***MP3 Audio Steganography with Improved Capacity and Robustness***

*Hemalatha S, Renuka A, Shamathmika*

*Department of Computer Science and Engineering*

*Manipal Institute of Technology, Manipal University, Manipal, India*

[*hema.shama@manipal.edu*](mailto:hema.shama@manipal.edu)*,* [*renuka.prabhu@manipal.edu*](mailto:renuka.prabhu@manipal.edu)*,* [*shamathmika76@gmail.com*](mailto:shamathmika76@gmail.com)

*Abstract*— Developments in inter networking made the digital transmission simple but at the same time posed serious threats to secure data transmission. Steganography is one of the methods used for secret communication. It is a means of furtive transmission which hides the existence of the secret information. The carrier such as an image, an audio, or a video is used to hide the secret information. Audio files are common and existing abundantly. One of the properties being excessively used for concealing the secret information in audio files is that they have large space. In audio steganography, the carrier is an audio and the secret information can be a text file, an image, or an audio. In this paper, an MP3 audio steganography is proposed with improved capacity and robustness. Signal to Noise Ratio and Squared Pearson Correlation Coefficient are the metrics used for measuring the performance of the technique.

Keywords— Steganography; MP3; Capacity; Robustness; Audio

# Introduction

MP3 audio steganography is challenging because of its compressed form. The secret information should be retained under compression. Usually, the audio files are stored in a compressed form and the most commonly used form is MP3. When the audio frames are used for steganography appropriate frames should be selected to hide the secret information so that during compression the secret information is not lost [1], [2].

An MP3 file consists of several frames that comprise a header and a data block. The header is 4-byte long and the beginning of a valid frame is identified by a sync word in the header. The first eleven bits (first twelve bits in MPEG version 2.5) of a frame header are called frame sync that are always set. The checksum, if exists, follows the frame header which is 16-bit long. Next field is a 32-byte side information segment (17-byte if the MP3 file is mono), followed by a main data segment (length varies). The file may begin and end with tags that provide the Title, Artist, Year, Genre and other great information.

The main data section in a particular frame, contains two pieces: granule1 and granule2. If MP3 file is of mono type then each granule provides data for one channel and if it is of stereo type then each granule provides data for two channels. Between the frames there may be space that does not contain any specific information, called ancillary data [2].

If the information is hidden in the main data segment there is no guaranty that it is not lost during encoding. It is not possible to predict accurately, the loss during compression. It depends upon the encoder used and the compression algorithm involved. Therefore, data frames are not safe for steganography, if the stego signal has to withstand compression. Thus, the only available portion is the header. In the header also the possible fields that can be used for steganography are padding byte, unused header bits such as private bit, copyright bit, original bit, mode extension bits and emphasis bits. Mode extension bits can be used if the channel mode is not set to stereo mode. Similarly, overwriting emphasis bits produces distortion on some audio players. Another field that can be used to hide information is ancillary data. Thus, the number of bits per frame that can be used for steganography is very less.

# Literature Review

Most of the data hiding methods for MP3 audio that have been proposed in the literature have low embedding capacity. For MP3 audio, MP3Stego is the most widely used tool among a few stego tools viz., MP3Stego, UnderMP3Cover, and MP3Stegz. It has been proved that MP3Stego detection is possible and the continuity of the coefficient distribution in the adjacent frames is affected by MP3Stego. Later, the same authors proposed an MP3 steganography method by exploiting the rule of window switching during encoding [3], which exhibits very low capacity. The challenges of MP3 steganography and the potential hiding methods in MP3 files are explained in detail in [1], [2]. In MP3 files, the data may be hidden before compression, during compression, or after compression. If the embedding is done before compression, the hidden data may be lost in the compression process. Data hiding can be combined with the various compression steps, but it is very complex and the capacity is very less. Even in the post compression embedding process, the possibilities of bit modification are very few. Watermarking algorithms in MP3 files robust to compression are proposed in [4], [5]. The watermark is embedded during encoding by manipulating the redundant bits in the frames. S. Rekik et al., proposed audio in audio steganography technique and used Wavelet and Fourier transforms to hide the secret audio [6]. The low-pass spectral proprieties of the high-frequency regions of the cover audio signal are exploited to hide the secret audio signal.

# Proposed Method

To protect the secret information against MP3 compression and to get good capacity, the secret information is hidden in the main data in duplicates. To overcome data loss, the audio samples with more than 50% of the maximum sample value are selected for hiding and the data is hidden in duplicates. While extracting majority evaluation is done to approximate the hidden data. In MP3, samples are stored in single precision (32-bit) floating-point format. While reading, the samples are read in their native format and the data is hidden in the significand field. The sample values range between -1 and +1.

## Embedding Procedure

Up to 5 bits per sample can be hidden without any bit error. This improves the capacity. The 3rd, 4th, 5th, 6th and 7th bit of the significand field from the most significant bit side are replaced by the secret bits. The positional weights of these bits add up to less than 0.25. Therefore, even if all the bits are modified to one, then the value will not be increased to more than one. Since only the sample values with more than 50% of the maximum value (i.e. more than 0.5) are considered for steganography, the 1st bit of the significand is 1. Dynamic adjustment is done so that the sample value is changed by only minimum value. To achieve this, if all the bits to be modified are changed from 0 to 1, then 2nd bit of the significand is made 0. If all the bits to be modified are changed from 1 to 0, then 2nd bit of the significand is made 1. Two extreme cases are illustrated in the example below.

If the sample value is 0.87890619, its 32-bit floating-point representation is

0 01111110 11**00000**1111111111111111; bits in bold are the bits to modify.

Now suppose all the bits to be modified are changed to 1 (one extreme case),

0 01111110 11**11111**1111111111111111; value increases to 0.99993896.

Then the 2nd bit is changed to 0 so that the value becomes 0.87108875

Similarly, if the sample value is 0.87500000, its 32-bit floating-point representation is

0 01111110 11**00000**0000000000000000; bits in bold are the bits to modify.

Now suppose all the bits to be modified are changed to 1 (other extreme case),

0 01111110 11**11111**0000000000000000; value increases to 0.99609375.

Then the 2nd bit is changed to 0 so that the value becomes 0.87109375

Similarly it can be shown that the change in the sample value is minimum by making the 2nd bit of the significand to 1 when all the bits to be modified are changed from 1 to 0.

The secret audio is transformed to time frequency domain using Integer Wavelet Transform (IWT). This produces approximation and detail coefficients of the secret audio. IWT can be applied using lifting wavelets [7], [8], [9]. Approximation coefficients have significant features and using only these coefficients, entire secret audio can be reconstructed. So only approximation coefficients are hidden in the cover. In the extraction process, detailed coefficients are taken as zeros during inverse transformation. Hiding the frequency coefficients of the secret audio improves the security because it is difficult to detect for the eavesdroppers

The technique is referred as MP3AS and the algorithms are given in Figs. 1 and 3. Fig. 2 gives the block diagram representation.

|  |
| --- |
| **Algorithm: Embed-(MP3AS).**Embeds the secret audio in the cover audio.  **Input:**   * Cover audio *C.mp3* * Secret audio *S.mp3*   **Output:**Stego audio, *G.mp3.*  **Method:**   1. Read cover audio *C* and secret audio *S* and convert the secret audio to .wav format 2. *LS* 🡨**liftwave** *( ‘haar’, 'Int2Int' )* 3. *[CAs, CDs]* 🡨 **lwt***(double(S),LS)* // Obtain IWT of the secret audio to get approximation (*CAs*) and detail (*CDs*) coefficients 4. *CAs\_bin* 🡨 **dectobin***(CAs)* 5. *N* 🡨 No. of bits in *CAs\_bin* 6. *Nbits* 🡨 **dectobin***(N)* // Convert *CAs* coefficients to binary and obtain the no. of secret bits *N*. Convert *N* to binary 7. Copy *Nbits* to an array *D*. Duplicate *CAs\_bin* four times and append to *D*. 8. **for** i= 1 to length(*C*) do 9. **while**(N>0) do 10. **if** (*C*(i) >0.5 ) then 11. *B* 🡨 *floattobin(C(i))* 12. *B(3:7)* 🡨secretbits from *D* 13. *C(i)* 🡨 *bintofloat(B)* 14. ***end if*** 15. *N* 🡨 *N*-4 16. **end while** 17. **end for** // Hidethe secret bit array D in the 3rd, 4th, 5th, 6th, and 7th bits of the significand of the cover samples whose value is greater than 0.5. 18. *audiowrite(C, ‘G.mp3’)* // Write stego audio file 19. **return**stego audio *G*. |

Fig. 1. MP3AS embedding algorithm

Approximate coefficients

Convert to binary and duplicate

Secret audio

Detailed coefficients

IWT

Cover audio

Convert cover samples with values greater than 0.5 to binary

Replace 3rd, 4th, 5th, 6th, and 7th bits of the significand by the secret bits

Robust stego audio

Test the robustness

Stego audio with optimal capacity and security

Fig. 2. Block diagram – MP3AS (Embedding)

## Extracting Procedure

The hidden secret audio data can be extracted by performing the operations of embedding in the reverse order. The algorithm is shown in Fig. 3.

|  |
| --- |
| **Algorithm: Extract-(MP3AS).**Extracts the secret audio from the stego audio.  **Input:** Stego audio *G.mp3*  **Output:**Secret audio *Sextracted.mp3.*  **Method:**   1. Read stego audio *G* 2. Convert elements of *G* with value greater than 0.5, to 32 bit floating point binary. Extract 3rd, 4th, 5th, 6th and 7th bits of significand. The first *N* bits will give the no. of secret bits of *CAs*. 3. *CAs\_bits* 🡨**bintodec***(N)* 4. Extract four times *CAs\_bits* from the elements of *G*. 5. Perform the majority evaluation in the four groups of the CAs\_bits and obtain the coefficients of CAs by converting to decimal. 6. *Sextracted* 🡨**ilwt** *(CAs, 0, LS)* // Obtain the secret audio by taking the inverse transform considering detailed coefficients as zero. 7. **return**secret audio *Sextracted.mp3. //* Write to a file and return |

Fig. 3. MP3AS extracting algorithm

# Results and Discussion

To write MP3 files LAME 3.99.5 encoder tool is used. Results show that high capacity with reasonably good performance metrics are obtained compared to the existing MP3 steganography techniques. If the header information is used for hiding the information, only 5 to 6 bits per frame can be hidden. Compared to that, the proposed technique gives very high capacity with good security. Table I shows the performance metrics for constant cover sample and different secret bits per sample. Quality of stego audio is analyzed using SNR and SPCC.

SNR indicates the deviation between the original and the modified signals. It is given by (1) [10].

SNR = 10 (1)

where

*x* and *y* are the original and the modified signals respectively.

*N* is the number of signal samples

*xi* is the value of the ith sample in x

*yi* is the value of the ith sample in y.

A larger SNR value indicates a better quality. The recommended SNR for audio signal is above 30dB.

SPCC is a metric used for measuring the quality of an audio signal which depends on the correlation of samples. The high value of SPCC indicates good quality. Its range is between 0 and 1and it is given by (2) [10].

(2)

where

*x* and *y* are the input and the output signals respectively.

and are the average of the input and the output signals respectively.

Table I. SNR and SPCC values for different secret sample sizes (MP3AS)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cover audio samples | No. of secret bits per sample | SNR in dB | | SPCC | |
| Stego | Extracted Secret | Stego | Extracted Secret |
| 25530623 out of which 159496 with value more than 0.5 | 3 (29905 samples) | 70 | 32 | 0.9725 | 0.8041 |
| 4 (39874 samples) | 67 | 31 | 0.9225 | 0.795 |
| 5 (49842 samples) | 62 | 30 | 0.9000 | 0.7823 |

This table shows that with 5 secret bits per cover sample the metrics of the stego and the extracted secret audio are well within the range. When it is increased more than this, the secret audio cannot be extracted properly.

## Performance Against Attacks

The common attacks on stego audio signals are the compression, filtering, resampling, normalization etc. [11]. The cover audio samples with the secret audio bits are written as MP3 audio file and it is possible to retrieve the secret audio signal. Therefore, this technique withstands MP3 compression. Table II shows the performance against other common attacks.

Table II. Performance against common attacks (MP3AS)

|  |  |  |  |
| --- | --- | --- | --- |
| Attack Type | Extracted audio performance metrics | | |
| SNR in dB | SPCC | BER (%) |
| Cropping | 25 | 0.7581 | 12 |
| Low pass filtering | 29 | 0.8030 | 8.5 |
| High pass filtering | 28 | 0.8015 | 9 |
| Resampling | 30 | 0.8875 | 6 |
| Normalization | 32 | 0.8876 | 5 |

Bit Error Rate (BER) is the percentage of the bit error. It is found that the BER gives the appropriate measurement when the technique is tested for robustness. If the stego audio file is cropped at the beginning and ending, it will not affect the hidden data because MP3 files have some amount of silence period at both the ends. It is cropped 10% in time (at the beginning and ending) and by 1dB in amplitude. Then the bit error rate of the extracted secret audio is 12%. If the cropping is increased, the secret audio cannot be retrieved. The secret audio can be extracted without much bit error when filtering (both low pass and high pass) with 1 kHz cut off frequency is applied. Resampling with a difference in the sampling rate of 0.5 kHz and normalization up to plus or minus 10% gain will not affect the hidden data. Outside this range, the extracted audio will be distorted.

## Comparison with the existing techniques

Table III compares the performance metrics of the proposed audio steganography technique with the other existing related techniques. The comparison is done taking the percentage payload capacity as the reference. Percentage payload capacity is taken as the percentage of the cover signal used for embedding. In some of the papers, the SPCC is not calculated, so it is entered as NI (Not Implemented) in the table. The proposed MP3 steganography technique, MP3AS is compared with the method of Y. Diqun et al. [60] that hides message bits in MP3 audio, since the techniques of hiding audio in MP3 audio are not found in the literature. In this case, the proposed method achieves 30.9% improvement in the capacity with 4dB increase in SNR. Here the performance is improved because the secret information is hidden in the data frames using floating point representation for data samples. 5 most significant mantissa bits per sample are used for information hiding and hence the capacity is improved significantly. Since dynamic adjustment is done to keep the bit changes minimum, SNR also increases.

Table III. Performance comparison of the proposed techniques with that of other related published work

|  |  |  |  |
| --- | --- | --- | --- |
| Technique | Payload capacity in % | SNR in dB | SPCC |
| Y. Diqun et.al. [3] | 0.1 | 58 | NI |
| Proposed (MP3AS) | 31 | 62 | 0.9 |

# Conclusion

Audio signals can be efficiently used for steganography because of its ease of storage and transmission. This paper proposes an MP3 audio steganography technique which is challenging since MP3 format is complex. The results of this proposal outperforms compared to existing techniques in terms of capacity and security. MATLAB 8.2 is used for implementation.

# References

|  |  |
| --- | --- |
| [1] | A. Mohammed Salem, I. Subariah, S. Ghazali, Z. Akram and A. Adamu, "Exploring the Challenges of MP3 Audio Steganography," in *Proceedings of the International Conference on Advanced Computer Science Applications and Technologies*, 2013. |
| [2] | Z. Mikhail, L. William and C. Robert, "MP3 Files as a Steganography Medium," in *Proceedings of the Second annual conference on Research in information technology*, Orlando, Florida, USA, 2013. |
| [3] | Y. Diqun, W. Rangding, Y. Xianmin and Z. Jie, "Steganography for MP3 Audio by Exploiting the Rule of Window Switching," *Computers and Security ,* vol. 31, pp. 704-716, 2012. |
| [4] | C. Bingwei and Z. Jiying, "An Adaptive and Secure Audio Watermarking Algorithm Robust to MP3 Compression," in *Proceedings of the International Instrumentation and Measurement Technology Conference*, Singapore, 2009. |
| [5] | Y. Bailong, W. Penghui, J. Yaque and M. Jing, "Lossless and Secure Watermarking Scheme in MP3 Audio by Modifying Redundant Bit in the Frames," in *Proceedings of the International Conference on Information Management, Innovation Management and Industrial Engineering*, 2013. |
| [6] | S. Rekik, D. Guerchi, S. Selouani and H. Hamam, "Speech Steganography using Wavelet and Fourier Transforms," *EURASIP Journal on Audio, Speech, and Music Processing,* 2012. |
| [7] | L. Baiying, S. Ing Yann, Z. Feng, L. Zhen and L. Haijun, "A Robust Audio Watermarking Scheme based on Lifting Wavelet Transform and Singular Value Decomposition," *Signal Processing,,* vol. 92, pp. 1985-2001, 2012. |
| [8] | D. Fatiha, A. Beghdad, M. Karim Abed and H. Hamam, "Comparative Study of Digital Audio Steganography Techniques," *EURASIP Journal on Audio, Speech, and Music Processing,* 2012. |
| [9] | S. Hemalatha, A. U. Dinesh and A. Renuka, "Audio data hiding technique using integer wavelet transform," *International Journal of Electronic Security and Digital Forensics,* vol. 8, no. 2, pp. 131-146, 2016. |
| [10] | B. L. Dora M. and M. A. Juan M., "Highly Transparent Steganography Model of Speech Signals using Efficient Wavelet Masking," *Expert Systems with Applications,* vol. 39, no. 10, pp. 9141-9149, 2012. |
| [11] | M. Nutzinger, "Real-time Attacks on Audio Steganography," *Journal of Information Hiding and Multimedia Signal Processing,* vol. 3, no. 1, pp. 47-65., 2012. |