# Secret Message Integrity of Audio Steganography Using Bi-LSB Embedding

**Mahmood Maher Salih† , Mudhafar Al-Jarrah††**

† Tikrit University, †† Middle East University

**Summary**
The main issue for the majority of internet users is the information security. To ensure having secured use of internet, several steganography methods have been proposed. Those methods are mainly utilized to solve the security problems using two main processes; embedding and extracting. However, the current steganography methods, cannot verify the presence of attacks in secret messages. Therefore, this work introduces the development of an advanced Least Significant Bit (LSB) technique; Bi-LSB to solve the low security and capacity problems of the traditionally used LSB techniques. The performance of the developed Bi-LSB technique is evaluated based on adding an Additive White Gaussian noise (AWGN) noise to the stego file before extracting the embedded messages and comparing the extracted messages with the original ones using three techniques; checksum, hash function and frequency methods. Results demonstrated that both the PSNR values and correlation percentage for the three integrity methods are decreased after adding the AWGN attack.

***Key words:***
*Steganography; Least Significant Bit (LSB); Bi Least Significant Bit (Bi-LSB); embedding; extractin; integrity; checksum; hash function; frequency*

## 1. Introduction

Information is shared globally through the Internet, in digital form [1]. There are issues and challenges regarding the security of information in transit from senders to receivers. The major issue is the protection of digital data against any form of intrusion, penetration, and theft. The major challenge is developing a solution to protect information and ensure their security during transmission. Three components of information security are confidentiality, integrity, and availability [2]. Confidentiality ensures that information is kept secret from any unauthorized access. This could be done through information hiding techniques, namely cryptography and steganography [3].

Cryptography involves the act of encryption and decryption of a digital data. The major weaknesses of such techniques are that even though the message has been encrypted, it still exists. Steganography dwells on concealing any digital data in an innocuous digital carrier, the word steganography is derived from an old Greek word which means covered writing [4].

Steganography has been used for concealing secret messages during ancient times [5]. It was used by Histiaeus, the tyrant of Miletus, who, in 499 BC, tattooed the scalps of his slaves with a hidden message with a command for his men to attack the Persian [6], [7]. The message became hidden when the slaves' hair grew back. According to researchers, steganography can be described as a study of the means of hiding secondary information within primary information without affecting the size of information nor the cause of any form of distortion which could be perceived [8], [9].

Steganography is one of the two techniques used for covert communication. However, watermarking is the second technique that can embed watermark into host cover to keep copyright for the hosts. Steganography typically establishes point-to-point data security [10]. The strength of steganographic technique, in keeping the data in the carrier medium against attacks or alteration is weak during transmission, storage or format conversion is weak [4].

The remaining of the paper is organized as follows; section II introduces the research scope, section III introduces some of the recent works related to MP3 steganography methods. Section IV introduces the system model, including the methodology that is followed during the system implementation and designing. Section V investigates and discusses the obtained results. Section VI concludes the whole work.

## 2. Research Scope

This paper focuses on Bi LSB technique on MP3 audio files. The unit of analysis is the security of the proposed technique. The following are the highlights; applying AWGN attack to the stego file before extracting the embedded messages and then checking the integrity based on comparing the extracted messages with the original ones using three techniques; checksum, hash function and frequency method

Manuscript received July 5, 2015
Manuscript revised July 20, 2015

## 3. Previous Research

Frame headers of the MP3 are made up of fields like the private bit, copyright bit, original bit, and emphasis bit. However, their use is commonly omitted in a number of MP3 players. Such fields are the important aspect of the frame, which aids the interpretation of information that is concealed in an audio signal. They can be properly applied to embed undisclosed massage where they replace the undisclosed massage bit stream through the bits in the field. But, if in the process of replacing bit stream with bits in the field, there is a failure the actual content of the secret message received within the frame is lost and this makes signal recovery to be more challenging [11].

According to [12], stuffing of padding byte was recently established as being one of steganography techniques. Its approach is relatively straightforward in terms of implementation. It represents regular and fine storage capability and contains the ability to program 1 byte of information for any frame so long as there is accessibility of padding bytes. An MP3 file is a given example of the medium of material that can well utilize the method of padding byte stuffing because it can allow for hundreds of frames in a secret message, specifically when the filling bytes are not able to take any audio information.

Authors in [13], developed BAF, their approach embeds into MP3 file the embeds text files. The text file is encrypted by the use of the RSA algorithm in order to increase the undisclosed secret message security. Encrypted information is filled in the first frame. This process is done time and again sequentially until the frame headers are filled. Approximately 15 KB is used when encryption algorithm is utilized; otherwise, it takes around 30 KB for MP3 file. Even with the chances of the secret message to be sniffed, there are many advantages in using this approach; for example, the method of padding and the unused bit even after the frames must have been filled, provides more encoding capability.

Authors in [13], developed the technique of Steganography that embeds between frames (BF). It as well embeds the text file to MP3 file like the BAF and information is encrypted in bits format by use of the RSA algorithm to provide extra protection for that concealed secret message. There is a difference between BF and BAF that exists in the way text files are inserted into the frames. This does not start with the first frame seen; but it selects the frame it chooses. Again, in terms of capacity, compared to BAF, BF uses around d40 MB with encryption algorithm, it however requires 80 MB with the original format. Although BF provides a higher capacity for embedding text file, it is still prone to attack. Literature review shows that the embedding information method after compression is a hard task since the process of embedding is done after the compression and the text file are located in the location of unused bits and not in audio data. The platform provided by this technique is prone to attack as a result of the content of the sent secret message which can easily be deciphered by third party sniffing by use of the communication link. It also provides limited capacity for hiding the message that is secret. However, the problem with capacity would be resolved if the LSB technique used 2, 3 and 4 bit exchange in the audio data (8-bit for sample) to insert speech in MP3 file. While addressing the security problem, the use of key as a lock for the concealed secret message is a viable approach that could achieve maximum security for concealed secret messages.

Authors in [14], proposed a new method for imperceptible audio data hiding for an audio file with wav or mp3 format. This approach is based on the Mod 16 Method (M16M) designed for image, the Mod 4 Method (M4M), along with Number Sequence Generator Algorithm to avoid embedding data in the consecutive indexes of the audio, which would eventually assist prevent distortion in the quality of audio. The input messages exist in any digital form, and are often treated as bit streams. The positions of embedding are selected on the basis of some mathematical function that de-ends of the data value of the digital audio stream. Data embedding is performed by mapping every two-bit of the secret message in each of the seed positions based on the remainder of the intensity value while dividing by 4. The extraction process starts with the selection of those seed positions required during embedding. On the receiver’s side, a different reverse operation is carried out to the extract of the original information.

Authors in [14], proposed a new imperceptible method of audio data hiding for an audio file with mp3 or wav format. The approach is based on the method of Mod 16 (M16M) designed for image, the Mod 16 Method for audio (M16MA), and used alongside with a Number Sequence Generator Algorithm that help avoid embedding data in consecutive indexes of the audio, this would help prevent distortion of audio quality. Input message can exist in any digital form and is commonly treated as a bit stream. The positions of emending are selected based on some mathematical function that de-ends of the data value of the digital audio stream. The embedding of data is performed by mapping every four-bit of the secret message in each of the seed positions based on the remainder of the intensity value when divided by 16. Extraction process starts by selecting those seed positions required during embedding. On the receiver’s side, a different reverse operation is carried out to extract the original information.

Authors in [13], developed (SSAS) the most significant and public standard for audio compression on the Internet is the “secret digital music MP3 files”. To ascertain high rates of compression and shrunk the main file to the smallest possible size, the MP3 uses a devastating technologies for compression. In the process of protecting the “digital media files”, there are many algorithms that used in the data hiding, are known the steganographic algorithms, which defined as covered writing. This algorithm has three types; “public key steganography”, the “secret key steganography”, and “pure steganography”. Applications, which are used in the data hiding, there is a requirement for hiding algorithms on the side of the sender, and detecting algorithms and mechanism on the side of the receiver. The authorized persons just can retrieve the hiding data and messages. The data hiding pivotal parameters are; capacity, invisibility, complexity, security, and reliability. The authors discussed a new method on steganographic based on the algorithm of “novel data-embedding”, to embed information between the frames of MP3.

## 4. Research Method

### 4.1 Introduction

This section explores the conducted methodology in this work to offer an advanced Bi Least Significant Bit (Bi-LSB) MP3 audio steganography technique to solve the security problem of traditional LSB techniques and offer an efficient method to hide audio information in a more secured way with the use of the MATLAB program. All the conducted procedures and principles in this work to achieve the main purpose of this study are proposed in details.

### 4.2 Research Framework

In this section, the proposed method is explained in details. The framework of this research includes three main phases; preprocessing, embedding and extracting and message validation. The preprocessing is applied to enhance the security of messages to be hidden in an MP3 file. The embedding and extracting stage includes designing the proposed algorithm for MP3 files to solve the current security problem of the traditional LSB technique, while the message validation stage is applied to develop another method to hide messages to recognize the efficiency of hidden messages in MP3 files.

### 4.2.1 Preprocessing Phase

Both the input cover (C) and the Message (M) are MP3 files. In this work, the secret message is validated and preproced based on implementing three methods; checksum, hash function and frequent comparison in order to generate the codebook. The resultant preprocessed C and M files are then used in the second stage.

### 4.2.2 Embedding and Extracting Phase

This phase includes two main processes; embedding and extracting. In the embedding process, both the preprocessed C and M files are processed to offer a Stego Object (SO). After that, the Bi-LSB method is used to embed the secret message on the cover. In the extracting processes, the SO is then processed to extract the M file.

### 4.3 Available Datasets

Both the cover and secret messages in this work are MP3 files since they achieve a good data compression when considering the limitations in the human hearing to eliminate information without affecting the sound quality perception. Since there are no efficient number of researchers who used MP3 files as cover messages, there are available standard data set to apply embedding and extracting algorithms to generate results. This is because most researchers depend on using .wav file, where this results in the available standard data set for it. In this data set, wav file contains 12 different genres; (Classical, Jazz, Country, and R&B, Rap, Reggae, Pop, Rock, Blues, Hip-hop, Dance and Metal).

The generation of MP3 files depends on using a certain program to convert each genre from wav file to MP3 one, such as the Free Make Audio converter. On the other hand, there are five different bit rate encoding compression methods to compress MP3 files; 320 kbps, 256 kbps, 196 kbps, 128 kbps and 96 kbps, where they differ in their impact on the sound quality. In other words, the increase in the number of bits per sample results in an increase in the quality of sound. The sampling frequency for bit rate for 320, 256 and 192 kbps is 48 KHz, while it is 44.1 KHz for the 128 kbps and 22.050 KHz for the 96 kbps. The following table illustrates the MP3 standard data set, which generated to be used in this work.

Table 1: Cover dataset.

| Name of genre | Time | Size of file (WAV) | Size under 320 kbps | Size under 256 kbps | Size under 192 kbps | Size under 128 kbps | Size under 96 kbps |
|---|---|---|---|---|---|---|---|
| | Minute | (MB) | (MB) | (MB) | (MB) | (MB) | (MB) |
| Classica | 2:54 | 14.7 | 6.67 | 5.33 | 4 | 2.66 | 2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| l | | | | | | | |
| Jazz | 3:12 | 16.2 | 7.34 | 5.87 | 4.4 | 2.93 | 2.2 |
| Country | 3:42 | 18.7 | 8.48 | 6.78 | 5.08 | 3.39 | 2.54 |
| R&B | 3:51 | 19.4 | 8.81 | 7.05 | 5.29 | 3.52 | 2.64 |
| Rap | 3:59 | 20.1 | 9.14 | 7.31 | 5.48 | 3.65 | 2.74 |
| Reggae | 3:59 | 20.1 | 9.14 | 7.31 | 5.48 | 3.65 | 2.74 |
| Pop | 4:00 | 20.2 | 9.16 | 7.33 | 5.49 | 3.66 | 2.75 |
| Rock | 4:33 | 23 | 10.4 | 8.35 | 6.26 | 4.17 | 3.13 |
| Blues | 4:41 | 11.8 | 10.7 | 8.59 | 6.44 | 4.29 | 3.22 |
| Hip-hop | 5:27 | 27.5 | 12.4 | 9.98 | 7.48 | 4.99 | 3.74 |
| Dance | 6:12 | 31.3 | 14.2 | 11.3 | 8.53 | 5.68 | 4.26 |
| Metal | 6:28 | 32.6 | 14.8 | 11.8 | 8.88 | 5.92 | 4.44 |

From the table above, it can be concluded that the size of wav file is bigger than that of the MP3 files. Furthermore, the difference in sizes between an MP3 file depends on the time of music and the number of bits per sample.

## 4.4 Designed System

The proposed system is designed using the MATLAB program. The designed system aims to evaluate the performance of the developed Bi-LSB technique based on adding an AWGN attack to the stego file before extrcating messages to investigate its effect on the PSNR values and then checking the integiry of the extracted messages.

### 4.4.1 Applying the Embedding Process

In this stage, the user is asked to verify the desired type of LSB technique; traditional or improved one. After that, the stego-object is initialized with the cover vector. Then, a GR value, which a random number in the first 10000 bytes of the cover message are chosen, where then the secret message in binary code is reshaped in one row in order to be used in the LSB algorithm in an ease way based on determining the number of LSBs and initializing two counters; k and w. A for-loop in message embedding is then generated to start at GR and with using a step of mod(i,100) to embed the secret message in the cover audio. For the traditional technique, in each one of the cycle iterations, the selected byte is modified using the following logic:

- If the 4-LSB is used, then bits from 2nd to 5th bits are substituted with the first available 4 bits in the secret message.
- If the 2-LSB is used, then bits 2nd to 3rd are substituted with the first available2 bits in the secret message.
- If the 1-LSB is used, only the 2nd bit is substituted with the first bit available in the secret message.

For the improved Bi-LSB technique, in each one of the cycle iterations, bits are hidden as follows:

- Hiding 1 bit in the first byte
- Hiding 2 bits in second byte
- Hiding 2 bits in third byte
- Hiding 1 bit in fourth byte and repeat

After that, the modified byte is converted back from binary to decimal to generate the stego object. In each one of the iterations, the counter k is updated depending on the chosen LSB algorithm.

### 4.4.2 Applying the Distortion Evaluation of the Message

In this stage, the Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) are calculated. Both represent the two error metrics used for comparing cover quality. The PSNR and MSE can be calculated using the following formulas, respectively:

$$MSE = \frac{1}{N} * \sum_{i=1}^{N}((X(i) - Y(i))^2 \quad .....................(1)$$

$$PSNR = 20 \log_{10} \frac{255}{\sqrt{MSE}} \quad ...........................(2)$$

Where, X is the original cover, Y is the stego-object, N is the size of the cover and MAX is the maximum variation in the input cover.

### 4.4.3 Applying the Integrity Validation message

In this stage, different intgerity methods are applied. The checksum technique offers integrity certification to simplify the recognition of packets, which carry secret data at the end of received message. The hash function represents the mapping of digital data with random size to data. The offered values by this function are known as hash codes, hash sums, has values or hashes. The frequency technique represents counting the number of ones in the message and comparing it with that number in the extracted message.

## 5. Results and Discussion

### 5.1 Attack Part Results

In this part, an AWGN attack is added to the stego file before extracting the message to extract the message and compare it with the original one based on computing the PSNR percentage.

### 5.1.1 Results of Adding AWGN Attack to the First Secret Audio Message

An AWGN attack is added to the Blues_96_msg1.mp3 secret message. The following tables show the obtained PSNR values for the attack for 0.1 bits/sec/Hz and 0.3 bits/sec/Hz variance, for the all band of the stego file.

Table 2 Results of adding AWGN to the first secret message - awgn=0.1

| Cover messages | Embedded secret message | PSNR without attack | PSNR with attack | Degredation percentage |
|---|---|---|---|---|
| Blues_cover1_320.mp3 | Blues_96_msg1.mp3 | 95.2852 | 92.5227 | 2.89% |
| Classical_cover2_320.mp3 | Blues_96_msg1.mp3 | 75.8143 | 70.5173 | 6.98% |
| Jazz_cover3_320.mp3 | Blues_96_msg1.mp3 | 102.7913 | 98.6331 | 4.04% |
| Pop_cover4_320.mp3 | Blues_96_msg1.mp3 | 73.4542 | 70.5211 | 3.99% |
| R&B_cover5_320.mp3 | Blues_96_msg1.mp3 | 101.4737 | 97.5227 | 3.89% |

Table 3 Results of adding AWGN to the first secret message - awgn=0.3

| Cover messages | Embedded secret message | PSNR without attack | PSNR with attack | Degredation percentage |
|---|---|---|---|---|
| Blues_cover1_320.mp3 | Blues_96_msg1.mp3 | 95.2852 | 90.1174 | 5.42% |
| Classical_cover2_320.mp3 | Blues_96_msg1.mp3 | 75.8143 | 68.1577 | 10.09% |
| Jazz_cover3_320.mp3 | Blues_96_msg1.mp3 | 102.7913 | 94.9447 | 7.63% |
| Pop_cover4_320.mp3 | Blues_96_msg1.mp3 | 73.4542 | 64.5411 | 12.13% |
| R&B_cover5_320.mp3 | Blues_96_msg1.mp3 | 101.4737 | 93.9547 | 7.40% |

It can be concluded that there is an obvious degradation in PSNR value after adding an AWGN. As shown in the tables above, the degredation in the PSNR value increases with the increase in the noise variance value.

### 5.1.2 Adding AWGN Attack to the Second Secret Audio Message

This section explores the obtained results after adding an AWGN attack to the Jazz_128_msg2.mp3 secret message with 0.1 and 0.3 bits/sec/Hz variance. The resultant degredation in the PSNR values are shown in the following two tables.

Table 4 Results of adding AWGN to the second secret message -awgn=0.1

| Cover messages | Embedded secret message | PSNR without attack | PSNR with attack | Degredation percentage |
|---|---|---|---|---|
| Blues_cover1_320.mp3 | Jazz_128_msg2.mp3 | 94.5494 | 91.4884 | 3.23% |
| Classical_cover2_320.mp3 | Jazz_128_msg2.mp3 | 79.9468 | 75.5149 | 5.54% |
| Jazz_cover3_320.mp3 | Jazz_128_msg2.mp3 | 103.2509 | 97.6331 | 5.44% |
| Pop_cover4_320.mp3 | Jazz_128_msg2.mp3 | 98.6931 | 94.5227 | 4.22% |
| R&B_cover5_320.mp3 | Jazz_128_msg2.mp3 | 76.7305 | 71.5227 | 6.78% |

Table 5 Results of adding AWGN to the second secret message awgn=0.3

| Cover messages | Embedded secret message | PSNR without attack | PSNR with attack | Degredation percentage |
|---|---|---|---|---|
| Blues_cover1_320.mp3 | Jazz_128_msg2.mp3 | 94.5494 | 88.5144 | 6.38% |
| Classical_cover2_320.mp3 | Jazz_128_msg2.mp3 | 79.9468 | 71.7444 | 10.25% |
| Jazz_cover3_320.mp3 | Jazz_128_msg2.mp3 | 103.2509 | 94.5144 | 8.46% |
| Pop_cover4_320.mp3 | Jazz_128_msg2.mp3 | 98.6931 | 90.7594 | 8.03% |
| R&B_cover5_320.mp3 | Jazz_128_msg2.mp3 | 76.7305 | 66.5447 | 13.27% |

The tables above illustrate that there is a clear degradation in the PSNR values after adding the AWGN noise. In addition, it is obvious that the degradation in the PSNR values is directly related to the noise variance value, where it is more for 0.3 variance value than that of the 0.1 value.

## 5.2 Integrity Part Results

In this stage, the hidden secret message in the cover in the presented two cases are extrcated and compared with the original secret message. This is performed using three techniques; checksum, hash function and frequency methods.

### 5.2.1. Results of Applying Checksum, Hash Function and Frequency Methods to the First Secret Audio Message

The following tables represent the achieved correlation percentage among the original first message and the extracted one without and with adding noise.

Table 6 Comparison between different integrity methods without adding noise

| Cover | msg | checksum | hash function check | frequency check |
|---|---|---|---|---|
| Blues_cover1_320.mp3 | Blues_96_msg1.mp3 | 100% | 100% | 100% |
| Classical_cover2_320.mp3 | Blues_96_msg1.mp3 | 100% | 100% | 100% |
| Jazz_cover3_320.mp3 | Blues_96_msg1.mp3 | 100% | 100% | 100% |
| Pop_cover4_320.mp3 | Blues_96_msg1.mp3 | 100% | 100% | 100% |
| R&B_cover5_320.mp3 | Blues_96_msg1.mp3 | 100% | 100% | 100% |

Table 7 Adding AWGN with 0.1 variance

| Cover | msg | checksum | hash function check | frequency check |
|---|---|---|---|---|
| Blues_cover1_320.mp3 | Blues_96_msg1.mp3 | 20.9695 % | 86.2071 % | 50 % |
| Classical_cover2_320.mp3 | Blues_96_msg1.mp3 | 21.0825 % | 91.4787 % | 50 % |
| Jazz_cover3_320.mp3 | Blues_96_msg1.mp3 | 20.9695 % | 86.2071 % | 50 % |
| Pop_cover4_320.mp3 | Blues_96_msg1.mp3 | 20.9869 % | 90.6557 % | 50 % |
| R&B_cover5_320.mp3 | Blues_96_msg1.mp3 | 20.9925 % | 81.8492 % | 50 % |

As shown, the correlation among the original message and the extrcated one is 100% when there is no added noise for the message. On the other hand, it is decreased obviously after adding a noise, where the best achieved correlation percentage for the three cover messages is by the has function check method, while the minimum achieved results are for the checksum one.

### 5.2.2. Results of Applying Checksum, Hash Function and Frequency Methods to the second Secret Audio Message

The tables below display the achieved correlation percentage among the original second message and the extracted one without and with adding noise

Table 8 Comparison between different integrity methods without adding noise

| Cover | msg | checksum | hash function check | frequency check |
|---|---|---|---|---|
| Blues_cover1_320.mp3 | Jazz_128_msg2.mp3 | 100% | 100% | 100% |
| Classical_cover2_320.mp3 | Jazz_128_msg2.mp3 | 100% | 100% | 100% |
| Jazz_cover3_320.mp3 | Jazz_128_msg2.mp3 | 100% | 100% | 100% |
| Pop_cover4_320.mp3 | Jazz_128_msg2.mp3 | 100% | 100% | 100% |
| R&B_cover5_320.mp3 | Jazz_128_msg2.mp3 | 100% | 100% | 100% |

Table 9 Adding AWGN with 0.1 variance

| Cover | msg | checksum | hash function check | frequency check |
|---|---|---|---|---|
| Blues_cover1_320.mp3 | Jazz_128_msg2.mp3 | 73.4203 % | 92.0659 % | 87.5 % |
| Classical_cover2_320.mp3 | Jazz_128_msg2.mp3 | 73.6589 % | 91.3738 % | 87.5 % |
| Jazz_cover3_320.mp3 | Jazz_128_msg2.mp3 | 73.6275 % | 92.3676 % | 87.5 % |

| | | | | |
|---|---|---|---|---|
| Pop_cover4 _320.mp3 | Jazz_128_ms g2.mp3 | 73.6275 % | 91.3676 % | 87.5 % |
| R&B_cover 5_320.mp3 | Jazz_128_ms g2.mp3 | 73.3995 % | 89.5476 % | 87.5 % |

As shown, the correlation among the original message and the extrcated one is 100% when there is no added noise for the second message. On the other hand, it is decreased after adding noises, where the best achieved correlation percentage for the three cover messages is by the hash function check method, while the minimum achieved results are for the checksum one.

## 6. Summary and Concluding Remarks

This work introduces the development of an advanced Least Significant Bit (LSB) technique; Bi-LSB to solve the low security and capacity problems of the traditionally used LSB techniques, which do not provide a step for encrypting data, and if secret message is sequentially or randomly embedded and attackers know this pattern of embedding the message, they can obtain the message. In addition, the validation code of those techniques is stored in the stego object. Therefore, the Bi-LSB technique is developed in this work to solve those problems and offer an efficient method to hide audio information in a more secured way with the use of the MATLAB program.

The developed technique includes three main steps; preprocessing, embedding and extracting and message validation. In the first stage, the main purpose is to improve the security of messages to be hidden in an MP3 file. In the second stage, the proposed algorithm is designed for MP3 files to solve the current security problem of the traditional LSB technique. In the final stage, another method is implemented to hide messages to recognize the efficiency of hidden messages in MP3 files.

The performance of the developed Bi-LSB technique is evaluated based on adding an AWGN to the stego file before extrcating messages to analyze its effect on the PSNR values and then comparing the extracted message with the original one based on checking the integrity

Results show that there is an obvious reduction in the PSNR values after adding the AWGN attack. In addition, the achieved correlation percentage for the three integrity methods; checksum, hash function and frequency check without noise is 100% for all cover messages, while it decreased with the presence of noise in which the best correlation percentage was for the hash function check method, while the minimum one was for the checksum method. Furthermore, it is shown that the degradation in the PSNR values is directly related to the noise variance value.

## References

[1] Fricker, R., and Schonlau, M., “Advantages and Disadvantages of Internet Research Surveys: Evidence from the Literature”, Field Methods, vol. 14, no. 4, PP. 347-367, doi:10.1177/152582202237725 2002.

[2] Feruza, Y., and Kim, T., “IT Security Review: Privacy, Protection, Access Control, Assurance and System Security”, International Journal of Multimedia and Ubiquitous Engineering, vol. 2, no. 2, PP. 17-32, 2007.

[3] Lenti, J., Steganographic Methods”, Department Of Control Engineering and Information Technology, Budapest University. Periodica Poltechnica Ser. El. Eng. 44, PP. 249–258, 2000.

[4] Katzenbeisser, S., and Petitcolas, F., “Information Hiding Techniques for Steganography and Digital Watermarking”, Artech House Inc, 2000.

[5] Rahim, L., B., A., Bhattacharjee, S., and Aziz, I., B., “An Audio Steganography Technique to Maximize Data Hiding Capacity along with Least Modification of Host”, In Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013) Springer Singapore , PP. 277-289, 2014.

[6] Huayin, S., and Chang-Tsun, L., “Maintaining Information Security in E-Government through Steganography”, Department of Computer Science, University of Warwick, UK, 2008.

[7] Emelia, A., Sugathan, S. K., & Ho, A., “Receiver Operating Characteristic (Roc) Graph to Determine the Most Suitable Pairs Analysis Threshold Value”, Advances in Electrical and Electronics Engineering, PP. 224-230, 2008.

[8] Francia, G. A., and Gomez, T. S., “Steganography Obliterator: An Attack on the Least Significant Bits”, Information Security Curriculum Development Conference, PP. 85-91, 2006.

[9] Liu, Q., Sung, A. H., and Qiao, M., “Novel Stream Mining for Audio Steganalysis”, ACM Multimedia Conference, PP. 95-104, 2009.

[10] Mandala, J., Kotagiri, S., and Kapala, K., “Watermarking Scheme for Color Images”, International Jo€urnal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 2 , no. 5, PP. 179-182, 2013.

[11] Maciak .L, Ponniah. M., and Sharma. R., “MP3 STEGANOGRAPHY”, 2008.

[12] Zaturenskiy. M., “Behind The Music: MP3 steganography”, [online]: available at: http://www.cpd.iit.edu/netsecure09/MIKHAIL_ZATURENSKIY.pdf, 2009.

[13] Atoum, M. S., Suleiman, M., Rababaa, A., Ibrahim, S., and Ahmed, A., “A Steganography Method Based on Hiding secrete data in MPEG / Audio Layer III”, Journal of Computer Science, vol. 11, no. 5, PP. 184-188, 2011.

[14] Bhattacharyya, S., and Sanyal, G., “Audio Steganalysis of LSB Audio Using Moments and Multiple Regression Model”, International Journal of Advances in Engineering & Technology, vol. 3, no. 1, PP. 145-160, 2011.