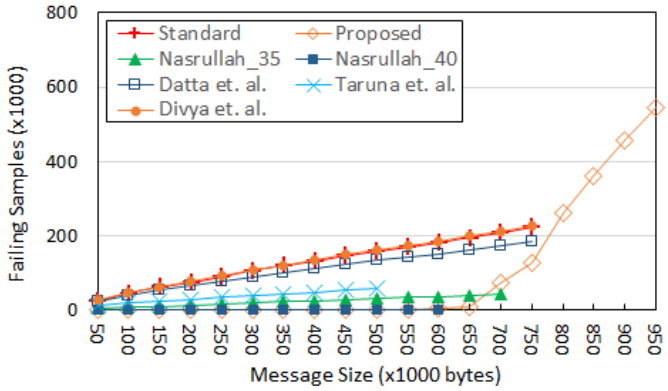

FIGURE 8. Number of samples that have an SNR below 40 dB for each message size in Carrier_U.

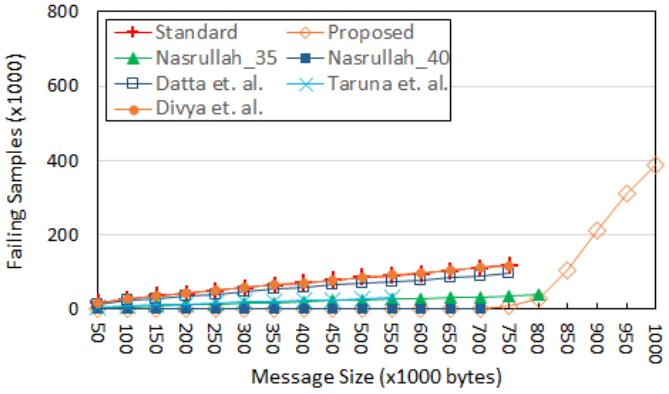

FIGURE 9. Number of samples that have an SNR below 40 dB for each message size in Carrier_H.

from capacity perspective. In addition, it can be seen that as the carrier amplitude increases the CZD increases as well due to the reduced effect of the embedded message with respect to the carrier. Taken together, the results of the three metrics indicate a prominent advantage of the proposed algorithm over the other algorithms from capacity

and perceptual transparency. The high variety of the number of embedding bits in each sample in proposed method against other algorithms gives more robustness.

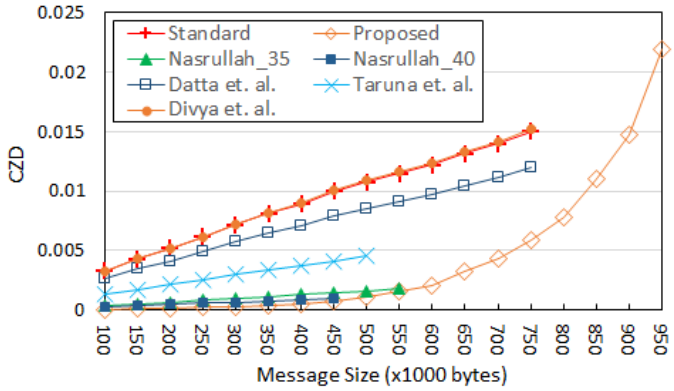

FIGURE 10. Czekanowski Distance (CZD) as a function of the message size in Carrier_L.

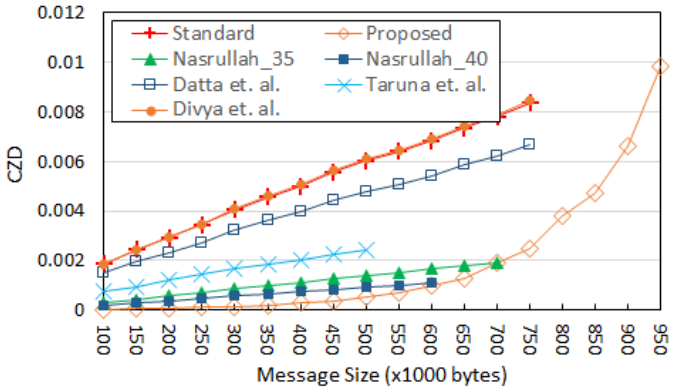

FIGURE 11. Czekanowski Distance (CZD) as a function of the message size in Carrier_U.

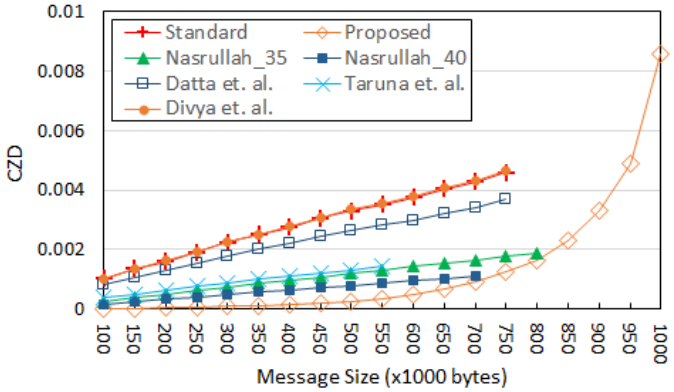

FIGURE 12. Czekanowski Distance (CZD) as a function of the message size in Carrier_H.

7. **Conclusion.** The results from previous section show that the proposed method outperforms other methods in main evaluation metrics. It clearly can handle large message size as opposed to the other algorithms that provide limited capacity. Also the comparison results show that the proposed method is more efficient in small message size, because it distributes message bits among all the carrier samples. Therefore this method gives more flexibility in terms of message and carrier sizes with acceptable performance.

REFERENCES

[1] J. S. Pan, H. C. Huang, L. C. Jain, and W. C. Fang (eds.), *Intelligent Multimedia Data Hiding: New Directions*, Springer, Berlin-Heidelberg, Germany, 2007.

[2] K. Gopalan, Audio steganography using bit modification, in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 2, pp. II–421, 2003.

[3] A. Ali, M. Mokhtar, and L. George, Enhancing the hiding capacity of audio steganography based on block mapping, vol. 95, pp. 1441–1448, April 2017.

[4] M. Asad, J. Gilani, and A. Khalid, An enhanced least significant bit modification technique for audio steganography, in *International Conference on Computer Networks and Information Technology*, pp. 143–147, 2011.

[5] T. Taruna and R. Jain, Message Guided Adaptive Random Audio Steganography using LSB Modification, *International Journal of Computer Applications*, vol. 86, pp. 6–9, Jan. 2014.

[6] N. Cvejic and T. Seppanen, Increasing the capacity of lsb-based audio steganography, in *2002 IEEE Workshop on Multimedia Signal Processing.*, pp. 336–338, 2002.

[7] N. Cvejic and T. Seppanen, Increasing robustness of lsb audio steganography using a novel embedding method, in *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, vol. 2, pp. 533–537, 2004.

[8] M. M. Lafta, Data hiding in audio wave file, *J. Of College Of Education for Women*, vol. 19, pp. 237–243, January 2008.

[9] S. Shirali-Shahreza and M. Shirali-Shahreza, Steganography in silence intervals of speech, in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 605–607, 2008.

[10] M. Shirali-Shahreza, Real-time and mpeg-1 layer iii compression resistant steganography in speech, *IET Information Security*, vol. 4, issue. 1, pp. 1–7, March 2010.

[11] M. A. Ahmed, M. L. Mat Kiah, B. Bahaa, and A. Zaidan, A novel embedding method to increase capacity and robustness of low-bit encoding audio steganography technique using noise gate software logic algorithm, *Journal of Applied Sciences*, vol. 10, pp. 59–64, January 2010.

[12] S. S. Divya, M. Ram, and M. Reddy, Hiding text in audio using multiple lsb steganography and provide security using cryptography, *International Journal of Scientific & Technology Research*, vol. 1, no. 6, pp. 68–70, 2012.

[13] B. Datta, S. Tat, and S. K. Bandyopadhyay, Robust high capacity audio steganography using modulo operator, in *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, pp. 1–5, 2015.

[14] D. C. Kar, A. M. Nakka, and A. K. Katangur, A new statistical attack resilient steganography scheme for hiding messages in audio files, *International Journal of Information and Computer Security*, vol. 10, no. 2-3, pp. 276–302, 2018.

[15] M. A. Nasrullah, Lsb based audio steganography preserving minimum sample snr, *International Journal of Electronic Security and Digital Forensics*, vol. 10, no. 3, pp. 311–321, 2018.

[16] Detection of audio covert channels using statistical footprints of hidden messages, *Digital Signal Processing*, vol. 16, no. 4, pp. 389–401, 2006.