LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

Presented To

Mr. Md. Maruf Hassan
Associate Professor
Daffodil International University
Department of SWE

Presented By

Teertho Kamol Goshami Likhon
ID: 201-35-3032 Sec: A
Daffodil International University
Department of SWE

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

Teertho Kamol Goshami Likhon
Department of Software Engineering, Daffodil International University, Bangladesh.

Abstract:
Steganography is the practice of concealing data and is well known for masking communication's existence. This method involves inserting the covert message into a suitable medium, such as writing or audio. Due to their widespread use on the Internet, digital writing (text file) messages are the most popular among these carriers. The secret information can be incorporated in the cover message's least significant bit using the LSB alteration approach, although this method is easily exploitable. Steganography has been implemented using MATLAB software. The Least Significant Bit (LSB) alteration technique was employed in this work. There are two methods that are suggested, both of which rely on the sample bits used at the moment. These methods disseminate the covert message throughout various portions of the sample. AES encryption was used in the process as a fallback in the event that the steganography procedure failed. This won't compromise the point-to-point link between the two PCs or the confidentiality of the secret information.

Keywords: LSB, AES, Steganography, Encryption Introduction.

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

I. INTRODUCTION

The term "steganography" is a compound of the Greek words "stegos," which means "cover," and "grafia," which means "writing" [1]. Invisible communication, or covered writing, is both an art and a science. Regarding digital security, steganography and cryptography complement one another [2]. Steganography primary goal is to conceal sensitive information inside the cover item in such a way that no one else can even tell that there is a hidden message inside [3].

Text, audio, and video are just a few of the numerous types of material that can use steganography. Due to the availability of redundancy, audio and video files are regarded as ideal steganography carriers [4]. Because the human auditory system (HAS) is more perceptive than the human visual system (HVS), audio steganography is regarded as being more challenging than video steganography [5]. The human auditory system (HAS) is more sensitive than the human visual system, making audio steganography the most difficult and effective method [6]. Through the use of trusted third-party indexing keys and a secondary key that the encoder will supply, this work aims to increase security against cryptanalysis attacks. Successful audio steganography requires a technique that defeats the human auditory system (HAS). Capability and transparency are two challenges that the audio steganography technique has to address [3]. Since there is little impact on the original audio, the least significant bits (LSB) of the audio sample are substituted with the secret message bits. To improve robustness, the target bits' index might be randomly chosen [7]. Other techniques for concealing a secret message within an audio file include LSB substitution, eco-hiding, phase coding, and spread spectrum [8]. The secret message was hidden by the hairs and kept under the hairs, so only the intended recipient could decipher it.

The fundamental spread-spectrum technique makes an effort to disperse secret information as widely as feasible across the cover message frequency spectrum. The program that applies the LSB change and evenly distributes the message bits throughout the cover message Contrary to LSB modification, the spread spectrum method disperses the cover message's frequency spectrum with the hidden message using a process unrelated to the signal itself. A paradigm is put forth that increases the steganographic system's capability while also improving the confidentiality of the hidden message [13]. Additionally, a generic algorithm is put forth that chooses to bury the secret message content in the carrier audio file's most inaccessible regions [14]. By changing a few other bits, the modification error is reduced [15]. The method must satisfy transparency in order to achieve this goal. The ability is also a problem because an efficient strategy is one that can fit more secret information into the same number of cover messages. There is still a trade-off between competence and transparency, though. To sustain a technique's capacity and transparency, a proper balance needs to be maintained.

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

How it’s works?

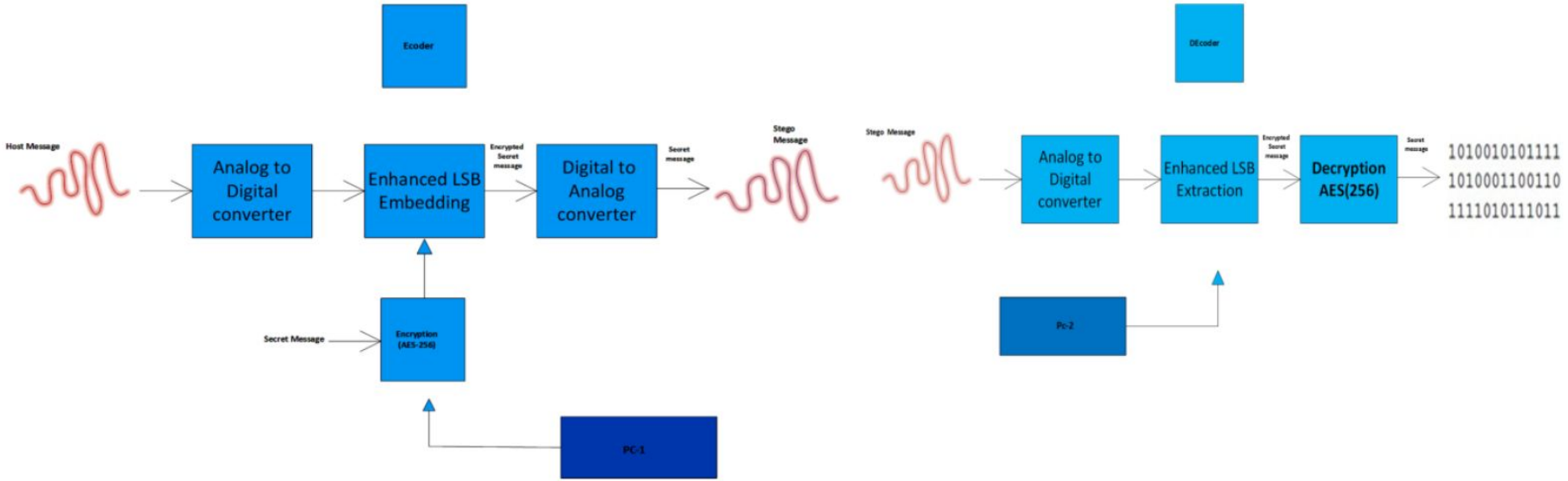


Fig.1 Block diagram

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

II. METHODOLOGY

Both the hardware and software are covered in this essay. Because it processes data more quickly and is more effective, the software-side LSB modification technique has gained popularity [16]. According to the approach, the least significant bit of each sample of the cover message should be swapped out for a bit from the hidden message. In this manner, the secret message is concealed using the LSBs of several samples from the cover message. The method has been changed to strengthen the algorithm's security and protect confidential data from prying eyes. The least significant bit of each sample of the cover audio signal was changed with a bit from the secret message using this paper's implementation of the traditional LSB modification approach.
An analog-to-digital converter processes an input audio host signal, sampling the audio at a specific sampling rate and often at 8 or 16 bits per sample. Then, a small hidden information message is put in the LSB of each sample. All of the secret message bits are concealed within the LSBs of various host message samples. The suggested methodology includes the two strategies below to combat steganalysis. It was found that modifying the first, second, and third least significant bits in a sample had no effect on the auditory characteristics of any cover audio signal. Due to their low weightage relative to other bits, altering these bits also results in roughly the same spectral visibility. However, altering other bits not only causes a spectral shift but also allows a person to hear the audio.

The suggested method uses all three of the sample's least significant bits in light of this observation. The secret information bit could be found in any one of these three bits. Therefore, a sample has no set bit that may hold secret information. Which bit in that sample would carry the secret information would be determined by the first and second most significant bits (MSBs). Assume that the 8-bit sample has the value "01001010" when using the selection of the first-bit technique. Find the secret information bit that corresponds to the first two MSBs of the number 01. The secret information bit will be placed at the second LSB in the same sample if the first two MSBs have a value of 01 in them.

TABLE I
BIT SELECTION IN FIRST TECHNIQUE

First MSB	Second MSB	Secret Information Bit
0	0	First LSB
0	1	Second LSB
1	0	Third LSB
1	1	Fourth LSB

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

The methodology used includes a steganography-supporting encryption mechanism. In the event that the steganography method is broken, encryption can be used to prevent the intrusive party from accessing confidential data. The largest key length of 256 bits is used for this purpose, making Advanced Encryption Standard (AES) the most secure encryption technique currently available.
Before being included in the cover message, the secret information is encrypted using the AES-256 technique. The AES algorithm has remained impenetrable up until this point, with keys as short as 128 bits. On the one hand, the AES also needs the additional 256 bits of the key to be supplied covertly when the data is being transmitted encrypted.

An encoder on the transmitter side takes in an audio cover signal at the input and sends it to the ADC. The audio signal is converted by the ADC into a series of samples, each of which has 8 bits. The secret data is first encrypted using AES before being inserted into these samples. The secret information bit will then be embedded in a location determined by the initial two bits. Once the entire secret message has been incorporated into the cover message, the cover message is run through a DAC to produce an audio signal from the samples. The encoder transmits the stego message, an audio signal, at the output.

On the receiver side, an audio stego message signal is received by the decoder at the input port. To obtain audio signal samples, the audio signal is transmitted through an ADC. The samples are used to extract the encrypted secret information.
The LSB extraction procedure is the same as the one used for LSB embedding on the sender side. The required secret message is created on the receiver's end by passing the extracted encrypted secret information through the decryption block.

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

III. RESULTS AND SIMULATION

The simulation section is carried out using MATLAB. For steganography, a decoder and an encoder have been created in MATLAB. The encoder's function is to transform the secret message into a format that can be communicated via a point-to-point link using AES-256. The encoder must carry out a number of tasks, including bit insertion, secret message encryption, and analog-to-digital conversion.

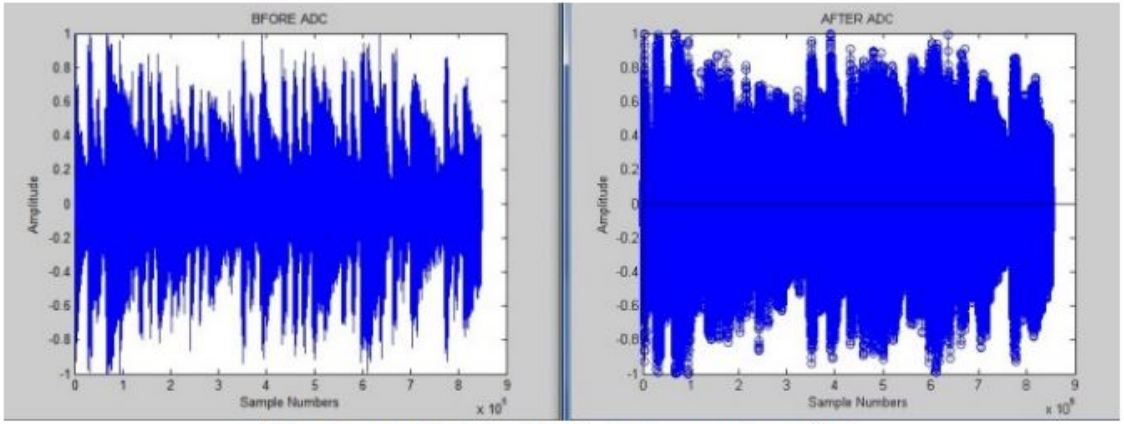
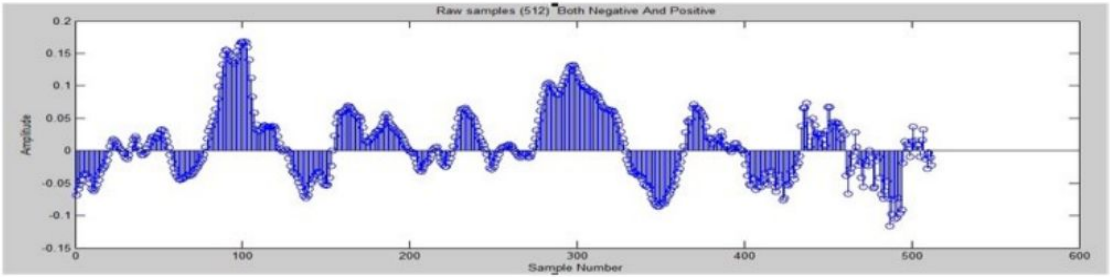


Fig. 2. Analog to Digital Conversion Plot

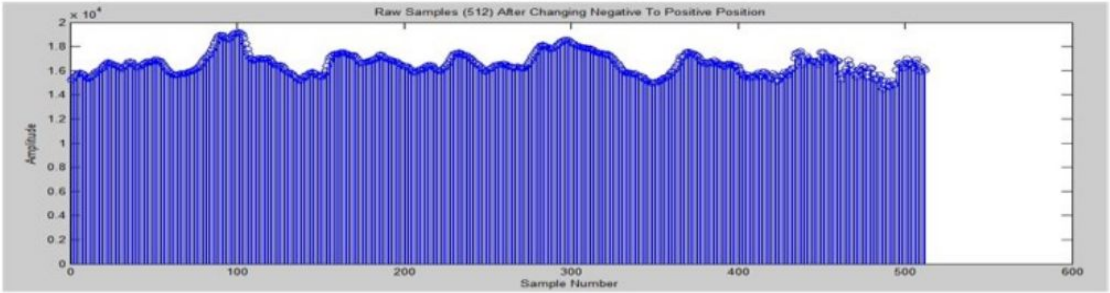
LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

A few variables that will be needed during the encoding process are being initialized on the encoder side. The most crucial variable, sec-info, contains the secret message that needs to be sent from one party to the other. Consider the scenario where the value Send Alpha Bravo Charlie to SWAT" for a terrorist operation is assigned to the variable sec-info. Steganography is being employed, and the cover message is an audio file called "host-message.wav".
Since the operations need a bit stream and sec-info is in Roman alphabets, the char type variable is transformed to double type. The conversion will be done at 8 bits per character in accordance with the ASCII representation. As a result, the entire sec-info variable, which can contain up to 64 characters, has a 512-bit bit stream.

The first function of the encoder converts a continuous cover message called "hostmessage.wav" to binary format. Each sample in the binary format has 8 bits, making up the format's number of samples. The first MSB of a sample's eight bits corresponds to the signed bit. Its amplitude is negative if its value is 1, and vice versa.
The signal is converted from analog to digital using quantization, which uses the remaining 7 bits to represent 27 levels.
The analog signal is converted by the ADC process into 128 levels, each of which is represented by a combination of 7 bits. In the following figure, the analog signal is converted to a digital bit stream using 256 defined levels.



(a).Raw samples Plot both negative and positive



(b).Raw samples Plot both negative and positive

Fig. 3. Raw samples Plot both negative and positive

However, this plot is made up of 9 million data points, in which we have inserted our message between 2000 and 2511 raw samples. As these samples fall between samples with positive and negative values.
Therefore, we execute some calculations on both sides, both before and after the LSB approach, to turn these negative samples into positive ones. The calculations' results are depicted in Fig. 3.
The encryption of the secret message kept in a sec-info variable comes after ADC. The Advanced Encryption Standard (AES) technique was used for the encryption procedure, and the largest key size that may be used is 256 bits. In AES-256, the plaintext block and ciphertext block are always 128 bits long, whereas the key is 256 bits long with 14 rounds. Each round will consist of the following 4 steps:

1. SubBytes, 2. ShiftRows, 3. MixColumns, and 4. AddRoundKey.

The 14th round does not include MixColumns, and AddRoundKey is executed prior to the start of the first round. The key and sec-info are both inputs to the AES block. Before and after the AES procedure, the value of sec-info is provided. Although the inputs and outputs from the encryption block are in binary form, the values are expressed as hexadecimal digits and characters to help with understanding.From Round 1 through Round 14, SubBytes transformation is always carried out first. Each byte in the state is changed by another byte throughout this transition. One byte is mapped to another byte in accordance with the substitution box (s-box), which has 16 rows and 16 columns. This box's creation requires the Galois field GF(28), which is a finite field and is invertible. The discussion of the Galois field and the associated abstract algebra is outside the purview of this paper. Every byte has eight bits, the final four of which are used as the row value and the first four of which are used as the column value. As indices, these row and column values are used to choose a fresh byte from the s-box.

Consider a byte with the value 10010100 as an example. In an s-box, the last four bits (0100) stand for a row, and the first four bits (1001) for a column. By utilizing the indices 4 for the row and 9 for the column, the bytes 10010100 will be replaced by 00111011. In a similar manner, 00001010 will replace another byte, 00111010. Figure 2 depicts the sub-byte translation of the left-side state matrix into the new right-side state matrix. The shift-rows transformation is depicted in Figure 4 as-

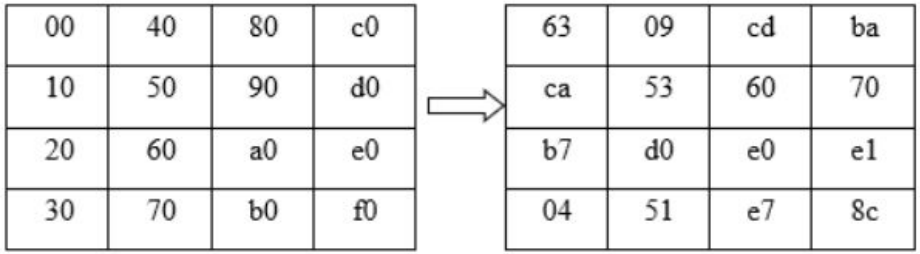
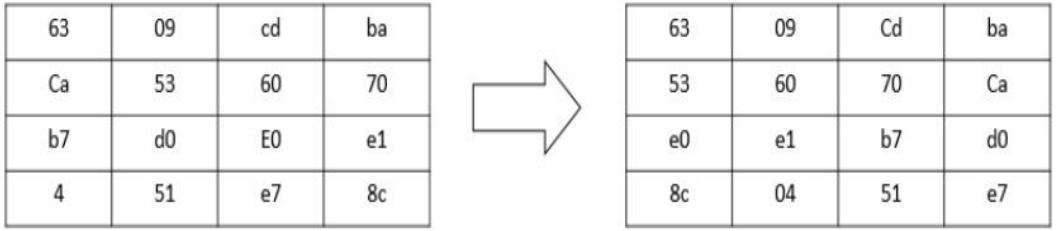


Fig. 4. Sub Bytes transformation performed on the input State matrix

from Round 1 to Round 14's second transformation, which is always carried out at the value of the state output from the SubBytes transformation. Each row, with the exception of the first row, is shifted to the left circularly. The state matrix's initial row stays the same. A left-circular shift of one, two, or three bytes is applied to the second, third, and fourth rows, respectively.
On creating the new state matrix on the right side, the ShiftRows transformation is applied to the left-side state matrix.



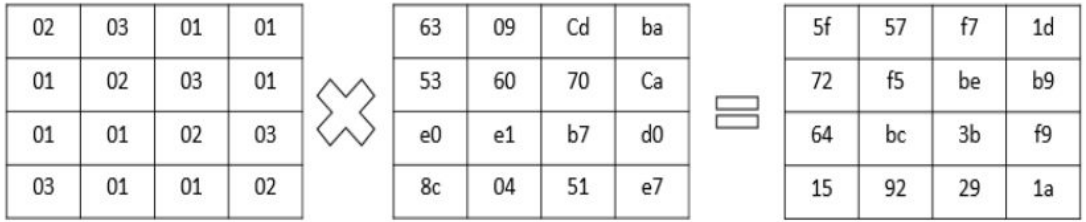
63	09	cd	ba
Ca	53	60	70
b7	d0	E0	e1
4	51	e7	8c

63	09	Cd	ba
53	60	70	Ca
e0	e1	b7	d0
8c	04	51	e7

Fig. 5. ShiftRows transformation performed on the input State matrix

The Mix Columns transformation, which is always applied to the output of the ShiftRows transformation, is absent from all but the final round. Each byte is changed to a different byte, and each column is independently transformed. A fixed matrix was obtained from the Galois field GF (28), multiplied by the state matrix. The matrix multiplication used in the Mix Columns transformation is distinct from regular multiplication. The bitwise XOR operation is used in place of adding all the partial multiplication results, as shown in Fig. 5.

Additionally, if the left-shifted bit is equal to 1, any partial multiplication outcome is bitwise XORed with "00011011." A typical multiplication operation produces the final multiplication result by summing the partial results of bit 1 multiplications.



02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

×

63	09	Cd	ba
53	60	70	Ca
e0	e1	b7	d0
8c	04	51	e7

=

5f	57	f7	1d
72	f5	be	b9
64	bc	3b	f9
15	92	29	1a

Fig. 6. Multiplication of fixed matrix with State in Mix Columns transformation

Every round's final transformation is always AddRoundKey. This metamorphosis is carried out in Round 0, an additional round, in addition to Rounds 1 through 14. The two inputs are bitwise XOR in this transformation, and each input is dependent on the round. Plain text and the 128 most important bits of the initial key are the two inputs for Round 0. Round 1's inputs are the least significant 128 bits of the starting key and the output of the Mix Columns transformation. Round 2 through the second-to-last round, the output of the Mix Columns transformation and the round key serve as inputs, and in the final round, the output of the ShiftRows transformation and the round key serve as inputs. After performing a bitwise XOR on the two inputs to this transformation, Figure 12 displays the results of the AddRoundKey transformation.

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

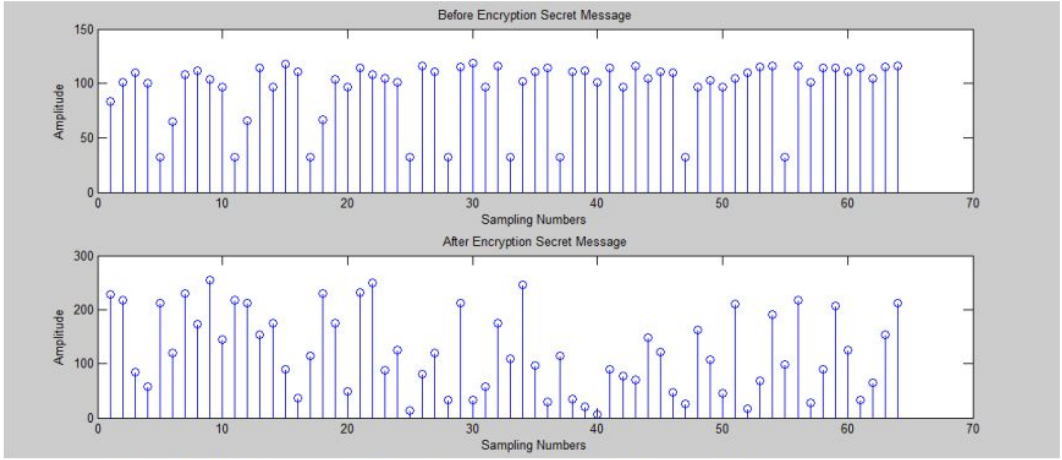


Fig. 7. Before and after Encryption of Secret Message

Simply doing a bitwise XOR between the two inputs (depending on the round) is what the AddRoundKey transformation entails. There is no distinction between the inverse AddRoundKey transformation and the AddRoundKey transformation since the bitwise XOR operation has its own inverse.

Before Encryption:

section (char) = Send Alpha Bravo Charlie to swat for an operation against terrorist section (Hexadecimal) = 83 101 110 100 32 65 108 112 104 97 32 66 114 97 118 111 32 67 104 97 114 108 105 101 32 116 111 32 115 119 97 116 32 102 111 114 32 111 112 101 114 97 116 105 111 110 32 97 103 97 105 110 115 116 32 116 101 114 114 111 114 105 115 116

Applying Key:

Keyh (In HEX) = ‘60’ ‘3d’ ‘eb’ ‘10’ ‘15’ ‘ca’ ‘71’ ‘be’... ‘2b’ ‘73’ ‘ae’ ‘f0’ ‘85’ ‘7d’ ‘77’ ‘81’... ‘1f’ ‘35’ ‘2c’ ‘07’ ‘3b’ ‘61’ ‘08’ ‘d7’... ‘2d’ ‘98’ ‘10’ ‘a3’ ‘09’ ‘14’ ‘df’ ‘f4’;

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

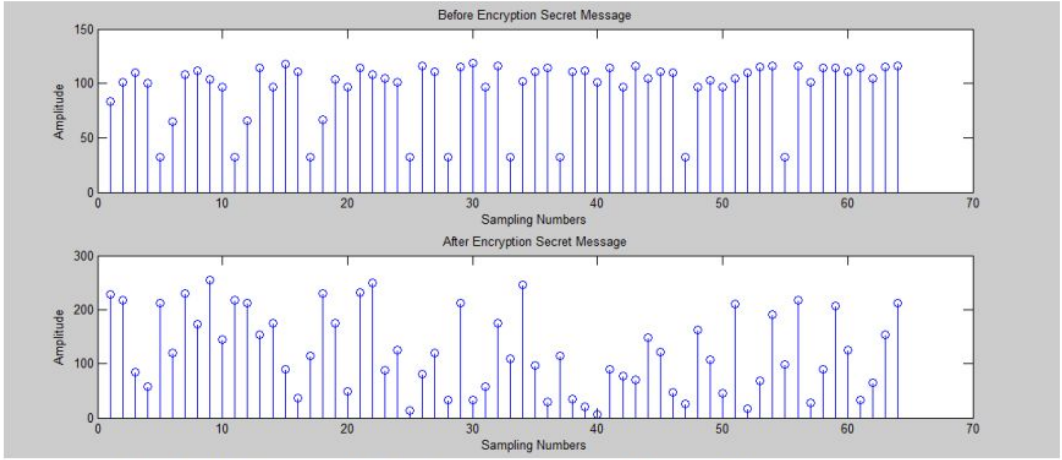


Fig. 7. Before and after Encryption of Secret Message

After Encryption:

section (Hexadecimal) = 228 218 84 57 212 120 229 173 255 145 218 213 153 175 89 37 115 229 174 48 231 249 87 126 13 80 119 33 212 33 58 174 110 246 97 29 114 34 20 7 90 77 71 148 122 47 26 163 108 45 211 17 68 190 99 218 27 89 206 125 33 65 154 212

The variable section's various values before and after encryption demonstrate the intricate link between the input and output of the encryption block. Without having access to a key, it cannot be broken. The primary operation of the encoder block, illustrated in Fig. 7, involves using the LSB replacement technique to insert the encrypted secret message into the host message.

The contents of the host message undergo a nondifferentiable change when the first three LSBs are replaced. This means that in Fig. 8, it is impossible to distinguish between a host message and a stego message if the first three LSBs are modified. Figures 9 and 10 alone are enough to back up the data used to develop our suggested methodology.

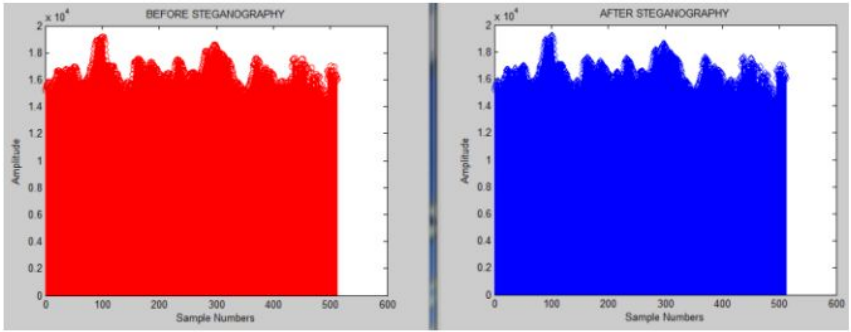


Fig. 8. Third LSB Replacement Plot

Now, when the suggested methodology is used, an improved LSB embedding is carried out. As you can see, the graph's shape is unchanged before and after the steganography.

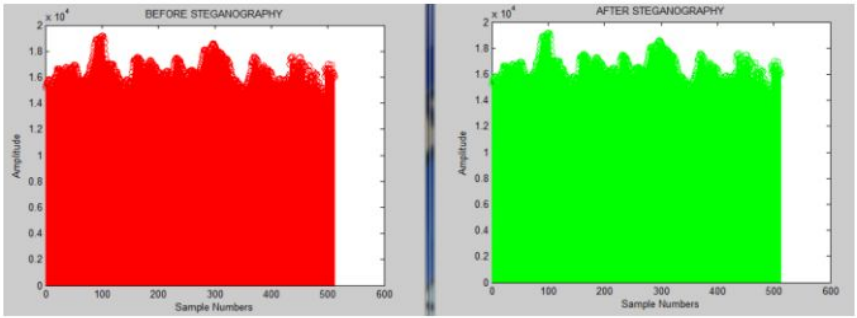


Fig. 9. Second LSB Replacement Plot

The stego message must be converted back to an audio waveform after the secret information has been incorporated into the host message and the resulting stego message has been created.

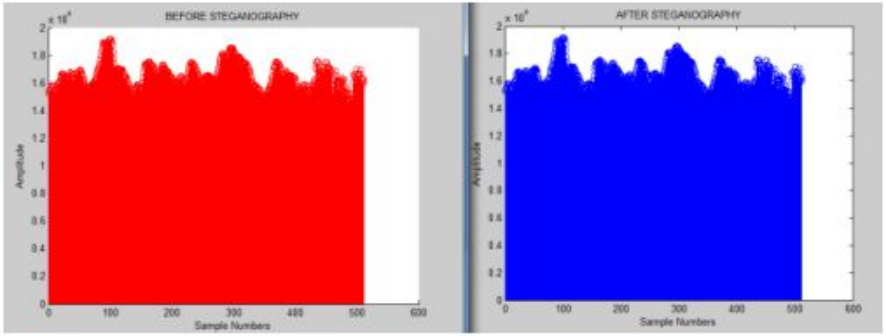


Fig. 10. Proposed Methodology Plot

Analog-to-digital conversion is carried out in reverse. The samples are mapped back in digital-to-analog conversion to create an audio file. The operation of the digital-to-analog converter is shown in Fig. 11.

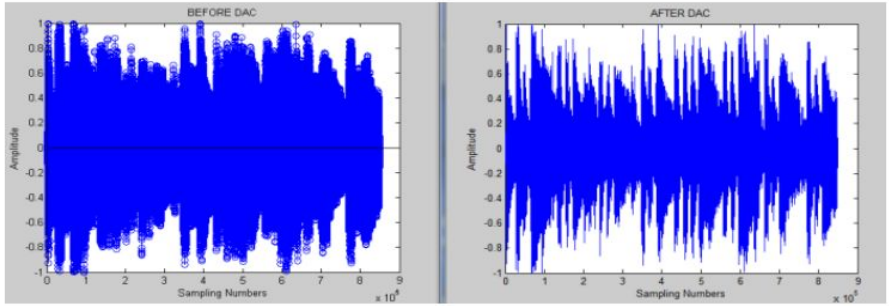


Fig. 11. Digital to Analog Conversion Plot

LSB Modification-based Audio Steganography using Advanced Encryption Standard (AES-256) Technique

IV. CONCLUSION AND FUTURE WORK

The LSB modification technique is easy to use and effective because it requires little processing and is straightforward. However, this method merely concerns itself with replacing the least significant bit in order to conceal the hidden information. In the instance of LSB, the issue that needs to be resolved is security. It becomes simple for the intruder to extract the entire secret information from the cover message if they can determine which part of the cover message has been updated with the secret information. Two strategies are described as an improved LSB methodology for the insertion of bits at random points in order to solve this issue.

Because of this, the intruder must determine the location of each bit in the cover message, which necessitates that he possess a complete understanding of the algorithm and takes more time and computer resources to defeat. The adoption of the Advanced Encryption Standard (AES) as a backup mechanism further enhances the security of the secret message. The implementation and testing of the simulation's findings on point-to-point link communication is the ultimate goal of our study. Future development of the project could result in a finished product that security organizations and others interested in secure communications could use.

REFERENCES:

[1] T. Morkel, J.H.P. Eloff, and M.S. Olivier, ”An Overview of Image Steganography,” in Proceedings of the 5th Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, June/July 2005.

[2] Dr. M. Umamaheswari, Prof. S. Sivasubramanian, and S. Pandarajan, Analysis of Different Steganographic Algorithms for Secured Data Hiding, International Journal of Computer Science and Network Security (IJCSNS), Vol. 10, No. 8, August 2010.

[3] Amritpal Singh, Harpal Singh, An Improved LSB-based Image Steganography Technique for RGB Images, IEEE International Conference on Electrical, Computer, and Communication Technologies (ICECCT), pp. 1–4, 2015 and others.

Thank You.