Jurnal
Teknologi

Full Paper

A NEW DATA EMBEDDING METHOD FOR MPEG LAYER III AUDIO STEGANOGRAPHY

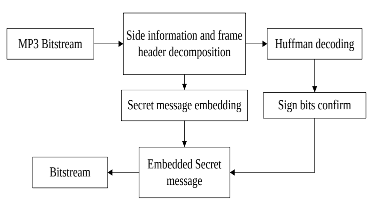
Mohsen Bazyar, Rubita Sudirman*

Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

Article history
Received
30 October 2015
Received in revised form
3 March 2016
Accepted
28 March 2016

*Corresponding author
rubita@utm.my

Graphical abstract



Abstract

A new method of MP3 steganography is proposed with emphasis on increasing the steganography capacity of the carrier medium. This paper proposes a data embedding algorithm to hide more information for compressed bitstream of MP3 audio files. The sign bits of Huffman codes are selected as the stego-object according to the Huffman coding characteristic in region of Count1. Embedding process does not require the main MP3 audio file during the extraction of hidden message and the size of MP3 file cannot be changed in this step. Our proposed method caused much higher information embedding capacity with lower computational complexity compared with MP3Stego tools. Experimental results show an excellent imperceptibility for the new algorithm.

Keywords: Information hiding, Huffman coding, MP3 Steganography, capacity

Abstrak

Kaedah baru MP3 steganography adalah dicadangkan dengan penekanan ke atas meningkatkan kapasiti steganography medium pembawa. Kertas ini mencadangkan algoritma data terbenam untuk menyembunyikan maklumat lanjut bagi aliran bit Mampat fail audio MP3. Bit menandakan Huffman kod dipilih sebagai stego-objek mengikut ciri rantau pengkodan Huffman Count1. Melakukan proses tidak memerlukan fail audio MP3 yang utama semasa pengekstrakan mesej tersembunyi dan saiz fail MP3 tidak boleh ditukar dalam langkah ini. Cadangan kaedah kami disebabkan maklumat lebih tinggi pemupukan keupayaan dengan kekompleksan pengiraan yang lebih rendah berbanding dengan alat-alat MP3Stego. Keputusan eksperimen menunjukkan imperceptibility yang sangat baik bagi algoritma yang baru.

Keywords: Maklumat bersembunyi, Pengkodan Huffman, MP3 Steganography, Kapasiti

1.0 INTRODUCTION

Nowadays, rapid development of Internet and digital information revolution causes significant changes in global society, ranging from the influence on the world economy to the people's daily communication. So, in order to transfer the data securely to the destination without any modifications, there are many approaches like cryptography and steganography. Information coding and cryptography is essential, but efficient privacy has been given by encryption and information hiding methods that can be misused for covering criminal activities. Therefore is important to develop tools and methods for forensic analysis [1].

Steganography and cryptography are normally connected together. Cryptography is effective in the usage of the key and the message is somehow coded. If it is sent insecurely, an attacker will notice it immediately and will try to decode it. However there is a steganography, which helps with the secure transfer of encoded messages. It codes a message inside of an audio or another multimedia file. If you see a steganographic audio, you will not recognize the secret message inside of audio file. And this is the point. Crackers will go through and will not pay attention to the message. Therefore it is necessary to have a method for its detection. To decode a message itself is another challenge, this thesis is aimed to reveal a secret message inside the audio. Steganography, as art of hiding information, has been known for over 2500 years. Back then steganography was mainly used for diplomatic, military and a very few people used it for personal purposes along with cryptography. Steganography as well as cryptography have a goal to secure transmitted information between the sender and the recipient, but both systems are used in a different way [2-3].

MPEG-I Layer 3, known as MP3, has generated a significant popularity for distributing digital music over the Internet. MP3 compresses digital music with high ratio while keeping high sound quality. However, copyright issue is raised because of illegal copy, redistribution and various malicious attacks. Digital watermarking is a technology that allows users to embed some imperceptible data into digital contents such as image, movie and audio data. Once a watermark is embedded into the original MP3 signal, it can be used to identify the copyright holder in order to prevent illegal copy and to verify the modification from the original content. Hiding capacity, perceptual transparency, and robustness are three important parameters which are known as 'the magic triangle' and use in designing steganography techniques [4-6]. Some methods attempt to be robust against various attacks like MPEG-1 layer III compression, whereas, achieving high hiding capacity is the goal of some other steganography techniques [7-8]. In recent years, Most of watermarking techniques which have been proposed for MP3 audio formats can be easily modified for the purpose of steganography. The goal of this study is to develop an application which can hide data in MP3 files and detect the presence of hidden messages in MP3 files using an appropriate analysis method.

Remaining of this paper is organized as follows. In the next section we describe the structure of MP3 file format with encoder and decoder of MPEG audio layer III. Then, we explain our proposed method, followed by discussing the experimental results.

2.0 MP3 BITSTREAM

MP3 audio files are composed of many parts, every one of them is called a frame. Each frame in the file is an encoded chunk of 1152 time domain audio samples. After that, these samples are divided into two granules with 576 samples. Each frame contains a header, ancillary data, audio data, and side information. The header frame is a frame with 32 bits long. For example, the side frame is 17 bytes long for MP3 mono audio files. The ancillary data and audio data have variable length, but the length of these frames cannot exceed the length available determined by the encoded bit rate [9].

MP3 encoded bitstream is divided into frames. The frame is a central concept when decoding the MP3 bit stream. Therefore, watermarking on MP3 frames is the major contribution of this thesis. An MP3 frame consists of five parts including header, CRC, side information, main data and ancillary data.

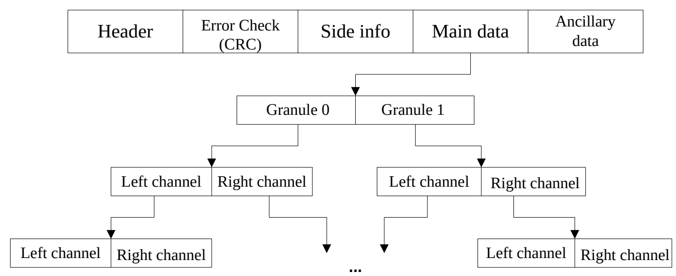


Figure 1 MP3 frame structure

Figure 1 shows MP3 frame structure. The size of each frame is usually fixed. A frame usually consists of 1 or 2 granules. Each granule is further divided into 32 subband blocks of 18 frequency lines. Therefore, in stereo audio, each frame includes two granules which are left and right granules and each granule is a 32 subband block of 18 frequency lines. For mono audio, a frame only includes one granule [10].

Therefore, in mono audio a frame is the same as a granule. In this research, the proposed MP3 watermarking algorithm is a generic method for MP3 frames. However, in order to easily present, the proposed watermarking algorithm is implemented on one granule in a frame. Therefore, for stereo audio, the new algorithm is implemented on its right granules in a frame; for mono audio, the proposed algorithm is implemented in its entire frame. The experimental results of the proposed watermarking algorithm are on the mono audio only. The size of an MP3 frame depends on the audio bit rate and sampling rate. Following formula calculates the total size in bytes for any given frame:

FrameSize = 144 x BitRate/'(SampleRate + Padding)

For instance, if encoding a file at 128 kbps, the original sample rate is 44.1 kHz, and no padding bit has been set, the total size of each frame will be 418 bytes (144x128000/(44100 + 0)). The following explains each of the MP3 frame components [11].

2.1 Header

For each frame, the size of the header is fixed. It is a 32-bit 0/1 stream, which includes the sync word and ancillary data for decoder.

2.2 Error Check

In MP3, the frame uses 16 bit CRC to do error check as an optional feature by users. This error check is used for checking transmission errors of the most sensitive data which is defined to be both side information and header by the standard. The whole frame will be distorted if these values are incorrect since an error in the main data only corrupts a part of the frame. A distorted frame also can be replaced or muted by the previous frame.

The side information part of the frame consists of information needed to decode the main data. The size depends on the encoded channel mode. Figure 2 shows the different parts of the side information [12].

Main data	Private bits	Scale factotor selectoin	Side info granule 0	Side info granule 1

Figure 2 MP3 frame side information

2.3 Main

Data MP3 frame main data are made up by granules. For mono audio, which has one channel, the frame main data include only one granule. For stereo audio, which has two channels, a frame main data include two granules. Figure 3 shows the MP3 frame format, subband blocks, frequency lines and granules. Each frame is divided into one (mono) or two (stereo) granules. Each granule contains 32 subbands, and each subband contains 18 frequency lines (MDCT values) [13].

2.4 Ancillary Data

In MP3 frame, ancillary data are optional and user definable. The number of ancillary bits equals the available number of bits in an audio frame minus the number of bits actually applied for header, audio data, and error check [14]. In Layer III, the number of ancillary bits corresponds to the distance between the end of the Huffman code bits and the location in the bitstream where the next main data beginning pointer points to.

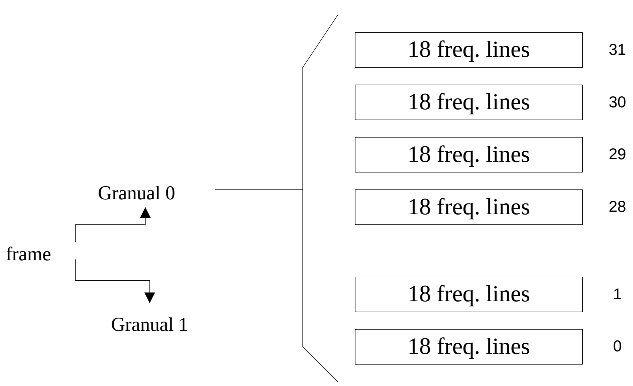


Figure 3 MP3 audio frame format, subband blocks, granules and frequency lines

3.0 MP3 AUDIO ENCODING

MP3 audio file which takes on smaller distortion and higher compressed ratio adopts the third layer of MPEG-I audio standard pattern. It has been the most popular audio file format in internet and very suitable for network audio transmission. Analysis filterbank, psychoacoustic model, quantization and bitstream formatting are four parts of MP3 encoding. The quantized coefficients adopt the lossless variable length of Huffman coding to improve compressed efficiency in MP3 encoder [15-17].

In Mp3 bitstream, Huffman coding is only applied to the quantized frequency lines. The Huffman coding is based on the assumption that frequency lines in the higher frequency regions have lower absolute values than frequency lines in the lower frequency regions, and that the absolute value of adjacent quantized frequency lines not differs significantly. In order to maximize the coding efficiency the quantized frequency lines are Huffman coded in smaller partitions. First each granule is partitioned into three partitions as shown in Figure 4. Starting from right, i.e. the higher frequencies, adjacent pairs of quantized frequency lines equal to zero are counted. This number is named rzero. Next adjacent quadruples of quantized frequency lines with absolute value less than or equal to 1 are counted, and the number is named count1. The remaining quantized frequency lines are paired and the number of pairs is called big_values which is further divided into two or three regions: region0, region1 and region2 [18-19].

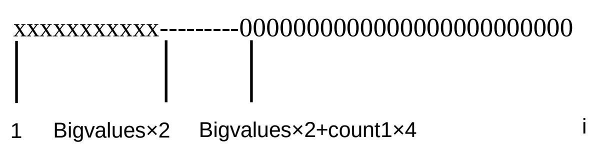


Figure 4 Organization of MP3 quantized values

4.0 THE DATA EMBEDDING ON BITSREAM OF MP3 AUDIO

As the mention before, the quantized coefficients based on the frequency scope are divided in three regions. Count1 region is located in intermediate frequency area where the coefficients value is much smaller consisting of -1, 0 or 1. It would be little effect on quality of MP3 audio if the coefficients are modified slightly in this region. For this reason, Count1 region is selected to hide secret message [20].

Figure 5 shows the Huffman coding stream structure. In Huffman coding, the frequency coefficients make up of each quaternary coefficient {x,y,v,w} and they gain their absolute value at first in Count1. The Huffman code is represented by Hcod and the sign bits of nonzero value of quaternary frequency coefficients are respectively expressed by Sign-x, Sign-y, Sign-v and Sign-w. The length of the sign bits changes from 0 to 4. By analyzing the format of huffman code, Hcod which is defined strictly represents a code word. Although, it is very difficult for Hcod to embed secret message, but there is no any redundancy in itself. So it is a good target which can be chosen to hide secret message since the sign bits which is followed by Hcod express the minus or plus character of frequency coefficients and there are no any correlations between Hcod and sight bits[21].

...	Hcod	Sign-x	Sign-y	Sign-v	Sign-w	...

Figure 5 The bitstream structure of Huffman coding in Count1

A random sequence which is called *m* is brought from 0 to 1 according to an input key. Extension of secret message around the whole MP3 bitstream using a density factor, α is ensured by the random sequence. The destiny factor range is changed from 0 to 1. It is more efficient for secret message to be hidden and dispersed when the value of α is smaller. But it could affect the capacity of data hiding if the value of α is too small. Therefore a suitable value of α is required in different tests. $S(i),\ i \in (1,\ldots,N\text{-}1)$ shows the sign bit of bitstream in MP3 audio, where *N* is the length of sign bits. The following formula embeds the binary secret message within the sign bit [22]:

$$S^{'}(i) = \begin{cases} S(i) & when \quad m(i) > \alpha \\ Zero[S(i)] + w(j) & when \quad m(i) \geq \alpha \end{cases} \quad (1)$$

S'(i) represents the sign bit passing through data embedding. *Zero()* is a function to let sign bit clear zero. This function can make the embedded message to replace with the sign bit when secret message is hidden. It also ensures that the MP3 audio bitstream length cannot be altered at the same time. Figure 6 shows the embedding data procedure within MP3 audio bitstream.

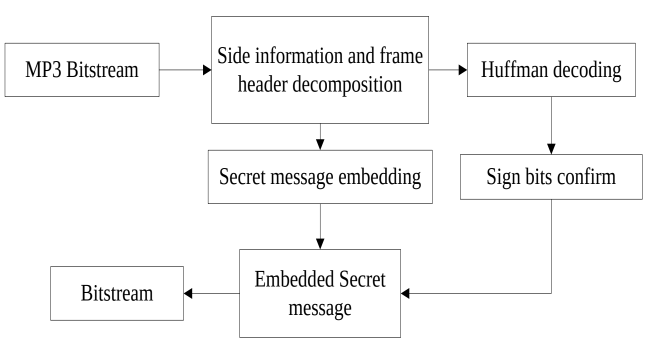


Figure 6 Secret message embedding

The following steps express the embedding steps procedure [23]:

- The secret information required to be pretreatment via being encrypted and scrambled to ensure that the hidden message is not able to be lost largely against the proposed method attacked.
- Reading a MP3 audio data frame, side information and header is parsed. The sign bits of Huffman codes in Count1 region could be confirmed by the Huffman and scale factor decoding.
- By formula (1), the secret message can be hidden to the sign bits after the sign bits are confirmed.
- The embedding data procedure would be achieved finally by repeating the second and third steps with audio data reading in next frame.

The extracting procedure is the same with embedding process when hidden message is extracted. Following formula is used to secret information extraction:

$$w(j) = S^{'}(i) \quad when \quad m(i) \geq \alpha \quad (2)$$

The main hidden message can be achieved after decryption. And it is clear that the proposed approach doesn't require main MP3 audio file and can obtain blind detection.

5.0 EXPERIMENTAL RESULTS

Different text files are selected as the secret message in this test. Four different type of music (Piano, Song, Pop and Jazz), are used to evaluate the proposed method performance with the following characteristics: time of music is 10s, 385 frames, mono channel and with sampling frequency of 44.1 KHz. Embedding capacity and imperceptibility are the most important goals of experiments to evaluate the performance of the proposed method. 64kbps, 128kbps, 192kbps, 256kbps and 320kbps are bitrates which used in these experiences.

5.1 The Data Embedding Capacity

All Huffman code sign bits in Count1 are noted by analyzing the bitstream of MP3 audio before embedding hidden message and the maximum information hiding capacity is acquired. The average maximum data capacity of each frame can be also obtained by the full number of MP3 frames at the same time. In this test, each MP3 file selects five bitrates. Table 1 shows the experimental results.

Table 1 Data hidden capacity for different bitrates

MP3 samples	The embedding bits average				bit/frame
	64 kbps	128 kbps	192 kbps	256 kbps	320 kbps
Piano	62.37	39.23	12.45	0.75	0.21
Song	59.60	43.38	4.37	0.08	0.01
Pop	57.35	31.22	9.11	1.45	0.44
Jazz	72.1	49.86	21.67	0.97	0.08

According to Table 1, the data hidden capacity is decreased with audio bitrates increase. However, the data embedding capacity is higher in lower audio bitrates. And the data embedding capacity is nearly close to zero when the bitrate is 320kbps. After MP3 audio encoding, the MP3 audio imperceptibility becomes much better with audio bitrate increase. On the other hand, the quantized is decreased due to this fact that the quantized precision requires to be much bigger. Therefore, the region of Count1 becomes less and less since the values of -1, 0 or 1 are small and quantized coefficients are much higher when the Huffman sign bits are also small. More acceptability for MP3 audio format in low bitrate is one of disadvantages of this proposed method. But the proposed approach still holds strong application in current network since MP3 audio usually adopts lower bitrate. Nevertheless, the embedding capacity for low bitrate, such as 64kbps or 128kbps can be 30 bits per frame. The proposed method capacity in comparison with the MP3Stego tools is much higher.

5.2 Perceptive Quality of Audio

To evaluate the objective quality of MP3 audio, PEAQ (Perceptual Evaluation of Audio Quality) is used in this study. PEAQ duo to simulating the human ear characteristic is very precise to audio quality test which is includes advanced and basic version [24]. The different between basic and advance model is that the first one works on higher real time but advanced model keeps higher accuracy. Basic model in used in this experiment. Objective difference grade (ODG) and model output variable (MOV) are the outputs of basic model of PEAQ. One variable which is called total noise to mask ratio (Total NMR) is selected from all MOVs that is present in [25-26]. The audio quality of this variable is better than others. Following formula defines Total NMR:

$$NMR_{tot}[n]10\log_{10}\frac{1}{N}\sum_{n}\left(\frac{1}{Z}\sum_{k=0}^{Z-1}\frac{P_{noise}[k,n]}{M[k,n]}\right) \quad (3)$$

where n is the number of current frame belongs to the total audio frame number which is shown with N. Masking patterns and noise patterns are defined as and respectively and Z is the frequency bands number. The ODG corresponds to subjective difference grade in the subjective domain. The audio quality is inversely proportional to Total NMR while directly proportional to the ODG.

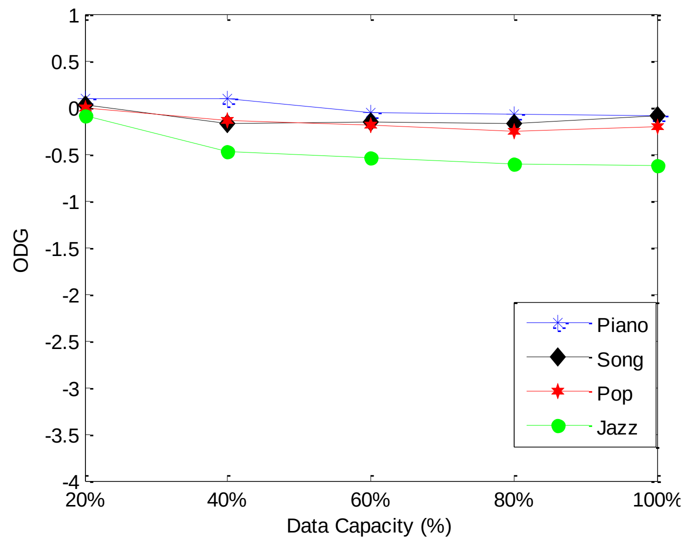


Figure 7 test of ODG using audio 64 kbps

MATLAB software is used to evaluate the quality of MP3 audio using PEAQ in this test for different MP3 audio bitrates (64kbps and 128kbps). Change of embedding capacity effects on quality of MP3 audio. Figure 7 and 8 show the values of ODG.

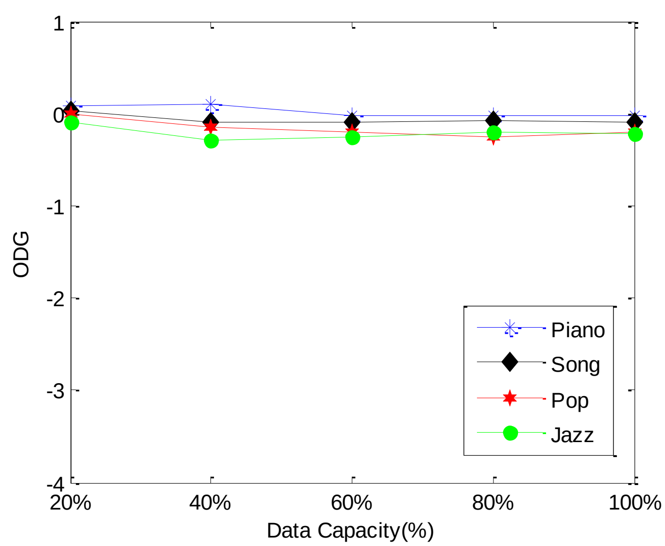


Figure 8 Test of ODG using audio 128 kbps

According to Figures 7 and 8 for the two ODG curves, the value of ODG becomes smaller with embedding capacity incensement. It is show that the distortion of MP3 file is increased with increasing the capacity of data embedding. On the other hand, it is clear that the whole values of ODG in Figure 7 in comparison with Figure 8 are smaller. The quality of MP3 audio in higher bitrate is better than one in lower bitrate after MP3 encoding. Therefore, the ODG value in 128 kbps is higher than one in 64 kbps that is between 0 and -1 for both Figures. It shows that the proposed algorithm keeps imperceptibility properly.

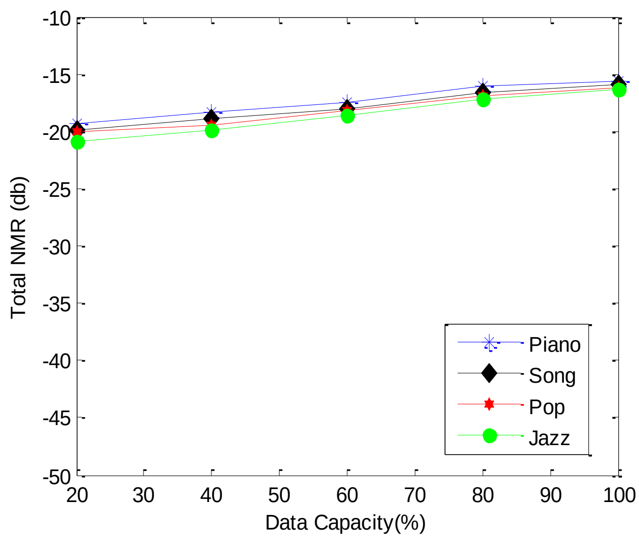


Figure 9 Test of total NMR using audio 64 kbps

However, as shown in Figures 9 and 10, the value of Total NMR is higher and higher with data embedding capacity increment. And this indicates that the audio perceptive quality is low.

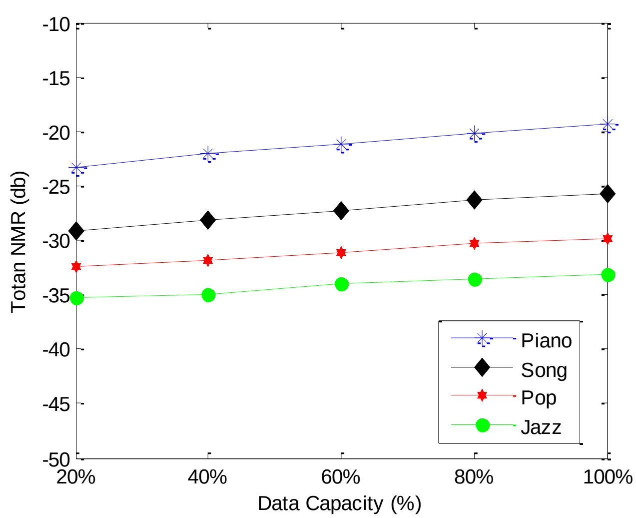


Figure 10 Test of total NMR using audio 128kbps

As shown in Figures 7 and 8, the result is similar as one reflected by ODG. It is presented that the audio perceptive quality change using ODG and Total NMR is the same to test the same audio.

6.0 CONCLUSION

A new data hiding approach for bitstream of MP3 audio after compression is proposed based on the properties of Huffman coding in Count1 region of MP3 audio encoding. Availability and simpleness are the most advantage of this proposed method. The Huffman sign bits in region of Count1 are replaced with hidden secret message during the embedding phase. The size of MP3 audio file is constant since this algorithm not effect on MP3 bitstream lengths.

Considerable imperceptibility and high data embedding capacity is obtained by proposed approach which is shown in the experimental results. However, low robustness against decoding attack is the most disadvantage of this method, but attackers generally don't use this kind of attack which leads to decrease the MP3 quality.

Acknowledgement

The authors would like to thank Universiti Teknologi Malaysia for funding the research under vot 05H37. We also would like to thank our research group in supporting and giving positive comment to improve our paper.

References

[1] Cox, I. Miller, M. Bloom, J. Fridrich, J. and Kalker, T. 2007. *Digital Watermarking and Steganography*. Morgan Kaufmann.

[2] Jayaram, P. Ranganatha, H. R. and Anupama, H. S. 2011. Information Hiding using Audio Steganography–A Survey. *The International Journal of Multimedia & Its Applications (IJMA)*. 3: 86-96.

[3] Dong, X. Bocko, M. F. and Ignjatovic, Z. 2004. Data Hiding via Phase Manipulation of Audio Signals. *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*. 5: V-377.

[4] Rabah, K. 2004. Steganography-the Art of Hiding Data. *Information Technology Journal*. 3(3): 245-269.

[5] Bazyar, M. and Sudirman, R. 2014. A Recent Review of MP3 Based Steganography Methods. *International Journal of Security and Its Applications*. 8(6): 405-414.

[6] Atoum, M. S. Ibrahimn, S. Sulong, G. Zeki, A., & Abubakar, A. 2013. Exploring the Challenges of MP3 Audio Steganography. *2013 International Conference on IEEE Advanced Computer Science Applications and Technologies (ACSAT)*.

[7] Moon, S. K. and Kawitkar, R. S. 2007. Data security using Data Hiding. EEE *International Conference on Computational Intelligence and Multimedia Applications*. 4: 247-251.

[8] Bazyar, M. and Sudirman, R. 2015. A New Method to Increase the Capacity of Audio Steganography Based on the LSB Algorithm. *Jurnal Teknologi*. 74(6).

[9] Atoum, M. S. Rababaa, A. Suleiman, M. Ibrahim, S and Abdulgader Ahmed, O. 2011. A Steganography Method Based on Hiding Secrete Data in MPEG/Audio Layer III. *International Journal of Computer Science and Network Security*. 11(5): 184-188.

[10] Wang, Y. and Vilermo, M. 2001. A Compressed Domain Beat Detector using MP3 Audio Bitstreams. *Proceedings of the*

Ninth ACM International Conference on Multimedia. 194-202.

[11] Brandenburg, K. and Stoll, G. 1994. ISO/MPEG-1 Audio: A Generic Standard for Coding of High-Quality Digital Audio. *Journal of the Audio Engineering Society*. 42(10): 780-792.

[12] Al-Rababah, D. O. A. 2010. A Steganography Method Based on Hiding Secrete Data in MPEG/Audio Layer III. College of Alkamel-Computer Department King Abdul Aziz University-110 Alkamel, 21931.

[13] Musmann, H. G. 2006. Genesis of the MP3 Audio Coding Standard. *IEEE Transactions on Consumer Electronics*. 52(3): 1043-1049.

[14] Gang, L. Akansu, A. N. and Ramkumar, M. 2001. MP3 Resistant Oblivious Steganography. *ICASSP.* 1365-1368. IEEE.

[15] Shikano, H. Suzuki, Y. Wada, Y. Shirako, J. Kimura, K. and Kasahara, H. 2006. *Performance Evaluation of Heterogeneous Chip Multi-Processor with MP3 Audio Encoder. IEEE Symposium on Low-Power and High Speed Chips (COOL Chips IX)*. 349-363.

[16] Mahalleh, S., Behtaji, V., Selamat, H., and Sandhu, F. 2015. The Effects of Distance in Dynamic Psychological Factors of One-Dimensional Queue. *10th IEEE Asian Control Conference (ASCC).* 1-6.

[17] Bazyar, M. and Sudirman, R. 2015. A Robust Data Embedding Method for MPEG Layer III Audio Steganography. *International Journal of Security and Its Applications*. 9(12): 317-327.

[18] Thuresson, M. Spracklen, L. and Stenstrom, P. 2008. Memory-Link Compression Schemes: A Value Locality Perspective. *IEEE Transactions on Computers*. 57(7): 916-927.

[19] Tsai, T. H. Liu, C. N. and Chen, W. C. 2004. A Low-Power VLSI Implementation for Variable Length Decoder in MPEG-1 Layer III. *Electronic Imaging 2004*. 30-39.

[20] Sherigar, B. and Ramanujan Valmiki, K. 2004. Huffman Decoder used for Decoding Both Advanced Audio Coding (AAC) and MP3 Audio. *U.S. Patent Application*. 10: 880,695.

[21] Kim, D. H. Yang, S. J. and Chung, J. H. 2004. Additive Data Insertion into MP3 Bitstream using Linbits Characteristics. *(ICASSP'04). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings.* 4: iv-181.

[22] Sharma, M. 2010. Compression using Huffman Coding. *IJCSNS International Journal of Computer Science and Network Security*. 10(5): 133-141.

[23] Torrubia, A. and Mora, F. 2002. Perceptual Cryptography on MPEG Layer III Bit-Streams. *IEEE Transactions on Consumer Electronics.* 48(4): 1046-1050.

[24] Thiede, T. Treurniet, W. C. Bitto, R., Schmidmer, C. Sporer, T. Beerends, J. G. & Colomes, C. 2000. PEAQ-The ITU standard for Objective Measurement of Perceived Audio Quality. *Journal of the Audio Engineering Society*. 48(1/2): 3-29.

[25] Huber, R. & Kollmeier, B. 2006. A New Method for Objective Audio Quality Assessment using a Model of Auditory Perception. *IEEE Transactions on Audio, Speech, and Language Processing.* 14(6): 1902-1911.

[26] Kabal, P. 2002. An Examination and Interpretation of ITU-R BS. 1387: Perceptual Evaluation of Audio Quality. TSP Lab Technical Report, Dept. Electrical & Computer Engineering. 1-89.