

## Research Article

### A Novel and High Capacity Audio Steganography Algorithm Based on Adaptive Data Embedding Positions

<sup>1,2</sup>Haider Ismael Shahadi, <sup>2</sup>Razali Jidin and <sup>2</sup>Wong Hung Way

<sup>1</sup>Electrical Engineering Department, University of Babylon, Hilla, Babil, Iraq

<sup>2</sup>Electronic and Communication Engineering, Tenaga National University (UNITEN), Putrajaya, Malaysia

**Abstract:** In this study, a novel and high embedding capacity audio steganography scheme based on Lifting Wavelet Transform (LWT) and adaptive embedding positions is proposed. Specifically, the message data is inserted in the imperceptible positions that chosen from the coefficients of detail sub-bands taking advantage of our proposed Weighted Block Matching (WBM). The WBM is preceded by preparing the cover audio in order to select the bits-positions that can possibly be used for embedding from each detail coefficient based on coefficient amplitude then copy the contents of the selected bits-positions and arrange them in blocks of bits. Also, the message data is arranged in blocks of bits after preprocessed and encrypted. The WBM computes the matching between each message block and whole extracted cover blocks to find the similarity between them. This process help to provide optimal locations to hide the message blocks. These locations are considered as a stego-key that is ciphered and hided within the final detail sub-band which is specified for this purpose. The proposed approach attains higher security than other fixed embedding positions approaches because the random positions for the embedded message blocks based on adaptive selection for embedding positions. Experimental results show that the proposed technique is not only has very high embedding capacity (exceed 300 kbps) with excellent transparency (above 35 dB for the cover to noise ratio), but also achieve lossless message retrieved. Comparisons with the related audio steganography algorithms also show that our proposed scheme outperforms all the selected algorithms.

**Keywords:** Adaptive embedding positions, audio steganography, high embedding capacity, Lifting Wavelet Transform (LWT), Weighted Blocks Matching (WBM)

## INTRODUCTION

Cryptography and steganography are two important technologies that are used in secure communications to prevent data hacking and eavesdropping (Cheddad *et al.*, 2010; Nissar and Mir, 2010). Cryptography encrypts a message in such a way that it becomes incomprehensible, whereas steganography hides a secret message in a cover signal without attracting attention. Sending of an encrypted message may create suspicions for an eavesdropper, whereas a hidden message in a cover signal does not do so. Nevertheless, both these technologies can be combined for a higher level of message protection (Cheddad *et al.*, 2010).

Unlike cryptography, steganography benefits from the perception limitations of human auditory or visual system, which fails to recognize difference between host and stego-signals (Huang *et al.*, 2010). Usually, steganography uses media files such as, image, audio or video as host signals to hide data. In general, using the audio signal as a steganography cover signal is less popular than image, because Human Auditory System (HAS) is more sensitive to noise in signal than Human

Visual System (HVS) (Bender *et al.*, 1996; Ercelebi and Batakc, 2009).

Each steganography algorithm requires some features which depend on the applications and transmission media. The most important requirements are transparency, security, high embedding capacity and robustness (Wang and Wang, 2004; Wang *et al.*, 2011).

Recently many audio steganography methods have been proposed. The earliest technique in the audio steganography has utilized a single LSB within the time domain to embed one message bit in each cover sample (Bender *et al.*, 1996). To increase embedding capacity, Cvejic (2004) used 3 and 4 LSBs of each cover sample to embed message bits. However, the perceptual quality of the output signal is lessened. Generally, the LSB method is very sensitive to the additive noise; therefore, many researchers have attempted to increase the robustness of LSB methods, by altering the LSB position such as proposed by Cvejic and Seppnen (2004a), they have adopted the 6<sup>th</sup> LSB bit of each sample with 16 bits resolution, to embed the message bits.

Many other high embedding capacity audio steganography hide secret data within the LSBs of cover coefficients in the transform domain such as, Fourier and Discrete Wavelet domains. Cvejic and Seppanen (2004b) have investigated LSBs coding for different domains (time, frequency and wavelet), in terms of data hiding capacity and perceptual transparency. They have employed Discrete Fourier Transform (DFT) to convert each frame of 1024 samples into frequency domain. In this method, on message bits are embedded in the LSBs of the cover coefficients after scaling and converting the coefficients to binary, followed by descaling and inverse of DFT to reconstruct the stego signal. They have attained embedding capacity of 150 kbps. Cvejic and Seppanen (2002) have proposed an audio steganography algorithm based on DWT. The cover signal is framed to 512 samples per frame; then, each frame is decomposed by using five levels of Haar-DWT; next, all the 512 wavelet coefficients are scaled according to the maximum value of each sub-band and then rounded to the nearest integers. After that, all the scaled coefficients are converted to binary. Subsequently, the message bits are embedded in the LSBs of the binary scaled coefficients. The reconstruction of the stego signal is achieved by descaling and Inverting the DWT (IDWT) processes. The embedding capacity achieved in those works is about 150-200 kbps (for 16 bits per sample resolution and 44100 Hz audio signal).

Although high embedding capacity and good transparency can be achieved in both, DFT and DWT steganography algorithms, these techniques still have disadvantages because of the data type conversions (from integer-to-floating and vice versa). They require scaling and rounding operations, before to the data embedding and descaling and rounding after the data embedding. Consequently, some of the hidden information may be lost (Djebbar *et al.*, 2011). To reduce or eliminate the data type conversion errors, which may be occurred in the retrieved hidden data, algorithms which have been developed below literatures adopt integer transform domain.

Geiger *et al.* (2006) have proposed an audio algorithm bases on Integer Modified Discrete Cosine Transform (IntMDCT). In order to maintain transparency, the bits of each integer DCT coefficient are divided into significant and insignificant bits. This is realized by considering the highest non zero bit (leading bit) with a constant number of lower bits as significant bits and all other lower bits as insignificant. The insignificant bits are used for data embedding. In that study, an embedding capacity of about 140 kb/sec with good transparency has been achieved.

Delforouzi and Pooyan (2006) have inserted an encrypted message into the LSBs of the cover audio coefficients in the integer wavelet domain. A human hearing threshold is employed to specify the maximum hiding rate. The number of the LSBs bits, which are used for embedding in each coefficient, is calculated

according to coefficient value and its sub-band (approximation or detail) thresholds. This algorithm has embedding capacity reaches to 250 kb/sec, with a 44100 Hz sampling frequency and cover to noise ratio after embedding about 35 dB.

Pooyan and Delforouzi (2007) have employed five levels of packet integer-LWT to decompose the cover audio signal. Then, the hearing threshold is calculated for each sample in the integer-LWT domain according to its sub-band. Based on the calculated threshold, data bits are embedded in the LSBs of the integer-LWT coefficients. Inverse-LWT is applied on the modified coefficients to construct the stego audio signal in time domain. In that study, an embedding capacity higher than 200 kbits/sec with full data recovery has been achieved.

Shahreza and Shalmani (2007, 2008) have proposed a steganography scheme, based on the integer-LWT. The data are embedded in the LSBs of the detail coefficients after decompose cover audio by integer-LWT. In Shahreza and Shalmani (2008) the number of LSBs bits to be used in the embedding is calculated for each cover detail coefficient  $c$  by finding the biggest power of 2, called  $p$ , which should be smaller than the value of  $c$ , where,  $p$  can be found by the inequality  $2^p < c < 2^{p+1}$ . Then, the number of bits that can be inserted is equal to  $(p-OBH)$ ; where, OBH (Original Bit to Hold) is a constant that has a minimum value, equals to one. In that study, an embedding capacity about 20% of the input speech signal with acceptable transparency and full recovery has been accomplished.

In our study (Shahadi and Jidin, 2011), we have adopted Wavelet Packet Transform (WPT), to decompose an audio cover signal to L-levels and after scaling and converting to binary, we have selected the LSBs of the details coefficients, which can be possibly used in the embedding process based on its strength. Subsequently, the bits block matching between the message bits and the LSBs of the host details coefficients are carried out, to seek optimal positions, to insert the message bits. Then, the modified details components coefficients are de-scaled and inversion of WPT is performed, to reconstruct the stego-signal. In that study, we have achieved a very high embedding capacity (about 300 kbps), with at least 50 dB SNR for the output perceptual quality. Although, this algorithm has a very high embedding capacity and excellent perceptual transparency, unfortunately its robustness is being compromised due to the multiplication scaling. This type of scaling may cause losses in the recovered data, in the case of transmitting or saving the stego-signal by the same number of bits/sample for the input cover signal.

To solve above problem while maintaining other advantages, we present in this study a novel approach that employs three levels of integer-to-integer Lifting Wavelet Transform (LWT) and adaptive embedding positions. The LWT is a lossless transformation and it has less complexity compared with conventional DWT

(Dewine and Comelis, 1997; Lee and Ko, 2011; Lei position method depends on the amplitudes of the cover coefficients in the integer-LWT domain and the results of Weighted Block Matching (WBM). The WBM is attained to find similarities between each block of the secret message and whole extracted blocks from the cover detail coefficients based on their amplitudes. The proposed approach provides a high embedding capacity with an excellent transparency, also the retrieved data is fully recovered without any lost.

### PROPOSED AUDIO STEGANOGRAPHY APPROACH

**Hiding phase:** The hiding phase includes five main stages as described in Fig. 1. These stages are demonstrated in the following sub-sections.

**Message preprocessing and encryption stage:** The proposed steganography has been tested by using three types of input messages: audio, images and text. The steps in this stage involves in preprocessing and encrypting an input secret message as follows:

- An input message is arranged in a one Dimensional (1D) vector called as  $M$ , then the vector  $M$  is segmented into  $G$  segments and each segment ( $M_i$ ,  $i = 1, 2, \dots, G$ ) has length of  $Q$  samples.
- Each segment  $M_i$  is converted into binary form and then converted to the two Dimensional (2D) matrix named as  $Mb_i$ , which has size equal to  $N_1 \times N_2$ , where  $N_1$  is the number of rows in the  $Mb_i$  and  $N_2$  is the number of bits in each row.

et al., 2012). The proposed adaptive embedding

- In order to encrypt each binary message segment ( $Mb_i$ ), Pseudo Random Generator (PRG) is used to generate a random binary matrix, which has ( $N_1 \times N_2$ ) size, follows by logical exclusive or (XOR) operation between  $Mb_i$  and PRG output. The output values of PRG matrix can be changed according to an input cipher key, which is entered to the PRG as a secret key by the user. The matrix produced from the encryption process is named as  $Emb_i$ .
- Finally, each binary matrix of the encrypted message segment ( $Emb_i$ ) is fed to the embedding process stage (third stage), to hide it in one of the cover signal segments.

**Audio cover signal decomposition and preparation stage:** The steps of this stage are as follow:

- An input cover audio signal  $C$  is segmented into  $G$  segments; each segment ( $C_i$ ,  $i = 1, 2, \dots, G$ ) has length of  $Z$  samples. Then, 3-levels of Int2Int Haar LWT are used to decompose each segment  $C_i$ . The LWT decomposition results four sub-bands: first and second detail sub-bands ( $D_{1i}$  and  $D_{2i}$ ) have length  $Z/2$  and  $Z/4$  coefficients, respectively. As for the third detail ( $D_{3i}$ ) and smooth ( $S_{3i}$ ) sub-bands they each have length of  $Z/8$  coefficients.
- For each decomposed audio cover segment, the first and second detail sub-bands are combined into a one vector ( $D_{12i}$ ,  $i = 1, 2, \dots, G$ ), to be used for embedding a one segment from the encrypted message ( $Emb_i$ ). While, the third detail component

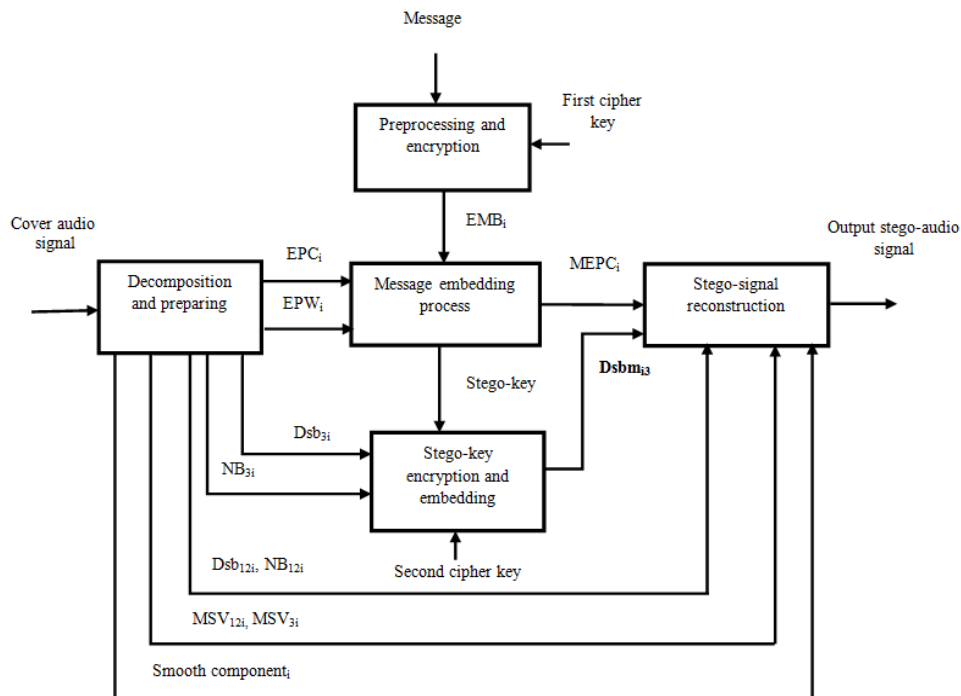


Fig. 1: The general block diagram of the proposed hiding phase

( $D_{3i}$ ,  $i = 1, 2, \dots, G$ ) is employed to carry a stego-key which will be generated after the matching process (in the next stage).

- A Map Sign Vector (MSV) is constructed for each of the vectors  $D_{12i}$  and  $D_{3i}$ , where their values equal to (-1), if the coefficient has negative value, or 1 otherwise. The two vectors ( $MSV_{12i}$  and  $MSV_{3i}$ ) are required later in the reconstruction stage.
- Absolute values of both  $D_{12i}$  and  $D_{3i}$  are computed and then an integer scaling process is performed as in Eq. (1):

$$Ds_{12i} = \{abs(d_{12i}(j)) + s_{12}, 1 \leq j \leq (\frac{z}{2} + \frac{z}{4}), 1 \leq i \leq G\} \quad (1)$$

$$Ds_{3i} = \{abs(d_{3i}(j)) + s_3, 1 \leq j \leq \frac{z}{8}, 1 \leq i \leq G\}$$

The scaling in Eq. (1) is optional; if we do not need very high embedding capacity, then the values of scale factors  $s_{12}$  and  $s_3$  are set to zeros. If high embedding capacity is required, then the values of  $s_{12}$  and  $s_3$  can be chosen within a range of  $\{0, s_{max}\}$ , where  $s_{max}$  can be calculated as the following:

$$\begin{aligned} s_{max\ 12i} &= 2^b - 2^{r_{12i}} - 1, s_{max\ 3i} = 2^b - 2^{r_{3i}} - 1, \\ r_{12i} &= \lceil \log_2 (Max (D_{12i})) \rceil, i = 1, 2, \dots, G \\ r_{3i} &= \lceil \log_2 (Max (D_{3i})) \rceil, i = 1, 2, \dots, G \end{aligned} \quad (2)$$

where,  $[A]$  rounds the value of A to the nearest integers, greater than or equal to A. The rounding is used to limit the integer coefficients values in the range  $\{0, 2^{b-1} - 1\}$  after scaling, where;  $b$  indicates number of bits per sample for the audio cover signal. The above scaling method guarantees that all scaled and de scaled coefficients remain as integer. Unlike the multiplication scaling approach, this integer scaling is used to specify minimum number of bits which can be possibly used for embedding in those coefficients which have absolute amplitudes smaller than scalar value:

- Convert each of  $Ds_{12i}$  and  $Ds_{3i}$  into binary, to obtain  $Dsb_{12i}$  and  $Dsb_{3i}$ , respectively that they have number of bits per coefficient equal to  $b-1$ .
- Calculate Number of Bits (NB) for each coefficient of  $Dsb_{12i}$  and  $Dsb_{3i}$ , which can be used for embedding. The value of NB is dependent on each coefficient amplitude as shown in Eq. (3):

$$NB_{12i} = \{p_{12i}(j) - sb_1 - sb_2 - 1, 1 \leq j \leq (\frac{z}{2} + \frac{z}{4}), 1 \leq i \leq G\} \quad (3)$$

$$NB_{3i} = \{p_{3i}(j) - sb_1 - sb_2 - 1, 1 \leq j \leq \frac{z}{8}, 1 \leq i \leq G\}$$

where,  $p_i(j)$  is the first position from the Most Significant Bits (MSB) direction in the coefficient  $dsb_i(j)$ , which has value equal to logic 1;  $sb_1$  is the number of bits, which is not used for embedding

and they have positions starting directly after  $p_i(j)$  bit position, towards the LSB direction. We use  $sb_1$  to minimize the difference between the original (before embedding) and modified (after embedding) coefficients. As for  $sb_2$ , it is the number of bits which is not used for embedding and it has positions starting from the first LSB towards the  $p_i(j)$  position. The  $sb_2$  is used to increase depths of the embedding message bits, which indicates the increase in the power of embedding data. As a result, the embedded data resistivity is higher, against the additive noise.

The following example shows NB calculation process: let  $dsb_{12i}(j) = 0000101011000101$ , Assume  $sb_1 = sb_2 = 2$ . From  $dsb_{12i}(j)$  we find  $p_{12i}(j) = 12$  then  $nb(j) = 7$ .

|     |   |   |   |   |   |                 |        |   |   |   |   |   |   |   |                 |
|-----|---|---|---|---|---|-----------------|--------|---|---|---|---|---|---|---|-----------------|
| 0   | 0 | 0 | 0 | 1 | 0 | 1               | 0      | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1               |
| MSB |   |   |   | p |   | sb <sub>1</sub> | nb = 7 |   |   |   |   |   |   |   | sb <sub>2</sub> |

- Then, an Embedding Position Contents vector (EPC<sub>i</sub>) is constructed by coping bits values based on  $NB_{i12}$  positions which are greater than zero, it is as follows:

Initialize the counter  $k = 1$ , if  $nb_{12i}(j) > 0$  then

$$EPC_i(k : k + nb_{12i}(j) - 1) = \{dsb_{12i}(j, sb_2 + 1 : nb_{12i}(j) + sb_2), \quad (4)$$

$$k = k + nb_{12i}(j),$$

$$1 \leq j \leq (\frac{z}{2} + \frac{z}{4}), 1 \leq i \leq G\}$$

For the above example,  $epc(1:7) = \{1, 0, 0, 0, 1, 1, 0\}$  for coefficient  $dsb_{12i}(j) = 0000101011000101$ .

- Then, Embedding Position Weight vector (EPW) is constructed and it is depending on the  $NB_{i12}$  values which are greater than zero, EPW is obtained as follows:

Initialize the counter  $k = 1$ , if  $nb_{12i}(j) > 0$  then

$$EPW_i(k : k + nb_{12i}(j) - 1) = \{2^{sb_2}, 2^{sb_2+1}, 2^{sb_2+2}, \dots, 2^{sb_2+nb_{12i}(j)-1}\}, \quad (5)$$

$$k = k + nb_{12i}(j),$$

$$1 \leq j \leq (\frac{z}{2} + \frac{z}{4}), 1 \leq i \leq G\}$$

For the above example,  $epw(1:7) = \{4, 8, 16, 32, 64, 128, 256\}$ .

|        |      |      |      |     |     |     |    |    |    |   |   |   |   |                   |   |
|--------|------|------|------|-----|-----|-----|----|----|----|---|---|---|---|-------------------|---|
| 0      | 0    | 0    | 0    | 1   | 0   | 1   | 0  | 1  | 1  | 0 | 0 | 0 | 1 | 0                 | 1 |
| MSB... | 4096 | 2048 | 1024 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Positions Weights |   |

- Finally, each of  $EPC_i$  and  $EPW_i$  vectors is converted into 2D matrix that has size  $N_3 \times N_2$  and the final remainder in the matrix is to be neglected if it has number of bits less than  $N_2$ .

**Message embedding stage:** Each encrypted message segment is embedded into a one  $EPC_i$  matrix, which is obtained in the previous stage. As mentioned earlier, the embedding position is performed adaptively and it requires three processes. First, to extract the bits of the positions which can be possibly used to embed message bits for each selected cover coefficient, as explained in above section. Second, the weighted block matching is conducted and finally, the message blocks (rows of  $EPC_i$ ) are embedded in optimal matched positions. The message embedding steps are as follows:

- First step:** The weighted block matching is calculated by comparing each block (row in matrix) of the Encrypted Message data matrix ( $EMb_i$ ) with whole rows of the Embedding Positions Contents matrix ( $EPC_i$ ) as follows:

If  $emb_i(j, k) = epc_i(l, k)$ , then  $match_{ij}(l, k) = epw_i(l, k)$  else  $match_{ij}(l, k) = 0$

$$1 \leq k \leq N_2, 1 \leq l \leq N_3, 1 \leq j \leq N_1, 1 \leq i \leq G$$

The summation of all values of  $match_{ij}$  over each block gives the final match value between  $j$  row from  $EMb_i$  and  $l$  row from  $EPC_i$  matrix. Thus, the final Block Matching ( $BM$ ) matrix is computed as in Eq. (6):

$$bm_i(j, l) = \sum_{k=1}^{N_2} match_{ij}(k), 1 \leq l \leq N_3, 1 \leq j \leq N_1, 1 \leq i \leq G \quad (6)$$

$$BM_i = \{bm_i(j, l), 1 \leq l \leq N_3, 1 \leq j \leq N_1, 1 \leq i \leq G\}$$

- The second step is to compute the index of the maximum value for each row in  $BM_i$  matrix. Then all the elements in  $BM_i$  matrix, which have indices equal to  $ind_i(j)$ , starting from row  $(j+1)$  till to  $N_1$ , are replaced by negative values (for example -1 as shown in Eq. (7)), to avoid overwriting, during the embedding process, if there is any similarity in the matching result between two or more blocks:

$$ind_i(j) = \text{index}(\text{Max}(bm_i(j, 1 : N_3))), \text{let} \quad (7)$$

$$BM_i(j+1 : N_1, ind_i(j)) = \{-1\}, 1 \leq j \leq N_1, 1 \leq i \leq G,$$

$$IND_i = \{ind_i(j), 1 \leq j \leq N_1, 1 \leq i \leq G\}$$

where,  $\text{index}(\text{Max}(bm_i))$  indicates a function that calculates the index of the maximum value in the vector  $bm_i$ .

- Now, the computed index vector above represents optimal positions to embed the  $EMb_i$  blocks into the  $EPC_i$  matrix and it represents the stego-key as in Eq. (8):

$$Key_i = \{ind_i(j), 1 \leq j \leq N_1, 1 \leq i \leq G\} \quad (8)$$

- Next, replace  $EPC_i$  rows which have indices equal to  $Key_i$  contents by  $EMb_i$  rows as follows:

$$EPC_i(key_i(j)) = \{emb_i(j, k), 1 \leq k \leq N_2, 1 \leq j \leq N_1, 1 \leq i \leq G, emb_i(j, k) \in \{0, 1\}\} \quad (9)$$

- Finally, the above Modified  $EPC_i$  which is named ( $MEPC_i$ ) is fed to stage of stego-signal reconstruction. As for the stego-key ( $Key_i$ ), it is forwarded to stage of stego-key encryption and embedding.

**Encryption and embedding of the stego-key stage:** The steps of this stage are:

- A stego-key is converted to binary form and its results are arranged into a 1D binary vector. Then, Pseudo Random Generator (PRG) is used to generate a random binary vector, which has the same length of the binary stego-key. Then, the XOR operation is performed between the two vectors, to obtain an encrypted stego-key and it is named as  $ESK_i$ . The output of PRG can be altered by changing the second secret key. Finally, an extra 40 bits are added to the beginning of the  $ESK_i$  binary vector as a header, which represents the message size (16 bits for number of rows and 16 bits for number of columns) and the number of the bits per sample for the stego-key (8 bits). These 40 bits are to be used for retrieval of stego-key and secret message at the receiver side.
- The binary vector produced in the first step ( $ESK_i$ ) is inserted into the LSBs of the last details component coefficients ( $D_{s3i}$ ) based on the positive values of  $NB_{3i}$  as the following:

Initialize the counter  $k = 1$ , if  $nb_{12i}(j) > 0$  then

$$Dsb_{3i}(j, sb_2 + 1 : nb_{3i}(j) + sb_2) = \{esk_i(k : k + nb_{3i}(j) - 1), \quad (10)$$

$$k = k + nb_{3i}(j),$$

$$1 \leq j \leq U, 1 \leq i \leq G\}$$

where,  $U$  is the smallest value of  $j$ , which make the summation ( $\sum_j^U nb_{3i}(j)$ ) greater than or equal to the total number of bits in the  $ESK_i$  vector. The modified third component, named ( $Dsbm_{i3}$ ) is fed to the stego-signal reconstruction stage.

**Stego-signal reconstruction stage:** This is the final stage in the proposed hiding algorithm and it is used to reconstruct the output stego-audio signal. The first step in this stage is to convert the Modified Position Contents matrix ( $MEPC_i$ ) for each cover segment, into a one dimensional vector, followed by returning of each bit to its original place in the coefficient matrix  $Dsb_{12i}$ , as follows:

$$\begin{aligned}
 & \text{Initialize} \rightarrow Dsbm_{12i} = Dsb_{12i} \\
 & \text{The counter } k = 1, \text{ if } nb_{12i}(j) > 0 \text{ then} \\
 & Dsbm_{12i}(j, sb_2 + 1 : nb_{12i}(j) + sb_2) = \{mepc_i(k : k + nb_{12i}(j) - 1), \\
 & k = k + nb_{12i}(j), \\
 & 1 \leq j \leq (\frac{Z}{2} + \frac{Z}{4}), 1 \leq i \leq G\}
 \end{aligned} \tag{11}$$

where,  $Dsbm_{12}$  is the  $Dsb_{12i}$  matrix after modification (data embedding).

The next step is to convert each modified binary details components ( $Dsbm_{12}$  and  $Dsbm_3$ ) to decimal and descaled as follows:

$$\begin{aligned}
 Dm_{12i} &= \{dsm_{12i}(j) - s_{12}, 1 \leq j \leq (\frac{Z}{2} + \frac{Z}{4}), 1 \leq i \leq G\} \tag{12} \\
 Dm_{3i} &= \{dsm_{3i}(j) - s_3, 1 \leq j \leq \frac{Z}{8}, 1 \leq i \leq G\}
 \end{aligned}$$

where, identifiers  $s_{12}$  and  $s_3$  are the same as those used in the host signal preparation stage.

Next step is to return the original sign of each coefficient in the details components, by multiplying each element from  $Dsbm_{12}$  and  $Dsbm_3$  by the corresponding element from  $MSV_1$  and  $MSV_2$ , respectively. After that, the signed modified details component vector ( $Dm_{12i}$ ) is separated into  $Dm_{1i}$  and  $Dm_{2i}$ , which have lengths of  $Z/2$  and  $Z/4$  samples, respectively.

Finally, all the modified details components ( $Dm_{1i}$ ,  $Dm_{2i}$  and  $Dm_{3i}$ ) and non-modified smooth component are passed to the Inverse Lifting Wavelet Transform (ILWT), to reconstruct the output stego-audio segment number  $i$ . The final output stego-audio signal is reconstructed by combining all  $G$  segments in a one signal.

**Message recovery phase:** The message recovery algorithm has four main stages: the first stage consists of the stego-audio signal decomposition and preparation. The steps of this stage are exactly the same as those in the preparing stage for the audio hiding algorithm. The second stage is the recovery of the stego-key stage. The stage starts by extracting the encrypted stego-key from the LSBs positions of  $DS_{3i}$ , based on  $NB_i$  that has been calculated in the first stage. Next, the encrypted stego-key is rearranged in a 1D vector according to the length ( $N_i$ ) and number of bits per sample, which are extracted from the 40-bits header. Then the encrypted key is forwarded to the stego-key deciphering steps, which are similar to the stego-key ciphering steps as explained in above subsections. The resulting binary stego-key is arranged and converted to decimal, to obtain the final stego-key.

The third stage is the recovery of the encrypted message information stage. In this stage, the Embedding Position Contents ( $EPC_i$ ) matrix for each segment is extracted, based on the positive values of  $NB_{i12}$  elements as given in Eq. (4). Next step is to

convert the binary  $EPC_i$  into a 2D matrix that has size of  $N_3 \times N_2$  (where,  $N_2$  was provided by the previous stage). The reminder in the matrix is neglected if the number of bits is less than  $N_2$ . The final step in this stage is to extract the  $EPC_i$  blocks (rows) that have indices equal to the values of stego-key, to construct the encrypted message information as given in Eq. (13):

$$\begin{aligned}
 Emb_i(j) &= \{pc_i(key_i(j), k), 1 \leq k \leq N_2, \\
 & 1 \leq j \leq N_1, 1 \leq i \leq G, pc_i(j, k) \in \{0, 1\}\}
 \end{aligned} \tag{13}$$

The final stage in the message recovery algorithm is the message reconstruction. In this stage, the Encrypted Message Information Matrix ( $Emb_i$ ) needs to be decrypted, using identical procedure as described in the message encryption process. The resulting message segment matrix is converted to a one dimensional vector of integer decimal and then all the recovered segments are combined together. Finally, the vector is converted to the required data type to obtain the required message.

## RESULTS AND DISCUSSION

In this section, we present experimental results for the proposed system, as well as some comparisons with some of the latest related works to ours.

### Tests of perceptual transparency and embedding capacity:

Perceptual transparency or perceptual quality means inability of human hearing to perceptible difference in the cover signal before and after embedding. It can be measured either by hearing to both the cover and stego (cover after message embedding) signals (by several people whose have excellent hearing), or by using mathematical measurements. A good mathematical way to measure a signal quality is to calculate the Signal-to-Noise power Ratio (SNR) (Souvic *et al.*, 2012), by regarding the difference between the cover and stego-signal as a noise, as shown in Eq. (14):

$$SNR = 10 \log_{10} \frac{\sum_{i=1}^G \sum_{j=1}^Z C_i^2(j)}{\sum_{i=1}^G \sum_{j=1}^Z [C(j) - C'(j)]^2} \tag{14}$$

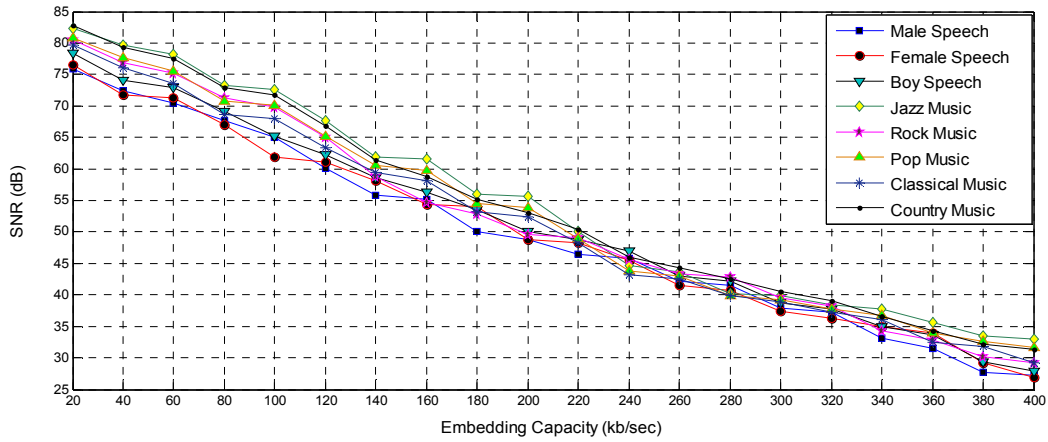
where,

$C$  and  $C'$ : The cover stego signals

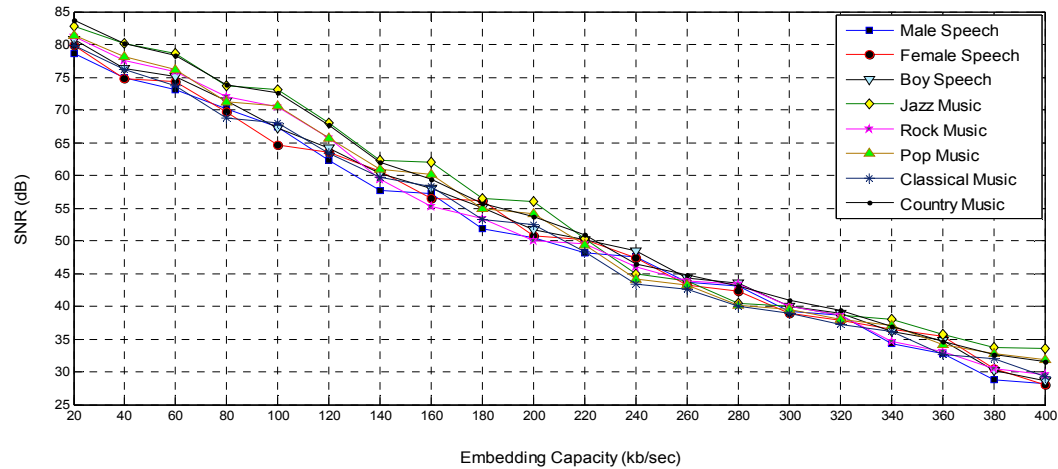
$Z$ : The number of samples in each segment

$G$ : The number of segments

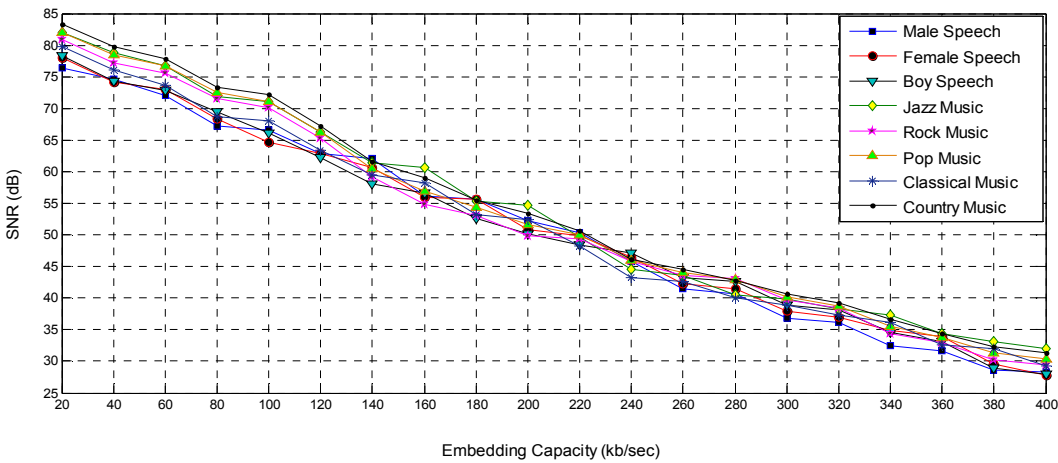
The second important feature for a steganography is embedding capacity or data rate. The embedding capacity represents an amount of embedded data in a cover signal per unit of time, such as bits per second (bps). Also it can be measured as a ratio of message to the cover data size. We have conducted experiments for varying data hiding capacities in order to test the



(a) Audio Message



(b) Image Message



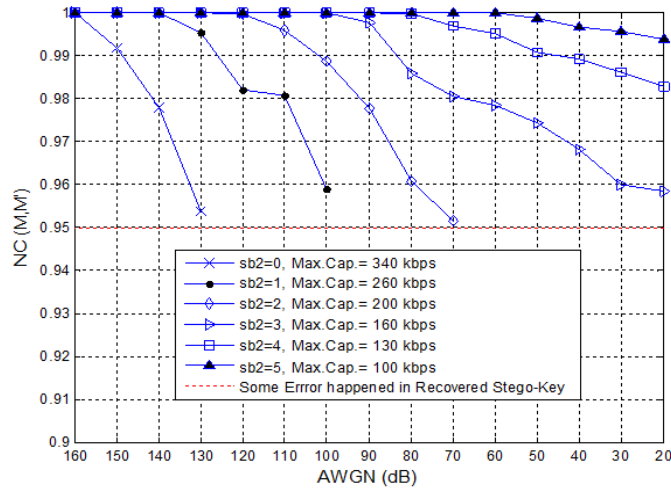
(c) Text Message

Fig. 2: The tests of embedding capacity versus output perceptual quality

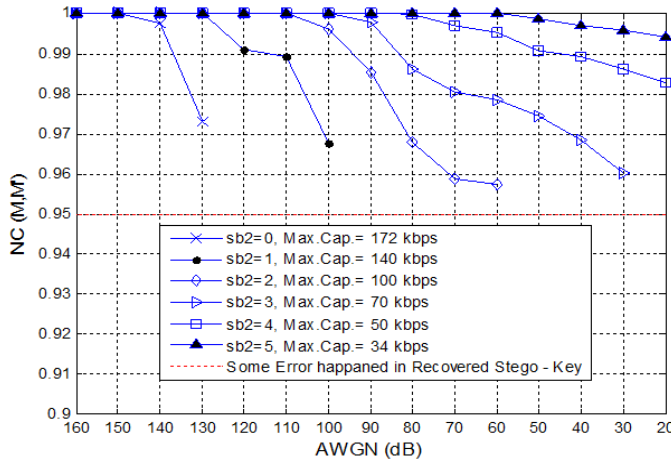
perceptual quality in terms of *SNR* and parts of these experiments are shown in Fig. 2.

For the tests shown in Fig. 2, we have used eight audio signals (5 music and 3 recorded speeches) with

16 bits/sample resolution, at 44100 Hz sampling frequency and 8 sec duration. Additionally, in terms of different type of messages, we have used image (boats image that has size 512×512), audio (recorded speech)



(a) With scaling (s = 4096)



(b) With scaling (s = 0)

Fig. 3: Robustness tests for the retrieved hidden message by computing Normalized Cross correlation (NC) for different channel AWGN  
 Max. Cap.: The maximum capacity with at least 35 db of perceptual quality at least 35 db of perceptual quality

and text. Each one of the messages is embedded in all the five audio host signals as described in sub-figures of Fig. 2. In these tests, we have used the following attributes:  $G = 1024$  samples/segment,  $N_2 = 64$  bits/block,  $s_{12} = s_3 = 1024$  and  $sb_2 = 0$ . As the value of  $sb_1$  that influence the data hiding capacity, it starts from 13 bits for the case of 20 kbps embedding capacity and it ends at 4 bits for the case of 400 kbps. The test results have demonstrated that, the perceptual is excellent (above 35 dB) until the embedding capacity reaches 340 kbps independence of audio host sources.

**Robustness tests:** Robustness means an ability to recover a hidden message from a host signal without or with an acceptable distortion after a stego signal has been affected by factors such as channel additive noise. Similarity between the recovered and original messages is the usual method being used to measure robustness. The most popular method is Normalized Correlation (NC) presently being adopted to measure the

similarities. The NC formula for one dimensional message signal such as the audio is shown in Eq. (15):




































$$NC(M, M') = \frac{\sum_{k=1}^{QG} M(k)M'(k)}{\sqrt{\sum_{k=1}^{QG} M(k)^2} \sqrt{\sum_{k=1}^{QN} M'(k)^2}} \quad (15)$$

where,  $M$  and  $M'$ , are the original and recovered secret message, respectively;  $QG$  indicates number of samples in each one of them. For the two dimensional embedded message such as an image, the NC formula is shown in Eq. (16):

$$NC(M, M') = \frac{\sum_{i=1}^{Q1} \sum_{j=1}^{Q2} M(i, j)M'(i, j)}{\sqrt{\sum_{i=1}^{Q1} \sum_{j=1}^{Q2} M(k)^2} \sqrt{\sum_{i=1}^{Q1} \sum_{j=1}^{Q2} M'(k)^2}} \quad (16)$$



Table 1: Robustness tests for image embedding message against AWGN for different values of  $sb_2$

| The value of $Sb_2$ | Without adding noise  | AWGN with SNR = 140 dB   | AWGN with SNR = 130 dB   | AWGN with SNR = 100 dB  | AWGN with SNR = 60 dB  | AWGN with SNR = 30 dB   | AWGN with SNR = 20 dB   |
|---------------------|---|--|--|---|--|---|---|
| $Sb_2 = 0$          | <br>NC = 1 | <br>NC = 0.9988 | <br>NC = 0.9525 | <br>Some errors in recovered stego-key | <br>Some errors in recovered stego-key | <br>Some errors in recovered stego-key | <br>Some errors in recovered stego-key |
| $Sb_2 = 1$          | <br>NC = 1 | <br>NC = 1.000  | <br>NC = 0.9922 | <br>NC = 0.9540                        | <br>Some errors in recovered stego-key | <br>Some errors in recovered stego-key | <br>Some errors in recovered stego-key |
| $Sb_2 = 2$          | <br>NC = 1 | <br>NC = 1.000  | <br>NC = 1.000  | <br>NC = 0.9946                        | <br>NC = 0.9592                        | <br>Some errors in recovered stego-key | <br>some errors in recovered stego-key |
| $Sb_2 = 3$          | <br>NC = 1 | <br>NC = 1.000  | <br>NC = 1.000  | <br>NC = 1.000                         | <br>NC = 0.9786                        | <br>NC = 0.9622                        | <br>Some errors in recovered stego-key |
| $Sb_2 = 4$          | <br>NC = 1 | <br>NC = 1.000  | <br>NC = 1.000  | <br>NC = 1.000                         | <br>NC = 0.9944                        | <br>NC = 9802                          | <br>NC = 0.9667                        |

where,

- $M$  and  $M'$  : The original and recovered secret message
- $Q_1$  : The number of rows
- $Q_2$  : The number of pixels in each row for an image matrix

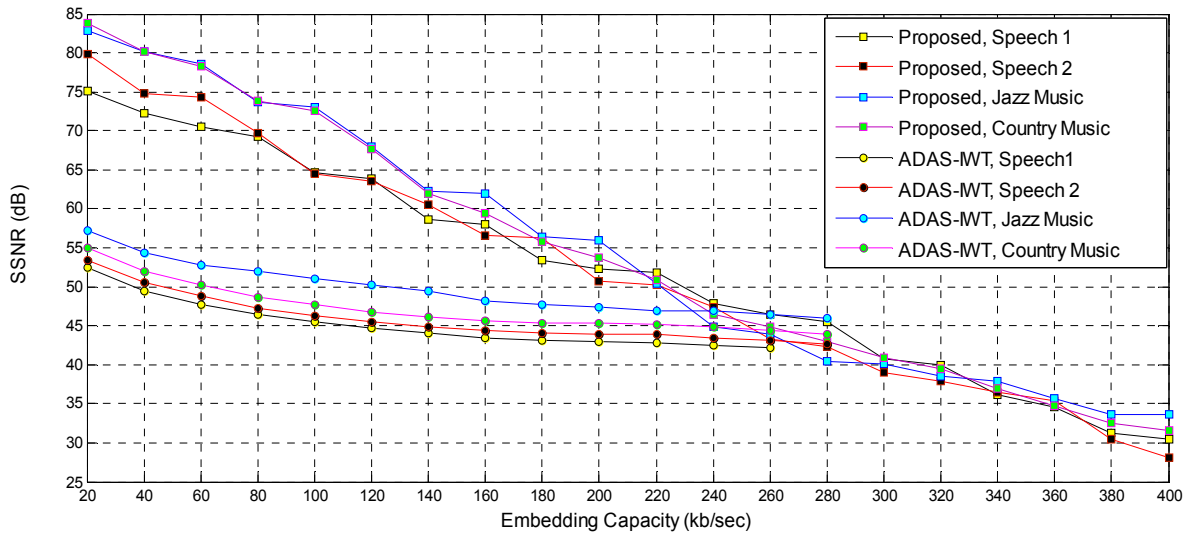
The primary robustness required for a steganographic system is against additive channel noise. In the robustness experiments of our proposed system, we have performed two groups of tests. The first group is shown in Fig. 3 for tests with a scaling factor ( $s = 4096$ ) and without scaling ( $s = 0$ ). In these tests, we have used the recorded speech as an audio cover signal, with resolution of 16 bits/sample, at 44100 Hz sampling frequency and duration time about 5 sec. Also we have utilized an image that has size  $(256 \times 256)$  pixels as a secret message. The two parts of Fig. 3 show the relation between normalized cross correlation ( $NC(M, M')$ ) and the channel AWGN, for various values of  $sb_2$ . In addition, the maximum capacity with minimum perceptual transparency (35 dB) for each case of tests is calculated and provided as the legends within the sub-figures. In these figures, the end point of each curve means the worst SNR value of AWGN, where, the embedded stego-key can be recovered without errors. The stego-key can be retrieved without errors if the values of  $NC$  are above 0.95 and there is no guarantee, if the value of  $NC$  is less than 0.95. Generally, the robustness of the stego-key can be improved by increasing the level depth of the LWT decomposition and choosing the last details component for stego-key embedding, so as to maintain of key bits energy. However, this is not necessary, as in all the tests conducted that have  $sb_2 = 3$  and more, the values of  $NC$  are very high and the secret message, which has transmitted over very bad channel (channel with  $SNR = 20$  dB or less) can successfully retrieved without errors.

Table 1 provides the second group of tests for the robustness. In these tests, we have embedded the image of a man's face ( $70 \times 70$  pixels) in the speech signal

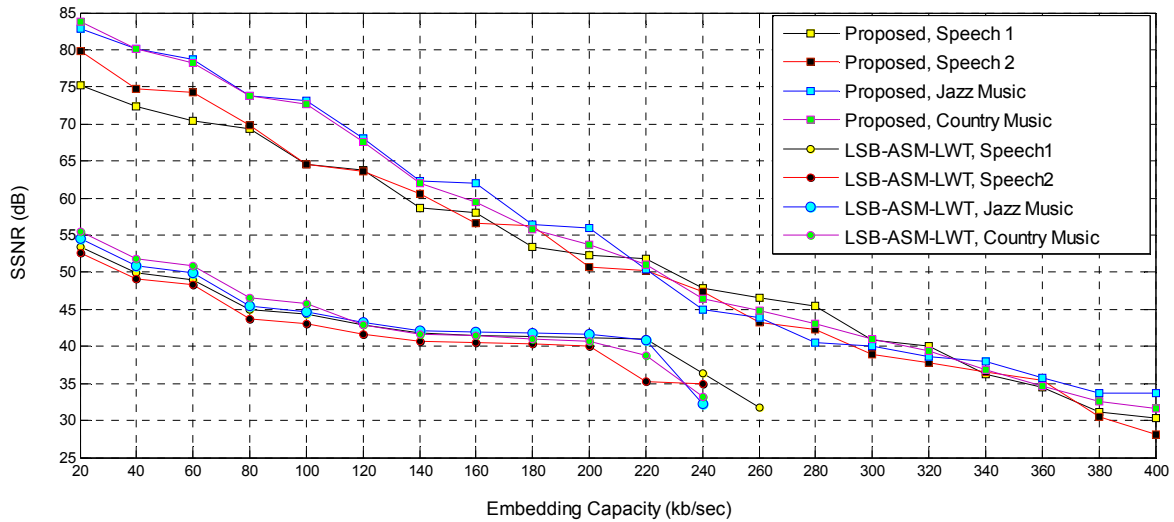
sourced from a female speaker. Table 1 shows the recovered image, after adding different SNR values of channel WGN for  $sb_2 = 0$  until  $sb_2 = 4$ . The results show that, the proposed approach has high resistivity against AWGN, when  $sb_2$  has values equal to 3 or higher.

**Comparisons of the proposed approach with some other related works:** This sub-section compares our proposed approach with some of the latest audio steganography algorithms that utilize DWT. These comparisons focus on the related DWT-based algorithms, which have high embedding capacity and perceptual quality as that discussed in related work section.

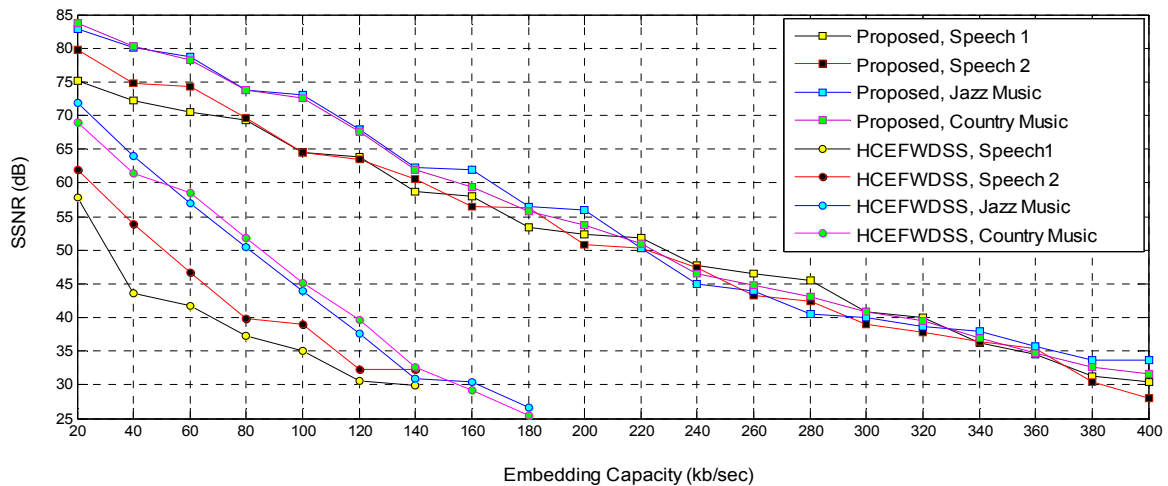
An Adaptive Digital Audio Steganography based on Integer Wavelet Transform (ADAS-IWT) have been proposed by Delforouzi and Pooyan (2006), adopts the threshold hearing to determine the number of embedding bits for each integer wavelet coefficient of the cover audio signal. Pooyan and Delforouzi (2007) have proposed LSB-based Audio Steganography Method, based on the Lifting Wavelet Transform (LSB-ASM-LWT). It is almost similar to the ADAS-IWT in terms of the number of embedded bit into each audio-host coefficient, which depends on the hearing threshold, except using five levels of packet LWT instead the integer domain of the convolution wavelet transform. Shahreza and Shalmani (2008) have proposed a High Capacity Error Free Wavelet Domain Speech Steganography algorithm (HCEFWDSS), where the message data is embedded in the LSBs of details coefficients, according to the coefficients values. Shahadi and Jidin (2011) have proposed High Capacity and Inaudible Audio Steganography Scheme (HCIASS) with two key steps to embed the message data. First, the strength of the details coefficients of L-level WPT are computed and in the second step, the result of comparison between blocks of message bits and contents of LSBs bits are used to determine positions for message data insertion.



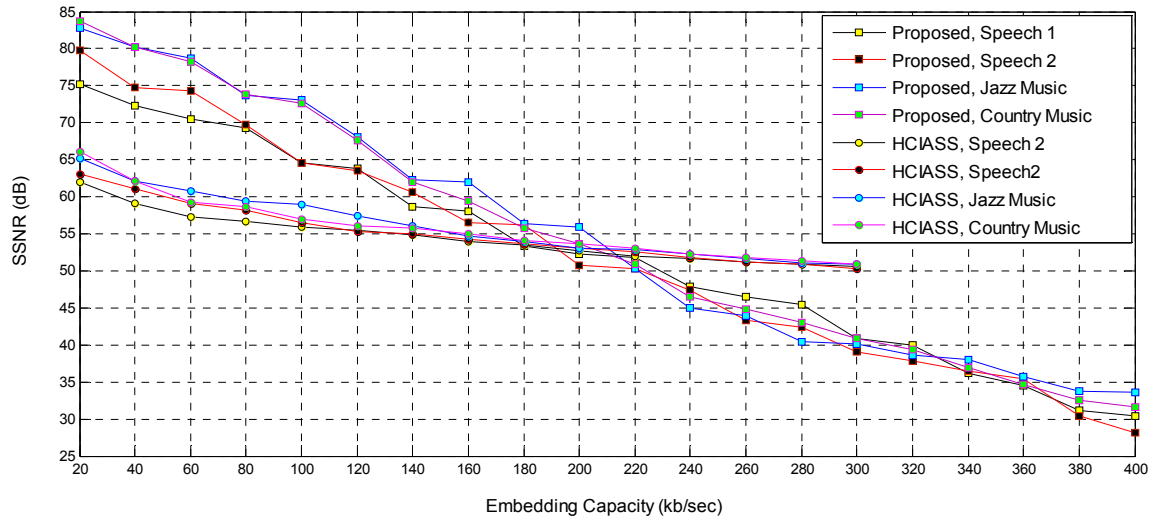
(a) Comparison Between Proposed and ADAS-IWT Approaches



(b) Comparison Between Proposed and LSB-ASM-LWT Approaches



(c) Comparison Between Proposed and HCEFWDSS Approaches



(d) Comparison Between Proposed and HCIASS Approaches

Fig. 4: Comparison of our proposed with the other four methods in term of perceptual quality vs. embedding capacity

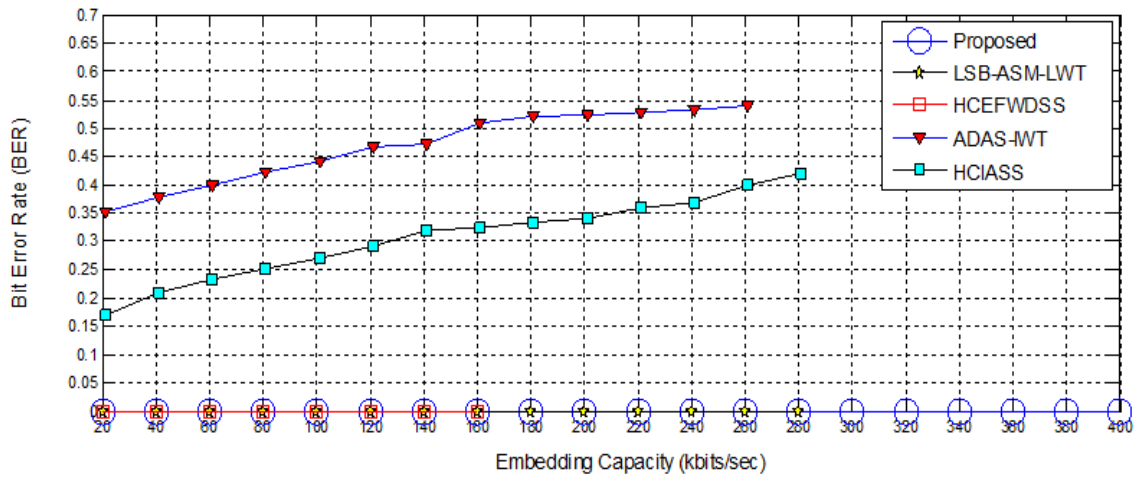


Fig. 5: Comparison of our proposed with the other four methods in term of BER vs. embedding capacity after saving stego-signals at a 16 bits/sample WAV files

In order to evaluate our algorithm with the above four algorithms, we have implemented all of them in MATLAB. The result of the tests for all the algorithms, in terms of embedding capacity versus the perceptual transparency, embedding capacity versus the bit error rate (BER, where BER is the ratio of error retrieved bits to the total numbers of retrieved message-bits) and robustness against the AWGN are shown in Fig. 4 to 6, for each respective performance measure of a steganography.

Figure 4 demonstrates perceptual transparency (in terms of *SNR* by *dB*) versus the embedding capacities tests. In these experiments, four audio signals (2 speech and 2 music signals) have been used to compare each one of the four algorithms with ours, as depicted in each individual sub-figures (Fig. 4a to d). The results

show superiority of our proposed method over all the other four methods in the embedding capacity, which can be achieved with excellent perceptual quality.

In Fig. 5 experiments have been conducted to find the Bit Error Rate (BER) in the retrieved data, after saving the audio stego signals as a WAV files at 16 bits/sample and then we have used them as an input signals to the recovery algorithms. The experimental tests have been conducted for all four methods and ours as well. The results show that, our proposed method, as well as the LSB-ASM-LWT and HCEFWDSS have zero BER ( $BER = 0$ ) for all embedding capacities. While, the other two (ADAS-IWT and HCIASS) have large values of BER, especially, at the high embedding capacities (above 30% at 200 kbps). In these tests, we have found that, all the three LWT methods that do not

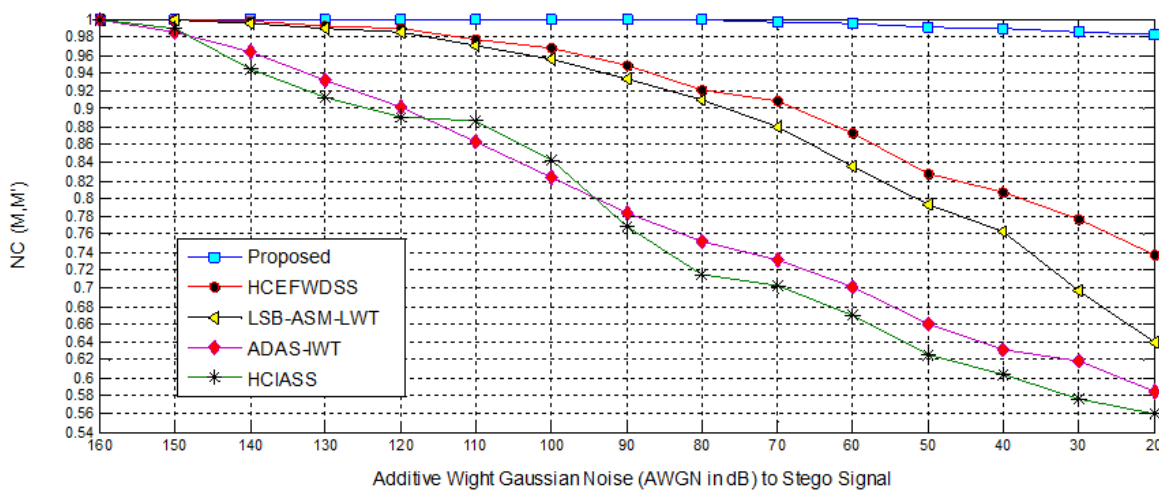


Fig. 6: Comparison between our proposed and the other four algorithms in term of robustness against AWGN

employ the multiplication scale, have retrieved data without any errors, whereas the other methods that use the multiplication scale and data conversion, have high distortion in their retrieved data.

In Fig. 6, experiments have been performed to test robustness against the AWGN for each individual five algorithms. All experimental tests have achieved embedding capacity about 130 kb/sec for all the methods. As for our algorithm, we have chosen  $sb_2 = 4$ . The tests evaluated different values of AWGN starting with the small additive channel noise ( $SNR = 160$  dB) to high channel noise ( $SNR = 20$  dB) and in each case, the cross correlation between original embedded message and retrieved message has been calculated. The results in Fig. 6 demonstrate the superiority of our proposed method, over all the other four methods.

### CONCLUSION

We have presented a novel Int2Int LWT based audio steganography. The approach attains positions for data embedding based on WBM between message blocks and extracted cover blocks to satisfy high embedding capacity and enhanced security. The proposed embedding technique reduces the errors between cover and stego signals, thus the output transparency is maintained. The Int2Int LWT upholds the robustness of the secret message, as data scaling and rounding is no longer required and this enables retrieval of the embedding data without any distortion even after adding noise to stego signal. Moreover,  $sb_2$  factor is used to improve the immunity of embedding data against the AWGN by increasing the depth of embedding positions. The experimental results have shown that, the embedding capacity can reach up to 340 kbps with excellent perceptual quality for output stego-audio signals.

### REFERENCES

Bender, W., D. Gruhl, N. Morimoto and A. Lu, 1996. Techniques for data hiding. *IBM Syst. J.*, 35(3-4): 313-336.

Cheddad, A., J. Condell, K. Curran and M.C. Kevitt, 2010. Digital image steganography: Survey and analysis of current methods. *Signal Process.*, 90(3): 727-752.

Cvejic, N., 2004. Algorithms for audio watermarking and steganography. Ph.D. Thesis, Department of Electrical and Information Engineering, Information Processing Laboratory, University of Oulu, Oulu, Finland.

Cvejic, N. and T. Seppanen, 2002. A wavelet domain LSB insertion algorithm for high capacity audio steganography. *Proceeding of 10th IEEE Digital Signal Processing Workshop and 2nd Signal Processing Education Workshop*, pp: 53-55.

Cvejic, N. and T. Seppanen, 2004a. Reduced distortion bit-modification for LSB audio steganography. *Proceeding of 7th International Conference on Signal Processing Proceedings (ICSP)*, Beijing, China, pp: 2320-2323.

Cvejic, N. and T. Seppanen, 2004b. Channel capacity of high bit rate audio data hiding algorithms in diverse transform domains. *Proceeding of International Symposium on Communications and Information Technology (ISCIT)*, pp: 84- 88.

Delforouzi, A. and M. Pooyan, 2006. Adaptive digital audio steganography based on integer wavelet transform. *Circ. Syst. Signal Pr. J.*, 27(2): 247-259.

Dewine, S. and J. Comelis, 1997. Lossless integer wavelet transform. *IEEE Signal Proc. Let.*, 4(6): 158-160.

Djebbar, F., B. Ayad, H. Hamam and K. Abed-Meraim, 2011. A view on latest audio steganography techniques. *Proceeding of IEEE International Conference of Innovations in Information Technology (IIT)*, pp: 409-414.

- Ercelebi, E. and L. Batakc, 2009. Audio watermarking scheme based on embedding strategy in low frequency components with a binary image. *Dig. Signal Process.*, 19(2): 265-277.
- Geiger, R., Y. Yokotani and G. Schuller, 2006. Audio data hiding with high data rates based on intMDCT. *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing*. Toulouse, pp: 205-208.
- Huang, X., Y. Abe and I. Echizen, 2010. Capacity adaptive synchronized acoustic steganography scheme. *J. Inform. Hiding Multimedia Signal Process.*, 1(2):72-90.
- Lee, C. and C. Ko, 2011. Short-term load forecasting using lifting scheme and ARIMA models. *Expert Syst. Appl.*, 38(5): 5902-5911.
- Lei, B., I. Soon, F. Zhou, Z. Li and H. Lei, 2012. A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition. *Signal Process.*, 92(9): 1985-2001.
- Nissar, A. and A. Mir, 2010. Classification of steganalysis techniques: A study. *Digit. Signal Process.*, 20(6): 1758-1770.
- Pooyan, M. and A. Delforouzi, 2007. LSB-based audio steganography method based on lifting wavelet transform. *Proceeding of 7th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT'07)*, pp: 600-603.
- Shahadi, H.I. and R. Jidin, 2011. High capacity and in audibility Audio steganography scheme. *Proceeding of 7th International Conference on Information Assurance and Security (IAS)*, pp: 104-109.
- Shahreza, S. and M. Shalmani, 2007. Adaptive wavelet domain audio steganography with high capacity and low error rate. *Proceeding of IEEE International Conference on Information and Emerging Technologies (ICIET)*, pp: 25-29.
- Shahreza, S. and M. Shalmani, 2008. High capacity error free wavelet domain speech steganography. *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp: 1729-1732.
- Wang, H. and S. Wang, 2004. Cyber warfare: Steganography vs. steganalysis. *Commun. ACM*, 47(10): 76-82.
- Wang, J., R. Healy and J. Timoney, 2011. A robust audio watermarking scheme based on reduced singular value decomposition and distortion removal. *Signal Process.*, 91(1): 1693-1708.