Multimed Tools Appl
DOI 10.1007/s11042-016-3851-y

Data embedding in audio signal using multiple bit marking layers method

Van Tam Vu [1] · Duc-Tan Tran [2] · Trong Hanh Phan [3]

Received: 24 July 2015 / Revised: 23 June 2016 / Accepted: 9 August 2016

Abstract One of the biggest challenges in data embedding is that the confidential data need to be in the 'transparency' after being embedded into the audio signal. Therefore, embedding methods must reduce the influence of embedded data onto the original audio signal. In this paper, the multiple bit marking layers (MBML) method has been proposed to fulfill this requirement. This method reuses the results from the previous embedding time (layer) as the input data to continue embedding it into audio signals (i.e. the next layer). The quality of the proposed method is evaluated through embedding error (*EE*), signal-to-noise ratio (*SNR*), embedded capacity (*EC*) and contribution error (*CE*). Experimental results have shown that the proposed method provides better quality of *EE*, and *SNR* than any other proposed embedding methods such as: LSB (Least Significant Bit), ELS (Embedding Large Sample.), BM (Bit Marking), and the BM/SW (Sliding Window) method with a single layer.

Keywords Multiple bit marking layers · Embedding signal · Least significant bit · Bit marking layers · Sliding window · Embedding error

✉ Duc-Tan Tran
tantd@vnu.edu.vn

Van Tam Vu
tamt36bca@gmail.com

Trong Hanh Phan
tronghanhmai@yahoo.com

[1] University of Technology and Logistics, Hanoi, Vietnam

[2] VNU University of Engineering and Technology, Hanoi, Vietnam

[3] Le Quy Don Technical University, Hanoi, Vietnam

Published online: 31 August 2016

1 Introduction

Confidential data embedding to multimedia signals is one of the hot topics of many researchers' interest. It can be used for the purpose of copyright protection or hiding confidential data [3, 5, 6, 9]. Currently, there are two main techniques being used: (i) *Watermarking* [9]. This technique is usually used for the purpose of copyright protection in the field of publishing. Thus, embedded data are usually small. The sustainability requirements of the embedded data is the highest priority, and it is possible for the multimedia signal can be altered substantially.

(ii) *Steganography* [1, 3, 4, 13–15, 21]. This technique is usually used in the field of security and defense. Storage of the embedded data may be large but high demands on safety and security for the embedded data. Therefore, the method of embedding data must meet a requirement that is nearly unchanged multimedia signals. It means that *EE* must be small and *SNR* must be large.

Multimedia signal can be text, audio, picture, video [9]. In the case of the signal is audio, it is easy to expose the confidential data embedded [6, 16] because the human auditory system is very sensitive to the amplitude, and frequency of the sound. Therefore, confidential data embedding in the audio signals is often more difficult than the others.

2 Related works

The most popular method for data embedding in audio signal is the direct overwriting bit. It will replace bits of confidential data to unimportant bits or unimportant parameters of audio signals. Thus, *BE* is still high; it makes the performance parameters such as *EE*, and *SNR* of embedding signal are not high. Some improvements such as LSB [10] and improved LSB [3, 11, 12, 17] could offer better of *EC*, but *EE, SNR* and *CE* are still small. Other methods such as PC (Parity Coding) [2, 7, 11] and Ph_C (Phase Coding) [7, 12] provide low *EC*. Echo Coding [7, 8] increase the size of audio signals, thus it is easy to expose. The method ELS (Embedding Large Sample) [18] only overwrite the confidential data to high sample of the audio signal. It makes *EE, SNR, CE* are improved. However, if the quantity of confidential data increases, *EE, SNR, CE* degrades faster. Therefore, it is necessary to find out a method which do not use (or spare use) this direct overwriting bit method.

In this field, we have proposed BM embedding method [16] where the embedding process is performed as follows: in the original audio signal, find q segments whose size is Ω [bit] which is coinciding with q segments of the confidential data need to embed. We use q segment whose size is $\mho$ [bit] ($\mho < \Omega$) to mark the position of the bit segments found. After that, ($q\mho$) marking bits is overwritten on the suitable position of the original audio signal. Thus, the number of bits overwritten on the original audio signal decreases β_1[bit].

$$\beta_1[\text{bit}] = \text{q}\Omega - \text{q}\mho = B_{\text{data}} - S_1, \tag{1}$$

Where $B_{\text{data}} = \text{q}\Omega$ is the confidential data which need to embed, $S_1 = q\mho$ is the marked bit string which is overwritten on the original audio signal.

In [19], we have proposed to use Sliding Window (SW) technique to improve the work in [20]. A window whose size is Ω [bit] is sliced in original audio signal. Data in the window is compared to the segment of confidential data embedded and is embedded by BM method. This technique has increased the embedding capacity. However, works in [19, 20] are done with only one embedded layer (i.e. when all the B_{data} is embedded, the string S_1 is overwritten on the original audio signal and the embedding process is completed). Thus, β_1[bit] is still small that lead to *EE*, and *SNR* are not high. From the above discussion, we can see that it is very necessary to find out a method to reduce the number of overwritten bits and hence we can improve the performance of data embedding in audio signals.

In this paper, we propose MBML method to effectively embed the confidential data in audio signals. With this method, we can exploit the advantage of β_1 [bit] > 0, thus when all B_{data} is embedded, we do not overwrite S_1 string on original audio signal to complete the procedure. S_1 is embedded into original audio signal to create S_2, S_2 is embedded to create S_3, etc. When $S_\omega < \Omega$ [bit], the embedding is finished and S_ω is used as the confidential key which help the receiver can de-embed ($Key = S_\omega$). It will reduce a large number of bits need to overwrite on audio signals, which in turn decrease *EE*, increase *SNR*, and especially enhances the safety of confidential data.

The rest of the paper is organized as follows: the system model and the working principles are presented in Section 3. The performance parameters are shown in Section 4. Experimental results and discussion are clarified in Section 5. Conclusions and further work are presented in Section 6.

3 System model

3.1 Model structure

Figure 1 is embedding and de-embedding model by the proposed method, the parameters are described as follows:

- C is the original audio signal (2) which consists of n audio samples C_j [19, 20], C may be taken from the storage device or from the digital transmission system (i.e. real-time embedding). C is divided into p segments C_∂ [19, 20], and each C_∂ has ξ audio samples (3, 4, 5).

$$\mathrm{C} = \{\mathrm{C}_j;\ j = 1, 2, \ldots, n\}, \tag{2}$$

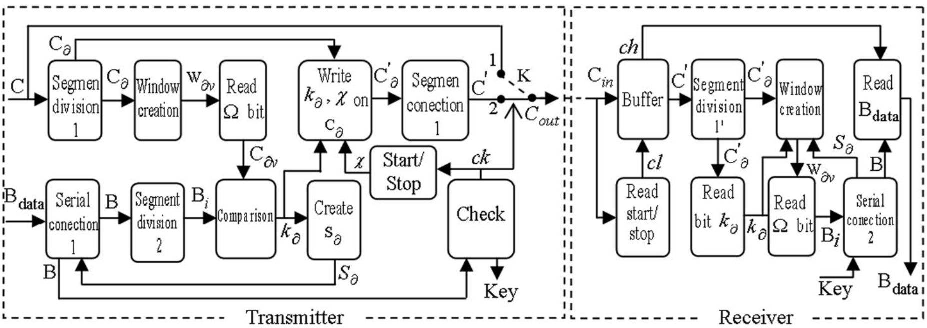


Fig. 1 Embedding and de-embedding model in the proposed method

$$C = \{C_\partial;\ \partial = 1, 2, \dots, p\},\tag{3}$$

$$p = \frac{n}{\zeta},\tag{4}$$

$$C_\partial = \{C_j;\ j = 1, 2, \dots, \zeta\}.\tag{5}$$

- + $W_{\partial v}$ is the sliding window in C_∂ by using the sliding window [19], v is the ordinal numbers of the window, $C_{\partial v}$ is the data of $W_{\partial v}$. B_{data} is the confidential data which need to embed into C, B_i is the segment of data embedded (8). S_∂ is the bit sequence that mark the location of $W_{\partial v}$, S_∂ has $\mho$[bit] and its decimal value is v (6), the serial merger will join the bit string of S_∂ with the bit string of B_{data} (7).

$$S_\partial[\text{bit}] = \mho,\tag{6}$$

$$B = S_\partial + (\text{serial})\ B_{\text{data}},\tag{7}$$

$$B = \{B_i;\ i = 1, 2, \dots, q\} \quad \text{with} \quad q = \frac{B}{\Omega}.\tag{8}$$

- k_∂ is bit resulted from the comparison between $C_{\partial v}$ and B_i [19], which is corresponding to two following cases:

$$k_\partial = \begin{cases} 1 \Leftrightarrow \exists\ C_{\partial v}\left(\text{with}\ \ v=0\ \ \text{or}\ \ 1\ \text{or} \dots \text{or}\ 2^{\mho}-1\right) \equiv B_i \\ 0 \Leftrightarrow \forall\ \ C_{\partial v}\left(\text{with}\ \ v=0 \dots 2^{\mho}-1\right) \neq B_i \end{cases}.\tag{9}$$

- ck is bit that control the electronic key K, corresponding to the two following cases:

$$ck = \begin{cases} 1 \Leftrightarrow B[\text{bit}] \geq \Omega \\ 0 \Leftrightarrow B[\text{bit}] < \Omega \end{cases}.\tag{10}$$

- K is the electronic key which has 2 states as follows:

$$K = \begin{cases} 1\ \ (\text{No embedding}) \Leftrightarrow ck=0 \\ 2\ \ (\text{Embedding}) \qquad \Leftrightarrow ck=1 \end{cases}.\tag{11}$$

- χ is the bit string used to mark the start ("*Start*") or stop ("*Stop*") embedding positions, corresponding to the two following cases:

$$x = \begin{cases} \text{"}Start\text{"} \Leftrightarrow ck \quad \text{from 0 to 1} \\ \text{"}Stop\text{"} \Leftrightarrow ck \quad \text{from 1 to 0} \end{cases}.\tag{12}$$

- k_∂, χ will be overwritten at the least significant bit of C_j in the segment C_∂ (see Fig. 2) [20]. In (5), each C_j would be overwritten by ≤ 1 [bit], thus *Start*[bit] $\leq \zeta$ and *Stop*[bit] $\leq \zeta$.
- C'_∂ is the segment of audio signal embedded (13), and C' is audio signal embedded (14):

$$C'_\partial = \{C'_j;\ \ j = 1, 2, \dots, \zeta\},\tag{13}$$

$$C' = \{C'_\partial;\ \partial = 1, 2, \dots, p\}.\tag{14}$$

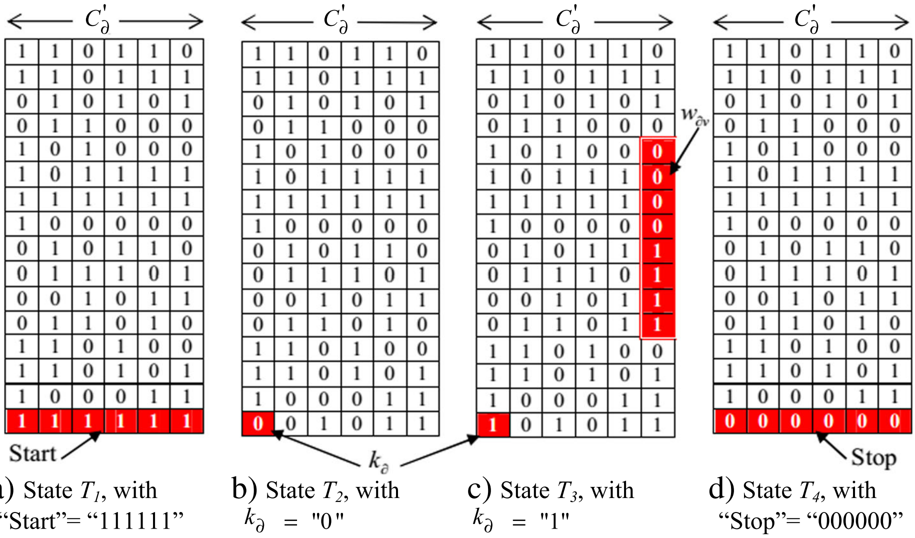


Fig. 2 The illustration of the overwriting step of the bit strings k_∂, *Start*, *Stop* with BitsperSample = 16[bit], $\Omega = 8$[bit], $\mho = 5$[bit], $\zeta = 6$[sample], $B_i = "00001111"$

- C_{out} is the output audio signal of the transmitter, C_{out} is brought to the storage devices or to the digital transmission system, and sent to the receiver $C_{\text{out}} = \begin{cases} C \Leftrightarrow K=1 \\ C' \Leftrightarrow K=2 \end{cases}$.
- *ch* is a reading control bus of B_{data} when the buffer has no embedding data, it works as follows:

$$ch = \begin{cases} 1 \Leftrightarrow C'=0 \\ 0 \Leftrightarrow C'>0 \end{cases} \tag{15}$$

$$\text{Read} \;\; B_{\text{data}} = \begin{cases} \text{Enable} \Leftrightarrow ch = \text{from} \;\; 0 \;\; \text{to} \;\; 1 \\ \text{Disable} \Leftrightarrow ch = \text{from} \;\; 1 \;\; \text{to} \;\; 0 \end{cases}. \tag{16}$$

- *cl* is bit that control the writing or reading function of the buffer, corresponding to the two following cases:

$$cl = \begin{cases} 1 \Leftrightarrow \text{Read} \;\; Start/Stop = "Start" \\ 0 \Leftrightarrow \text{Read} \;\; Start/Stop = "Stop" \end{cases}, \tag{17}$$

$$\text{The buffer} = \begin{cases} \text{Write} \Leftrightarrow cl = \text{from} \;\; 0 \;\; \text{to} \;\; 1 \\ \text{Read} \Leftrightarrow cl = \text{from} \;\; 1 \;\; \text{to} \;\; 0 \end{cases}. \tag{18}$$

3.2 Embedding and de-embedding process

- Embedding process

– **Initial state (T_0):** B_{data} hasn't been put into the system, (B_{data}[bit] = 0) and S_∂[bit] = 0. According to (7), B [bit] = 0 and according to (10), $ck = 0$. As a result, according to (11) K key is at position 1, $C_{\text{out}} = C$.

– **Start state (T_1):** $B_{\mathrm{data}}[\mathrm{bit}] \geq \Omega$ is put into the system. According to (7):

$$B = S_{\partial} + (\mathrm{serial})B_{\mathrm{data}} = 0 + (\mathrm{serial})B_{\mathrm{data}} = B_{\mathrm{data}}. \tag{19}$$

According to (10), ck is from level 0 to level 1, as a result K key moves to position 2. Simultaneously, due to ck from level 0 to level 1, according to (12) χ = "*Start*". Bit string χ = "*Start*" is overwritten on the position of the least significant bits of the first audio segment (C_{∂}, with $\partial = 1$) to obtain C'_{∂}, with $\partial = 1$ (Fig. 2a). Consequently, $C_{\mathrm{out}} = C'$, the system switches to embedding state with $\partial = 2$ and $i = 1$.

– **Embedding State (T_2):** B_i is alternately compared to $C_{\partial v}$(with $v = 0 .. 2^{\mho} - 1$), corresponding to $2^{\mho}$ windows created in C_{∂}. According to (9), there are two cases:

(i) If $k_{\partial} = 0$ (B_i is not embedded successfully), bit "0" will be overwritten on the least significant bit of the first audio sample in segment C_{∂} [19] (Fig. 2b) and $\partial = \partial + 1$.

(ii) If $k_{\partial} = 1$ (B_i is embedded successfully), bit "1" will be overwritten on segment C_{∂} (Fig. 2c), $i = i + 1$ and $\partial = \partial + 1$. Simultaneously, the marking bit string $S_{\partial} = \mho[\mathrm{bit}]$ (S_{∂} has the equal decimal to v) is brought to serial merger block 1. We combine (6, 7, 8) and (19) to have:

$$B[\mathrm{bit}] = B_{\mathrm{data}} - B_i + S_{\partial} = B_{\mathrm{data}} - \Omega + \mho = [B_{\mathrm{data}} - (\Omega - \mho)]. \tag{20}$$

Comparing (19) with (20), it can be seen that whenever B_i is embedded successfully, the capacity of B decreases ($\Omega - \mho$) [bit], comparing to its previous capacity. According to (1), we have:

$$\beta_{\mathrm{Embedding}}[\mathrm{bit}] = -(\Omega - \mho), \tag{21}$$

where $\beta_{\mathrm{Embedding}}$ is called bit reduction coefficient when embedded. When q times embedding B_i successfully is performed (with $q = \frac{B_{data}}{\Omega}$),

$$B[\mathrm{bit}] = [B_{\mathrm{data}} - q(\Omega - \mho)], \tag{22}$$

From (8) and (19), the Eq. (22) is rewritten as follows:

$$B[\mathrm{bit}] = \left[B_{\mathrm{data}} - \frac{B_{\mathrm{data}}}{\Omega}(\Omega - \mho)\right] = \frac{B_{\mathrm{data}}\Omega - B_{\mathrm{data}}\Omega + B_{\mathrm{data}}\mho}{\Omega} = \frac{B_{\mathrm{data}}\mho}{\Omega} = q\mho. \tag{23}$$

From (23), it can be seen that after q times embedding B_i successfully, the confidential data (B_{data}) is all embedded ($B_{\mathrm{data}} = 0$). The remain data (at the output of the serial merger block 1) is the address of B_{data} in C'. We choose $\mho < \Omega$ then combine with (1) (6) (8) (19), we have

$$\mho < \Omega \rightarrow q\mho < q\Omega \rightarrow q\mho < B_{\mathrm{data}} \rightarrow qS_{\partial} < B_{\mathrm{data}} \rightarrow S_{\mathrm{Layer}\ 1} < B_{\mathrm{data}}, \tag{24}$$

where $S_{\mathrm{Layer}\ 1} = \{S_{\partial}, \partial = 1, 2, \ldots, q\}$ is the address bit string of B_{data} in C'. It is also the first layer embedding address.

From (24), it can be seen that the embedding address is always smaller than the corresponding embedded data. Therefore, if $S_{\mathrm{Layer}\ 1}[\mathrm{bit}] \geq \Omega$ and $S_{\mathrm{Layer}\ 1}$ is considered as B_{data} to continue embedding into C (embedding to layer 2), $S_{\text{Layer 2}} < S_{\text{Layer 1}}$, similar to $S_{\text{Layer } \omega - 1} < S_{\text{Layer } \omega}$ (with ω is the number of embedded layers). As a result, the number of bit of embedding address $S_{\text{Layer L}}$ (with $L = 1, 2, \ldots, \omega$) will reduce gradually to the value which is smaller than Ω.

– **Stop state (T_3):** When $S_{\text{Layer } \omega}[\text{bit}] < \Omega$, according to (7) and due to $B_{\text{data}}[\text{bit}] = 0$, $B = S_{\text{Layer } \omega} < \Omega$. As a result, according to (10, 12), $\chi = "Stop"$, *Stop* bit string is overwritten on the position of the least significant bit of C_∂ (Fig. 2d). The remaining bit string of B is used as a confidential key ($Key = S_{\text{Layer } \omega}$).

  Ending the Stop state, due to $ck = 0$, according to (11) K key moves to position 2 (the initial state). $C_{\text{out}} = C$, the transmitter is ready for the next data embedded session. So, the process of embedding B_{data} into C can be described by the diagram of algorithm as follows:

• De-embedding process

– **The initial state (T'_0):** After beginning, $cl = 0$, $ch = 0$, $B_i = 0$. The buffer and the reading block B_{data} are at the state of being triggered (activated). So, $C' = 0$, $B_{\text{data}} = 0$. The confidential key $Key = S_{\text{Layout } \omega}$ is put into the serial merger 2, we have

$$B = B_i + Key = Key. \tag{25}$$

– **The state of writing to the buffer (T'_1):** *Start/Stop* reading block will load audio samples at the input (C_{in}) to find the *Start/Stop* string. When the marked bit string "*Start*" is found, according to (17), $cl = 1$ and according to (18), the buffer works in recording mode to record C'. When the *Start/Stop* reading block finds out the marked bit string "*Stop*", $cl = 0$. As a result, the buffer changes to reading mode (i.e. the order is first in, later out).

– **State of de-embedding (T'_2):** C' is divided into some of C'_∂ corresponding with the transmitter (13, 14). k_∂ bit of each segment C'_∂ will be read out to check, corresponding to the two following cases:

  (i) If $k_\partial = 0$ (segment C'_∂ has no embedding data), the receiver will move to the other k_∂ of the next segment C'_∂, where ($\partial = \partial + 1$).

  (ii) If $k_\partial = 1$ (segment C'_∂ has embedding data), the window $W_{\partial v}$ is reconstructed in the position S_∂ (S_∂ is taken from the serial merger 2). Embedded data (B_i) in $W_{\partial v}$ is taken to the serial merger 2, then the receiver would de-embedd on the next segment C'_∂, where ($\partial = \partial + 1$).

– Thus, when B_i is de-embedded, data in the serial merger 2 will be:

$$B = Key - S_\partial + B_i = Key - \Omega + \mho = Key + (\Omega - \mho). \tag{26}$$

– From (25, 26), it can be seen that each time B_i is de-embedded successfully, the capacity of B increases ($\Omega - \mho$) [bit], so

$$\beta_{\text{De-Embedding}}[\text{bit}] = (\Omega - \mho), \tag{27}$$

– Where $\beta_{\text{De - Embedding}}$ is a increasing bit coefficient in the de-embedding process. From (21), we have

$$\beta_{\text{De-Embedding}} = -\beta_{\text{Embedding}}, \tag{28}$$

– From (27, 28) and the de-embedding process, it can be seen that if the state (T'_2) is repeated until $C' = 0$, the data in the serial merger 2 will be B_{data}.
– **The Stop state (T'_3):** When $C' = 0$, according to (15) $ch = 1$. Consequently, according to (16), the reading block B_{data} is triggered to read the confidential data from the serial merger 2, ($B_{\text{data}} = B$); Concurrently, the receiver moves to T'_0 state in order to be ready for the next de-embedding session.

4 Calculation of parameters

4.1 The number of times B_i is embedded successfully (λ [times])

To embedd B_{data} and bit sequence $S_{\text{Layer L}}$ (with $L = 1, 2, \ldots, \omega - 1$) into C, λ must be the sum of all (B_i) that need to embedd into C. From Fig. 3 and (8) we have

$$\begin{aligned}\lambda[\text{times}] &= q + \sum_{L=1}^{\omega-1} q_L = \frac{B_{\text{data}}}{\Omega} + \sum_{L=1}^{\omega-1} \frac{S_{\text{Layer } L}}{\Omega} \\ &= \frac{B_{\text{data}}}{\Omega} + \frac{S_{\text{Layer 1}}}{\Omega} + \frac{S_{\text{Layer 2}}}{\Omega} + \cdots + \frac{S_{\text{Layer } \omega\text{-}1}}{\Omega}\end{aligned}. \tag{29}$$

where $q_L = \frac{S_{\text{Layer } L}}{\Omega}$ is the number of bit segments at the L^{th} layer.

According to (8) and (10), only when $B_{\text{data}}[\text{bit}] \geq \Omega$, embedding is performed and according to (21), for each B_i is embedded successfully, B decreases ($\Omega - \mho$) [bit]; thus, λ is calculated as follows:

$$\lambda[\text{times}] = \text{Floor}\left(\frac{B_{\text{data}} - \Omega}{\Omega - \mho} + 1\right), \tag{30}$$

where "Floor" is the rounded-down function.

4.2 The length of the code key (*Key* [bit])

From (20, 30), it can be seen that after λ times B_i is embedded successfully, B decreases $\lambda(\Omega - \mho)$[bit]. This makes B [bit] < Ω and $Key[\text{bit}] = S_{\text{Layer } \omega} = B$; the remain segment after the embedding procedure is computed as

$$Key[\text{bit}] = B_{\text{data}} - \lambda(\Omega - \mho). \tag{31}$$

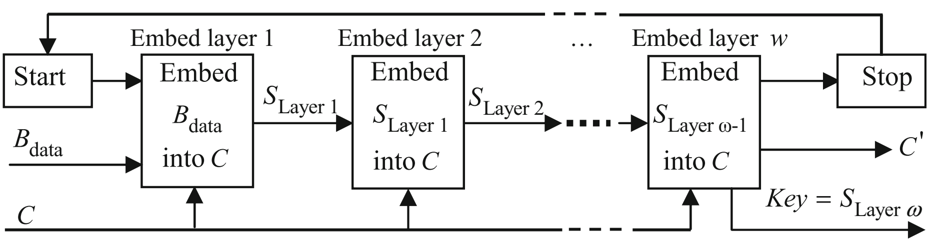


Fig. 3 Algorithm to embed data into audio signals by the proposed method

4.3 The total bit overwritten on audio file (φ [bit])

In both cases of B_i is embedded successfully or not, the segment C_∂ is also overwritten 1 bit (Fig. 2). Thus, φ [bit] is calculated as follows:

$$\varphi[\text{bit}] = Start + \lambda + \gamma + Stop, \tag{32}$$

where γ is the times that B_i is not embedded successfully (Fig. 2b).

From (32), it can be seen that bit string " *Start* " = const, " *Stop* " = const; with the certain values of B_{data} and Ω, advance according to (30), λ = const. In order to decrease φ [bit] which leads to the decreasing of *EE* and the increasing of embedding capacity, γ must be decrease (in this proposed model, we used the sliding window technique [19] to decrease γ).

4.4 The maximum data embedding capacity ($B_{\text{data max}}$[bit])

Data embedding capacity would be maximum when all p segments C_∂ of C are successfully embedded. Consequently, the successful embedding times (λ) equals to number of segments (p) of $C(\lambda = p)$. Thus, $B_{\text{data}} = B_{\text{datamax}} \Leftrightarrow \lambda = p$, and combine with (3, 4, 30), we have

$$B_{\text{data}} = B_{\text{datamax}} \Leftrightarrow \lambda = p \Rightarrow p = \frac{B_{\text{data max}} - \Omega}{\Omega - \mho} + 1, \tag{33}$$

We have

$$B_{\text{data max}}[\text{bit}] = p(\Omega - \mho) + \mho. \tag{34}$$

4.5 Embedding error (*EE*)

In the general cases [5, 13, 18, 20] *EE* is calculated as follows:

$$EE = |C' - C| = \sum_{j=1}^{n} |C'_j - C_j| = \sum_{j=1}^{n} EE_j. \tag{35}$$

However, with the proposed method, due to bit strings " *Start* ", k_∂, " *Stop* " are overwritten on the position of least significant bit (20) in C, *EE* calculated by amplitude is equal to *EE* calculated by bit (*BE* = *EE*) and *EE* is calculated by φ [bit] as follows:

$$EE = BE, \tag{36}$$

$$BE = \varphi. \tag{37}$$

Simultaneously, because a number of bit of φ may coincide with some bit of C at the overwritten position so

$$BE \le \varphi \Rightarrow EE \le \varphi. \tag{38}$$

4.6 Signal- to-noise ratio (*SNR*)

In the general cases [5, 13, 20]

$$SNR = \frac{\sum_{j=1}^{n} |C_j|^2}{\sum_{j=1}^{n} |EE_j|^2} = \frac{\sum_{j=1}^{n} |C_j|^2}{\sum_{j=1}^{n} |C'_j - C_j|^2} = 10\log_{10}\left[\frac{\sum_{j=1}^{n} |C_j|^2}{\sum_{j=1}^{n} |C'_j - C_j|^2}\right][\mathrm{dB}]. \tag{39}$$

From Fig. 2 and (32), we can see that only φ audio sample are overwritten 1 bit, so (39) is rewritten as follows:

$$SNR[\mathrm{dB}] = 10\log_{10}\left[\frac{\sum_{j=1}^{n} |C_j|^2}{\sum_{j=1}^{\varphi} |C'_j - C_j|^2}\right] = 10\log_{10}\left[\frac{\sum_{j=1}^{n} |C_j|^2}{\sum_{j=1}^{\varphi} |1|^2}\right]. \tag{40}$$

Combine (40) with (38), we have

$$SNR[\mathrm{dB}] \geq 10\log_{10}\left[\frac{\sum_{j=1}^{n} |C_j|^2}{\varphi}\right]. \tag{41}$$

5 Experiment results

5.1 Experiment scenarios

We have used the Visual Basic programming software to implement the proposed method (see Fig. 4). The parameters are represented as:

- The original audio signal (C) consists of 50 different audio files $\{C_g;\ g = 1, 2, \ldots, 50\}$ in wave format, coded 16 [bit] (BitsperSample = 16[bit]).
- $\Omega = 8$[bit], (because $B_i = \Omega$ [bit] (8), and we have to compare B_i with $C_{\partial v}$, thus $C_{\partial v} = \Omega$[bit] (9). However, $C_{\partial v} = W_{\partial v}$, $W_{\partial v} \subset C_j$, $C_j = BitsperSample$[bit], thus we have 2[bit] $\leq C_{\partial v} \leq$ 16[bit]=> 2[bit] $\leq \Omega \leq$ 16[bit]. Finally, because we use audio file C in this experiment, we choose $\Omega = 8$[bit]).
- $\mho = 5$[bit], ($\mho$ is chosen based on (24), $\mho = 5$[bit] will provide 2^5 window locations in the segment C_∂),
- $\zeta = 6$[sample] (ζ is large enough to set up 2^5 window locations in the segment C_∂, each C_j will create 6 window locations [19], thus $\zeta \geq \frac{2^5}{6} \to \zeta = 6$ [sample]).
- " *Start* " = " 111111 ", " *Stop* " = " 000000 " (as shown Fig. 2a, d), " *Start* " [bit] $\leq \zeta$, and " *Stop* " [bit] $\leq \zeta$.

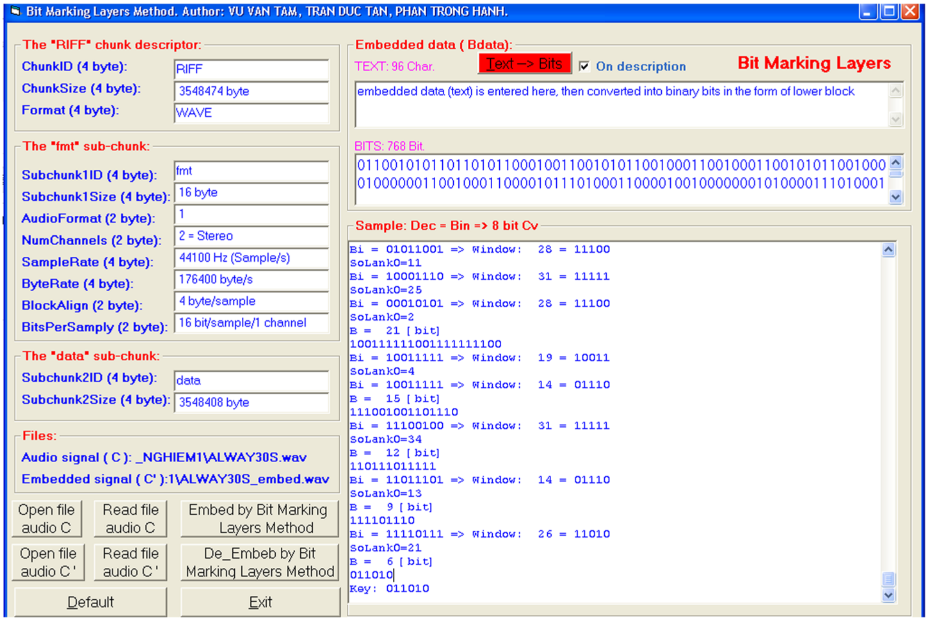
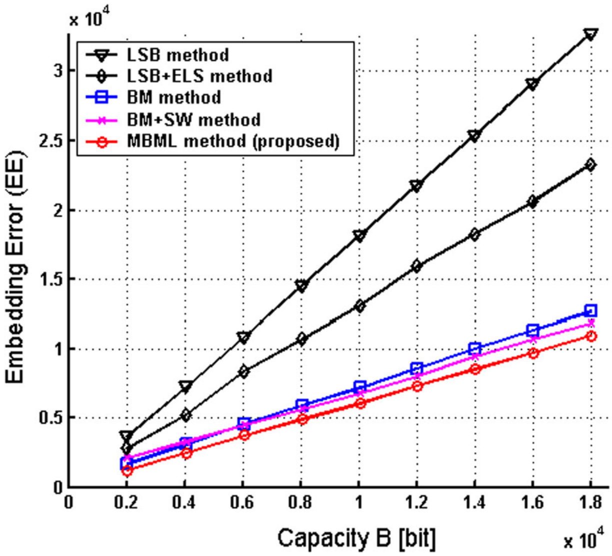


Fig. 4 The interface of the program that implement the proposed method

- The confidential data (B_{data}) consists of 09 text paragraphs $\{B_{\text{data }h}; h = 1, 2, \ldots, 9\}$ with the size varied from 200[bit] to 1800[bit]. Each audio file is embedded and de-embedded with 09 text paragraphs to get 450 embedded signal files $\{C'_{gh};\ g = 1, 2, \ldots, 50; h = 1, 2, \ldots, 9\}$.
- The formulas (35, 39) are used to calculate $\{EE_{gh};\ g = 1, 2, \ldots, 50; h = 1, 2, \ldots, 9\}$ and $\{SNR_{gh};\ g = 1, 2, \ldots, 50; h = 1, 2, \ldots, 9\}$. We compute the average parameters C_g, $B_{\text{data }h}$,

Fig. 5 EE performance



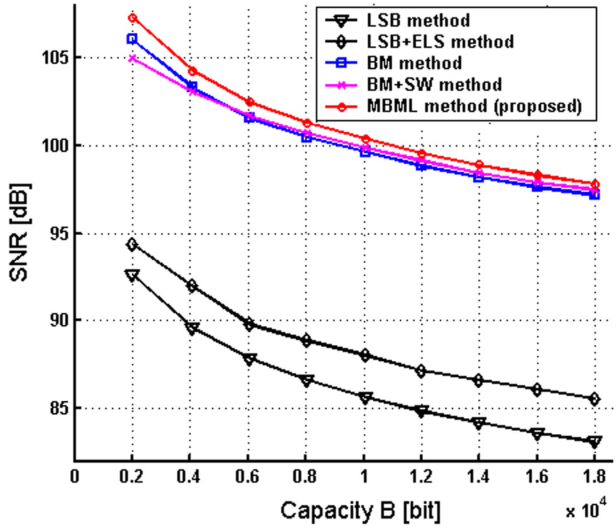


Fig. 6 SNR performance

EE_{gh},SNR_{gh} to compare with conventional methods such as LSB [10, 12, 17], ELS [18], BM [20] and the single layer bit marking method combining with window sliding technique BM + SW [19].

5.2 Results

1) Figures 5 and 6 present the performance of *EE* and *SNR* using the proposed and other conventional methods (LSB, ELS, BM, BM + SW [19]). The result shows that

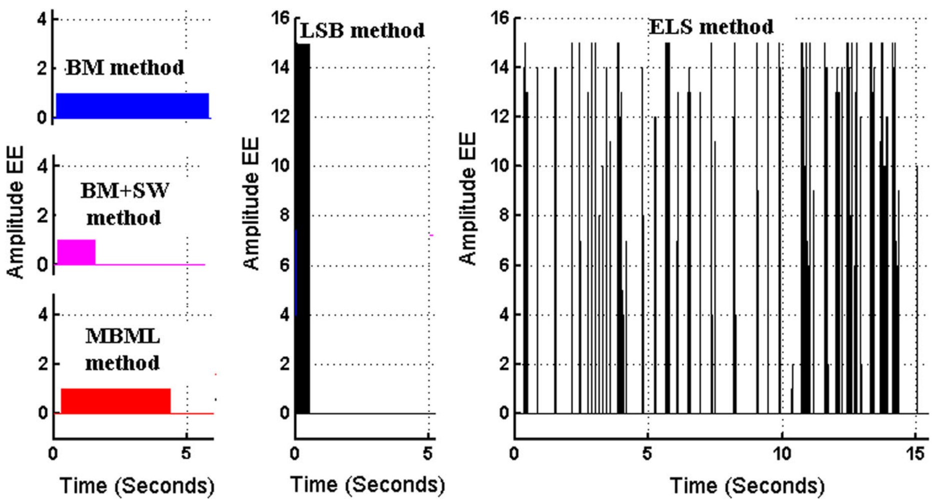
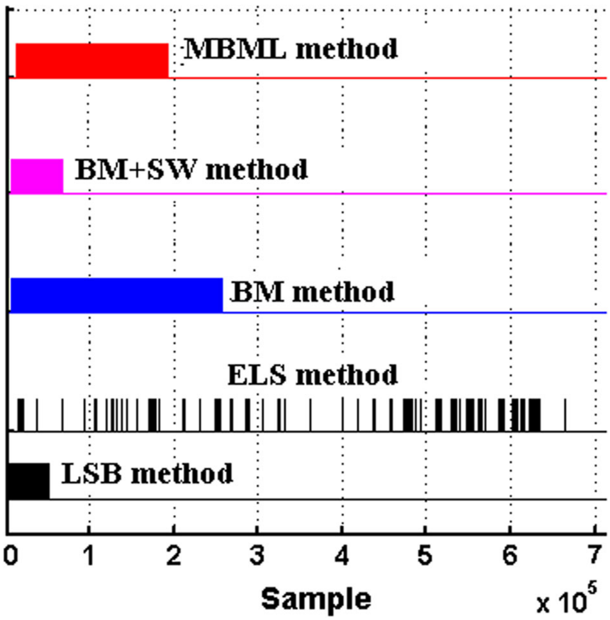


Fig. 7 The amplitude and contribution of EE

the proposed method improves significantly the quality of *EE*, and *SNR*[dB]. These results can be explained by several reasons: (i) In the embedding process, only 1[bit] (k_∂ = " 1 " or k_∂ = " 0 ") is overwritten on the original audio segment (C_∂) (with the BM/SW method, from 1 to 6 [bit] is overwritten on (C_∂)). Because the number of bit overwritten on the original audio signal decreases, the quality of *EE*, and *SNR*[dB] improve. (ii) SW technique [19] makes the times of unsuccessful embedding λ to decrease, which in turn make to φ decreases (according to (32)), then *EE* decreases (according to (37, 38)), and finally *SNR*[dB] increases (according to (41)). The improvement of *EE*, *SNR*[dB] have proved the superiority of the proposed method. It also increases the "transparency" of embedding signal, and reduce the enemy's suspicion in the transmission or storage of C'. Therefore, the safety of the confidential data embedded is enhanced.

2) Figure 7 shows the amplitude and distribution of *EE* versus to the time (in seconds) of the proposed and other conventional methods. The result shows that the proposed method has a good error amplitude as well as BM and BM/SW embedding methods, it is better than the error amplitude of LSB and ELS methods. The reason is that at the embedding process, 1 [bit] is overwritten on C_j at the position of the least significant bit, thus, at the embedded audio sample: $|C'_j - C_j| = |EE_j| \le 1$.
3) *EE* distribution of the proposed method is also improved compared to the LSB and BM/SW methods. The reason is that, each segment C_∂ has 1 bit overwritten (Fig. 2b, c), so $|C'_\partial - C_\partial| = |EE_\partial| \le 1$. Thus, segment C'_∂ consists of ζ audio samples (13) which only have 0 or 1[bit] error. Because *EE* is spread more widely and equally than the BM/SW method [19], the degradation of the sound quality is equally distributed and less, thus the enemy's suspicion decreases.

Fig. 8 Embedding capacity



4) Figure 8 shows the comparison of embedding capacity among these the methods. Although many layers are embedded in the proposed method, it still has better embedding capacity than BM method. By using the sliding window technique [19], we can reduce the number of unsuccessful embedding times (λ). Therefore, the number of original audio samples (C_j) used to embed confidential data decreases.
5) The proposed method still has some limitations as follows: (i) *EE* contribution is not as good as one of BM and ELS methods (Fig. 7). (ii) The embedded capacity of the proposed method is not as high as one of BM/SW and LSB methods. (iii) The proposed method has a more relatively complicated processing than other methods; hence it needs to have better hardware when implementing real-time processing. The limitations mentioned above will be our motivation to continue research in the field of embedding data in the audio signal.

6 Conclusions

In this paper, we proposed the multiple bit marking layers method to effectively embed the confidential data into audio signals. The experimental results show that the proposed method significantly improves the quality of embedding signal in comparison with the LSB [10, 12, 17], ELS [18], BM [20], BM/SW [19] methods. Because the confidential data is embedded with several layers along with we must use the confidential key during the embedding process; so the proposed method also enhances the safety of data embedded into audio signals. The proposed method MBML works at a base band frequency, thus, in our future work, it can also work with multimedia signal such as text, picture, video, etc. We will also develop Android/iOS applications to implement this technique in smart phones.

References

1. Al-Dalah RA, AL-Anani AM, AL-Khalid RI, Massadeh SA (2012) An efficient technique for data hiding in audio signals. American Academic & Scholarly Research Journal Special Issue 4(5)
2. Anderson RJ, Petitcolas F (1998) On the limits of steganography. IEEE J Sel Areas Commun 16: 474–481
3. Babu L, Jais John S, Parameshachari BD, Muruganantham C, DivakaraMurthy HS (2013) Steganography method for data hiding in audio signal with LSB &DCT. Int J Comput Sci Mob Comput 2(8):54–62
4. Bagade AM, Talbar SN (2014) A high quality steganographic method using morphing. J Inf Process Syst 10(2):256–270
5. Baluja Y, Mishra S, Saini TS, Patil MV (2003) Frequency domain based data hiding technique for audio signal. Int J Innov Res Sci, Eng Technol 2(5):1564–1569
6. Bhattacharyya D, Dutta P, Kim T-h (2009) Secure data transfer through audio signal. J Secur Eng 6:187–194
7. Dutta P, Bhattacharyya D, Kim T (2009) Data hiding in audio signal. Int J Database Theory Appl 2(2):1–8

8. Gruhl D, Lu A, Bender W (1996) Echo hiding. Information Hiding, Springer Lecture Notes in Computer Science, vol. 1174, pp 295–315
9. Jang B-J, Lee S-H, Lim SH, Kwon K-R (2015) Biological infectious watermarking model for video copyright protection. J Inf Process Syst 11(2):280–294
10. Kekre HB, Archana AA (2008) Information hiding using LSB technique with increased capacity. International Journal of Cryptography and Security 1(2)
11. Kekre HB, Athawale A, Rao S, Athawale U (2010) Information hiding in audio signal. Int J Comput Appl 7(9):14–19
12. Kumar S, Barnali B, Banik Y (2012) LSB modification and phase encoding technique of audio steganography revisited. Int J Adv Res Computer Commun Eng 1(4):1–4
13. Malviya S, Nayak KK, Saxena M, Khare A (2012) Audio steganography in a nutshell. Int J Electron Comput Commun Technol 2(5):219–222
14. Moein S, Gebali F, Traore I (2014) Analysis of covert hardware attacks. J Convergence 5(3):26–30
15. Petrovic R, Winograd JM, Jemili K, Metois E (1999) Data hiding within audio signals. Facta Universitatis Series: Electron Energetics 12(2):103–122
16. Ralf G, Yokotani Y, Schuller G (2006) Audio data hiding with high data rates based on INTMDCT. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp V-205–V-08
17. Saroha K, Singh PK (2010) A variant of LSB steganography for hiding images in audio. Int J Comput Appl 11(6):12–16
18. Van Tam V, Hanh PT (2014) A new method to embed data into audio signals. J Army Sci Technol Res 33: 13–19 (In Vietnamese)
19. Van Tam V, Tan TD, Thuy NT, Hanh PT (2015) Embedding data into audio signal by combining sliding window technique with a novel marking bit method. international conference on multimedia and ubiquitous engineering (MUE-15), printed at advanced multimedia and ubiquitous engineering. Lect Notes Electr Eng 352:191–197
20. Van Tam V, Tan TD, Tien ND, Thuy NT, Hanh PT (2015) Data embedding in audio signal by a novel bit marking method. Int J Advancements Computing Technol 7(1):67–77
21. Verma N, Jain VK (2013) Audio steganography—a review. Int J Adv Res Computer Commun Eng 2(1):49–55

Van Tam Vu was born in 1973. He received his B.Sc, M.Sc. degrees respectively in 2004, 2009 at the Le Quy Don Technical University, Vietnam. University of Technology and Logistics, Vietnam, where he has been a lecturer since 2010. His present research interest is in DSP applications.

Duc-Tan Tran was born in 1980. He received his B.Sc, M.Sc, and PhD. degrees respectively in 2002, 2005, and 2010 at the University of Engineering and Technology (UET), Vietnam National University – Hanoi, Vietnam (VNUH), where he has been a lecturer since 2006. He was the recipient of the Vietnam National University, Hanoi, Vietnam Young Scientific Award in 2008. He is currently an Associate Professor with the Faculty of Electronics and Telecommunications, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam. He is the author and coauthor of 50 papers on signal processing and ít applications.

Trong Hanh Phan was born in 1967. He received his B.Sc, M.Sc degrees respectively in 1989, 1998 at the Le Quy Don Technical University, Vietnam. He received the PhD. degree in 2007 at the Moscow Aviation Institute, Russian Federation. His current research interests include DSP applications and theoretical physics.