ENHANCED KNOWLEDGE IN SCIENCES AND TECHNOLOGY
e-ISSN: 2773-6385

EKST

Vol. 6 No. 1 (2026) 628-637
https://publisher.uthm.edu.my/periodicals/index.php/ekst

Design and Implementation of a Cryptographic Framework for Secure Communication Using Audio Steganography in Digital Music

Rachel Liew Ruo En[1], Afishah Alias[1]*

[1] *Department of Physics and Chemistry, Faculty of Applied Sciences and Technology, UTHM Kampus Cawangan Pagoh, Hab Pendidikan Tinggi Pagoh, KM 1, Jalan Panchor, 86400 Pagoh, Muar, Johor, MALAYSIA.*

*Corresponding Author: afishah@uthm.edu.my
DOI: https://doi.org/10.30880/ekst.2026.06.01.069

Article Info

Received: 4 January 2026
Accepted: 19 January 2026
Available online: 9 July 2026

Keywords

Cryptography, Audio Steganography, Secure Communication, Implementation Design, Digital Music

Abstract

In this study, a cryptography and audio steganography combined framework was designed and implemented as a channel for secure communication. With the proliferation of digital information transmission, data confidentiality is essential. To achieve this, the reseach framework performs data exchange between a sender and a receiver by embedding an encrypted plaintext within digital music. The methods used in this framework are the Advanced Encryption Standard (AES) algorithm, phase coding, and Least Significant Bit (LSB) embedding. Though these methods, a modified audio signal referred to as the stego audio is produced. At the receiver's end, decoding and decryption processes were carried out to retrieve the plaintext contents. Analysis of the framework's performance was conducted by evaluating the modified audio signal's audio quality. The analysis included Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Bit Error Rate (BER), and Cepstral Distance (CD) computations and Mel-Frequency Cepstral Coefficient (MFCC) difference heatmap plot. The resulting quantitative and graphical results demonstrated that the stego audio has acceptable overall distortion. Hence, the framework was able to transmit data effectively and confidentially. This fulfils the secrecy and authentication requirements of secure communication. Future work may include further audio quality analysis such as formal listening tests.

1. Introduction

Secure communication systems enable reliable digital data exchange and transmission. As such, these systems are required to provide secrecy, authentication, integrity, and non-repudiation [1]. To safeguard communication systems against threats of eavesdropping, data tampering and unauthorized access, enforcement strategies were researched.

A commonly used enforcement strategy is cryptography [2]. Cryptography is defined as the practice of securing information through mathematical and logical encryption algorithms [3]. In a cryptographic system, digital plaintext is encrypted into a ciphertext before transmission. Upon receiving the ciphertext, the decryption process is carried out.

Similarly, steganography is also a technique used to ensure secure communication. Steganography is the science of communicating through the transmission of message contents concealed within a cover media.

Steganography enables a form of undetectable communication. Both cryptography and steganography are strategies that can be used together to increase communication security [4].

All steganography techniques have three competing properties that offer different advantages and disadvantages in data transmission. These properties are payload capacity, imperceptibility and robustness [5]. Using audio signals as the cover media, audio steganography techniques use the Human Auditory System's (HAS's) to remain undetected [6].

Compared to standalone cryptographic or steganographic strategies, combining cryptography with audio steganography is a more effective form of secure communication. Examples of researches that combined cryptography with audio steganography are [7] and [8].

In this study, the combination of cryptography and audio steganography is implemented through the Advanced Encryption Standard with a 128-bit key (AES-128) algorithm, phase coding and Least Significant Bit (LSB) embedding. This study is carried out with three research objectives. These objectives are to develop the cryptographic framework for confidential message transmission, implement phase coding and LSB embedding to encode encrypted messages into digital music without significantly affecting the audio quality or raising detection, and lastly, asses the research framework's effectiveness and security.

The analysis of the research framework's performance is conducted through Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Bit Error Rate (BER), and Cepstral Distance (CD) computations and Mel-Frequency Cepstral Coefficient (MFCC) difference heatmap plot. By evaluating the overall audio distortion of the modified audio signal, referred to as the stego audio, and the plaintext retrieval accuracy, the research framework's effectiveness and security can be determined. Additionally, the analysis also provided a simple comparison between phase coding and LSB embedding in terms of their payload capacity, imperceptibility, and robustness.

This study aims to contribute to fields such as protected digital communication and copyright marking. Some practical examples of technologies used to achieve protected digital communication are Virtual Private Network (VPN) and password protection [9]. Overall, this study contributes to the advancing field of secure communication.

2. Research Methodology

The research methodology is based on the research framework. The research framework consists of a pre-processing stage, followed by the encoder, decoder, and performance analysis.

2.1 Research Framework

The flow of the research framework is as shown in Fig. 1. A plaintext and digital music audio file are used as the framework inputs. The framework is designed to take a digital music with a general duration of three minutes in Waveform Audio (WAV) file format.

During the audio pre-processing stage, the audio file is prepared for the framework's encoder. The audio pre-processing produces an audio output referred to as the original audio. Ciphertext pre-processing is carried out after the plaintext has been encrypted by the encoder. Through pre-processing, the ciphertext data type will become compatible with the audio file's data type.

In the encoder, the plaintext is first encrypted by the Advanced Encryption Standard with a 128-bit key (AES-128) algorithm. Before audio steganography is applied to the original audio signal, the audio's capacity is computed. As phase coding and Least Significant Bit (LSB) embedding has their own distinct payload capacities within a fixed audio duration, the total capacity is the sum of both the computed capacities. The ciphertext bit length must be able to fit within the audio capacity before audio steganography is applied. Then, the original audio will undergo the encoder's internal digital signal processing steps. The ciphertext bits are first encoded into the original audio using phase coding, followed by LSB embedding.

The decoder requires several inputs. These inputs are the stego audio and the metadata. The metadata includes the key, number used once (nonce), and authentication tag from AES-128 encryption, the ciphertext bit length, the selected frequency bins indices, and the reference phases. The metadata are the decryption parameters, the stopping point, and the phase decoding inputs.

Before decoding is carried out, the stego audio first undergoes the same audio pre-processing steps. This is to arrange the audio samples into the same frame order as the original audio. Then, phase decoding and LSB extraction are carried out. The ciphertext outputs then undergo the reversed operations of the ciphertext pre-processing stage. When the data type of the ciphertext matches the data type required for AES-128 decryption, the plaintext can be retrieved. This completes the research framework. The framework performance is then analysed.

Enhanced Knowledge in Sciences and Technology Vol. 6 No. 1 (2026) 628-637

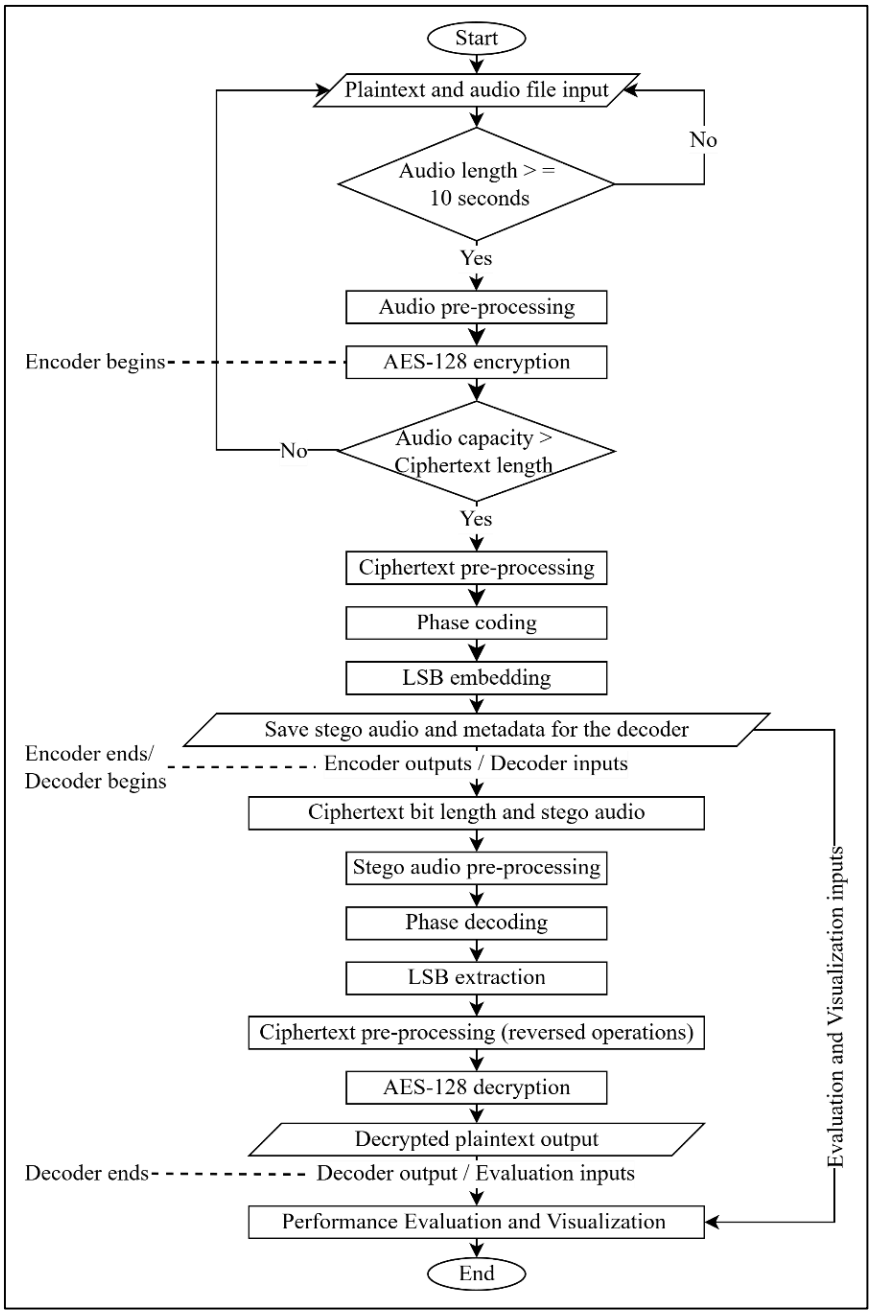

Fig. 1 Research Framework

2.2 Audio and Ciphertext Pre-processing

The audio pre-processing operations are sampling, cropping, and frame grouping. The sampling rate used is 16 kHz, the audio is cropped to 10 seconds, and the frame size used is 1024 audio samples. The design choice of using 10 seconds of audio in the framework is to limit the total encoding capacity. Encrypting long plaintexts using AES-128 requires long computational time. Therefore, using a 10 seconds duration balances between the total payload capacity for message transmissions and computational time.

After the plaintext underwent encryption, ciphertext pre-processing is carried out. The ciphertext goes through byte-to-bit data type conversion and redundancy addition. The redundancy addition is carried out by repeating each binary bit of the ciphertext a fixed number of times. The research framework uses a redundancy of three to increase the error tolerance of the decoder without taking up too much audio capacity.

The reversed ciphertext pre-processing operations are bit-to-byte conversion and redundancy removal. After the ciphertext bits are extracted in the decoder, these operations are carried out. After that, the ciphertext bytes can be decrypted.

Encoder

The Advanced Encryption Standard with a 128-bit key (AES-128) algorithm is used to encrypt the plaintext. AES-128 carries out encryption in 16 byte blocks. Therefore, the algorithm will separate the plaintext into blocks of 16 bytes, equivalent to 18 bits.

AES-128 encryption requires a key, a mode, and a plaintext. To produce a different output for each encryption, a random key is generated for each encryption. The Encrypt-then-Authenticate-then-Translate (EAX) mode is chosen for the AES-128 algorithm. the EAX mode generates a number used once (nonce) and an authentication tag for each encryption. The nonce is a random number that changes with each encryption, and the tag is generated for verification when decryption is carried out. Thus, AES-128 decryption can only be carried out when the same key, nonce, and tag are used. Because of this, the key, nonce, and tag are saved as metadata for decryption.

Before the audio steganography techniques can be applied to the original audio, the audio must undergo the encoder's internal digital signal processing steps. During these steps, the original audio is separated into two audio segments. Each audio segment is 5 seconds. The first audio segment is then prepared for phase coding by undergoing Fast Fourier Transform (FFT). The second audio segment is prepared for Least Significant Bit (LSB) embedding by undergoing data type conversion.

FFT convert the audio frames of the first audio segment from the time domain into the frequency domain. This turns the audio samples into frequency bins. The frequency bins have their own magnitudes and phases [10].

Within each audio frame, ten frequency bins are selected for phase coding. To produce the reference frame, the Root Mean Squared (RMS) of each audio frame in the first audio segment is first computed. Then, the reference frame is obtained by averaging the audio samples of the three frames with the highest RMS values. The RMS of an audio frame is computed using equation (1).

$$\text{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \tag{1}$$

where n = number of audio samples in the frame
x_i = amplitude of each audio sample
i = audio sample index

Phase coding modifies the selected frequency bins' phases to encode ciphertext bits. The phases are replaced with either a positive or negative phase offset of $\frac{\pi}{2}$ radians from the predefined reference phase to represent the binary 0 or 1 respectively. The modified phase values are computed as the new phase in equation (2):

$$\text{New phase} = \text{reference phase} \pm \frac{\pi}{2} \tag{2}$$

For real-valued time domain reconstruction, the conjugate symmetry of the spectrum must be maintained. The conjugate of the frequency bins also undergoes phase changes but do not represent the ciphertext bits. After phase coding is complete, the Inverse FFT (IFFT) is applied to obtain the phase coded audio frames.

When the phase coding capacity is completely occupied, the remaining ciphertext bits are encoded into the second audio segment using LSB embedding. This audio steganography technique is implemented through bitwise operations. For each audio sample in the designated LSB embedding frame array, the original sample's least significant bit is cleared using the AND operation with NOT 1. The sample's assigned ciphertext bit is then embedded into the sample using the OR operation. This process is repeated sequentially across samples and frames until all remaining ciphertext bits are embedded.

The encoder produces a stego audio by concatenating the two modified audio segments. The stego audio, carrying encrypted data bits, is transmitted to the decoder to retrieve the original plaintext.

2.3 Decoder

The decoder is designed to reverse the encoder operations to decode the stego audio and retrieve the original plaintext message. After stego audio pre-processing is carried out, the stego audio is separated into two audio segments. Each audio segment is 5 seconds long. Phase decoding is applied on the first stego audio segment. It is carried out by comparing the phase difference between each phase coded frequency bin to its corresponding reference phase. The phase difference equation is shown in (3).

$$\text{Phase difference} = \text{phase} - \text{reference phase} \tag{3}$$

The decoded bit is determined by comparing the absolute phase difference between the computed phase difference and the phase offsets of $\pm\frac{\pi}{2}$ radians. A computed phase difference closer to $+\frac{\pi}{2}$ radians is decoded as binary bit 0, while a computed phase difference closer to $-\frac{\pi}{2}$ radians is decoded as binary bit 1. This comparison ensures that all encoded bits within the first stego audio segment are decoded correctly.

In the second stego audio segment, Least Significant Bit (LSB) extraction is carried out. During LSB extraction, the audio samples undergo AND operation with 1. This gives the outputs of the audio samples' least significant bits. From the encoder, the least significant bits are the encoded ciphertext bits. LSB extraction is carried out following the same audio sample and frame arrangement used in the encoder.

The phase decoded bits and the LSB extracted bits are combined. The ciphertext bits undergo the reverse ciphertext pre-processing operations before proceeding to decryption. Using the saved key, nonce, and tag, AES-128 decryption is carried out. The decoder outputs the original plaintext. Therefore, the framework is completed.

2.4 Performance Analysis and Visualisation

The performance analysis and visualisations are carried out using the encoder and decoder outputs. The framework's effectiveness and security is influenced by the amount of audio distortion in the stego audio and the plaintext retrieval accuracy. Audio distortion is analysed in terms of mathematical distortion and perceptual distortion. The analysis will also compare the payload capacity, imperceptibility, and robustness of phase coding and Least Significant Bit (LSB) embedding.

In the analysis, the research framework is tested with four different audio files. These four audio files consist of male vocal, female vocal, heavy instrumentals, and string instrumentals respectively. The use of different audios is to determine the framework effectiveness. Additionally, the effect of the ciphertext bit length on stego audio quality is evaluated using three ciphertexts with different bit lengths are used. To determine the framework security, the plaintext retrieval accuracy is determined.

To analyse the stego audio's mathematical distortion, the Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are computed. The MSE measures the difference between between two different audio signals [11]. The PSNR is the representation of MSE in the logarithmic decibel scale (dB scale) [12]. (4) and (5) are the equations for MSE and PSNR respectively.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{4}$$

where $x = \{x_i | i = 1,2, \dots, N\}$
$y = \{y_i | i = 1,2, \dots, N\}$
x_i = audio sample value of the first audio signal
y_i = audio sample value of the second audio signal
i = audio sample index
N = number of audio samples in the audio

$$\text{PSNR} = 10 \cdot log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) \tag{5}$$

where MAX = maximum possible sample value of the first audio signal
MAX = 1
MSE = Mean Squared Error

In this framework, the audio data type used is float64 and the amplitude range of the audio samples are within the range of -1 to 1. Therefore, the value of MAX used for all PSNR computations is 1. Both MSE and PSNR are computed for (A, B), (A, C), and (B, C). A denotes the original audio, B denotes the audio after undergoing audio pre-processing and the encoder's internal digital signal processing steps, and C denotes the stego audio. These different types of audio outputs are used to quantify the distortion introduced by different stages of the framework.

The Bit Error Rate (BER) is used to determine the plaintext retrieval accuracy by comparing the encoded message bits and the extracted message bits. BER measures the ratio of different bits or error to the total number of transmitted bits. This is shown in equation (6).

$$\text{BER} = \frac{\text{Number of Bit Errors}}{\text{Total Number of Bits Transmitted}} \tag{6}$$

Perceptual audio distortion is evaluated using the Mel-Frequency Cepstral Coefficient (MFCC) difference plot and the Cepstral Distance (CD) measure between the original and stego audios. The MFCC difference is computed from the first thirteen cepstral coefficients (C_0 to C_{12}) of the original audio and the stego audio, while CD measures the root-mean-squared difference between the cepstral coefficients of the original audio and the stego audio on a frame-by-frame basis [13]. The CD of a frame, CD(k) is computed using the equation in (7).

$$\mathrm{CD(k)} = \sqrt{\sum_{i=1}^{M} [C_X(i,k) - C_Y(i,k)]^2} \tag{7}$$

where $C_X(i,k) = i$-th cepstral coefficient of the original audio at frame k
$C_Y(i,k) = i$-th cepstral coefficient of the stego audio at frame k
k = audio frame index
i = cepstral coefficient index
M = number of cepstral coefficients

The overall Cepstral Distance is measured by averaging CD(k) over all the audio frames. The equation of the overall Cepstral Distance is shown in (8).

$$\mathrm{CD} = \frac{1}{N}\sum_{k=1}^{N} \mathrm{CD(k)} \tag{8}$$

where N = number of audio frames

CD(k) computation generally excludes the zero-th coefficient, C_0 as it represents the overall log-energy of the audio, rather than the audio's spectral characteristics (Kubichek, 1993). Hence, in this analysis, the CD between the original audio and the stego audio are computed using C_0 to C_{12}, denoted as $CD(C_0 \text{ to } C_{12})$, then repeated with C_1 to C_{12}, denoted as $CD(C_1 \text{ to } C_{12})$, to separate the overall log-energy distortion from perceptual audio distortion.

From the outputs of the computed metrics, a low, near-zero MSE value and a PSNR value greater than 30 decibels (dB) are generally accepted to indicate minimal mathematically measured distortion between the two compared audios. However, there are some studies in audio steganography, such as the research by [12] and [14], that compute PSNR using a MAX value of 255, which differs from the MAX value used in the analysis.

Due to this variation in PSNR computation methods, the analysis supplements PSNR evaluation with the Cepstral Distance (CD) measure. The computed $CD(C_0 \text{ to } C_{12})$ is expected to be consistently higher than $CD(C_1 \text{ to } C_{12})$. This behaviour indicates that audio distortion is mainly contributed by changes in the overall signal energy rather than changes in the audio's timbre, formant, and harmonic structure.

Additionally, informal audio playback of the A, B, and C audio outputs are included as supplementary qualitative validation during framework demonstration. A formal listening test with perceptual scoring was not conducted, and the audio playback was not used for performance analysis.

To indicate that the research framework is secure, the Bit Rate Error (BER) is expected to be 0. This shows that the plaintext retrieval is fully accurate.

3. Results and Findings

The analysis of the research framework includes quantitative and qualitative results. A discussion is carried out to determine the effectiveness and security of the framework.

3.1 Quantitative Results

The quantitative results are tabulated in four tables. In Tables 1 and 4, the ciphertext bit length encoded into each audio file is 10056 bits. From the 10056 bits, the first 780 bits are encoded into the audio using phase coding. The remaining 9276 bits are encoded using Least Significant Bit (LSB) embedding.

The three different ciphertext bit lengths used in the framework performance analysis are specified in Table 2. The corresponding performance metrics computed under different payload capacities are shown in Table 3.

Table 1 Performance Metrics Between A, B, and C Audio Outputs

Audio	MSE		PSNR (dB)		BER
	(A, B)	(A, C)	(A, B)	(A, C)	
1	0.0004	0.0157	27.1818	11.5528	0
2	0.0009	0.0071	24.5318	15.4204	0
3	0.0013	0.0405	29.0121	14.1285	0
4	0.0011	0.0103	27.6663	17.8335	0

Table 1 presents the Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Bit Error Rate (BER) metrics. These metrics were evaluated between the original audio (A), the audio after undergoing audio pre-processing and the encoder's internal digital signal processing steps (B), and the stego audio (C), and were repeated across four different audio files.

From Table 1, all MSE(A, B) and MSE(A, C) values are consistently near zero. The MSE(A, B) values are closer to zero compared to the MSE(A, C) values. However, all PSNR(A, B) and PSNR(A, C) values fall below 30 decibels (dB). The PSNR(A, B) values are closer to 30 dB compared to the PSNR(A, C) values. Across the four tested audios, the BER was computed to be zero consistently.

Table 2 Encoding Capacity Distribution for Phase Coding (PC) and LSB Embedding using Audio 1

Bit Length			Capacity (%)		
PC	LSB	Total	PC	LSB	Total
768	0	768	98.46	0.00	0.95
780	9276	10056	100.00	11.61	12.47
780	79308	80088	100.00	99.29	99.30

Table 2 shows three different ciphertext bit lengths and their corresponding occupied audio capacity percentages. This part of the framework performance analysis was carried out using the first audio file only. In this table, phase coding is denoted as PC, and LSB embedding is simplified into LSB.

From Table 2, phase coding has a significantly lesser capacity compared to LSB embedding when both techniques were applied on an audio segment with a 5 second duration. The MSE, PSNR, and BER metrics resulting from the three different ciphertext bit lengths are shown in Table 3.

Table 3 Effect of Encoding Capacity on the Performance Metrics using Audio 1

Capacity (%)	MSE		PSNR (dB)		BER
	(A, B)	(A, C)	(A, B)	(A, C)	
0.95	0.0004	0.0170	27.1818	11.2307	0
12.47	0.0004	0.0157	27.1818	11.5528	0
99.30	0.0004	0.0158	27.1818	11.5504	0

Table 3 shows the MSE, PSNR and BER metrics obtained using the ciphertext bit lengths defined in Table 2. Based on this table, increasing the total occupied audio capacity from 0.95% to 12.47% to 99.30% produced minimal changes to the MSE(A, C) and PSNR(A, C), while the BER remained at zero.

Table 4 MFCC Cepstral Distances Between Original and Stego Audios

Audio	Cepstral Distance (CD)	
	$CD(C_0 \text{ to } C_{12})$	$CD(C_1 \text{ to } C_{12})$
1	26.2784	7.8121
2	27.2628	9.2192
3	21.1086	8.2000
4	28.0944	17.9136

Table 4 presents the Cepstral Distances (CD) between the original audio and the stego audio. The CD data were computed from the Mel-Frequency Cepstral Coefficient (MFCC) difference heatmap. $CD(C_0 \text{ to } C_{12})$ denotes

the use of cepstral coefficients ranging from the zero-th coefficient to the 13th coefficient, while $CD(C_1$ to $C_{12})$ denotes that CD computation started from the first cepstral coefficient up to the 13th coefficient.

Based on this table, all $CD(C_1$ to $C_{12})$ values are much lower than the $CD(C_0$ to $C_{12})$ values. The $CD(C_1$ to $C_{12})$ value of the fourth audio is shown to be higher than the $CD(C_1$ to $C_{12})$ values of the first to third audios.

3.2 Graphical Results

The differences between the original audio and the stego audio are visualised using an overlay audio waveform plot and a Mel-Frequency Cepstral Coefficient (MFCC) difference plot. The waveforms shown in Fig. 2 were plotted across the audio samples of the signal. The plot's x-axis represents the audio sample index. Since the audio has a duration of 10 seconds and a sampling rate of 16000 Hz, there is a total of 160000 audio samples. The first 80000 audio samples correspond to the first 5 seconds of the audio signal, where phase coding was applied. The remaining audio samples underwent Least Significant Bit (LSB) embedding. Fig. 3 shows the heatmap of Mel-Frequency Cepstral Coefficient (MFCC) difference between the original audio and the stego audio.

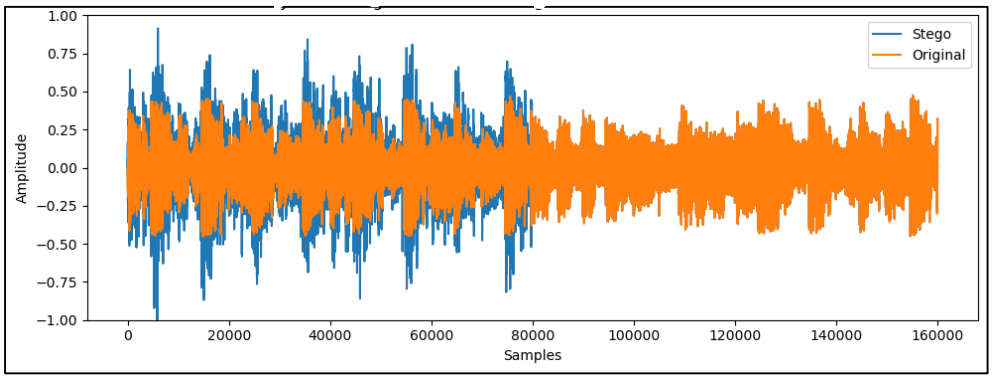

Fig. 2 Overlay of the Original Audio and Stego Audio Waveforms in the Time Domain

Fig. 2 shows the stego audio waveform overlaid with the original audio waveform in the time domain. This overlay provides a visual comparison of the amplitude changes across the audio samples before and after audio pre-processing and audio steganography were applied.

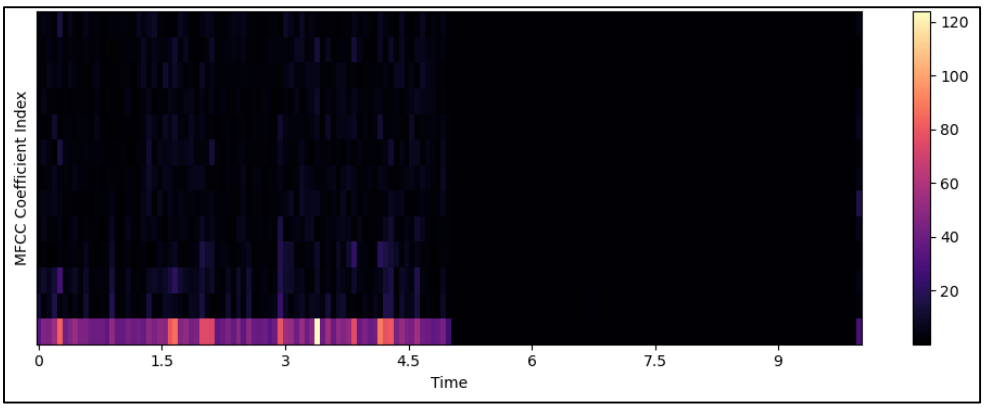

Fig. 3 MFCC Difference Between Original and Stego Audios

Fig. 3 illustrates the MFCC difference between the original audio and the stego audio using a heatmap. The y-axis of the heatmap represents the Mel-Frequency Cepstral Coefficient (MFCC) coefficients ranging from the zero-th coefficient to the 13th coefficient. The colour bar represents the magnitude of the differences between the corresponding cepstral coefficients of the two audios.

3.3 Discussion

From the quantitative results, the near zero Mean Squared Error (MSE) values indicates that the stego audios are mostly similar to their original audios. This also indicates that the applied audio steganography techniques do not deteriorate the stego audio quality significantly. Additionally, from Table 1, most of the audio distortion is produced by phase coding and Least Significant Bit (LSB) embedding. The audio pre-processing and the encoder's internal digital signal processing steps produced very little audio distortion in comparison.

Although the low, near-zero MSE values are aligned with the expectation of the research framework analysis, the Peak Signal-to-Noise Ratio (PSNR) values between the original audio and the stego audio are lower than 30 decibels (dB). This shows that there is measurable mathematical distortion after audio steganography is applied. However, direct PSNR comparison with other audio steganography studies is not carried out. This is because the MAX value of the PSNR computation in equation (5) is undisclosed or inconsistently defined in some of the studies. Therefore, in terms of mathematical distortion, the PSNR values fall below expected, while the MSE values are aligned with the research framework analysis's expectation.

Perceptual distortion within the stego audio is analysed through the Cepstral Distance (CD) values and informal audio playback of the A, B, and C audio outputs. From Table 4, it is indicated most of the perceptual distortion is resulted from overall audio signal energy changes. This suggests that according to human hearing, the detected difference between the original audio and the stego audio is mainly the audio loudness.

When the zero-th cepstral coefficient, C_0 is excluded from CD computation, the $CD(C_1 \text{ to } C_{12})$ values are comparatively lower. As the first cepstral coefficient, C_0 until the 13^{th} coefficient, C_{12} captures an audio's timbre, formants, and harmonic structure, this suggests that these audio characteristics are mostly maintained. Therefore, the changes in these characteristics are mostly undetected through human hearing. Additionally, the informal audio playback supports these findings.

From Table 4, the fourth tested audio has the highest $CD(C_1 \text{ to } C_{12})$ value. This is because the audio is composed of string instrumentals. This suggests that the audio distortion produced by audio steganography is more perceptible in softer audio signals.

Both the mathematical distortion and the perceptual distortion contribute to the overall audio distortion. Mathematical distortion is computed as absolute amplitude differences, while perceptual distortion replicates human auditorial perception of distortion. From the results, the MSE values, CD values, and informal audio playback indicate that the stego audio quality is mostly preserved, while the PSNR values points to the presence of measurable mathematical distortion.

As the PSNR is the representation of MSE in the logarithmic decibel scale (dB scale) and depends on the audio data type, the study considers PSNR values below 30 dB acceptable for the float64 audio data type and the $[-1, 1]$ audio sample amplitude range. Thus, the stego audio quality is acceptable, indicating framework effectiveness.

Throughout Tables 1 to 4, the Bit Error Rate (BER) remained at zero. This shows accurate plaintext retrieval for each tested audio files and ciphertexts. Hence, the framework is able to demonstrate secure message transmission. In the last part of the analysis, the payload capacity, imperceptibility, and robustness of phase coding and Least Significant Bit (LSB) embedding are discussed. Phase coding has a significantly smaller capacity compared to LSB embedding. Based on the graphical results, Fig. 2 and Fig. 3 show that most of the audio characteristics differences are located in the first 5 seconds of the audio. This indicates that phase coding produced most of the overall audio distortion. In contrast, LSB embedding has showed better imperceptibility.

In terms of robustness, LSB embedding is very susceptible to degradation from audio processing operations. In this aspect, phase coding has higher robustness as it does not degrade when audio processing operations are applied. In the research framework, the Advanced Encryption Standard-128 (AES-128) algorithm plays an important role in increasing the framework robustness.

Discussion of the research framework analysis determined the effectiveness and security of the framework. The framework produced a stego audio with acceptable overall audio distortion and accurate plaintext retrieval. Additionally, this discussion also highlighted the trade-offs between payload capacity, imperceptibility, and robustness in audio steganography techniques, and how using phase coding and LSB embedding can balance these properties for better framework performance.

4. Conclusion

From the research framework analysis, the Mean Squared Error (MSE) values indicated acceptable mathematical distortion within the stego audios. Although the computed Peak Signal-to-Noise Ratio (PSNR) values fall below the generally accepted threshold of 30 decibels (dB), the study accepted the PSNR values for the float64 audio data type and the $[-1, 1]$ audio sample amplitude range. From the perceptual audio distortion standpoint, the Cepstral Distance (CD) indicated that the differences between the original audio and the stego audio is mostly due to overall signal energy changes. The changes between the audio characteristics of the original audio and the stego audio are mostly undetected by human hearing. The informal audio playback also supports these findings.

Therefore, as the stego audio quality has acceptable overall distortion, the researcher framework is thus effective for undetected communication. The security of the framework is demonstrated by accurate plaintext retrieval. A Bit Error Rate (BER) of zero was achieved every time the decoding and decryption processes was carried out. Hence, the framework is secure for confidential message transmission. A limitation of the research framework is the design choice of saving the metadata externally and loading it into the decoder. The metadata contains information that is central for the decoder to function. Instead of saving it externally, it is recommended that the metadata is encoded as part of the ciphertext into the audio file. This recommendation may help to increase framework security and reduce any external dependencies. This recommendation can be included in

future work, along with formal listening test to support audio quality analysis. To conclude the results and findings, the research framework of the study fulfils the fundamental requirements of a secure communication system by ensuring secrecy and authentication during data transmission. At the same time, the study can be further improved through future work.

Acknowledgement

The authors would like to express gratitude to the Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Rachel Liew Ruo En, Afishah Alias;* ***data collection:*** *Rachel Liew Ruo En;* ***analysis and interpretation of results:*** *Rachel Liew Ruo En, Afishah Alias;* ***draft manuscript preparation:*** *Rachel Liew Ruo En, Afishah Alias. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Blahut, R. E. (2014). Cryptography and Secure Communication. Cambridge University Press. https://ahsanghazi.wordpress.com/wp-content/uploads/2017/03/252257089-blahut-cryptography.pdf

[2] Barker, Elaine, et al. (2013). A Framework for Designing Cryptographic Key Management Systems. Vol. NIST Special Publication 800-130, nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-130.pdf, https://doi.org/10.6028/nist.sp.800-130.

[3] Ramakrishnan, S. (2018). Cryptographic and Information Security Approaches for Images and Videos (1st ed.). CRC Press. https://doi.org/10.1201/9780429435461

[4] Ashok, Jammi, et al. (2010). STEGANOGRAPHY: AN OVERVIEW. International Journal of Engineering Science and Technology, vol. Vol. 2(10), ISSN: 0975-5462 (pp. 5985–5992).

[5] Macit, H. B., Koyun, A., & Güngör, O. (2018). A review and comparison of steganography techniques. IV. International Academic Research Congress (INES-2018), Alanya, Turkey.

[6] Kaur, Navneet, & Sunny Behal. (2014). Audio Steganography Techniques-A Survey. Navneet Kaur Int. Journal of Engineering Research and Applications Www.ijera.com, vol. 4, no. 6, 2014 (pp. 94–100), www.ijera.com/papers/Vol4_issue6/Version%205/P0460594100.pdf.

[7] Alwahbani, Samah M. H. & Elshoush, Huwaida T. I. (2018). Hybrid Audio Steganography and Cryptography Method Based on High Least Significant Bit (LSB) Layers and One-Time Pad—a Novel Approach. Studies in Computational Intelligence, Springer International Publishing AG (pp. 431–453). https://doi.org/10.1007/978-3-319-69266-1_21

[8] Paniker, Pooja, et al. (2020). Data Hiding Using Cryptography and Image/Audio Steganography. International Research Journal of Engineering and Technology, vol. 07, no. 05, (pp. 1565–1571), www.irjet.net/archives/V7/i5/IRJET-V7I5304.pdf.

[9] Reddy, Kukutla Tejonath. (2023). Cryptography in Action: Applications and Advantages. Insights2Techinfo, insights2techinfo.com/cryptography-in-action-applications-and-advantages/.

[10] Amin Gasmi. (2022). What Is Fast Fourier Transform? Société Francophone de Nutrithérapie et de Nutrigénétique Appliquée., hal.science/hal-03741810v1.

[11] KOYUN, Arif, & Hüseyin Bilal MACİT. (2018). GENERATING a STEGO-AUDIO DATA USING LSB TECHNIQUE and ROBUSTNESS TEST. Mühendislik Bilimleri ve Tasarım Dergisi, vol. 6, no. 1 (pp. 87–92), https://doi.org/10.21923/jesd.319330.

[12] Sayed, Mohamed H., & Talaat M. Wahbi. (2024). Information Security for Audio Steganography Using a Phase Coding Method. European Journal of Theoretical and Applied Sciences, vol. 2, no. 1, (pp. 634–647), https://doi.org/10.59324/ejtas.2024.2(1).55.

[13] Kubichek, Robert F. (1993). Mel-Cepstral Distance Measure for Objective Speech Quality Assessment. Pacific Rim Conference on Communications, Computers and Signal Processing, https://doi.org/10.1109/pacrim.1993.407206.

[14] Helmy, Mai, & Hanaa Torkey. (2025). Secured Audio Framework Based on Chaotic-Steganography Algorithm for Internet of Things Systems. Computers, vol. 14, no. 6 (pp. 207–207), https://doi.org/10.3390/computers14060207.