HIDING A MESSAGE IN MP3 USING LSB WITH 1, 2, 3 AND 4 BITS

Alaa AbedulsalamAlarood[1,2],Azizah Abed Manaf[1], Mohammed J. Alhaddad[2] and Mohammed Salem Atoum[3]

[1]Department of Computing, University Technology Malaysia, Johor Bahru.
[2]Department of information Technology, King Abdulaziz University, Jeddah
[3]Department of Computer Science, Irbid National University, Irbid

ABSTRACT

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. This paper presentsa new method which randomly selects position in MP3 file to hide a text secret messageby using Least Significant Bit (LSB) technique. The text secret message isused in start and ends locations a unique signature or key.The methodology focuses to embed one bit, two bits, three bitsor four bits from secret message into MP3 file by using LSB techniques. The evaluation and performancemethods are based on robustness (BER and correlation), Imperceptibility (PSNR) and hiding capacity (Ratio between Sizes of text message and MP3 Cover) indicators.The experimental results show the new method is more security. Moreover the contribution of this paper is the provision of a robustness-based classification of LSB steganography models depending on their occurrence in the embedding position.

KEYWORDS

Steganography, LSB, mp3 data set, hiding a message

1. INTRODUCTION

Steganography is the art and science of hiding information by embedding messages within others, seemingly harmless messages. Steganography means “covered writing” in Greek. As the goal of steganography is to hide the presence of a message and create a covert channel, it can be seen as the complement of cryptography, whose goal is to hide the content of a message [1]. Steganography basically aims at hiding communication between two parties from the attackers [3]. Steganography operates by embedding a secret message which might be a copyright mark, or a covert communication, or a serial number in a cover message such as a video film, an audio recording, or computer code in such a way that it cannot be accessed by any wrong person during data exchange.

The three types of Steganography include the first, Pure Steganography where there is no need for the key. , Second Secret Key steganography and lastly, Public Key Steganography is based on the concepts of public key cryptography. Public key steganography uses a public key and a private key to secure the communication between the parties [20].

Steganography technique used in the data hiding process must have important properties in order to secure data successfully. Some of these properties include robustness, imperceptibility and capacity. These properties are explained below. Robustness means resistance to “blind”, non-targeted modifications, or common image operations [6]. Imperceptibility is typically required for

DOI: 10.5121/ijcnc.2016.8305

secure covert communication. For example, if a steganography method uses the noise component of digital images to embed a secret message, it should do so while not making statistically significant changes to the noise in the carrier.Capacity or (Data Rate) refers to the amount of information that can be hidden relative to the size of the cover message [21].

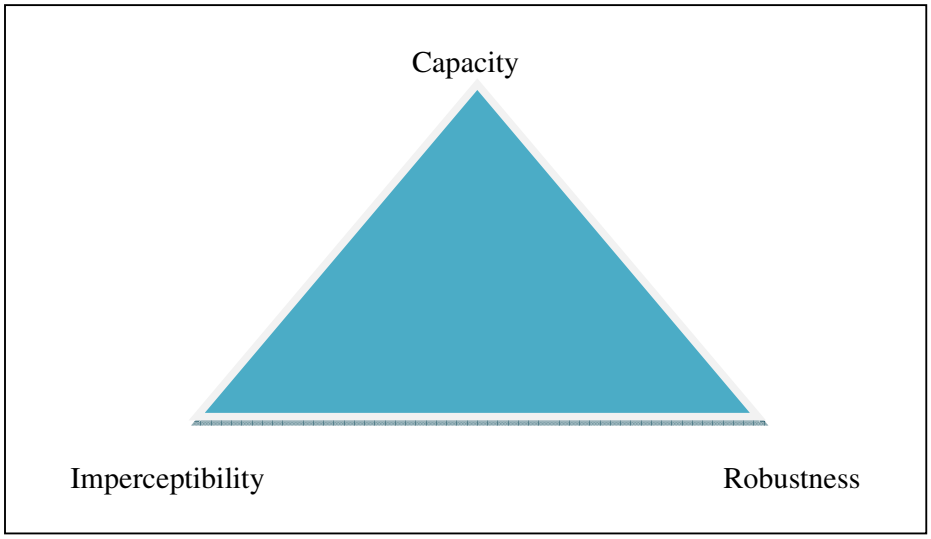


Figure 1. properties of steganography

2. STENOGRAPHIC METHODS

This section analyses the steganography which help understand the topic in a new perspective Steganography methods can be classified mainly into six categories, although in some cases exact classification is not possible [2].

- Substitution methods substitute redundant parts of a cover with a secret message (spatial domain). A number of methods exist for hiding information in various media. These methods range from LSB coding also known as bit plane or noise insertion toolsmanipulation of image or compression algorithms to modification of image properties such as luminance. Basic substitution systems try to encode secret information by substituting insignificant parts of the cover by secret message bits; the receiver can extract the information if he has knowledge of the positions where secret information has been embedded. Since only minor modifications are made in the embedding process, the sender assumes that they will not be noticed by a passive attacker.[14]

- Transform domain techniques embed secret information in a transform space of the signal (frequency domain): It has been noted early in the development of stenographic systems that embedding information in the frequency domain of a signal can be much more robust than embedding rules operating in the time domain. Most robust stenographic systems known today actually operate in some sort of transform domain.[15]

- Spread spectrum techniques: define spread spectrum techniques as "means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by means of a code which is independent of

the data, and a synchronized reception with the code at the receiver is used for dispreading and subsequent data recovery.[16]

- Statistical methods encode information by changing several statistical properties of a cover and use hypothesis testing in the extraction process.

- Distortion techniques store information by signal distortion and measure the deviation from the original cover in the decoding step.

- Cover generation methods encode information in the way a cover for secret communication is created.

3. DATA SET GENERATION AND PREPARATION

The dataset that is used in this research are cover files and secret messages. The cover files are MP3 files and the secret message is text. Most researchers who work in MP3 steganography used their own file for testing, and did not use standard data set. However, the types of MP3 file that are used have been generated from the standard dataset used in [17].

3.1 COVER DATASET

In the proposed Algorithm, the cover files are MP3 files. MP3 was created in 1993 by the Fraunhofer Institute and since then, it has become the most used methods for audio compression. The algorithm was standardized as MPEG-1 Layer III (ISO 11172-3). This algorithm achieves a good data compression when using the knowledge of the limitations in the human hearing to eliminate information without affecting the sound quality perception [18]. To generate MP3 file, standard data set uses a program to convert each genre from wave file to MP3 file. Many common programs are used to convert between different audio formats such as Free Make Audio converter. However, there are five different bit rate encoding compression methods in MP3 compression: 320 kbps, 256 kbps, 196 kbps, 128 kbps and 96 kbps. The differences between bit rate methods, encoding compression are impact of sound quality, where increasing the number of bits per sample means increasing the quality of sound. The sampling frequencies for bit rate for (320, 256 and 192 kbps) are 48 KHz and for 128 kbps is 44.1 KHz and for 96 kbps is 22.050 KHz respectively. Table 1 shows MP3 standard data set generated to be implemented in this paper [22].

From Table 1, it can be concluded that the size of a wave file is more than the size of MP3 files. Furthermore, the different sizes between MP3 file depend on the time of music and the number of bits per sample. If the quality is important, the size should be more.

Table 1. MP3 Dataset

Name of genre	Size of file (WAVE)	Size under 320kbps MP3	Size under 256kbps MP3	Size under 192kbps MP3	Size under 128kbps MP3	Size under 96kbps MP3
Classical	**14.7**	**6.67**	**5.33**	**4**	**2.66**	**2**
Jazz	**16.2**	**7.34**	**5.87**	**4.4**	**2.93**	**2.2**
Country	**18.7**	**8.48**	**6.78**	**5.08**	**3.39**	**2.54**
R&B	**19.4**	**8.81**	**7.05**	**5.29**	**3.52**	**2.64**
Rap	**20.1**	**9.14**	**7.31**	**5.48**	**3.65**	**2.74**
Reggae	**20.1**	**9.14**	**7.31**	**5.48**	**3.65**	**2.74**

Pop	**20.2**	**9.16**	**7.33**	**5.49**	**3.66**	**2.75**
Rock	**23**	**10.4**	**8.35**	**6.26**	**4.17**	**3.13**
Blues	**11.8**	**10.7**	**8.59**	**6.44**	**4.29**	**3.22**
Hip-hop	**27.5**	**12.4**	**9.98**	**7.48**	**4.99**	**3.74**
Dance	**31.3**	**14.2**	**11.3**	**8.53**	**5.68**	**4.26**
Metal	**32.6**	**14.8**	**11.8**	**8.88**	**5.92**	**4.44**

3.2 SECRET MASSAGE

The secret message is text; there are six different sizes of secret massage 100 KB, 200 KB, 400 KB, 800 KB, 1 MB and 2 MB.

Table 2. Text Datasets used as a secret message

Name of Text	Size	Size byte
100.txt	100 KB	102,590 bytes
200.txt	200 KB	205,180 bytes
400.txt	400 KB	410,364 bytes
800.txt	800 KB	819,932 bytes
1Mb.txt	1 MB	1,049,704 bytes
2Mb.txt	2 MB	2,099,408 bytes

4. PROPOSED ALGORITHM

To make stego MP3 file by reading an MP3 audio file and text message file then, embed the text file inside the audio MP3 to generate stego MP3 that contains a specific message.

4.1 PRE-PROCESSING FOR MP3 AND TEXT FILES

Will read the MP3 File and returns an output argument as the analog value of the audio samples, properties of the MP3 file, sampling frequency, and number of bits. In addition, we remove the header and timeframe.in parallel way read the text file, after that compare the size of MP3 file with text file if I can embedded or not.

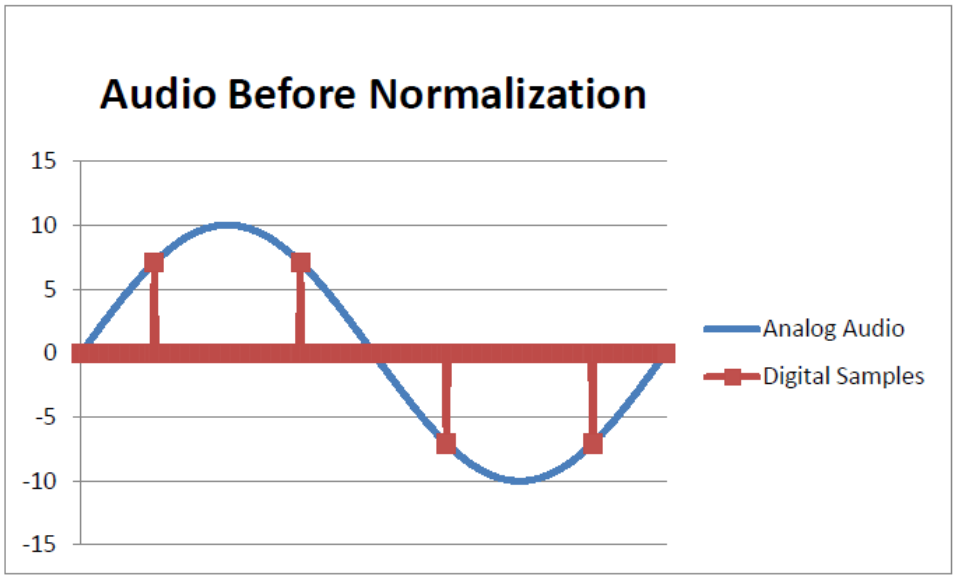


Figure 1 Pre-Processing MP3 files

The parameters of the MP3 file will be measured and estimated to determine the size of data, encoding and other parameters. The embedding is start from random location inside the audio file. The random start location is calculated as the following equation:

Irand=ceil (rand*fix (Espace/2)) +200 (1)

Where Espace is calculated as the following:

Espace=r-rb*cb/deg–200 (2)

Where, "r" is the number of samples in the MP3 file"rb*cb" is the size of text message file "deg" is the number insertion bits"rand" is a function that generates random number from 0 to 1.This will generates a random starting location of embedding where it ensures that, the end of insertion will be located inside the MP3 file.

4.2 Digitizing for MP3 and Text file

To enable digital handling of the text file, all text data then converted to digital format. The Text file is firstly converting to ASCII format, takes the string of the text data as input argument and returns the ASCII code for each character. Below is an example of this function

>>double ('Hello')

ans = 72 101 108 108 111

The result of this function is in decimal format not hexadecimal, and no need to convert it to hexadecimal format at old. Instead, it should be converted to binary directly. Each single character is being converted separately to binary. The result of binary conversion will be in a

matrix of two-dimension format, as shown in the example bellow of the "Hello" word. The row indicates the character, and the columns represent the binary code for the specific character.

```
>>dec2bin ('ans')

ans =
1001000
1100101
1101100
1101100
1101111
```

We used function to converts the decimal numbers to binary as the following equation where Bd_i is the binary digit index, and Dd_i+1 is the decimal digit division result. And the "rem" is the division remainder.

$$Bd_i=\text{rem}\ (Dd_i+1/2) \quad (3)$$

The results will be a two dimensional matrix with size of Rx8 where the 8 is the number of bits for ASCII character conversion to binary, and the R is the number of characters in the text file. R is counting not only alphabetic characters, but also any ASCII symbol including the space and carriage return.

4.3 Normalize MP3 and Text files

When the text file is processed and converted to binary, then the MP3 audio file should be processed tool. The first step of audio file processing is to normalize it. As the third party function that reads the MP3 file express it as analog value, then, each MP3 sample will have a value in the interval [-1, +1] with float number format, By theory, the float number is an approximation in digital system, and thus, any process over it will comprise an accumulated error. Therefore, to deal with this analog value with minimal error, that's limit approaches to zero, then, normalizing it to higher value is required. The following equations illustrate how to normalize the analog numbers of the audio MP3. Where the AiN is the ith normalized sample of the audio array A.

$$A_iN=(A_i+1)*10^6 \quad (4)$$

After normalized sample audio processing is to convert the normalized audio samples to binary format. The conversion function is the same that has been used for the text file conversion that is described before.

4.4 Build Stego-object

Once the audio is being normalized and converted to binary, and the text file has been converted to ASCII then to binary, then, the data is ready to build the stego file starting by embedding the binary that represents the text within the binary that represents the audio. If fact, because of the embedding starts from random location within the MP3 file, the embedded message should be bounded by its start and end locations. The start location follows start signature or key, and the end location is just before end signature or key. Another issue is that, multi-bits insertion is possible, so, the key should contain information about the number of insertion bits. The following signatures are used at start of every message embedding:

Single bit insertion ➔ 10101010101010
Two bits insertion ➔ 01010101010101
Three bits insertion➔10101010101010, 01010101010101
Four bits insertion ➔ 01010101010101, 10101010101010

The same signature is being used for start and end of embedding. So, the massage is being bounded within the same signature that indicates start and end of the message, in addition to number of insertion bits. The embedding is being done in terms of insertion. The insertion is a method that is simply removes a bit or number of bits from the carrier data and inserts new bit or bits from the message data.

The figure 2 bellow shows how to insert one bit, and two bits from the message to the carrier. The insertion is being done in terms of least significant bit. From (least significant), its noticeable effect of the actual digital value is negligible. So, the insertion will not highly affect the resulted audio data with respect to the person who hears the MP3 audio file.

There are four insertion scenarios depending on the number of bits those are inserted within the carrier, according to the developed system. Single bit insertion, two bits insertion, three bits insertion, and four bits insertion is possible and those are selected with the input arguments of the developed program. The figure above shows how to insert single bit, and two bits only. The described process above is continued until all text data bits inserted within the digital carrier data that represents the audio MP3 data.

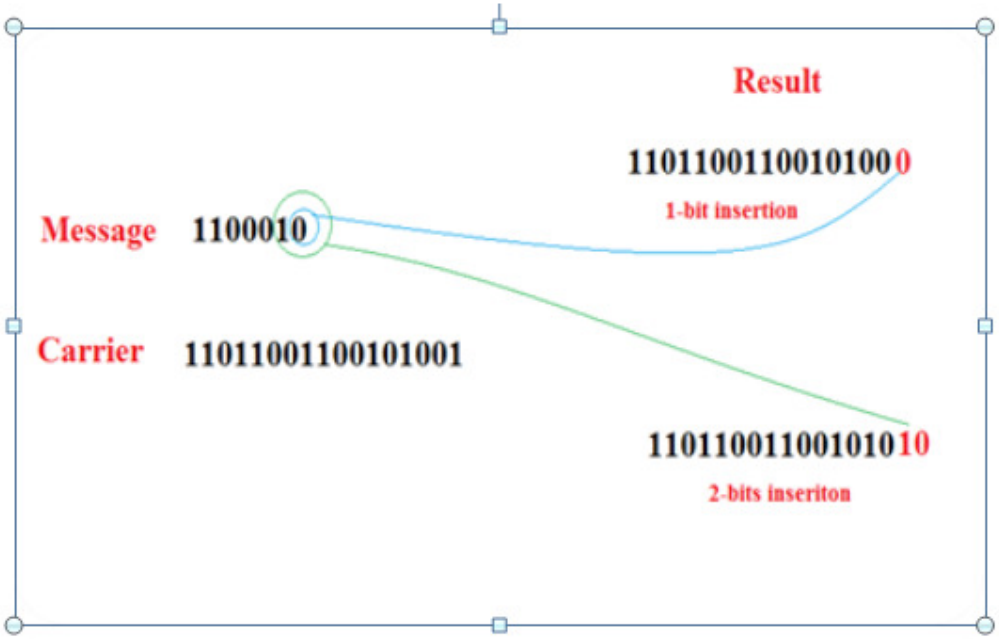


Figure 3. the insert 1 bit, and 2 bits from the message to the carrier

Once the all-binary samples of text file were inserted and the process is completed, an inverse process will be performed in the post-processing phase. The resulted digital array that represented the stego MP3 file will be converted to decimal again in an inverse process of that described above. Where above a decimal to binary conversion is being accomplished, but now, binary to decimal will be performed. The following equation illustrates how to convert the binary data to decimal.

$$Dd=\sum b_i*2_i \quad (5)$$

Where Dd is the decimal digit that is resulted after conversion, b_i is the $_i$th binary bit value, $_i$ is the index of the binary bit. The "$_i$" has values from 0 to 22. Hence, the maximum normalized decimal number is $2x10^6$, that's why the maximum number of i is 22.The resulted decimal data is normalized according to the normalization process that is described above. Thus, de-normalization is required to get the analog audio format again. The following equation is being used for de-normalization.

$$A_i= (A_iN*10^6)-1 \quad (6)$$

The latest step is finally to convert the stego audio data to an MP3 format file. The same toolbox for MP3 format handling is also used; it contains an MP3 write function. It takes the number of bits and MP3 encoding format to generate formal MP3 file. The same MP3 parameters those are gotten when read of the original MP3 file are used to write the new stego MP3 file.

In figure 4The following flowchart that explain process of the model that start to read the text and MP3 file and then Normalize MP3 file and convert the text file to ASCII format, and then Embeddedbits within the MP3 file, after finish that we reverse the process to restores the MP3 file as normal MP3.

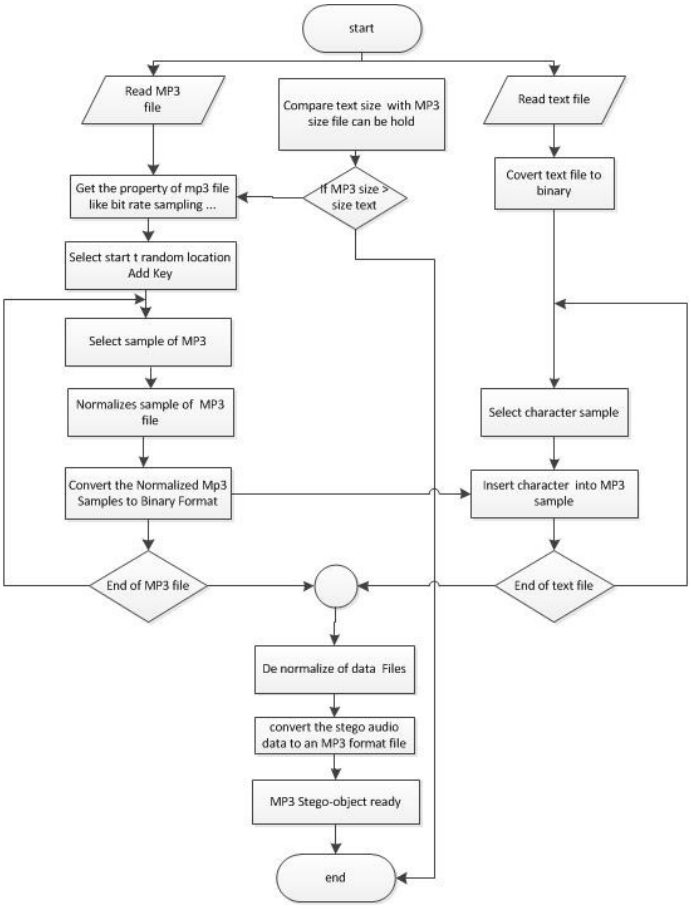


Figure 4. Process for embedding Text to MP3 fil

5. EXPERIMENTAL RESULTS

The Peak Signal-to-Noise Ratio (PSNR) is the ratio between a signal's maximum power and the power of the signal's noise. Engineers commonly use the PSNR to measure the quality of reconstructed signals that have been compressed. Signals can have a wide dynamic range, so PSNR is usually expressed in decibels as show Comparisonsbetween embedded different bits in figure 7. In statistics, the Mean Squared Error (MSE) of an estimator is one of many ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the average of the squares of the "errors." The error is the amount by which the value implied by the estimator differs from the quantity to be estimated, as show Comparisonsbetween embedded different bits in figure 6. [19].

Table 3. PSNR and MSE values of ten audio files at 200 kb text message s

Name of genre	PSNR	MSE	No. of Embedding
Classical	75.4490	0.0019	7714000
Jazz	63.0154	0.0325	8494000
Country	63.1514	0.0315	9805000
R&B	63.7896	0.0272	10195000
Rap	58.7419	0.0869	10573000
Reggae	60.8632	0.0533	10571000
Pop	65.4141	0.0187	10596000
Rock	65.4814	0.0184	12068000
Blues	66.9008	0.0133	12423000
Dance	64.1514	0.0175	13486000
Hip-Hop	66.2314	0.0168	13926000
Metal	63.4814	0.0135	14499000

In **Error! Reference source not found.**5 shows the signal structure of the audio file Rap.MP3 at before embedding. By using MP3 audio with 128kbps and the size of MP3 3.65 MB (3,836,190 bytes) and the genre Rap. Also use 200 KB the secret Message.

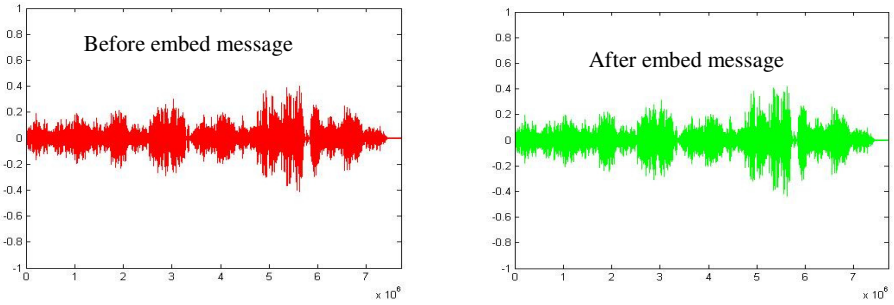


Figure 5. the signal level comparisons between a MP3 carrier file before and after the LSB with size 3.65 MB Text 200 KB

Table 4. PSNR and MSE values of ten audio files at 200 kb text message when embedded 1 bit and embedded 2 bits

Name of genre	embedded 1 bit			embedded 2 bits		
	PSNR	MSE	No. of Embedding	PSNR	MSE	No. of Embedding
Classical	78.1266	0.0010	7714	77.6796	0.0011	7714
Jazz	63.4224	0.0296	8494	63.4224	0.0296	8494
Country	63.5267	0.0289	9805	63.5267	0.0289	9805
R&B	64.1845	0.0248	10194	64.1845	0.0248	10194
Rap	59.1376	0.0793	10572	59.1376	0.0793	10572
Reggae	61.2687	0.0486	10571	61.2687	0.0486	10571
Pop	66.1688	0.0157	10595	65.7593	0.0173	10595
Rock	65.8772	0.0168	12067	65.8772	0.0168	12067
Blues	67.3400	0.0120	12422	67.3400	0.0120	12422
Dance	67.0223	0.0129	16443	65.2804	0.0193	16443
Hip-Hop	60.1148	0.0633	14423	58.8159	0.0854	14423
Metal	68.1388	0.0100	17114	66.8664	0.0134	17114

Table 5 PSNR and MSE values of ten audio files at 200 kb text message when embedded 3 bits and 4 bits

Name of genre	embedded 3 bits			embedded 4 bits		
	PSNR	MSE	No. of Embedding	PSNR	MSE	No. of Embedding
Classical	77.6796	0.0014	7714	77.2308	0.0012	7714
Jazz	63.4224	0.0267	8494	64.3006	0.0242	8494
Country	63.5267	0.0262	9805	64.3789	0.0237	9805
R&B	64.1845	0.0225	10194	65.0483	0.0203	10194
Rap	59.1376	0.0719	10572	59.9930	0.0651	10572
Reggae	61.2687	0.0439	10571	62.1421	0.0397	10571
Pop	65.7593	0.0142	10595	67.0086	0.0129	10595
Rock	65.8772	0.0152	12067	66.7580	0.0137	12067
Blues	67.3400	0.0108	12422	67.9554	0.0110	12422
Dance	65.2804	0.0193	16443	65.2817	0.0193	16443
Hip-Hop	58.8159	0.0854	14423	58.8168	0.0854	14423
Metal	66.8664	0.0134	17114	66.8681	0.0134	17114

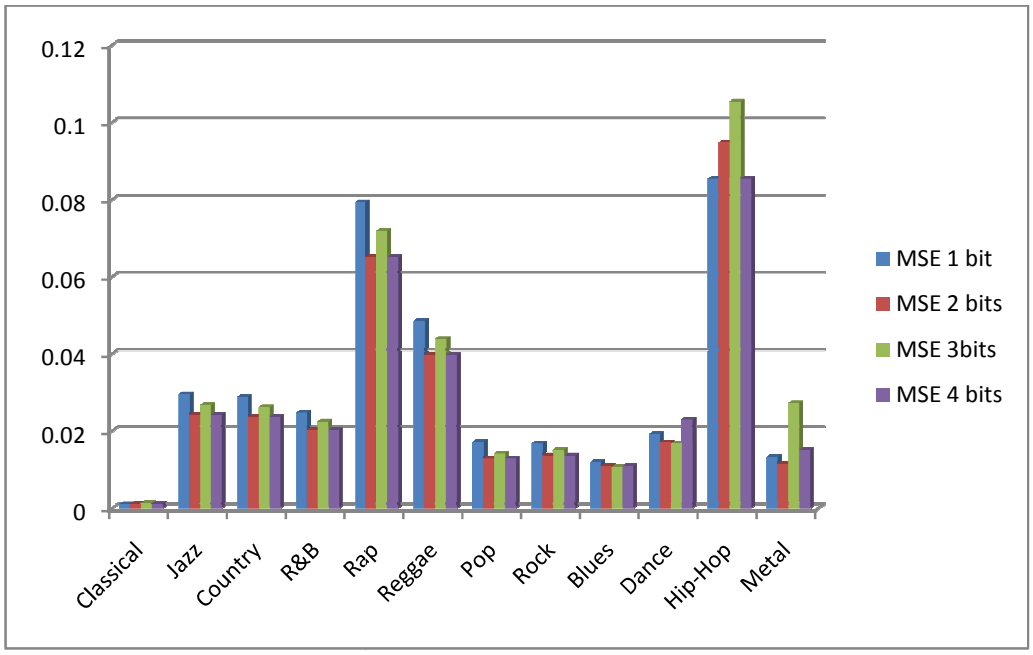


Figure 6. Comparisons between different results for Mean Squared Error (MSE)

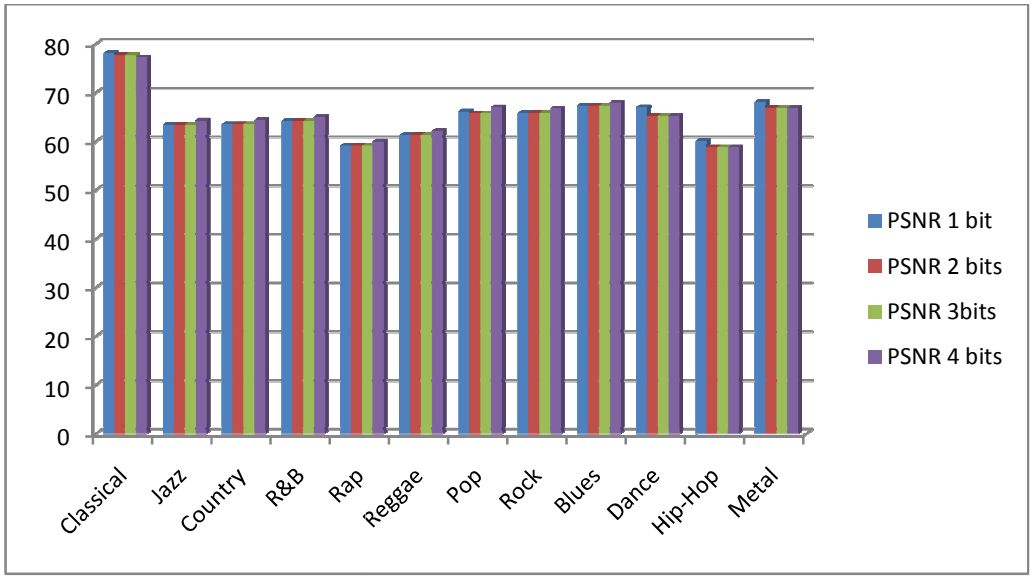


Figure 7. Comparisons between different results for Mean Squared Error (MSE)

6. CONCLUSIONS

This paper has explored and reviewed MP3 audio steganography, particularly with respect to MP3 files after compression. LSB in time domain has been developed to use randomly position from cover file to hide the secret message by using 1, 2, 3 and 4 bits. The new Model aims at meeting the three most important audio steganography requirements, which are imperceptibility, capacity, and robustness. Any technique tries to enhance the capacity or robustness should

preserve imperceptibility. A new method is increased the capacity and robustness as well as improved the imperceptibility. In this paper, we concentrate model that has been built, achieved hiding the data in Audio file, by keeping the accuracy of the audio file high, even though, one mange to discover the secret message, still extracting the message is challenging.

6. ACKNOWLEDGEMENTS

This Project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under grant no. (G-481-611-37). The authors, therefore, acknowledge with thanks DSR for technical and financial support

REFERENCES

[1] Katzenbeisser S., Peticotas F., “Information Hiding Techniques For Steganography And Digital Watermarking”, Artech House Inc.2000

[2] Stefan Katzenbeiser& Fabien A.P.Petitcolas(1999), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Computer Security series, Boston, London

[3] Kivanc M., “Information Hiding Codes And Their Applications To Images And Audio”, Phd. Thesis, University Of Illinois At Urbana-Champaign, 2002.

[4] Cacciaguerra S., Ferretti S., “Data Hiding: Steganography And Copyright Marking”, Department Of Computer Science, University Of Bologna, Italy, Url: Http://Www.Cs.Unibo.It/~Scacciag/Home-File/Teach/Datahiding.Pdf.

[5] Dunbar B., “A Detailed Look AtSteganographic Techniques And Their Use In An Open-Systems Environment”, Sans Institute 2002, Url: Http://Www.Securitydoc.Com/Library/1272.

[6] Fridrich, J. (2010). Steganography in Digital Media Principles, Algorithms, and Applications. Cambridge University Press: UK.

[7] Bender W., Gruhl D., Morimoto N., Lu A., "Techniques For Data Hiding “, Ibm System Journal, Vol. 35, No. 3&4,1996,Url: Http://Isj.Www.Media.Mit.Edu/Isj/Sectiona/313.Pdf.

[8] P.K. Singh, H. Singh, And K. Saroha, “A Survey On Steganography In Audio,” Audio, 2009.

[9] M. Wakiyama, Y. Hidaka, And K. Nozaki, “An Audio Steganography By A Low-Bit Coding Method With Wave Files,” 2010 Sixth International Conference On Intelligent Information Hiding And Multimedia Signal Processing, Oct. 2010, Pp. 530-533

[10] Kekre, H. B., Athawale, a, Rao, B. S., &Athawale, U. (2010). Increasing the Capacity of the Cover Audio Signal by Using Multiple LSBs for Information Hiding. 2010 3rd International Conference on Emerging Trends in Engineering and Technology, 196-201. IEEE. doi:10.1109/ICETET.2010.118

[11] Zamani, M., Taherdoost, H., Manaf, A. A., Ahmad, R. B., &Zeki, A. M. (2009). Robust Audio Steganography via Genetic Algorithm. Soft Computing, 0-4.

[12] Zamani, M., Manaf, A. A., Ahmad, R. B., Zeki, A. M., & Abdullah, S. (2009). A Genetic-Algorithm-Based Approach for Audio Steganography. Engineering and+ Technology, 360-363.

[13] Bhowal, K., Pal, a J., Tomar, G. S., &Sarkar, P. P. (2010). Audio Steganography Using GA. 2010 International Conference on Computational Intelligence and Communication Networks, 449-453. Ieee. doi:10.1109/CICN.2010.91

[14] Cox, I., et al., "A Secure, Robust Watermark for Multimedia," in Information Hiding: First International Workshop, Proceedings, vol. 1174 of Lecture Notes in Computer Science, Springer, 1996, pp. 185–206

[15] Koch, E., and J. Zhao, "Towards Robust and Hidden Image Copyright Labeling," in IEEE Workshop on Nonlinear Signal and Image Processing, Jun. 1995, pp. 452–455.

[16] Tirkel, A. Z., G. A. Rankin, and R. van Schyndel, "Electronic Watermark," in Digital Image Computing, Technology and Applications—DICTA 93, Macquarie University, 1993, pp. 666–673

[17] Rückert, C., R. Szczepanowski, et al.(2005) "Complete genome sequence of the actinobacterium< i>Actinoplanesfriuliensis</i> HAG 010964, producer of the lipopeptide antibiotic friulimycin." Journal of biotechnology 178: 41-42.

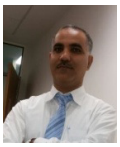
[18] Chanu, Y. J., K. M. Singh, et al. "Image steganography and steganalysis: A survey." International Journal of Computer Applications 52(2).2012

[19] Atoumet al., “A New Method for Audio Steganography Using Message Integrity” .Journal of Convergence Information Technology(JCIT),Volume8, Number14, September 2013

[20] Atoum, M. S., Ibrahim, S., Sulong,G. and Ahmed, A. (2012). MP3 Steganography: Review. Journal of Computer Science issues, 9(6).
[21] Atoum, M. S. (2015). A Comparative Study of Combination with Different LSB Techniques in MP3 Steganography. In Information Science and Applications (pp. 551-560). Springer Berlin Heidelberg
[22] Atoum, M. S. (2015, August). New MP3 Steganography Data Set. In IT Convergence and Security (ICITCS), 2015 5th International Conference on (pp. 1-7). IEEE.

AUTHORS

Mr.Alaaalarood He still continues his Ph.D. in UniversityTechnology Malaysia (UTM) Faculty of computingand Information Technology. He is Lecturer of king abed alaziz university, Department of Computing and Information Technology.His research interests are Information Security,steganalysis, steganography, Artificial Intelligence ANN, and Computer Graphics.



Prof. Azizah Abdul Manaf, Professor of Computer Science, Deputy Dean Academic Advanced Informatics School (UTM AIS) UniversitiTeknologi Malaysia (UTM) her research interests are Image Processing, Multimedia Security, Computer Forensics.

Mohammed J. Alhaddad Associate Professor in Faculty of Information Technology at King Abdulaziz University. Visiting Associate Professor at University Teknologi Malaysia, His research interests are: network Security, Artificial Intelligence, Robots, Brain Computer Interface BCI.

Mohammed Salem Atoum is assistant professor in Irbid National University. His research interests,Steganography, Watermarking, data hiding,cryptography,Information Security.