Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064370

ARTICLE

Interpolation-Based Reversible Data Hiding in Encrypted Audio with Scalable Embedding Capacity

Yuan-Yu Tsai[1,*], **Alfrindo Lin**[1], **Wen-Ting Jao**[1] **and Yi-Hui Chen**[2,*]

[1]Department of Communications Engineering, Feng Chia University, Taichung, 407102, Taiwan
[2]Department of Information Management, Chang Gung University, Taoyuan, 333323, Taiwan

*Corresponding Authors: Yuan-Yu Tsai. Email: yuanytsai@fcu.edu.tw; Yi-Hui Chen. Email: cyh@mail.cgu.edu.tw

Received: 13 February 2025; Accepted: 08 May 2025; Published: 09 June 2025

ABSTRACT: With the rapid expansion of multimedia data, protecting digital information has become increasingly critical. Reversible data hiding offers an effective solution by allowing sensitive information to be embedded in multimedia files while enabling full recovery of the original data after extraction. Audio, as a vital medium in communication, entertainment, and information sharing, demands the same level of security as images. However, embedding data in encrypted audio poses unique challenges due to the trade-offs between security, data integrity, and embedding capacity. This paper presents a novel interpolation-based reversible data hiding algorithm for encrypted audio that achieves scalable embedding capacity. By increasing sample density through interpolation, embedding opportunities are significantly enhanced while maintaining encryption throughout the process. The method further integrates multiple most significant bit (multi-MSB) prediction and Huffman coding to optimize compression and embedding efficiency. Experimental results on standard audio datasets demonstrate the proposed algorithm's ability to embed up to 12.47 bits per sample with over 9.26 bits per sample available for pure embedding capacity, while preserving full reversibility. These results confirm the method's suitability for secure applications that demand high embedding capacity and perfect reconstruction of original audio. This work advances reversible data hiding in encrypted audio by offering a secure, efficient, and fully reversible data hiding framework.

KEYWORDS: Reversible data hiding; encrypted audio; interpolation; sampling; multi-MSB prediction; Huffman coding

1 Introduction

With the rapid advancement of digital information technology, data hiding techniques have gained increasing prominence in the realms of information security and data protection. These techniques are extensively applied in domains such as copyright protection, digital signatures, and secure communications. Particularly within multimedia data—including audio, images, 3D models, and video—data hiding must ensure both the effective concealment of information and the preservation of multimedia quality.

Among these, image data, due to its accessibility, rich content, and ease of processing, has emerged as a primary focus for data hiding applications. The vibrant colors and intricate details inherent in image data enable the embedding of substantial amounts of hidden information without easy detection, making it a highly researched area in the field of data hiding. Furthermore, advancements in photography and image processing technology have facilitated the acquisition of high-quality images, propelling the evolution of image data hiding techniques. However, audio data also plays a critical role in our daily lives. It is omnipresent

in music, voice communications, broadcasting, and audiovisual entertainment, serving as an indispensable medium for information transmission and interpersonal communication. Audio plays a significant role in communication and user experience, with high-quality sound often contributing to listener engagement and perception. This underscores the importance of preserving the integrity and confidentiality of audio data in security-sensitive applications.

The need for audio data hiding arises from various considerations. In copyright protection, watermarks embedded in audio can prevent unauthorized copying and distribution. Watermarking techniques allow for the insertion of copyright information without compromising audio quality, effectively safeguarding audio data. In secure communications, confidential information can be embedded into ordinary audio, facilitating covert communication, protecting the privacy of the content, and reducing the risk of interception and decryption. Additionally, in fields such as law and medicine, where data integrity is paramount, audio data hiding techniques provide reliable methods for data verification, ensuring the authenticity and integrity of data during transmission.

As threats to information security intensify, merely hiding data in original media is no longer sufficient. While encryption techniques can effectively protect the privacy and security of multimedia data, they also complicate the embedding and recovery of data within the encrypted state. Traditional embedding methods often require multiple encryption and decryption operations, which reduce processing efficiency and potentially introduce additional security risks. Thus, research into reversible data hiding techniques within encrypted multimedia [1–4] has become increasingly critical. Traditional data hiding involves embedding messages into unencrypted media, aiming to conceal them without arousing suspicion from attackers. For this reason, the disguised media must closely resemble the original, making it difficult to be detected or noticed. However, this method, when applied to unencrypted media, remains vulnerable to advanced attacks or analysis that could reveal the presence of hidden information. In contrast, embedding messages in encrypted media introduces different considerations. Since encrypted media already appears as scrambled data, an attacker, even if in possession of it, cannot read the contents without decryption. The primary advantage of this approach lies in its enhanced security and privacy. In this scenario, the challenge shifts to how effectively one can increase the embedding capacity within the encrypted state while ensuring that the hidden message remains intact during decryption.

Implementing reversible data hiding techniques in encrypted audio is particularly significant. Reversible data hiding ensures that the original data can be fully recovered after embedding, which is crucial for applications demanding data authenticity. These techniques not only protect data privacy but also ensure the security and integrity of data during both embedding and extraction. This paper proposes, to the best of our knowledge, the first reversible data hiding method for encrypted audio, based on interpolation sampling techniques, capable of achieving scalable embedding capacity. Increasing the number of samples through interpolation enhances embedding capacity while maintaining audio privacy. Combining multiple most significant bit (multi-MSB) prediction and Huffman coding [5–7] further optimizes data compression efficiency and embedding capacity. Our approach enables efficient data hiding and ensures complete recovery of the original audio post-embedding.

The principal contributions of this paper are multifaceted:

- To the best of our knowledge, the proposed algorithm is the first to implement reversible data hiding specifically in encrypted audio, addressing the unique challenges posed by audio data.
- By combining multi-MSB prediction and Huffman coding, the algorithm achieves high embedding capacity while reducing transmission overhead, thereby ensuring efficient data compression and embedding.

- The algorithm employs interpolation sampling to increase the number of audio samples, thus achieving scalable embedding capacity while maintaining audio privacy.
- The proposed method ensures that the original audio can be fully recovered after data embedding, making it suitable for data authenticity and integrity applications.

The paper is organized as follows: Section 2 introduces the waveform (WAV) audio format and reviews existing data hiding techniques in audio. Section 3 presents the proposed algorithm in detail. Section 4 discusses the experimental results and analyzes the algorithm's performance. Section 5 concludes with key findings and suggests potential topics for future research.

2 Related Work

This section offers a comprehensive overview, beginning with the WAV audio format utilized in this study, highlighting its significance and structural attributes in preserving high-quality audio. Subsequently, we explore contemporary data hiding techniques tailored for audio, focusing on recent advancements and their practical applications.

2.1 Waveform Audio File Format

The WAV format, short for Waveform Audio File Format, is a standard digital audio format developed by Microsoft and IBM in 1991 as part of the Resource Interchange File Format (RIFF). Renowned for its ability to store uncompressed audio data, WAV files ensure high-quality sound reproduction, making them ideal for various audio applications. They support a range of bit resolutions, sampling rates, and audio channels, offering versatility in different contexts. A key feature of WAV files is their capacity to store raw, uncompressed audio in Pulse Code Modulation (PCM) format, which preserves the fidelity of the original sound recordings without any loss of information.

The WAV file header, structured according to the RIFF standard, defines the audio data's properties and structure. It begins with a 'RIFF' descriptor, followed by the file size and format type, typically 'WAVE'. Within this header, two critical sub-chunks are essential for audio playback: the fmt chunk and the data chunk. The fmt chunk, usually 24 bytes long, specifies audio format details such as the audio format, number of channels, sample rate, byte rate, block alignment, and bits per sample. The data chunk includes a descriptor and size, indicating the length of the audio data that follows and the actual sound data itself. While the standard WAV file header is 44 bytes, it can be larger due to additional metadata or extended format information, such as the Broadcast Wave Format (BWF) extensions used in professional applications. WAV files are renowned for their excellent sound quality, making them the preferred choice for audio professionals and enthusiasts who require lossless audio storage. Despite their large file size, their simplicity, robustness, and widespread compatibility ensure their continued relevance across various software and hardware platforms.

2.2 Data Hiding in Audio

Nassrullah et al. [8] introduced an adaptive audio steganography technique using the Least Significant Bit Substitution (LSBs) method that adjusts the number of bits embedded based on parameters such as message size and Signal-to-Noise Ratios (SNRs), effectively maximizing the carrier's capacity while maintaining low perceptual distortion. This method demonstrated robustness, supporting the embedding of large messages with minimal performance degradation. Ahmad and Fiqar [9] advanced traditional audio data hiding techniques by incorporating Newton's Divided Difference Interpolation (NDDI) to generate interpolated audio samples, significantly increasing embedding capacity. A post-embedding smoothing step minimized distortion between the original and modified samples, greatly enhancing stego audio quality.

Setiawan and Ahmad [10] employed high-order Lagrange interpolation combined with Reduced Difference Expansion (RDE) to maximize embedding capacity. This approach involved normalizing audio signals and generating new signal spaces between original samples for data embedding, followed by a smoothing algorithm to refine audio quality. Ahmad et al. [11] proposed another method focusing on discretizing and interpolating audio signals to create embedding spaces, with a subsequent smoothing step to mitigate embedding noise, substantially improving audio quality. Kuznetsov et al. [12] explored the use of direct spread spectrum technology in audio steganography, employing different spreading sequences, including Walsh-Hadamard sequences, to minimize distortion and maintain low bit error rates—ideal for applications requiring high fidelity and capacity. Feng et al. [13] presented a robust coverless audio steganography method that avoids modifying audio by mapping secret information to representative audio files selected via differential privacy clustering. It introduces an enhanced Mean-Fusion Mel-Frequency Cepstral Coefficient (MF-MFCC) feature extraction technique, combining local MFCCs with global statistics to improve resistance against attacks like time-stretching. Experimental results show over 97% extraction accuracy and up to 95% robustness, demonstrating the method's high security and practicality. Wang and Wang [14] proposed a novel audio steganography method that segments audio into foreground and background based on auditory sensitivity, allowing more bits to be embedded in the foreground with minimal perceptual impact. Using an adaptive threshold and random sampling for embedding, the method achieves high capacity (up to 24,574 bits) while maintaining strong imperceptibility and resistance to steganalysis, outperforming existing techniques in both security and efficiency.

Yu et al. [15] introduced a novel reversible data hiding method for audio signals by converting binary secret messages into novenary digits and embedding them using a magic matrix in dual-channel audio. This technique involved minimal adjustments to audio sample values, thereby preserving the original sound quality while achieving high data embedding rates. Ma et al. [16] proposed a reversible data hiding approach for audio using Code Division Multiplexing (CDM) with orthogonal spreading sequences derived from a Walsh-Hadamard matrix. The audio signal is divided into segments, and binary messages are converted and embedded using these orthogonal sequences. This technique enables high-capacity and repeated embedding while preserving audio quality, thanks to the orthogonal and zero-mean properties that also ensure full recovery of both the embedded data and the original audio. Building on this foundation, Ma and Xu [17] further developed a CDM-based reversible data hiding scheme that embeds multiple data streams without mutual interference by employing orthogonal spreading vectors. Their method incorporates a sparse prediction-error plane formed via adjacent sample prediction, allowing for enhanced embedding capacity and minimal distortion. Experimental results demonstrate the scheme's superiority over existing techniques, achieving SNRs over 83 dB at high payloads and supporting both accurate data extraction and complete audio recovery. Samudra and Ahmad [18] expanded upon prediction error expansion by integrating a dynamic sample distribution and mirroring technique, which discretized audio samples and embedded payload data with minimized distortion due to the mirroring scheme. This approach achieved high-quality stego audio and ensured the reconstruction of both the payload and the original audio, offering a practical solution for secure data transmission and multimedia security.

Hu and Zhang [19] developed Au_EliCsiNet, an advanced deep learning framework for embedding Channel State Information (CSI) within audio signals, targeting massive Multiple-Input Multiple-Output (MIMO) systems where feedback overhead is a critical concern. The framework utilized Convolutional Neural Networks (CNN) for feature extraction, embedding, and reconstruction, treating audio data as images through spectrogram conversion, and enabling efficient CSI embedding. This method significantly reduced uplink transmission resources, providing substantial benefits in real-time applications like autonomous driving by minimizing delays and optimizing resource usage. Integrating audio-based data hiding with deep

learning presents a novel approach that balances the demands for security, feedback accuracy, and system efficiency, with promising applications in modern telecommunications.

3 Proposed Algorithm

This section explores the comprehensive framework of the proposed reversible data hiding algorithm in encrypted audio. Fig. 1 presents a flowchart outlining each component, detailing their interconnections and functionalities. The discussion begins with an analysis of the preprocessing and sampling processes managed by the audio owner, which are essential for preparing the audio signal for subsequent similarity calculations and encryption. To minimize the auxiliary information required for data embedding, a label map encoding and embedding process is initiated, where all labels derived from the similarity calculation are encoded and embedded into the encrypted audio. This results in generating the encrypted audio with the embedded label map. On the data hider's side, after obtaining the labels for each audio sample, the data hiding process is described, emphasizing the use of the secret key and the secret message. Finally, the receiver's role in retrieving the original audio and the hidden message through precise audio recovery and data extraction processes is explained. Fig. 2 shows the pseudocode of all the steps for different roles in the proposed algorithm. Each aspect of this framework will be thoroughly elaborated upon in the following subsections.

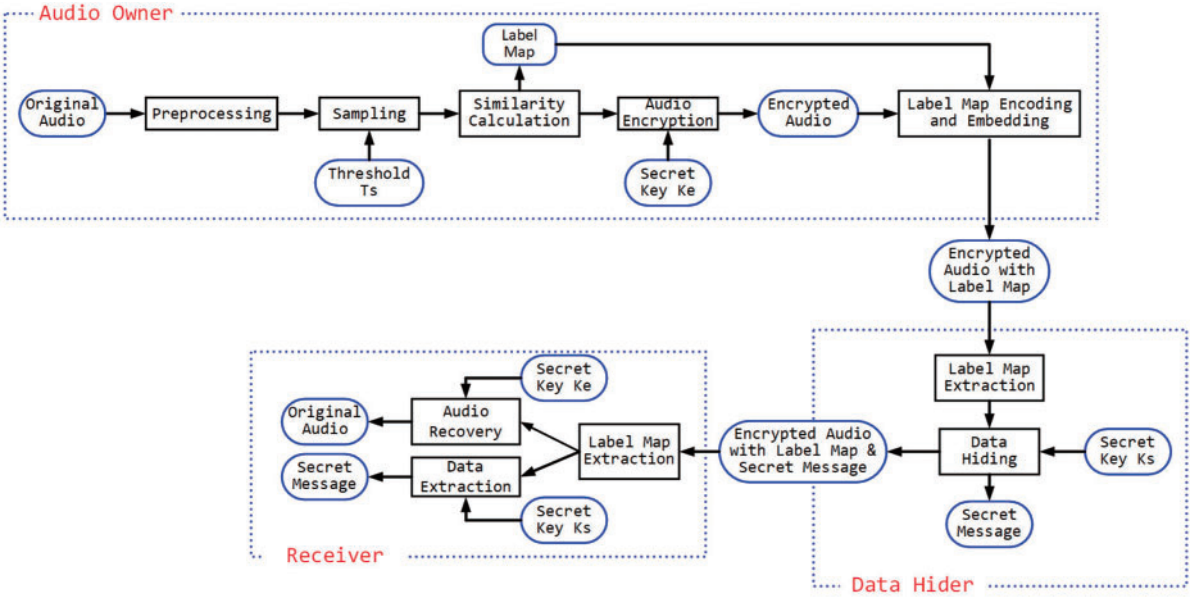


Figure 1: The comprehensive framework of the proposed reversible data hiding algorithm in encrypted audio

3.1 Preprocessing Process

In the preprocessing process, the original audio in WAV format is loaded, and detailed information about the audio is extracted. This includes metadata such as the sampling rate sr, bit depth b, and the number of channels ch. These details are critical, as they directly impact the subsequent processing of the audio data. Extracting this information allows our algorithms to dynamically adapt to various audio specifications, ensuring both broad applicability and optimal performance of the technique. The final step in the preprocessing process involves obtaining the audio sample information by breaking down the audio signal into K discrete samples DS_t, where t ranges from 1 to K. The value of K is determined by the audio's

length M in seconds, the sampling rate sr, and the number of channels ch. These samples are then analyzed to assess their suitability for data embedding.

//Audio Owner's Role
Input: Original Audio OA with length M seconds, Encryption Key K_e, Sampling Threshold T_s
Output: Encrypted Audio EA with Label Map LM Embedded
Process:
1. Preprocessing:
a. Extract metadata from OA, including sampling rate sr, bit depth b, and number of channels ch.
b. Obtain all the discrete samples DS_t from OA, where $t = 1, 2, \ldots, K$ ($K = M \times sr \times ch$).
2. Sampling:
a. For each sample with the original position p_o^i:
i. Calculate new position p_n^i using a linear mapping function shown in Eq. (1).
ii. Apply interpolation between samples DS_t to generate $(T_s - 1) \times K$ new samples.
b. Collect all samples AS_i (original + interpolated), where $i = 1, 2, \ldots, T_s \times K$.
3. Similarity Calculation:
a. For each pair of consecutive samples (AS_i, AS_{i+1}), where $i = 1, 2, \ldots, T_s \times K - 1$:
i. Add 2^{b-1} to ensure positivity of both samples.
ii. Convert both samples to b-bit binary strings.
iii. Compare the MSBs of two binary strings and calculate similarity index L_{i+1}.
iv. Calculate the embedding capacity $EC_{i+1} = min(b, L_{i+1} + 1)$ for $(i + 1)$th samples.
b. Store all the similarity index L_{i+1} in the label map LM.
4. Audio Encryption:
a. Generate a binary string using the encryption key K_e.
b. Perform XOR operation between AS_i and the binary string to produce encrypted audio EA.
5. Label Map Encoding and Embedding:
a. Compress label map LM using Huffman coding to generate a Huffman tree HT and the encoded result ER.
b. Insert Huffman Tree HT and the encoded labels of first ten audio samples (from ER) into the audio header.
c. Embed residual ER into EA using bit substitution, excluding the first audio sample as a reference.

//Data Hider's Role
Input: Encrypted Audio EA with Label Map LM Embedded, Data Hiding Key K_s, Secret Message SM
Output: Encrypted Audio EA with Label Map LM and Secret Message SM Embedded
Process:
1. Label Map Extraction:
a. Extract the Huffman tree HT and the encoded labels of first ten audio samples from the audio header.
b. Decode the encoded result ER to obtain the embedding capacity EC_i of each sample.
2. Data Hiding:
a. Encrypt the secret message SM using the data hiding key K_s to obtain the encrypted secret message SM'.
b. Embed the encrypted secret message SM' into the residual embeddable space of EA using bit substitution, according to the embedding capacity EC_i.

//Receiver's Role
Input: Encrypted Audio EA with Label Map LM and Secret Message SM Embedded, Encryption Key K_e, Data Hiding Key K_s, Sampling Threshold T_s
Output: Original Audio OA, Secret Message SM
Process:
1. Label Map Extraction:
a. Extract the Huffman tree HT and the encoded labels of first ten audio samples from the audio header.
b. Decode the encoded result ER to obtain the embedding capacity EC_i of each sample.
2. Data Extraction:
a. Extract the encrypted secret message SM' from the audio.
b. Decrypt encrypted secret message SM' using the data hiding key K_s to obtain the secret message SM.
3. Audio Recovery:
a. Recover the original audio OA by performing XOR operation on EA using the encryption key K_e.
b. Recover the MSBs of the processing sample by the value of previous audio sample.
c. Extract the original samples using the sampling threshold T_s to recover the original audio OA.

Figure 2: The pseudocode of all the processes for different roles

3.2 Sampling Process

Following the preprocessing process, sampling is critical in preparing the audio data for effective message embedding. Initially, K samples are extracted from the audio signal based on the preprocessing analysis. To enhance the embedding capacity of the proposed algorithm, we employ a linear interpolation technique to generate additional audio samples. Specifically, with a given sampling threshold T_s, we interpolate $(T_s - 1) \times K$ samples, resulting in a total of $T_s \times K$ samples. To ensure the reversibility of the original audio samples, we first map the samples from their original positions p_o^i in the audio to their new positions p_n^i in the interpolated audio using a linear mapping, as shown in Eq. (1), where $i = 1, 2, \ldots, K$. Following this, linear interpolation is applied to determine the values of the interpolated samples. This approach increases the embedding capacity and distributes the embedding load, helping preserve the original audio's perceptual characteristics.

$$p_n^i = \left\lfloor \frac{(p_o^i - 1) \times (T_s \times K - 1)}{K - 1} \right\rfloor + 1 \tag{1}$$

For instance, if we consider $K = 10$ samples from the original audio and select $T_s = 3$, the interpolation process will generate 20 additional samples. Table 1 illustrates the linear mapping results for the ten samples, displaying their original positions p_o^i and new positions p_n^i. The interpolated sample x_i^m within the ith interval, with two endpoints having the sample values $x_{p_o^i}$ and $x_{p_o^{i+1}}$, can be described by Eq. (2), where $m = 1, 2, \ldots, p_n^{i+1} - p_n^i - 1$. This equation ensures that each interpolated sample x_i^m is a linear combination of its neighboring original samples, smoothly bridging the gap between them. We refer to all the samples, including both the original and interpolated ones, as AS_i where $i = 1, 2, \ldots, T_s \times K$. This method not only increases the number of potential embedding positions but also minimizes the risk of introducing audible artifacts, as the interpolated samples maintain the natural progression of the audio waveform.

$$x_i^m = \left\lfloor x_{p_o^i} + m \times \frac{x_{p_o^{i+1}} - x_{p_o^i}}{p_n^{i+1} - p_n^i} \right\rfloor \tag{2}$$

Table 1: A linear mapping example for ten samples for their original positions p_0 and new positions p_n

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
p_o^i	1	2	3	4	5	6	7	8	9	10
p_n^i	1	4	7	10	13	17	20	23	26	30

3.3 Similarity Calculation Process

The objective of this process is to determine the embedding capacity of all the $T_s \times K$ samples. Initially, each sample, which is an integer value, is modified to ensure positivity by adding 2^{b-1} to each sample. This adjustment simplifies subsequent binary operations by avoiding complications associated with negative numbers and signed bit representations, thereby enabling a uniform approach to handling all audio samples, regardless of their original amplitude. Note that this process will be initialized in stereo audio's first and second channels, respectively.

Once all samples are adjusted, we define two consecutive samples, AS_i and AS_{i+1}, as a pair, where the first is termed the reference sample and the subsequent one the embeddable sample. These samples are then converted into their binary forms, each represented as a b-bit string (with $b = 16$ for standard audio bit depth).

The analysis involves a bit-by-bit comparison from the most significant bit (MSB) to the least significant bit (LSB). When comparing the binary strings of the reference and embeddable samples, we record the length of consecutive bits (from the MSB) that are identical between the two strings. This length, denoted as L_{i+1} where $i = 1, 2, \ldots, T_s \times K - 1$, is crucial because it represents the number of leading bits in the embeddable sample that match the reference sample. The embedding capacity for each sample is then defined as $L_{i+1} + 1$, implying that up to $L_{i+1} + 1$ MSBs of the embeddable sample can be modified arbitrarily. These changes in the MSBs can be accurately reconstructed using the reference sample, where the first L_{i+1} bits are identical, and the $(L_{i+1} + 1)$th bit is flipped to the bit value of the reference sample. Notably, the first audio sample is exempt from this similarity calculation process to preserve its original value, serving as a reference value for the decryption and recovery process. This strategy ensures that each subsequent sample can be decoded, and the original audio is fully reconstructed based on its preceding sample.

For example, consider a reference audio sample and an embeddable sample shown in Table 2, with values 12,345 and 12,360, respectively. We first add $2^{15} = 32768$ to each sample, obtaining modified values of 45,113 and 45,128, respectively. These values are then represented in binary form as 1011000001101001 and 1011000010001000. By comparing these binary strings from the MSB, we identify the first nine bits as identical, with differences beginning at the 10th bit, establishing L_{i+1} as 9. This results in an embedding capacity of $L_{i+1} + 1$, or 10, allowing the 10-bit space to be used for embedding data. This method ensures that the data can be accurately reconstructed using the nine preceding identical bits from the reference sample, with the 10th bit flipped to match the bit value of the reference sample during the extraction phase.

Table 2: Example of similarity calculation process for two consecutive samples AS_i and AS_{i+1}

	Original value	Modified value	Binary representation														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
AS_i	12345	45113	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1
AS_{i+1}	12360	45128	1	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0

Note: Green bits indicate matching positions used in multi-MSB prediction, while the red bit marks the first differing position.

After calculating the label for each sample, these labels are collectively organized into a label map. This map serves as a blueprint for the data hiding process, indicating precisely where data can be embedded within the audio stream. The label map is essential for both the embedding and extraction processes, ensuring that every modification made during data embedding can be precisely reversed during data extraction, thereby recovering both the original audio and the embedded data accurately.

3.4 Audio Encryption Process

In the audio encryption process of our algorithm, the audio samples are secured using a secret key K_e. This key is used to seed a pseudo-random number generator (PRNG), which generates a binary key stream tailored to match the length of the audio data. Each bit of this key stream is then applied to the corresponding bit of the modified audio samples through an exclusive-or (XOR) operation, resulting in an encrypted audio signal. This encryption mechanism ensures that the original audio content remains confidential and unintelligible without the correct key. Because the PRNG is deterministic, decryption is possible by regenerating the same binary stream using K_e and applying the XOR operation again. This method provides strong data confidentiality while seamlessly preparing the audio for reversible data hiding in the encrypted domain.

3.5 Label Map Encoding and Embedding Process

To efficiently embed the label map into the audio while minimizing transmission overhead between the audio owner, data hider, and receiver, we employ Huffman coding, a well-known lossless data compression technique, to compress the label map before embedding. Huffman coding optimizes the size of the label map by assigning shorter codes to more frequently occurring labels and longer codes to less frequent ones. This compression is essential, as it significantly reduces the additional data embedded in the audio, thereby preserving the capacity for embedding auxiliary information and decreasing the bandwidth required for transmission. Two encoding mechanisms are available for processing stereo audio. The first involves collecting all labels from both channels, generating a unified label map, and applying Huffman coding to the combined labels. The second method entails collecting labels from each channel separately, generating two distinct label maps, and applying Huffman coding individually to each map. The following section will discuss the experimental results comparing these two encoding mechanisms.

To facilitate the decoding process and ensure the integrity of the embedded data, the structure of the Huffman coding tree structure, along with the encoded labels of the first several audio samples, is inserted as partial auxiliary information in the audio file's header. This strategic placement allows immediate access to the decoding parameters at the beginning of the data extraction phase. Including this information in the header ensures that the receiver can efficiently reconstruct the Huffman tree and decode the label map upon receiving the audio file. After compressing the label map, the remaining encoded data is sequentially embedded into the audio samples, beginning with the second audio sample, using a bit substitution method to replace the calculated embedding length $L_{i+1} + 1$ of each processed sample AS_{i+1}. Notably, the first audio sample serves as a reference and is excluded from the data embedding process.

As a result, an encrypted audio file with the embedded label map is generated, following the subtraction of 2^{b-1} from each audio sample. The corresponding fields in the header are also modified to maintain consistency with the standard WAV format, including updating the new sampling rate $sr' = sr \times T_s$. This file not only contains the concealed data but also preserves high privacy to the original audio, with all modifications being reversible and the embedded label map optimally encoded for efficient transmission. This method ensures both security and data integrity throughout the process.

3.6 Label Map Extraction and Data Hiding Process

Once the data hider obtains the encrypted audio with the embedded label map, the data hiding process can be initiated. The initial step involves extracting the Huffman coding tree structure from the audio file header. This structure is essential for reconstructing the Huffman tree, which is pivotal in decoding the encoded label map. By understanding the Huffman coding structure for each label, we can efficiently decode the label information embedded within the audio samples. This process also includes extracting the labels for the first few audio samples, which are stored directly in the header, providing immediate data necessary for initiating the decoding process for subsequent labels.

As we continue to extract the encoded results, we systematically decode the label for each audio sample, with the modified value by adding 2^{b-1} to each sample, in the sequence. This ongoing extraction process is facilitated by the previously reconstructed Huffman tree, allowing for the accurate identification and understanding of the labeling of all audio samples. This comprehensive label map is crucial for accurately locating the embedding spaces within the audio where the secret message can be safely embedded without affecting the reversibility of each audio sample.

Following the extraction of the label map, the secret message is prepared for embedding. To ensure secure transmission, the message is first encrypted using a secret key K_s. This key is used to generate a binary

key stream via the previously described PRNG mechanism. The secret message is then XORed with this key stream to produce an encrypted message. Using a bit substitution technique, the encrypted message is then embedded into the residual spaces identified by the label map within the audio samples. This method ensures that the embedded data is integrated discreetly and securely within the audio, leveraging the embedding capacity dictated by the label map.

The culmination of this process is creating an audio file that includes the embedded label map and securely contains the encrypted secret message. This audio file stands as a testament to the efficacy of our data hiding technique, demonstrating the capability to embed substantial amounts of data within an encrypted audio file and ensuring the reversibility of both the original audio content and the embedded data upon decryption.

3.7 Audio Recovery and Data Extraction Process

In the audio recovery and data extraction process, the receiver can retrieve and reconstruct both the original audio and the embedded secret message, ensuring the integrity and security of the data using the possessed secret key. First, the receiver extracts the Huffman coding tree structure from the audio file header to reconstruct the Huffman coding scheme, which allows the decoding of the encoded label map. Concurrently, the labels for the first few audio samples, stored directly in the header, are also extracted. This initial label information provides the context for decoding subsequent audio samples' labels. To facilitate accurate recovery of the original samples, 2^{b-1} is added to each sample. This addition ensures that all samples remain positive, simplifying the reconstruction process and avoiding complications with signed integers during bit manipulation.

We then proceed to decode the remaining labels by continuously extracting the encoded results using the reconstructed Huffman tree. This process allows us to understand the labels of all audio samples, guiding us in identifying the residual embedding spaces within the audio. Following the extraction of the complete label map, we locate and extract the encrypted secret message from the residual embedding space. Using the secret key K_s, we decrypt this message to retrieve the original hidden data. The decryption process ensures the confidentiality and integrity of the embedded message, allowing it to be securely accessed by authorized parties.

With the labels and the encrypted message extracted, we begin the recovery of the original audio samples. Since the first audio sample was not altered, it serves as the initial reference sample. We recover the original $L_{i+1} + 1$ MSBs for each subsequent sample from the second to the last one, leveraging the reference relationship between consecutive samples. By sequentially recovering each sample, starting from the second, and using it to inform the recovery of the next, we ensure an error-free reconstruction of the audio.

Finally, to complete the recovery process, we subtract 2^{b-1} from each sample, returning them to their original range and values. This subtraction corrects the initial offset added earlier, ensuring that each audio sample is restored to its original state before any modifications are made. According to the sampling threshold T_s, we can extract the original audio samples from all the audio samples in the files and then alter the header content to recover the original audio files.

Since every process—interpolation, embedding, encryption, and recovery—is designed to be fully invertible, the proposed method guarantees lossless reconstruction of the original audio signal. This ensures that no distortion is introduced during the embedding and extraction phases, which is a critical requirement for reversible data hiding, especially in applications involving sensitive or high-integrity audio data.

4 Experimental Results

This section presents the experimental results of evaluating our reversible data hiding technique applied to encrypted audio. The experiments were conducted on a personal computer featuring an Intel Core i7-14700 3.40 GHz processor and 64 GB of RAM. We utilized the EBU SQAM CD as our test dataset, well-regarded for its comprehensive collection of audio samples specifically designed for subjective sound quality assessments. This dataset encompasses a diverse range of audio program signals, including 70 mono or stereo audio tracks, meticulously crafted to reveal specific impairments in analog and digital audio systems—12 of these tracks are mono, while the remaining 58 are stereo. The accompanying handbook provides detailed guidelines for evaluating various parameters such as A/D and D/A conversion linearity, aliasing distortion, bit errors, bit-rate reduction, dynamic range, frequency response, and more. We converted all tracks from FLAC to WAV format prior to conducting the algorithm evaluation. This section delves into four key aspects of our study: Section 4.1 presents the visual effects of reversible data hiding at various stages. Section 4.2 analyzes the distribution of embedding capacities as the sampling threshold T_s is varied. For these first two aspects, only the results of one test audio file are detailed, as the other audio files exhibited similar outcomes. We also examine the auxiliary information size, including Huffman tree structures and the encoded label information of the first ten samples stored in the audio header, along with the average total embedding capacity, average size of auxiliary information, and average pure embedding capacity. The detailed results for one mono and one stereo audio sample are provided, as well as the average values for all mono and stereo samples, given the consistent trends observed across the dataset. Lastly, Section 4.3 compares the performance metrics of our algorithm against existing methods.

4.1 The Visual Effects of Reversible Data Hiding at Various Stages

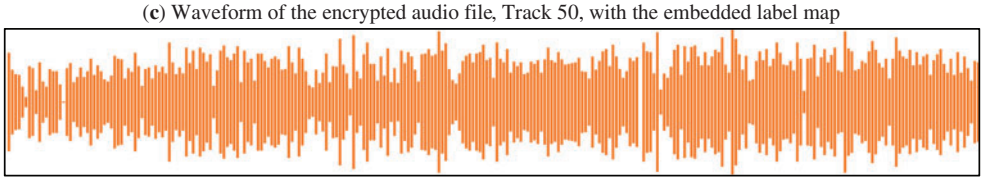
Fig. 3 illustrates the waveforms of an audio file (Track 50) at various stages of the proposed reversible data hiding algorithm. Fig. 3a presents the original audio file's unaltered waveform. In Fig. 3b, the waveform after encryption is displayed, showing a more randomized signal, which signifies that the content has been securely obfuscated. Fig. 3c depicts the waveform following the embedding of the encoded label map, introducing subtle alterations to the initial samples of the encrypted version, yet maintaining the encrypted structure's randomized appearance. Finally, Fig. 3d shows the waveform after both the encoded label map and the secret message have been embedded. This stage reveals additional modifications, further differentiating the waveform from previous stages while preserving the overall randomized pattern. This progression underscores that while the embedding process influences the waveform's structure, the proposed technique successfully conceals the data within the audio, ensuring the waveform remains visually complex and secure, albeit distinct from earlier stages.

(**a**) Waveform of the original audio file, Track 50

(**b**) Waveform of the encrypted audio file, Track 50

Figure 3: (Continued)

(**c**) Waveform of the encrypted audio file, Track 50, with the embedded label map



(**d**) Waveform of the encrypted audio file, Track 50,with the embedded label map and secret message

Figure 3: Waveform of the audio file at various stages of the proposed algorithm

4.2 Embedding Capacity Performance Evaluation

Fig. 4 illustrates the distribution of labels across audio samples as the sampling threshold T_s increases for the original audio files (Track 50). The labels on the X-axis represent different ranges of label values assigned to the processed audio samples, where lower labels (such as 1 and 2) indicate a lower degree of similarity between samples, and higher labels (such as 15 and 16) indicate a higher degree of similarity. As the sampling threshold T_s increases, the number of interpolated samples generated between each pair of consecutive original samples also increases. This interpolation enhances the similarity between adjacent samples, leading to a shift in the distribution towards higher labels. The Y-axis represents the proportion of each label within the entire set of samples, highlighting the changing distribution as T_s varies. For lower values of T_s, the distribution is more spread across a broader range of labels, indicating a more diverse set of similarities. However, as T_s increases, there is a noticeable concentration of samples within the higher label ranges (13–16), reflecting the improved similarity between samples due to the interpolation process. This trend underscores the impact of the threshold T_s on the embedding process, where higher values of T_s result in a greater proportion of samples being classified with higher labels. This distribution pattern is significant as it demonstrates the effectiveness of the interpolation process in maintaining the similarity between the audio samples while simultaneously increasing the embedding capacity.

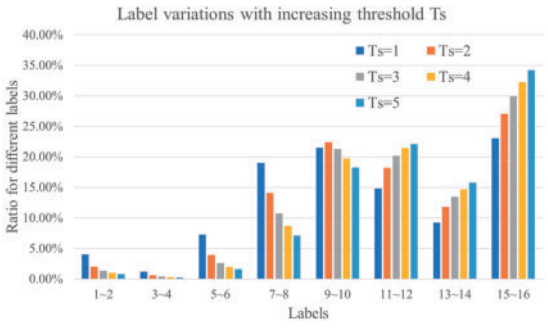


Figure 4: Label distribution with increasing threshold T_s for the original audio file, Track 50

Tables 3 and 4 comprehensively compare the experimental results for reversible data hiding in encrypted audio across both mono and stereo tracks, specifically Track 50 (mono) and Track 63 (stereo). The tables detail the impact of the sampling thresholds T_s on various performance metrics, including the number of embeddable samples, the auxiliary information size in the header (both the tree structure and the

first 10 samples), and total embedding capacity (TEC), auxiliary information (AUX), and pure embedding capacity (PEC). TEC, AUX, and PEC are represented by bits per sample. For Track 50, Table 3 shows how increasing the threshold T_s leads to a progressive increase in the number of samples, with corresponding adjustments in the performance metrics. Notably, as T_s increases, the TEC, AUX, and PEC values also increase, reflecting the algorithm's ability to embed more data. The average of all mono audios further emphasizes this trend, demonstrating a consistent increase in embedding capacity across different audio samples. Similarly, in Table 4, Channel 1 refers to the first channel and Channel 2 refers to the second channel within the stereo audio. The label map construction was conducted separately for the labels from Channel 1, Channel 2, and their combination (Channel 1 and Channel 2). Huffman coding was then applied to the labels collected from each channel individually and collectively. Although the relative experimental results for TEC, AUX, and PEC are similar across the different methods, applying Huffman coding to the labels from each channel separately shows slightly better performance. This is likely due to the subtle differences in label distribution between the two channels. As expected, the stereo audio has two channels with more samples to offer a larger embedding capacity. The increase in T_s results in more samples and higher embedding capacity, while the overhead in the header (tree and first 10 samples) remains relatively stable across different T_s values.

Table 3: The relative performance metrics for the mono audios in the dataset

Audio name	T_s	Samples	Header (Bits)		Related information (Bits per sample)		
			Tree	First 10 samples	TEC	AUX	PEC
Track 50	1	970,199	127	20	10.56	3.55	7.01
	2	1,940,399			11.42	3.43	7.99
	3	2,910,599			11.89	3.34	8.55
	4	3,880,799			12.22	3.26	8.96
	5	4,850,999			12.47	3.20	9.26
Average of All Mono Audios	1	1,374,449	127	17.50	10.80	3.32	7.47
	2	2,748,899		16.70	11.55	3.21	8.34
	3	4,123,349		15.83	11.96	3.12	8.84
	4	5,497,799		15.00	12.24	3.04	9.19
	5	6,872,249		15.00	12.45	2.98	9.46

Table 4: The relative performance metrics for the stereo audios in the dataset

Audio name	T_s	Channel	Samples	Header (Bits)		Related information (Bits per sample)		
				Tree	First 10 samples	TEC	AUX	PEC
	1	1	2,513,699		40	9.40	3.63	5.77
		2	2,513,699			9.20	3.69	5.51
		1+2	5,027,399			9.30	3.66	5.64
	2	1	5,027,399		30	10.39	3.58	6.81
		2	5,027,399			10.18	3.64	6.54
		1+2	10,054,799			10.28	3.61	6.67

(Continued)

Table 4 (continued)

Audio name	T_s	Channel	Samples	Header (Bits)		Related information (Bits per sample)		
				Tree	First 10 samples	TEC	AUX	PEC
Track 63	3	1	7,541,099	127	30	10.94	3.52	7.42
		2	7,541,099			10.73	3.59	7.14
		1+2	15,082,199			10.83	3.56	7.28
	4	1	10,054,799		30	11.32	3.48	7.84
		2	10,054,799			11.11	3.54	7.57
		1+2	20,109,599			11.21	3.51	7.71
	5	1	12,568,499		30	11.61	3.43	8.18
		2	12,568,499			11.40	3.50	7.90
		1+2	25,136,999			11.50	3.47	8.04
Average of All Stereo Audios	1	1	2,389,763	122.72	21.55	10.63	3.32	7.31
		2	2,389,763	123.14	21.72	10.60	3.33	7.27
		1+2	4,779,527	123.41	21.55	10.62	3.33	7.29
	2	1	4,779,527	122.86	19.83	11.43	3.20	8.23
		2	4,779,527	123.14	20.00	11.41	3.21	8.19
		1+2	9,559,054	123.41	19.83	11.42	3.21	8.21
	3	1	7,169,290	122.86	19.14	11.87	3.12	8.75
		2	7,169,290	123.14	18.97	11.85	3.13	8.72
		1+2	14,338,581	123.41	19.14	11.86	3.13	8.73
	4	1	9,559,054	122.86	18.45	12.17	3.06	9.11
		2	9,559,054	123.14	18.28	12.15	3.07	9.08
		1+2	19,118,109	123.41	18.45	12.16	3.06	9.10
	5	1	11,948,818	122.86	18.10	12.40	3.01	9.39
		2	11,948,818	123.14	17.76	12.38	3.02	9.36
		1+2	23,897,637	123.41	18.10	12.39	3.01	9.38

4.3 Performance Comparisons against Existing Methods

In the rapidly evolving field of data hiding techniques for audio files, our proposed algorithm marks a significant advancement, particularly highlighted in Tables 5 and 6 which compares the performance of existing and proposed algorithms. Two tables are crucial as it emphasizes the unique attributes and superior capabilities of our method, especially in terms of operation domain, embedding method, sample expansion, number of embeddable samples, embedding capacity, and reversibility. Firstly, our algorithm is, to the best of our knowledge, the first to operate on the encrypted domain, addressing a critical gap in audio data security. By implementing reversible data hiding in the encrypted domain, our method ensures the privacy of audio content while enabling the embedding of substantial additional messages, a feature not commonly supported by existing methods. This capability is particularly advantageous for secure communications where confidentiality is paramount. The tables also illustrate some existing and proposed algorithms performed for sample expansion to obtain more audio samples to significantly enhance the embedding capacity or achieve reversibility purposes. Unlike other algorithms that only perform data embedding on interpolated samples, our algorithm can perform data embedding both on the original

and interpolated samples, allowing for a more flexible and robust embedding strategy. The reversibility feature of our algorithm also stands out as a key advantage. It ensures that the original audio can be fully recovered after the extraction of the embedded data, making it suitable for applications requiring data authenticity and integrity such as legal and medical recordings where any alteration to the original content could be problematic. Overall, the combination of these features—operation on the encrypted domain, high embedding capacity, and complete reversibility—positions our algorithm as a pioneering solution in the field of reversible data hiding in encrypted audio. It protects audio privacy and provides a substantial capacity for secure message transmission, setting a new benchmark for future developments in the domain.

Table 5: Algorithm performance comparison between existing and proposed algorithms for audio files

Algorithm	[8]	[9]	[10]	[11]	[12]	[14]	[15]
Operation domain	Spatial	Spatial	Spatial	Spatial	Frequency	Spatial	Spatial
Sample expansion	N. A.	Interpolation	Interpolation	Interpolation	N. A.	N. A.	Dual-channel
Embedding method	LSB Substitution	Message Addition	Reduced Difference Expansion	Message Addition	Direct Spread Spectrum	LSB Substitution	Magic Matrix
Embeddable samples	Original Samples	Interpolated Samples	Interpolated Samples	Interpolated Samples	Original Samples	Original Samples	Original Samples
Embedding capacity	High	Medium	Medium	Medium	Low	Low	Medium
Reversibility	No	Yes	Yes	Yes	No	No	Yes

Table 6: Algorithm performance comparison between existing and proposed algorithms for audio files

Algorithm	[16]	[17]	[18]	[19]	[20]	**Proposed**
Operation Domain	Spatial	Spatial	Spatial	Frequency	Spatial	Encryption
Sample Expansion	N. A.	N. A.	N. A.	N. A.	N. A.	Interpolation
Embedding Method	Code Division Multiplexing	Code Division Multiplexing	Prediction Error Expansion	CNN	Message Addition	Bit Substitution
Embeddable Samples	Original Samples	Original Samples	Original Samples	Original Samples	Original Samples	All Samples
Embedding Capacity	Low	Low	Medium	Medium	Medium	High
Reversibility	Yes	Yes	Yes	No	Yes	Yes

5 Conclusion and Future Work

This study has presented a significant advancement in reversible data hiding for encrypted audio, establishing a new paradigm for secure audio data handling that preserves both confidentiality and integrity. By leveraging interpolation sampling, multi-MSB prediction, and Huffman coding, our approach achieves high embedding capacity while ensuring perfect recovery of the original audio. Crucially, the ability to operate directly within the encrypted domain enhances security and enables seamless integration into sensitive applications such as legal documentation and secure communications.

Looking forward, future work will focus on several key directions. First, although the system is modular, parameter tuning (e.g., sampling threshold T_s) and adaptation to various audio formats or bit depths may

require customization. We plan to investigate automatic parameter optimization and real-time adaptation to improve deployment across diverse applications, such as streaming or communication systems. Second, a comprehensive security analysis is needed to identify potential vulnerabilities in interpolation, label map embedding, and data hiding mechanisms. Third, while Huffman coding improves efficiency, its overhead may affect performance in real-time or resource-constrained scenarios; thus, further optimization will be considered. Fourth, we will evaluate the robustness of the embedding process under common signal processing operations (e.g., compression, noise, format conversion). Lastly, adaptive embedding strategies will be explored to dynamically adjust payload size without compromising reversibility or data integrity.

Acknowledgement: The authors gratefully acknowledge the thoughtful comments and guidance provided by the anonymous reviewers and the editor, which greatly contributed to the improvement of this work.

Funding Statement: This work was funded by the National Science and Technology Council of Taiwan under the grant number NSTC 113-2221-E-035-058.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Yuan-Yu Tsai and Alfrindo Lin; software, Yuan-Yu Tsai, Alfrindo Lin, Wen-Ting Jao, and Yi-Hui Chen; validation, Yuan-Yu Tsai, Alfrindo Lin, Wen-Ting Jao, and Yi-Hui Chen; writing—original draft preparation, Yuan-Yu Tsai, Alfrindo Lin, Wen-Ting Jao, and Yi-Hui Chen; writing—review and editing, Yuan-Yu Tsai and Yi-Hui Chen; supervision, Yuan-Yu Tsai and Yi-Hui Chen; project administration, Yuan-Yu Tsai; funding acquisition, Yuan-Yu Tsai. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding authors, Yuan-Yu Tsai and Yi-Hui Chen, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Chen CC, Chang CC, Chen K. High-capacity reversible data hiding in encrypted image based on Huffman coding and differences of high nibbles of pixels. J Vis Commun Image Represent. 2021;76(8):103060. doi:10.1016/j.jvcir.2021.103060.
2. Gao G, Zhang L, Lin Y, Tong S, Yuan C. High-performance reversible data hiding in encrypted images with adaptive Huffman code. Digit Signal Prog. 2023;133(1):103870. doi:10.1016/j.dsp.2022.103870.
3. Tsai YY, Liu HL. Integrating coordinate transformation and random sampling into high-capacity reversible data hiding in encrypted polygonal models. IEEE Trans Dependable Secur Comput. 2023;20(4):3508–19. doi:10.1109/TDSC.2022.3204291.
4. Tsai YY, Liu HL, Kuo PL, Chan CS. Extending multi-MSB prediction and Huffman coding for reversible data hiding in encrypted HDR images. IEEE Access. 2022;10(5A):49347–58. doi:10.1109/ACCESS.2022.3171578.
5. Yang Y, He H, Chen F, Yuan Y, Mao N. Reversible data hiding in encrypted images based on time-varying Huffman coding table. IEEE Trans Multimed. 2023;25(4):8607–19. doi:10.1109/TMM.2023.3238549.
6. Yin Z, She X, Tang J, Luo B. Reversible data hiding in encrypted images based on pixel prediction and multi-MSB planes rearrangement. Signal Process. 2021;187(2):108146. doi:10.1016/j.sigpro.2021.108146.
7. Yin Z, Xiang Y, Zhang X. Reversible data hiding in encrypted images based on multi-MSB prediction and Huffman coding. IEEE Trans Multimed. 2019;22(4):874–84. doi:10.1109/TMM.2019.2936314.
8. Nassrullah HA, Flayyih WN, Nasrullah MA. Enhancement of LSB audio steganography based on carrier and message characteristics. J Inf Hiding Multim Signal Process. 2020;11(3):126–37.
9. Ahmad T, Fiqar TP. Enhancing the performance of audio data hiding method by smoothing interpolated samples. Int J Innov Comput Inf Control. 2018;14(3):767–79.

10. Setiawan R, Ahmad T. Improving the capacity of data hiding by modifying the interpolation of audio samples. In: 2020 Proceedings of 2020 8th International Conference on Information and Communication Technology; 2020 Jun 24–26; Yogyakarta, Indonesia. Piscataway, NJ, USA: IEEE; 2020. p. 1–6.
11. Ahmad T, Amrizal MH, Wibisono W, Ijtihadie RM. Hiding data in audio files: a smoothing-based approach to improve the quality of the stego audio. Heliyon. 2020;6(3):e03464. doi:10.1016/j.heliyon.2020.e03464.
12. Kuznetsov A, Onikiychuk A, Peshkova O, Gancarczyk T, Warwas K, Ziubina R. Direct spread spectrum technology for data hiding in audio. Sensors. 2022;22(9):3115. doi:10.3390/s22093115.
13. Feng Y, Xu L, Lu X, Zhang G, Rao W. A robust coverless audio steganography based on differential privacy clustering. IEEE Trans Multimed. 2025. doi:10.1109/TMM.2025.3543107.
14. Wang J, Wang K. A novel audio steganography based on the segmentation of the foreground and background of audio. Comput Electr Eng. 2025;123(17):110026. doi:10.1016/j.compeleceng.2024.110026.
15. Yu H, Wang R, Dong L, Yan D, Gong Y, Lin Y. A high-capacity reversible data hiding scheme using dual-channel audio. IEEE Access. 2020;8:162271–8. doi:10.1109/ACCESS.2020.3015851.
16. Ma B, Hou JC, Wang CP, Wu XM, Shi YQ. A reversible data hiding algorithm for audio files based on code division multiplexing. Multimed Tools Appl. 2021;80(12):17569–81. doi:10.1007/s11042-021-10532-9.
17. Ma R, Xu J. A high-performance reversible data hiding scheme for audio streams base on code division multiplexing. IEICE Commun Express. 2025;14(3):107–10. doi:10.23919/comex.2024XBL0191.
18. Samudra Y, Ahmad T. Improved prediction error expansion and mirroring embedded samples for enhancing reversible audio data hiding. Heliyon. 2021;7(11):e08381. doi:10.1016/j.heliyon.2021.e08381.
19. Hu Y, Zhang R. Deep data hiding-based channel state information feedback within audio signals. Electron Lett. 2023;59(13):e12854. doi:10.1049/ell2.12854.
20. Samudra Y, Ahmad T. Segmentation embedding method with modified interpolation for increasing the capacity of adaptable and reversible audio data hiding. J King Saud Univ-Comput Inf Sci. 2023;35(8):101636. doi:10.1016/j.jksuci.2023.101636.