applied sciences

MDPI

Article

Imperceptible and Reversible Acoustic Watermarking Based on Modified Integer Discrete Cosine Transform Coefficient Expansion

Xuping Huang [1,2,*] and Akinori Ito [1]

[1] Department of Communications Engineering, Graduate School of Engineering, Tohoku University, Sendai 980-8577, Japan; aito.spcom@tohoku.ac.jp
[2] Interdisciplinary Faculty of Science and Engineering, Shimane University, Matsue 690-8504, Japan
* Correspondence: huang.xuping.q8@dc.tohoku.ac.jp

Abstract: This paper aims to explore an alternative reversible digital watermarking solution to guarantee the integrity of and detect tampering with data of probative importance. Since the payload for verification is embedded in the contents, algorithms for reversible embedding and extraction, imperceptibility, payload capacity, and computational time are issues to evaluate. Thus, we propose a reversible and imperceptible audio information-hiding algorithm based on modified integer discrete cosine transform (intDCT) coefficient expansion. In this work, the original signal is segmented into fixed-length frames, and then intDCT is applied to each frame to transform signals from the time domain into integer DCT coefficients. Expansion is applied to DCT coefficients at a higher frequency to reserve hiding capacity. Objective evaluation of speech quality is conducted using listening quality objective mean opinion (MOS-LQO) and the segmental signal-to-noise ratio (segSNR). The audio quality of different frame lengths and capacities is evaluated. Averages of 4.41 for MOS-LQO and 23.314 [dB] for segSNR for 112 ITU-T test signals were obtained with a capacity of 8000 bps, which assured imperceptibility with the sufficient capacity of the proposed method. This shows comparable audio quality to conventional work based on Linear Predictive Coding (LPC) regarding MOS-LQO. However, all segSNR scores of the proposed method have comparable or better performance in the time domain. Additionally, comparing histograms of the normalized maximum absolute value of stego data shows a lower possibility of overflow than the LPC method. A computational cost, including hiding and transforming, is an average of 4.884 s to process a 10 s audio clip. Blind tampering detection without the original data is achieved by the proposed embedding and extraction method.

Keywords: audio watermarking; modified integer DCT coefficient expansion; reversibility and imperceptibility

Citation: Huang, X.; Ito, A. Imperceptible and Reversible Acoustic Watermarking Based on Modified Integer Discrete Cosine Transform Coefficient Expansion. *Appl. Sci.* **2024**, *14*, 2757. https://doi.org/10.3390/app14072757

Academic Editor: Tomasz Figlus

Received: 12 February 2024
Revised: 6 March 2024
Accepted: 6 March 2024
Published: 25 March 2024

1. Introduction

Detecting tampering with digital data is essential for ensuring the authenticity and integrity of digital content, especially in surveillance and legal applications [1]. Various techniques for tampering detection have been developed in fields involving images [2], videos [3–5], audio [6], and 3D data streams [7].

We focus on audio tampering detection, a crucial aspect of digital forensics that has numerous applications in the legal and business fields. For instance, it can be used to verify the authenticity of audio evidence in court cases, detect fraudulent activities in financial transactions, and ensure the integrity of audio recordings in business meetings [8]. There are two types of audio tampering detection: passive methods [6] and methods based on information hiding [9–13]. Passive methods extract the environmental features recorded in the audio signal with its contents, such as the microphone's features [14], background noise [15], or electric network frequency [16,17]. The information-hiding-based methods

embed secret data in the audio signal, and the embedded data are used to verify the audio signal's integrity. Among these methods, we are developing a tampering detection method based on information hiding. In particular, we aim to develop a tamper detection method to locate the tampered part in the audio signal.

Information hiding methods for tamper detection must meet several requirements. Since the integrity of the original data is essential, the hiding algorithm should be reversible [18]. In this paper, we mainly focus on reversible audio watermarking, and we list the interpretations of the keywords of this paper as follows:

- Stego data: Stego data with hidden information are generated by an information-hiding algorithm. The hidden data may be secret media, a fingerprint, etc.
- Hash value: A hash value, a fixed-length unique numerical value, is generated by a particular cryptography algorithm, such as the MD5 and SHA1 algorithms. It can be used as a fingerprint for digital content.
- Residual: Residuals are floating numbers expressing predictor coefficients generated by the Linear Predictive Coding (LPC) algorithm. The value is the difference between the original value and the predicted value.
- Blind detection: Tampering detection can be processed without the original data.

To detect tampering with data with probative importance, the following tasks should be solved.

- Reversibility: After applying the proposed embedding and extraction algorithm, the original data can be extracted and re-constructed without data loss.
- Imperceptibility: This means that distortion should be controlled to be low enough after embedding the payload for integrity verification.
- High capacity: This means the positions reserved for embedding should be adequate to embed digested information for verification, such as hash value, fingerprint, and the necessary information for extraction. The algorithm for the hash value is as long as 128 bits, 160 bits, etc.

Reversible watermarking or reversible data hiding methods embed extra information in the cover data to completely recover the original cover data from the stego data. In addition to reversibility, the difference between the original and stego signals must be imperceptible so that the forgers do not perceive the anti-tampering information embedded in the audio signal. Other important aspects are blind detection and embedding capacity. Blind detection means we do not need additional information, such as the original signal, for detecting hidden information. Different embedding methods have different embedding capacities; we must ensure adequate capacity to embed the hash information for the input signal. On the contrary, robustness is not considered for this purpose because we expect the embedded information to be changed with a subtle manipulation such as splicing, encoding, and noise addition [19].

To guarantee the integrity of the original data with probative importance, Huang et al. proposed the principle of framewise tampering blind detection [12], which is a previous work presented at a conference by the first author of this paper. However, the method is so simple that complete reversibility is not guaranteed. Therefore, this paper extends this method to be entirely reversible. In addition, we investigated the most inaudible way of embedding the information.

This paper proposes an entirely reversible audio information-hiding method based on intDCT in a framework to achieve blind tampering detection imperceptibly with high capacity. Since the proposed method enables blind detection, tampering detection is achieved without sharing the original data in advance.

Figure 1 illustrates a flow diagram, or block diagram, to present the insertion and extraction algorithms. Integrity and reliability can be verified without the original data. On the verification side, the information for verifying the reconstructed original data is compared to the extracted information for verifying the original data. This makes blind verification possible.

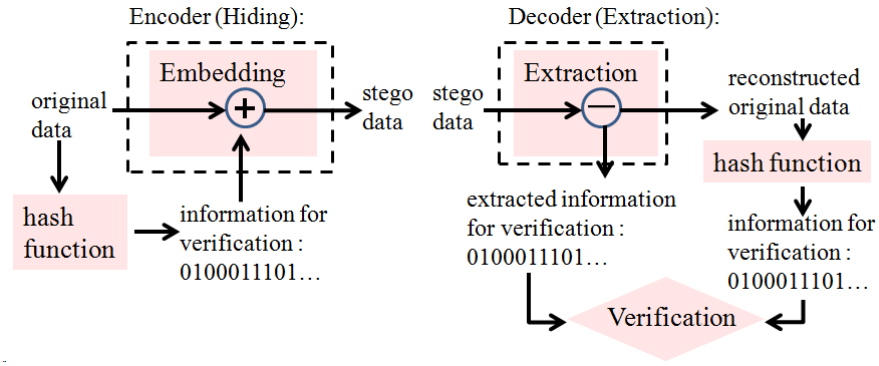

Figure 1. Flowchart of tampering detection using information hiding.

The novelty of the proposed paper is to propose a reversible audio information-hiding method based on intDCT in framewise to achieve blind tampering detection imperceptibly with high capacity. A location map is proposed to explore and record the appropriate locations for embedding, which can be extended and applied to hiding location optimization as further work. Location maps can determine whether or not to embed data into segments that may overflow after expansion. This is superior to the conventional works [12,20], which may have data loss when overflow occurs since the data is discarded if overflow, which harms reversibility. Blind tampering detection is also achieved without sharing original data in advance, since the payload for verification can be calculated from the reconstructed original data.

This paper is organized as follows. The approaches taken here and those of previous studies are discussed in Sections 1 and 2. Section 3 describes the conventional information-hiding method and its problems and proposes the proposed method in detail. Section 4 summarizes the experimental evaluation results. We discuss the results in Section 5 and conclude the paper in Section 6.

2. Conventional Work

2.1. Reversible Audio Data Hiding

Most conventional reversible audio data hiding methods hide information in the time domain [20–23]. We surveyed reversible information-hiding methods in the time domain as follows. Aoki proposed a technique to hide data in sign bits [21]. Yan et al. proposed a method to expand the residual between the predicted and original signals based on Linear Predictive Coding (LPC) [22]. Nishimura [20] extended the algorithm from Yan et al.'s work. Unoki proposed a method to hide data in phase information [23]. The LPC-based hiding method is an effective method with high capacity. However, the LPC-based method has several disadvantages for the tampering detection approach. The algorithm aims to expand the residual to reserve hiding space, which means multiplying the residual by two and adding embedded data to the results, since residual values are floating numbers expressing predictor coefficients for a certain length of data. For blind extraction, the predictor coefficients should be embedded as a part of the payload in the embedding phase. However, each frame generates floating residuals, which are difficult to embed into the original data. Therefore, the LPC-based method assumes that two parties at the sending and receiving sides share the residual signal beforehand, which means that the LPC-based algorithm cannot achieve blind embedding. Moreover, it is difficult or impossible to predict certain types of data with stable and invariable frequency, such as white noise. In this case, residuals cannot be expanded to reserve hiding space.

Alternatively, another method based on the discrete cosine transform (DCT) supplies an alternative domain to hide information. In the image field, the integer DCT (intDCT) is popularly used to achieve reversible information hiding [24–26] to embed data by modifying those integer DCT coefficients with peak amplitudes in each coefficient, which permits high capacity by coefficient expanding. Yan et al. [24] expanded DCT coefficients

with peak amplitude to achieve imperceptible hiding. Lin and Shiu [25] selected DCT coefficients in high-frequency components, which are supposed to have lower amplitude, to achieve high image quality. Additionally, Chang et al.'s scheme [26] uses the medium-frequency coefficients of DCT-transformed cover images to embed. Hiding data into higher DCT coefficients of images results in lower distortion, while hiding them in lower DCT coefficients promises better robustness.

Even though hiding in DCT coefficients by expansion has been proposed in the image field, only a few works discuss hiding in expanded DCT coefficients of audio data. Since the media data are different, distortion of images depends on visual perception, while audio distortion depends on auditory perception. Therefore, research on hiding with coefficient expansion for audio data may be worthwhile. When embedding information in the low, mid, and high DCT coefficients, the impact on the stego data may differ for image and audio. Therefore, it would be beneficial to investigate which band of DCT coefficients should be used to achieve high sound quality in data hiding for audio data.

The work in [27] embeds data based on a replacement algorithm, which is not reversible but explores good hiding performance with high capacity by hiding data in the modified DCT domain of audio data. The work in [28] is based on traditional MDCT, which has half of the window overlap between adjacent frames and is inappropriate for tampering detection, since the overlapped windows result in false detection but have a good hiding capacity.

2.2. Tamper Detection Using Data Hiding

Motivated by these works, Huang et al. developed a reversible audio data hiding method using intDCT [12]. In their method, the audio data are first segmented into frames with a fixed length to detect and localize tampering in a frame-wise unit. Figure 2 shows the framework of tamper detection using reversible data hiding. On the encoder side, information for verification (hash value) calculated from the audio signal of a frame is embedded into the frame itself reversibly (Figure 2a). When verifying the audio signal, the hash value is extracted from a frame; simultaneously, the original signal is recovered from the stego signal. Then, the hash value is calculated from the recovered signal (Figure 2b). If this hash value coincides with the embedded hash value, the recovered signal is identical to the original one. Using their method, Huang et al. also proposed a method to locate the tampered part of the audio signal [13].

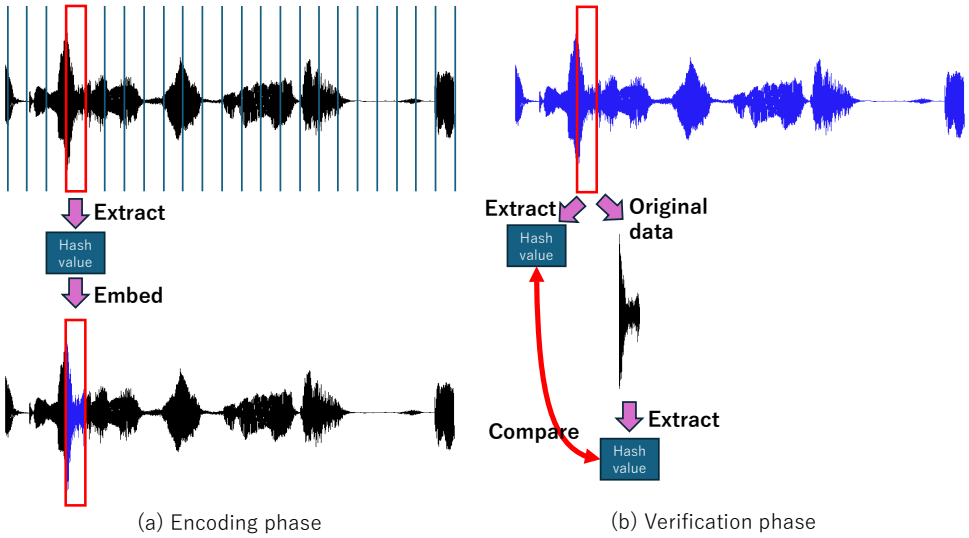

Figure 2. Tamper detection with reversible data hiding.

3. Reversible Watermarking Based on intDCT

In this section, we briefly introduce the information-hiding method proposed by Huang et al. [12], point out the limitation of their work, and propose an improvement of the hiding method.

3.1. Integer Discrete Cosine Transform

The integer discrete cosine transform (intDCT) is a transform similar to the discrete cosine transform (DCT) [29]. The unique feature of intDCT is that both the original and transformed signals are a series of integers. Since most media data, such as image, audio, and video, are encoded as integer values, the intDCT is used for lossless processing of those media data [24,30–32].

Let

$$h = (h(1)\, h(2)\, \ldots\, h(N))^T \tag{1}$$

$$H = (H(1)\, H(2)\, \ldots\, H(N))^T \tag{2}$$

be a time-domain signal at an N-point frame and its DCT coefficients, respectively. In a continuous case, we can obtain DCT coefficients H from a time-domain signal h by DCT-IV matrix as

$$H = C_N^{IV} h \tag{3}$$

where the (i,t)-th $(1 \le i \le N, 1 \le t \le N)$ elements of the DCT matrix C^{IV} are represented as

$$C_N^{IV}(i,t) = \sqrt{\frac{2}{N}} \left[\cos\left(\frac{(t+\frac{1}{2})(i+\frac{1}{2})\pi}{N} \right) \right] \tag{4}$$

Here, an input signal $\mathbf{h}$ is an integer, while C^{IV} is not. Although we can obtain the integers of $\mathbf{H}$ by applying the rounding operation to $C_N^{IV}\mathbf{h}$, it leads to information loss, which means H is irreversible. Here we need the reversible DCT.

In intDCT, the integer signal in the time domain is reversibly transformed into integer DCT coefficients. The reversibility of intDCT is based on the following factorization [33].

$$C_N^{IV} = R_1 R_2 S T_1 T_2 \tag{5}$$

where

$$R_1 = \begin{bmatrix} I_{N/2} & 0 \\ H_1 & I_{N/2} \end{bmatrix}, \tag{6}$$

$$T_1 = \begin{bmatrix} -D_{N/2} & K_2 \\ 0 & I_{N/2} \end{bmatrix}, \tag{7}$$

$$S = \begin{bmatrix} I_{N/2} & 0 \\ H_3 + K_1 & I_{N/2} \end{bmatrix}, \tag{8}$$

$$R_2 = \begin{bmatrix} I_{N/2} & H_2 \\ 0 & I_{N/2} \end{bmatrix}, \tag{9}$$

$$T_2 = \begin{bmatrix} I_{N/2} & 0 \\ K_3 & I_{N/2} \end{bmatrix}. \tag{10}$$

Here, $I_{N/2}$ is the identity matrix of order $N/2$. R_1, R_2, S, T_1, T_2 are block triangular matrices defined in [33]. $K_1, K_2, K_3, H_1, H_2, H_3, D$ are as follows:

$$K_1 = -(C_{N/2}^{IV} D_{N/2} + \sqrt{2} I_{N/2}) C_{N/2}^{IV} \tag{11}$$

$$K_2 = \frac{C_{N/2}^{IV}}{\sqrt{2}} \tag{12}$$

$$K_3 = \sqrt{2} C_{N/2}^{IV} D_{N/2} + I_{N/2} \tag{13}$$

$$H_1 = \begin{bmatrix} 0 & 0 & \cdots & -\tan\frac{(N-1)\pi}{8N} \\ \vdots & & \iddots & \vdots \\ 0 & -\tan\frac{3\pi}{8N} & \cdots & 0 \\ -\tan\frac{\pi}{8N} & 0 & \cdots & 0 \end{bmatrix},$$

$$H_3 = H_1, \tag{14}$$

$$H_2 = \begin{bmatrix} 0 & \cdots & 0 & \sin\frac{\pi}{4N} \\ 0 & \cdots & \sin\frac{3\pi}{4N} & 0 \\ \vdots & \iddots & & \vdots \\ \sin\frac{(N-1)\pi}{4N} & 0 & \cdots & 0 \end{bmatrix}, \tag{15}$$

$$D = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix}. \tag{16}$$

This equation shows that a DCT matrix can be factorized into the product of block triangular matrices with block identity diagonals. Multiplying a triangular matrix followed by a rounding operation can be reversible even if elements of the triangular matrix are not integers. This property also holds in the block matrix case. Therefore, iteratively multiplying triangular matrices in order and applying the rounding operation can be completely reversible [13].

3.2. Information Hiding with DCT Coefficient Expansion [12]

We introduce the hiding principle considering capacity and audio quality. The expansion technique is effective for reserving the hiding capacity [20,22,30,34–36]. Suppose one bit of information $b \in \{0, 1\}$ is embedded into the i-th DCT coefficient $H(i)$ of original data as

$$S(i) = 2H(i) + b, \tag{17}$$

where $S(i)$ is the i-th DCT coefficient of the stego data. Additionally, the original DCT coefficient $H(i)$ and the embedded information b can be extracted from $S(i)$ by

$$H(i) = \lfloor S(i)/2 \rfloor, \tag{18}$$

$$b = S(i) - 2H(i) \tag{19}$$

where $\lfloor \cdot \rfloor$ denotes the floor function.

Generally, the embedding location affects the quality of stego data. In the work by Huang et al. [12], when embedding K bits of information $b_1, \ldots, b_K \in \{0, 1\}$ in the coefficients ($K < N$), the last coefficients corresponding to the highest frequencies are used, i.e.,

$$S(i) = \begin{cases} H(i) & i \leq N - K \\ 2H(i) + b_{K-N+i} & i > N - K \end{cases} \tag{20}$$

As described above, the previous work used the fixed part of the intDCT coefficients. This framework has two problems. The first is that using the highest frequency coefficient might not be optimal. The second is that embedding the information could lead to an

overflow of the restored signal. Since the original signal is 16bit PCM format, a sample value should be between −32,768 and 32,767. However, the embedding process does not ensure that the sample values of the stego signal are within this interval.

3.3. Introduction of Location Map

To solve the above problems, we introduce the location map [22,37]. The location map is a bit array that expresses the positions of coefficients where the information is embedded. The location maps enable specifying for each frame whether the information is embedded in the frame and, if so, in which band. We can also select hiding coefficients in an adaptable way, which is reserved for future work.

The location map is embedded as a portion of the payload. The location map is stored into the coefficients in the highest frequency domain, according to Figure 3. In case of a frame length $N = 2048$, and the number of segments $M = 16$, then the location map is embedded into the [2032-nd, 2048-th] coefficients. For the reverse process, the 16-bit map in the highest frequency domain is referred to in order to extract the embedded data and reconstruct the original data. Thus, a shorter location map is desirable to save capacity. To shorten the location map, we divide a frame with N DCT coefficients into M blocks, where M is a divisor of N. Then, we select the appropriate blocks to expand to reserve hiding capacity. The location map is embedded into DCT coefficients from the highest frequency domain within the range of $N - M + 1 \leq i \leq N$.

Suppose m ($1 \leq m \leq M$) is the index of blocks, where each block includes $B = N/M$ DCT coefficients with i as the index of coefficients. We introduce a location map $\boldsymbol{\varphi} = (\varphi(1), \ldots, \varphi(M))$, where $\varphi(m) = 1$ means that the information is embedded in the m-th block, or $\varphi(m) = 0$. Suppose the number of expanded blocks is M_{emb} and the capacity is M_{emb}/M [bit/sample]. In our work, we use contiguous blocks for embedding. Therefore, $\boldsymbol{\varphi}$ is set as follows:

$$\varphi(m) = \begin{cases} 1 & M_b \leq m < M_b + M_{emb} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Here, $1 \leq M_b \leq M - M_{emb} + 1$ is the position of the first block for embedding.

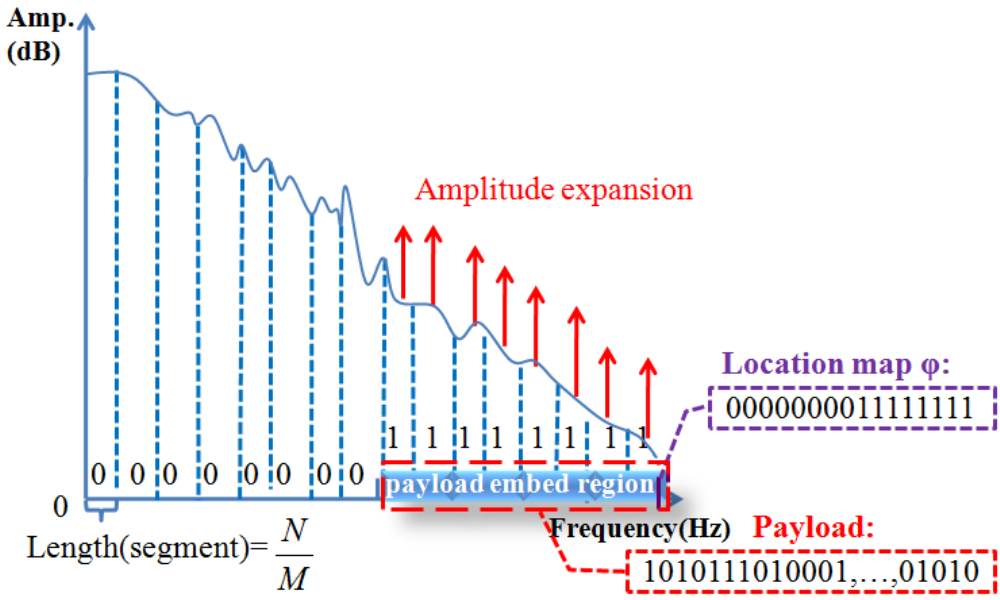

Figure 3. Illustration of expansion and hiding in expanded DCT coefficients with location map φ (length = M bits).

Since the inverse intDCT of the expanded coefficient does not ensure that the generated time-domain data fit within the limit of the 16-bit sample, we need to check whether

embedding data into a block causes overflow. To do this, we embed the data block by block and check if the time-domain data overflow. If the amplitude of time-domain data is estimated to be larger than 32,767 after expansion, the blocks that may result in overflow are not to be selected to embed data, i.e., $\boldsymbol{\varphi} = \mathbf{0}$. Note that even when the overflow occurs, we need to embed the M-bit location map into the DCT coefficients. When $\boldsymbol{\varphi} = \mathbf{0}$, we cannot embed the hash value into that frame, which means we cannot verify the integrity of that frame. However, since one frame is around 32 ms to 128 ms, the absence of a hash value in one frame has almost no effect on tamper detection as long as other frames have hash values. As the point of novelty, a location map is used to control the segments with or without embedding. The proposed method is superior to the previous works [12,20], because these methods do not achieve controllable embedding. The location map specifies 0 or 1 to determine the embedding in particular segments. If overflow can occur, the segment is skipped for embedding in the proposed method. Data loss occurs in the previous works [12,20], and the discarded data are irreversible. By the proposed method, in the case of overflow, the location map indicates the overflow segment with 0 in it. Theoretically, it is possible that a signal may have a location map full of zeros, and no verification is carried out if the data have high amplitude values. We discussed the controllable location map in Section 4.7 to explore how frequently this occurs. We prepare data with overflow and calculate the differences between the stego data and original data with different overflow probabilities.

Furthermore, to avoid overflow, we specify 0 for the blocks without expansion in the location map. Then, the metric of the robustness against tampering with different location maps should be discussed. For the case of location map {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}, the frame is ignored for verification. However, since one frame is around 32 ms to 128 ms, the absence of a hash value in one frame has almost no effect on tamper detection. However, considering the robustness between a location map of {0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1} and that with 0 in the high-frequency domain, such as {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1}, since location map values determine the capacities for hiding information, the location map should at least reserve 128 bits for embedding hash data for verification. Even if only one segment is set to one, 170 bits (in case $N = 2048$, $M = 16$) can be reserved for embedding hash value for verification. If two or more segments are available for embedding, the capacity is available to embed verification data with flexible combinations for verification data. For example, a combination of hash value with frame sequence number can detect the frame-by-frame replacement attack. A combination of various verifications is supposed to enhance the security utility for a stronger shield against tampering.

The embedding steps are summarized in Algorithm 1. First, we segment the audio signal into N-sample frames. Then, we apply the embedding procedure shown in Algorithm 1 frame by frame to obtain the stego signal of the frame. Finally, the stego signals are concatenated to make the final stego signal. The processes are shown in Figure 3.

3.4. Extracting Embedded Information and Reconstructing Original Data

In the extraction process, we extract the embedded data $b_1, \ldots, b_K$ and recover the original signal $\mathbf{h} = (h(1), \ldots, h(N))$ using the procedure shown in Algorithm 2. This extraction process is applied to all frames of the stego signal. Figure 4 plots the flowcharts for embedding, extraction, and verification.

Algorithm 1 Embedding procedure.

Require: A frame of the original signal $\mathbf{h} = (h(1) \dots h(N))$, block number M, secret data (hash value) $b_1, \dots, b_K$ where $K < BM_{emb} - M$

Transform $\mathbf{h}$ into $\mathbf{H} = (H(1), \dots, H(N))$ using intDCT
Determine $\varphi(1), \dots, \varphi(M)$
$k \leftarrow 1$
for $i \leftarrow N$ **to** 1 **do**
  if $i > N - M$ **then**
    $S(i) \leftarrow 2H(i) + \varphi(N - i + 1)$
  else if $\varphi(\lfloor (i-1)/M \rfloor + 1) = 1$ **then**
    if $k \leq K$ **then**
      $S(i) \leftarrow 2H(i) + b_k$
      $k \leftarrow k + 1$
    else
      $S(i) \leftarrow 2H(i)$
    end if
  else
    $S(i) \leftarrow H(i)$
  end if
end for
Transform $\mathbf{S}$ into $\mathbf{s}$ using inverse intDCT
Check if overflow occurs
if overflow occurs **then**
  for $i \leftarrow 1$ **to** N **do**
    if $i \geq N - M$ **then**
      $S(i) \leftarrow 2H(i)$
    else
      $S(i) \leftarrow H(i)$
    end if
  end for
  Transform $\mathbf{S}$ into $\mathbf{s}$ using inverse intDCT
end if

Algorithm 2 Extracting and recovering procedure.

Require: A frame of the stego signal $\mathbf{s} = s(1), \dots, s(N)$, block number M

Transform time-domain stego signal $\mathbf{s}$ into $\mathbf{S}$ using intDCT
for $i = N$ **to** $N - M + 1$ **do**
  $H(i) \leftarrow \lfloor S(i)/2 \rfloor$
  $\varphi(N - i + 1) \leftarrow S(i) - 2H(i)$
end for
$k \leftarrow 1$
for $i = N - M$ **to** 1 **do**
  if $\varphi(\lfloor (i-1)/M \rfloor + 1) = 1$ **then**
    $H(i) \leftarrow \lfloor S(i)/2 \rfloor$
    if $k \leq K$ **then**
      $b_k \leftarrow S(i) - 2H(i)$
      $k \leftarrow k + 1$
    end if
  else
    $H(i) \leftarrow S(i)$
  end if
end for
Transform $\mathbf{H}$ into $\mathbf{h}$ using inverse intDCT

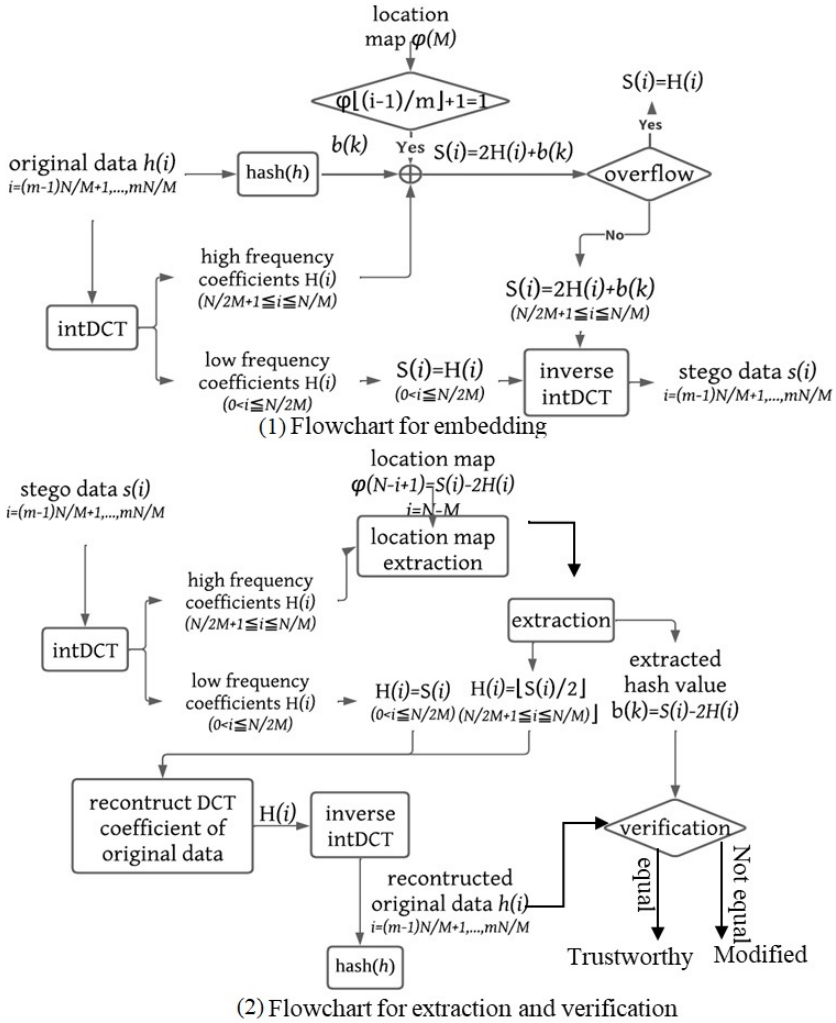

Figure 4. Flowchart for embedding, extraction, and verification.

4. Experimental Evaluation

4.1. Data Used for Experiments

We mainly use speech data for the target application scenario to guarantee data integrity with probative importance. We used the dataset from ITU-T Test Signals for Telecommunication Systems—Test Vectors Associated to Rec. ITU-T P.50 Appendix I [38] for evaluation. This dataset includes 16 kHz sampled and 16-bit quantized waveforms. We used 112 speech signals with 16 speakers in seven languages: American English, Arabic, Mandarin Chinese, Danish, French, German, and Japanese. The average length of each track was approximately 10 s, while we adjusted the data lengths to be an integer multiple of the frame length.

4.2. Analyzing Suitable Coefficients for Expansion

As described in Section 3.3, expanding the DCT coefficients may cause an overflow of the stego signal. Since the magnitude of the DCT coefficients becomes smaller when the frequency increases, the impact of DCT coefficient expansion may differ from band to band. Thus, we analyzed the maximum absolute amplitude of the stego signal generated by expanding different blocks and then embedding the hash data of each frame as the payload to generate the stego data and select the most appropriate coefficient blocks for hiding.

Here, with $M = 16$ and $M_{emb} = 8$, in the case of expanding from the second block ($M_b = 2$), the location map is {0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0}. Figure 5 shows the number of audio clips with maximum amplitude (32,767 or −32,768), suggesting that an overflow occurred to these clips. According to Figure 5, the higher the frequency of the expanded

coefficients located, the smaller the data overflow. No overflow occurred when expanding [9th, 16th] blocks ($M_b = 9$), indicating from 1025th to 2048th coefficients. Thus, selecting DCT coefficients at a higher frequency can avoid overflow.

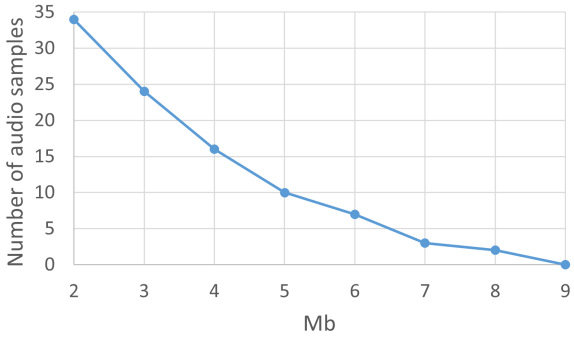

Figure 5. Number of overflowed audio samples with different M_b.

4.3. Evaluation of Audio Quality

4.3.1. Evaluation Criterion

To evaluate the audio quality of the stego data, we mainly use Perceptual Evaluation of Speech Quality (PESQ) and Segmental Signal-to-Noise Ratio (segSNR) in this paper, which have been extensively used to evaluate sound quality objectively [20,22]. PESQ is a standard comprising a test methodology for objective assessment of speech quality.

Mean Opinion Score—Listening Quality Objective (MOS-LQO) scores are used to evaluate the audio quality. SegSNR [dB] is a method of checking the distortion caused by differences in the time domain by comparing original and stego data sample by sample, which is a time-domain-based measurement. The higher the MOS-LQO and segSNR scores are, the better the audio quality is. To evaluate the listening quality of the speech data, we used MOS-LQO, which is an objective technique defined by ITU-T Recommendation P.862.1. MOS-LQO scores are obtained by mapping the distortion to MOS scores in the range from 1.02 (lowest quality) to 4.56 (highest quality). For the objective evaluation using MOS-LQO scores, we used PESQ version 1.2 [39]. We used AFsp package version 9.0 to assess the segSNR, defined as the average SNR value over segments.

4.3.2. Audio Quality Results with Different Hiding Locations

Basically, frame lengths and hiding capacities affect quality. Expanded blocks, which indicate the hiding locations, also affect audio quality. The number of DCT blocks M_{emb} determines the capacity.

The segSNRs between stego data and original data are shown in Figure 6, where the stego data are generated by expanding different blocks. Each frame is divided into M blocks, and the coefficients in the lowest frequency domain are specified as the first block. Different blocks from M_b to $M_b + M_{emb} - 1$ with $M_{emb} = 8$ are selected to explore the most suitable blocks for hiding. Figure 7 plots an example of the difference between stego data and original data in DCT coefficients due to different expansion block positions, including an example with overflow. This clearly illustrates the expanded blocks. According to the segSNRs in Figure 6, audio quality generally improves if higher coefficients are expanded, and [9th, 16th] are the most suitable locations for expansion. Figure 8 plots a graphic about the original sound, as shown in File Ja_m5.wav in Supplementary Materials and stego sound, as shown in File Ja_m5_9-16.wav in Supplementary Materials of some samples to compare how they change. When [9th, 16th] coefficients are expanded for embedding, by comparing the stego data and original data, the boundary for expansion and embedding might be visible.

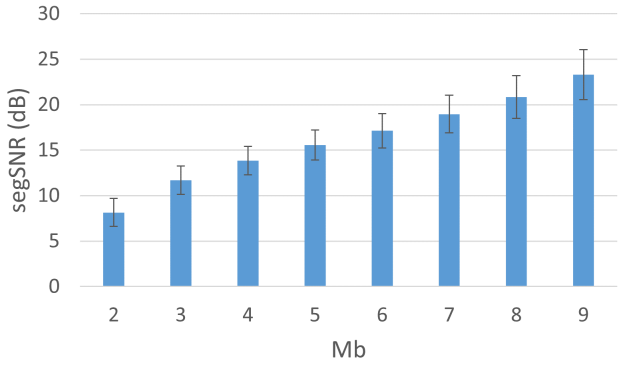

Figure 6. Segmental signal-to-noise (segSNRs, [dB]) ratio between stego data and original data according to different expanding blocks of 112 data.

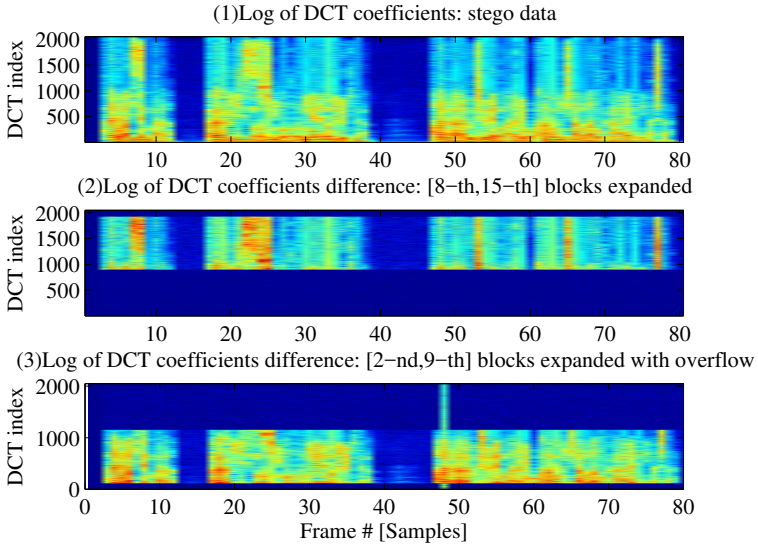

Figure 7. Log of DCT coefficient difference: expanding different blocks: Ja_m5.wav ($N = 2048$).

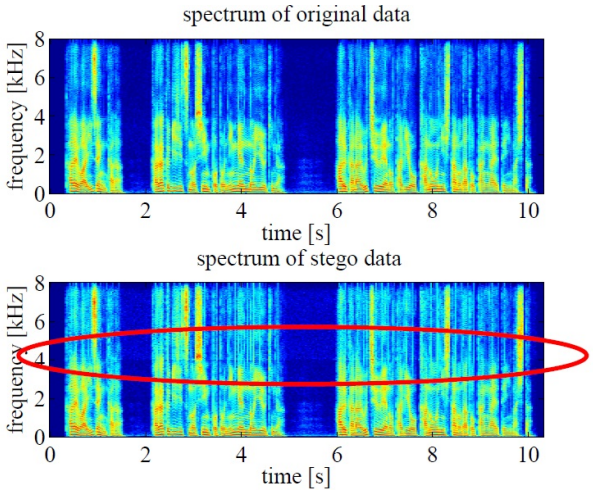

Figure 8. Log of DCT coefficient difference between original data and stego data with a red mark on samples to compare how they change: Ja_m5.wav ($N = 2048$).

4.4. Comparison of Different Frame Lengths and Embedding Capacity

Frame length is supposed to affect both the precision of localizing the tampered positions and audio quality. To determine which frame length is the best, we investigated

the quality of stego signals with different frame lengths, 2048, 1024, and 512. Here, the capacity is 8000 bps ($M = 16$, $M_{emb} = 8$, and $M_b = 9$).

We also compared the audio quality of the information-hiding method based on LPC [20]. Note that since the LPC-based method is not a blind watermark method, the LPC-based method is not an alternative to the proposed method. This experiment compares the absolute quality of the stego signal of different methods.

The comparison results are listed in Table 1. As shown, quality decreases with a shorter frame length. According to the results, the MOS-LQO scores of the proposed method are lower than the LPC-based method; however, the proposed method has better segSNR values. According the frame sizes $N = 512$, $N = 1024$, and $N = 2048$, the average MOS-LQO of the proposed method vs. LPC is 4.27 vs. 4.45, 4.34 vs. 4.48, and 4.41 vs. 4.5; however, according to ITU-T [39], the user satisfaction is at the same level when MOS-LQO is more than 4.3 for "very satisfied", which means that the proposed method has considerable objective listening quality scores compared to the LPC-based method. For segSNR score, the proposed method has better segSNR values in the level around 22 [dB], while the LPC-based method has segSNR values in the level around 16 [dB]. Theoretically, distortion may be caused by discontinuity at the border of two frames. Therefore, this result is consistent with the theoretical explanation because the number of frame borders is small when the frame length is long.

Table 1. Comparison with the conventional method of quality (average MOS-LQO and segSNR) of stego data for different frame lengths using 112 signals (capacity ≈ 8000 bps).

Frame Length	MOS-LQO		segSNR ([dB])	
	LPC [20]	Proposed	LPC [20]	Proposed
512	4.45	4.27	16.04	22.23
1024	4.48	4.34	16.11	22.99
2048	4.50	4.41	16.22	23.31

The stego data generated by the proposed method are obviously superior in terms of segSNRs to the LPC-based method, which means that the difference between the stego data and original data is smaller in the time domain. A possible reason is that the proposed method selects intDCT coefficients with lower amplitude for expansion. Only the modified intDCT-IV used by the proposed method has the feature whereby when frequency increases, the amplitude decreases among the typical seven types of DCT methods. Particularly, in the highest frequency domain, the amplitude became extremely low. Thus, the differences are small after inverse DCT from the frequency domain to the time domain. Meanwhile, the LPC expands the residual value by multiplying them and then adds the payload for embedding directly in the time domain. The residual depends on the adjacent data, and there are seldom residuals with extremely small values, which benefits expansion to achieve small differences in the time domain.

We also performed experiments using different capacities to determine the effect on audio quality. Capacities are set to be from 1000 bps to 7000 bps by adjusting $M_{emb} = 1, \ldots, 7$, hiding from the highest DCT coefficients to the lower one. A comparison of the results is shown in Table 2. As shown, larger capacities result in more distortion in stego data quality. The proposed method has an MOS-LQO of 4.55, the same as the LPC-based method, when capacity ≈ 1000 bps, and comparable average MOS-LQO values when capacity increases. The MOS-LQO scores of the proposed method are lower than those of the LPC-based method; however, the proposed method has better segSNR values. According to the capacity from 1000 bps to 8000 bps, the average MOS-LQO of the proposed method vs. LPC is better than 4.46, and according to ITU-T [39], the user satisfaction is at the same level when MOS-LQO is in a range greater than 4.3 for "very satisfied". All of the MOS-LQO scores are better than 4.46 for both the proposed method and the LPC-based method, which means that the proposed method has considerable objective listening quality scores compared to the LPC-based method. On the other hand, the proposed method has significantly better

segSNR values than that of the LPC-based method at all capacity levels. According to the data in Tables 1 and 2, and Figures 9 and 10, the proposed method has a considerable level of MOS-LQO and better segSNRs than the LPC-based method in general. The reason is supposed to be that the LPC-based method expands multiple data in the time domain, while the proposed method modified the data in the high-frequency domain, where the variation is insensitive for the human auditory system. The difference is difficult to perceive when MOS-LQO is more than 4.3. Furthermore, since the segSNRs score is calculated by the differences between the stego data and the original data, our proposed data selected the coefficients at high frequency for expansion with low amplitude, according to the feather of intDCT type IV. That is the reason for the advantage of segSNRs for the proposed method.

Table 2. Comparison with the conventional method of quality (average MOS-LQO and segSNR) of stego data for different capacities using 112 signals ($N = 2048$).

Capacities (bps)	MOS-LQO		segSNR ([dB])	
	LPC [20]	Proposed	LPC [20]	Proposed
1000	4.55	4.55	32.74	39.74
2000	4.55	4.54	27.03	32.93
3000	4.54	4.53	23.80	30.46
4000	4.54	4.52	21.46	29.00
5000	4.53	4.51	19.75	27.81
6000	4.52	4.49	18.13	26.63
7000	4.51	4.46	17.29	25.16

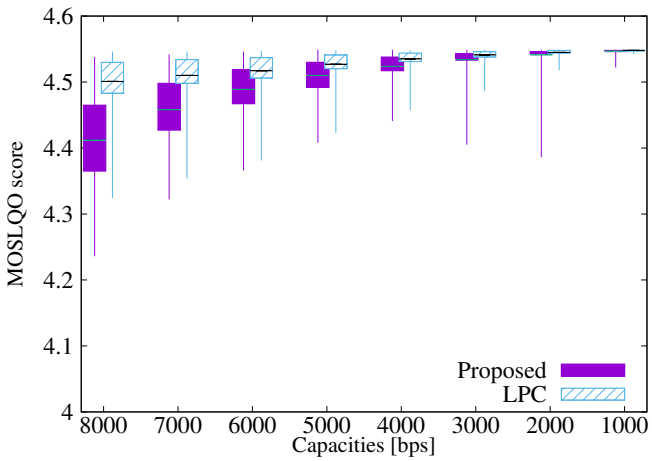

Figure 9. Comparison to LPC-based method [20]: MOS-LQO out of different capacities, $N = 2048$.

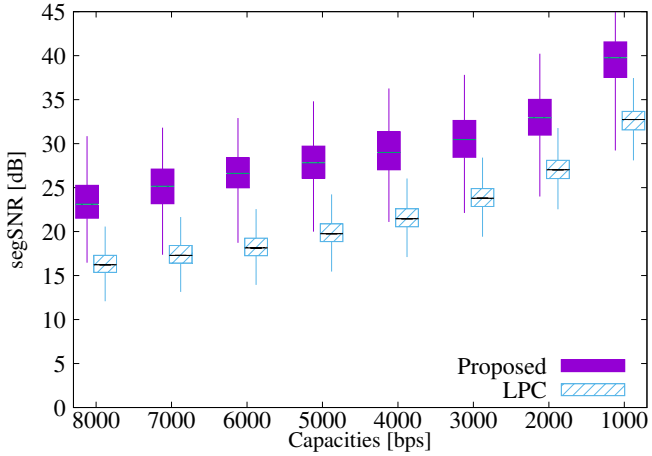

Figure 10. Comparison to LPC-based method [20]: segSNR out of different capacities, $N = 2048$.

We also compared with other similar approaches for watermarking using SNR. The work in [40] proposed a blind audio watermarking algorithm by discrete wavelet transform, which has a capacity of around 100 bps when $SNR = 21$ [dB], and the work in [41] proposed a robust audio watermarking scheme based on fractional Charlier moment transform, achieving SNR = 32 [dB], with capacity around 500 bps. As a popular criterion, MOS-LQO scores are used in [42], which refers to a reversible watermarking method based on variable error expansion of linear prediction applied to G711 μ-law-coded speech signals. The MOS-LQO is 4.13 for a capacity of 711 bps, 3.44 for 1253 bps, and 2.84 for 1995 bps. Additionally, as a reference of evaluation criteria, Unoki [43] et al. and [44–46] used perceptual evaluation of objective difference grades (ODGs), and log spectrum detection (LSD, the smaller, the better) as the criteria to evaluate the objective audio quality. LSD evaluates signals in short-term Fourier transforms of the original and watermarked signals. Typically, the LSD criterion for speech watermarking is less than or equal to 1 [dB]. The work in [46] proposed a watermarking scheme for tampering detection by modifying the line spectral frequencies (LSFs). The results showed scores for LSD $\leqq 1$ [dB] and ODGs $\geqq 3.0$ (slightly annoying). The work in [44] also achieved a similar result to [46]. The work in [45] proposed a blind method with spread spectrum using linear prediction residue, with MOS results located between 3 and 4, and LSD around 1 [dB], with a bit rate of 16 bps. The works in [44,45] aimed for irreversible watermarking.

4.5. Computational Cost of Embedding

Since the matrix calculation exists in intDCT, we calculated the computational cost for embedding. Even though calculation time depends on the computer's performance and the programming language for implementation, as a reference, we calculated the computational cost for embedding 112 data with expanded [9th,16th] blocks and the system configuration listed in Table 3. The average processing time is approximately 4.884 [s] for speech with an average length of 10 [s]. This result shows that real-time embedding is possible.

Table 3. System configuration and computational cost.

Configuration	Value
OS	Windows Vista Business
CPU	Intel Core 2 Quad CPU Q9650 @ 3.00 GHz (©Intel Corporation, Santa Clara, CA, USA)
Memory(RAM)	2 GB
Programming language	Matlab
Average time	4.884 s

4.6. Evaluation of Reversibility

We performed experiments to verify the reversibility of the proposed method by computing the differences between the reconstructed and original data. Data are read in the time domain as a matrix in GNU Octave version 3.4.2, and subtraction is applied. Different from the conventional works with semi-reversible or irreversible algorithms, the algorithm proposed in this work guarantees reversibility by rounding calculation; no data loss occurs by applying the proposed algorithm if there is no overflow. To examine the reversibility, we conducted an experiment to calculate the difference between the reconstructed data and the original data with 112 data. The results indicate that all 112 data resulted in differences of 0, which verified the proposed work's reversibility.

4.7. Overflow Analysis with Controllable Location Map When Embedding Data into Music Signals

As shown in Figure 3, using the highest eight blocks as the payload did not cause overflow in any of the speech materials. Here, the music signal is another target of tampering detection [47–49]. Music signals have more variation than speech signals, which may cause

overflow when all eight blocks are used for the payload [12]. Therefore, we experimented with embedding data into one to eight blocks at the highest frequency bands and calculated the probability of overflow of one frame.

We examined six music clips taken from YouTube, as shown in Table 4. Before the experiment, these music signals were mixed down into a single channel. The frame size of intDCT was $N = 2048$, and the number of blocks was $M = 16$. We changed the number of embedded blocks M_{emb}, and M_b was set as $M_b = M - M_{emb} + 1$.

Table 4. Music clips used in the experiment.

Genre	Music Clip Name	Length (s)
Classic	Bach—Fugue G minor BMV 578	194
Classic	Beethoven—Moonlight Sonata	900
Jazz	Dave Brubeck—Take Five	329
Jazz	John Coltrane—Giant Steps	289
Pops	Wah Wah World	206
Pops	DUNE ft.Miku Hatsune	239

Overflow of the signal depends on the amplitude distribution of the original signal. Thus, we investigated the effect of original amplitude on the overflow. First, we normalized the amplitude of all signals so that either the maximum value was 32,767 or the minimum value was −32,768. After the normalization, we multiplied a coefficient $0.5 \leq \alpha \leq 1$ by all samples to control the signal's amplitude distribution. Finally, we embedded data in one to eight blocks ($1 \leq M_{emb} \leq 8$) and observed the number of frames where overflow occurred. Then we divided the number of overflowed frames by the number of all frames to calculate the probability of overflow of one frame.

Figure 11 shows the experimental result when we changed α and embedded in eight blocks ($M_{emb} = 8$). The X-axis is α, and the Y-axis is the probability that a frame overflows. Note that the Y-axis is a log scale, but the bottom of the axis indicates zero. As shown in the figure, when we normalize the signal at the −3 [dB] level ($\alpha = 0.5$), we observe no overflow at all. When we increase α, the probability increases, and we observe overflowed frames for all signals when $\alpha = 1$. The overflow probability of classic and jazz music clips was lower than that of Pops. This result is caused by not only genre differences but also differences in music production methods. Classic and Jazz music consists of recordings of acoustic instruments, while Pops clips are computer-generated signals. Moreover, the amplitude of the music signals of Pops was almost as large as the maximum throughout of the signal, probably using a dynamic range compressor [50].

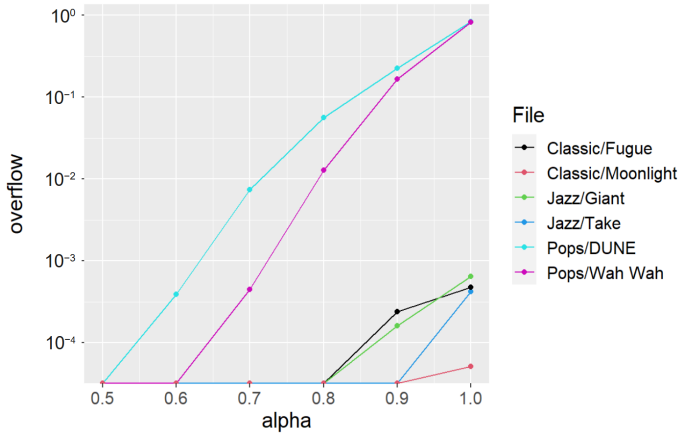

Figure 11. Overflow probability of a frame for music signals with respect to the normalization factor ($M_{emb} = 8$).

Figure 12 shows the overflow probability when we change the number of embedded blocks (M_{emb}) when $\alpha = 0.9$. We can see that no overflow occurs when $M_{emb} = 1$. If we embed all eight blocks as [12], we cannot avoid overflow, which makes the embedded signal not reversible. Because our work uses the location map, we can reduce the amount of embedded blocks when we expect overflow.

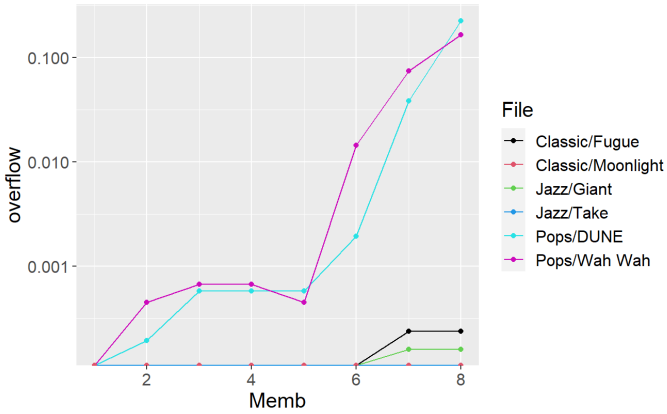

Figure 12. Overflow probability of a frame with respect to M_{emb} ($\alpha = 0.9$).

5. Discussion

Huang et al. [13] explored a target application for tampering detection based on digital watermarking, showing that a 4000 bps capacity is enough for tampering detection. We also reserve more capacity until 8000 bps to consider cases that use more information for verification, such as the combination of hash and sequence numbers. A comparison of the average MOS-LQO and segSNR scores given capacities that range from 1000 bps to 8000 bps ($N = 2048$) based on LPC, and the proposed method is plotted in Figures 9 and 10. According to the results, both methods have MOS-LQO scores better than 4.27, which falls between "imperceptible" and "perceptible but not annoying", where distortion is difficult to distinguish. The proposed method has comparable MOS-LQO values with LPC methods, and it is inferior when capacity increases to 7000 bps. According to the result shown in Figure 10, the proposed method had constantly higher segSNR than that of LPC, with the best average of 39.74 [dB] and the worst average of 22.23 [dB], which promises clear audio quality. There was no overlap between these two methods regarding segSNR, which means that the proposed method achieved a smaller difference between the stego and original data in the time domain.

6. Conclusions

This paper proposed and implemented a reversible watermarking method based on coefficient expansion transformed by modified integer DCT. Suitable coefficients are explored according to the audio feature. We evaluated this scheme by audio quality according to different frame lengths and capacities. We also objectively compared this to the LPC-based method regarding audio quality, possibility of overflow, and computational cost. An average of 4.41 (capacity = 8000 bps, and frame length $N = 2048$) for MOS-LQO is achieved for the proposed method, while MOS-LQO is 4.5 for the LPC method. Experimental results show that the proposed method has MOS-LQO scores comparable to those of the LPC method. An average value of 23.31 [dB] for segSNR out of 112 data is achieved for the proposed method, while segSNR is 16.22 [dB] for the LPC method. Furthermore, segSNR are better when frame length $N = 512$ with 22.23 [dB] (proposed) vs. 16.04 [dB] (LPC), and when frame length $N = 1024$) with 22.99 [dB] (proposed) vs. 16.11 [dB] (LPC). According to the results in Tables 1 and 2, the proposed method has notably better segSNR scores to achieve imperceptibility in the time domain and a lower possibility of overflow.

Due to the concentration of hiding locations in the high-frequency band, there is a risk that the hiding positions may be detected according to the borderline by spectrum analysis. To protect the hiding locations and achieve better audio quality, a more sophisticated algorithm is to be proposed for exploring adaptive hiding locations with distortion estimation.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app14072757/s1, original audio: Ja_m5.wav; stego audio: Ja_m5_9-16.wav.

Author Contributions: Conceptualization, X.H. and A.I.; methodology, X.H.; formal analysis, X.H. and A.I.; investigation, writing—original draft preparation, X.H.; writing—review and editing, A.I.; visualization, X.H. and A.I.; supervision, A.I.; project administration, X.H. and A.I.; funding acquisition, X.H. and A.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JSPS KAKENHI Grant Number JP18K18052.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The speech data used in this paper were ITU-T P.50: Artificial voices, which can be downloaded from https://www.itu.int/net/itu-t/sigdb/genaudio/Pseries.htm, accessed on 5 March 2024. The music data were downloaded from YouTube as follows: Bach—Fugue in G minor BWV 578: https://www.youtube.com/watch?v=PhRa3REdozw, accessed on 5 March 2024; Beethoven—Moonlight Sonata: https://www.youtube.com/watch?v=4Tr0otuiQuU, accessed on 5 March 2024; Dave Brubeck—Take Five: https://www.youtube.com/watch?v=vmDDOFXSgAs, accessed on 5 March 2024; John Coltrane—Giant Steps: https://www.youtube.com/watch?v=h6NCx0wcrC4, accessed on 5 March 2024; Wah Wah World: https://www.youtube.com/watch?v=okJ9Vk6owG8, accessed on 5 March 2024; DUNE ft. Miku Hatsune: https://www.youtube.com/watch?v=AS4q9yaWJkI, accessed on 5 March 2024.

Acknowledgments: We thank Echizen Isao from National Institute of Informatics, Nobutaka Ono from Tokyo Metropolitan University, and Akira Nishimura from Tokyo University of Information Sciences for there advice and support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bourouis, S.; Alroobaea, R.; Alharbi, A.M.; Andejany, M.; Rubaiee, S. Recent advances in digital multimedia tampering detection for forensics analysis. *Symmetry* **2020**, *12*, 1811. https://doi.org/10.3390/sym12111811.
2. Thakur, R.; Rohilla, R. Recent advances in digital image manipulation detection techniques: A brief review. *Forensic Sci. Int.* **2020**, *312*, 110311. https://doi.org/10.1016/j.forsciint.2020.110311.
3. Sitara, K.; Mehtre, B.M. Digital video tampering detection: An overview of passive techniques. *Digit. Investig.* **2016**, *18*, 8–22. https://doi.org/10.1016/j.diin.2016.06.003.
4. Echizen, I.; Yamada, T.; Tezuka, S.; Singh, S.; Yoshiura, H. Improved video verification method using digital watermarking. In Proceedings of the International Conference of Intelligent Information Hiding and Multimedia Signal Processing, Pasadena, CA, USA, 18–20 December 2006; pp. 445–448. https://doi.org/10.1109/IIH-MSP.2006.265037.
5. Ouyang, J.; Huang, J.; Wen, X.; Shao, Z. A semi-fragile watermarking tamper localization method based on QDFT and multi-view fusion. *Multimed. Tools Appl.* **2023**, *82*, 15113–15141. https://doi.org/10.1007/s11042-022-13938-1.
6. Bevinamarad, P.R.; Shirldonkar, M. Audio forgery detection techniques: Present and past review. In Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) , Tirunelveli, India, 15–17 June 2020; pp. 613–618. https://doi.org/10.1109/ICOEI48184.2020.9143014.
7. Agarwal, P.; Prabhakaran, B. Tamper Proofing 3d Motion Data Streams. In *Advances in Multimedia Modeling. MMM 2007*; Springer: Berlin, Heidelberg, 2007; pp. 731–740. https://doi.org/10.1007/978-3-540-69423-6_71.
8. Zakariah, M.; Khan, M.K.; Malik, H. Digital multimedia audio forensics: Past, present and future. *Multimed. Tools Appl.* **2018**, *77*, 1009–1040. https://doi.org/10.1007/s11042-016-4277-2.
9. Wu, C.P.; Kuo, C.C.J. Fragile speech watermarking based on exponential scale quantization for tamper detection. In Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, FL, USA, 13–17 May 2002; Volume 4, pp. 3305–3308. https://doi.org/10.1109/ICASSP.2002.5745360.
10. Petrovic, R. Digital watermarks for audio integrity verification. In Proceedings of the TELSIKS 2005 7th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services, Niš, Serbia, 28–30 September 2005; Volume 1, pp. 215–220. https://doi.org/10.1109/TELSKS.2005.1572095.

11. Nassar, S.S.; Ayad, N.M.; Kelash, H.M.; El-sayed, H.S.; El-Bendary, M.A.M.; Abd El-Samie, F.E.; Faragallah, O.S. Efficient audio integrity verification algorithm using discrete cosine transform. *Int. J. Speech Technol.* **2016**, *19*, 1–8. https://doi.org/10.1007/s10772-015-9312-6.
12. Huang, X.P.; Echizen, I.; Nishimura, A. A Reversible Acoustic Steganography for Integrity Verification. In *Digital Watermarking. IWDW 2010*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6526, pp. 305–316. https://doi.org/10.1007/978-3-642-18405-5_25.
13. Huang, X.P.; Ono, N.; Nishimura, A.; Echizen, I. Reversible Audio Information Hiding for Tampering Detection and Localization Using Sample Scanning Method. *J. Inf. Process.* **2017**, *25*, 469–476. https://doi.org/10.2197/ipsjjip.25.469.
14. Cuccovillo, L.; Mann, S.; Tagliasacchi, M.; Aichroth, P. Audio tampering detection via microphone classification. In Proceedings of the 2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP), Pula, Italy, 30 September–2 October 2013; pp. 177–182. https://doi.org/10.1109/MMSP.2013.6659284.
15. Meng, X.; Li, C.; Tian, L. Detecting audio splicing forgery algorithm based on local noise level estimation. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018; pp. 861–865. https://doi.org/10.1109/ICSAI.2018.8599318.
16. Wang, Z.F.; Wang, J.; Zeng, C.Y.; Min, Q.S.; Tian, Y.; Zuo, M.Z. Digital audio tampering detection based on ENF consistency. In Proceedings of the 2018 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Chengdu, China, 15–18 July 2018; pp. 209–214. https://doi.org/10.1109/ICWAPR.2018.8521378.
17. Zeng, C.; Kong, S.; Wang, Z.; Li, K.; Zhao, Y. Digital Audio Tampering Detection Based on Deep Temporal–Spatial Features of Electrical Network Frequency. *Information* **2023**, *14*, 253. https://doi.org/10.3390/info14050253.
18. Menendez-Ortiz, A.; Feregrino-Uribe, C.; Hasimoto-Beltran, R.; Garcia-Hernandez, J.J. A survey on reversible watermarking for multimedia content: A robustness overview. *IEEE Access* **2019**, *7*, 132662–132681. https://doi.org/10.1109/ACCESS.2019.2940972.
19. Fridrich, J. Image watermarking for tamper detection. In Proceedings of the 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269), Chicago, IL, USA, 7 October 1998; Volume 2, pp. 404–408. https://doi.org/10.1109/ICIP.1998.723401.
20. Nishimura, A. Reversible audio data hiding using linear prediction and Error Expansion. In Proceedings of the 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Dalian, China, 14–16 October 2011; pp. 318–321. https://doi.org/10.1109/IIHMSP.2011.76
21. Aoki, N. A technique of lossless steganography for G.711. *IEICE Trans. Commun.* **2007**, *E90-B*, 3271–3273. https://doi.org/10.1093/ietcom/e90-b.11.3271.
22. Yan, D.; Wang, R. Reversible data hiding for audio based on prediction error expansion. In Proceedings of the International Conference of Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, 15–17 August 2008; pp. 249–252. https://doi.org/10.1109/IIH-MSP.2008.27.
23. Unoki, M.; Miyauchi, R. Reversible watermarking for digital audio based on cochlear delay characteristics. In Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Dalian, China, 14–16 October 2011; pp. 314–317. https://doi.org/10.1109/IIHMSP.2011.99.
24. Yang, B.; Schmucker, M.; Niu, X.; Busch, C.; Sun, S. Reversible image watermarking by histogram modification for Integer DCT coefficients. In Proceedings of the Workshop on Multimedia Signal Processing, Siena, Italy, 29 September–1 October 2004; pp. 143–146. https://doi.org/10.1109/MMSP.2004.1436446.
25. Lin, C.; Shiu, P. High capacity data hiding scheme for DCT-based images. *J. Inf. Hiding Multimed. Signal Process.* **2010**, *1*, 220–240.
26. Chang, C.C.; Chen, T.S.; Chung, L.Z. A Steganographic Method Based upon JPEG and Quantization Table Modification. *Inf. Sci.* **2002**, *141*, 123–138. https://doi.org/10.1016/S0020-0255(01)00194-3.
27. Geiger, R.; Yokotani, Y.; Schuller, G. Audio data hiding with high data rates based on IntMDCT. In Proceedings of the International Conference on Acoustics, Audio, and Signal Processing (ICASSP), Toulouse, France, 14–19 May 2006; pp. 205–208. https://doi.org/10.1109/ICASSP.2006.1661248.
28. Nishimura, A. Reversible and Robust Audio Watermarking Based on Spread Spectrum and Amplitude Expansion. In *Digital-Forensics and Watermarking (IWDW 2014)*; Lecture Notes in Computer Science; Shi, Y.Q.; Kim, H.; Pérez-González, F.; Yang, C.N., Eds.; Springer: Cham, Switzerland, 2015; Volume 9023, pp. 215–229. https://doi.org/10.1007/978-3-319-19321-2_16.
29. Zeng, Y.; Cheng, L.; Bi, G.; Kot, A.C. Integer DCTs and fast algorithms. *IEEE Trans. Signal Process.* **2001**, *49*, 2774–2782. https://doi.org/10.1109/78.960425.
30. Alattar, A.M. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156. https://doi.org/10.1109/TIP.2004.828418.
31. Zhang, J.; Ho, A.T. An efficient digital image-in-image watermarking algorithm using the integer discrete cosine transform (IntDCT). In Proceedings of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia, Proceedings of the 2003 Joint, Singapore, 15–18 December 2003; Volume 2, pp. 1163–1167. https://doi.org/10.1109/ICICS.2003.1292643.
32. Geiger, R.; Sporer, T.; Koller, J.; Brandenburg, K. Audio coding based on integer transforms. In Proceedings of the Audio Engineering Society Convention 111, New York, NY, USA, 30 November–3 December 2001. Audio Engineering Society: New York, NY, USA. Available online: http://www.aes.org/e-lib/browse.cfm?elib=9891 (accessed on 5 March 2024).

33. Haibin, H.; Susanto, R.; Rongshan, Y. A fast algorithm of integer MDCT for lossless audio coding. In Proceedings of the IEEE International Conference on Acoustics, Audio and Signal Processing (ICASSP), Montreal, QC, Canada, 17–21 May 2004; pp. 177–180. https://doi.org/10.1109/ICASSP.2004.1326792.
34. Zhou, H.; Chen, K.; Ma, Z.; Wang, F.; Zhang, W. *Triangle Mesh Watermarking and Steganography*; Springer Nature: Berlin/Heidelberg, Germany, 2023.
35. Zeng, X.; Chen, Z.Y.; Chen, M.; Xiong, Z. Reversible video watermarking using motion estimation and prediction error expansion. *J. Inf. Sci. Eng.* **2011**, *27*, 465–479.
36. Tian, J. Reversible Data Embedding Using a Difference Expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. https://doi.org/10.1109/TCSVT.2003.815962.
37. Thodi, D.M.; Rodríguez, J.J. Expansion embedding techniques for reversible watermarking. *IEEE Trans. Image Process.* **2007**, *16*, 721–730. https://doi.org/10.1109/TIP.2006.891046.
38. ITU-T. Test Signals for Telecommunication Systems–Test Vectors Associated to Rec. ITU-T P.50 Appendix I. 2001. Available online: http://www.itu.int/net/itu-t/sigdb/genaudio/Pseries.htm (accessed on 3 October 2023).
39. ITU-T. Perceptual Evaluation of Audio Quality (PESQ): An Objective Method for End-to-End Audio Quality Assessment of Narrow-Band Telephone Networks and Audio Codecs. ITU-T Recommendation P.862, 2001. Available online: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=5374&lang=en (accessed on 5 March 2024).
40. Wang, X.; Wang, P.; Zhang, P.; Xu, S.; Yang, H. A norm-space, adaptive, and blind audio watermarking algorithm by discrete wavelet transform. *Signal Process.* **2013**, *93*, 913–922. https://doi.org/https://doi.org/10.1016/j.sigpro.2012.11.003.
41. Yamni, M.; Karmouni, H.; Sayyouri, M.; Qjidaa, H. Robust audio watermarking scheme based on fractional Charlier moment transform and dual tree complex wavelet transform. *Expert Syst. Appl.* **2022**, *203*, 117325. https://doi.org/https://doi.org/10.1016/j.eswa.2022.117325.
42. NISHIMURA, A. Reversible Audio Data Hiding Based on Variable Error-Expansion of Linear Prediction for Segmental Audio and G.711 Speech. *IEICE Trans. Inf. Syst.* **2016**, *E99.D*, 83–91. https://doi.org/10.1587/transinf.2015MUP0009.
43. Mawalim, C.O.; Unoki, M. Feasibility of Audio Information Hiding Using Linear Time Variant IIR Filters Based on Cochlear Delay. *J. Signal Process.* **2019**, *23*, 155–158. https://doi.org/10.2299/jsp.23.155.
44. Wang, S.; Yuan, W.; Wang, J.; Unoki, M. Speech Watermarking Based on Source-filter Model of Speech Production. *J. Inf. Hiding Multim. Signal Process.* **2019**, *10*, 517–534.
45. Isoyama, T.; Kidani, S.; Unoki, M. Blind Speech Watermarking Method with Frame Self-Synchronization Based on Spread-Spectrum Using Linear Prediction Residue. *Entropy* **2022**, *24*, 677. https://doi.org/10.3390/e24050677.
46. Chen, X.; Yuan, W.; Wang, S.; Wang, C.; Wang, L. Speech Watermarking for Tampering Detection Based on Modifications to LSFs. *Math. Probl. Eng.* **2019**, *2019*, 7285624. https://doi.org/10.1155/2019/7285624.
47. Gomez, E.; Cano, P.; Gomes, L.; Batlle, E.; Bonnet, M. Mixed watermarking-fingerprinting approach for integrity verification of audio recordings. In Proceedings of the International Telecommunications Symposium, Natal, Brazil, 8–12 September 2002.
48. Li, W.; Zhang, X.; Wang, Z. Music content authentication based on beat segmentation and fuzzy classification. *EURASIP J. Audio, Speech, Music Process.* **2013**, *2013*, 1–13. https://doi.org/10.1186/1687-4722-2013-11.
49. Renza, D.; Ballesteros L., D.M.; Lemus, C. Authenticity verification of audio signals based on fragile watermarking for audio forensics. *Expert Syst. Appl.* **2018**, *91*, 211–222. https://doi.org/10.1016/j.eswa.2017.09.003.
50. Giannoulis, D.; Massberg, M.; Reiss, J.D. Digital dynamic range compressor design—A tutorial and analysis. *J. Audio Eng. Soc.* **2012**, *60*, 399–408. http://www.aes.org/e-lib/browse.cfm?elib=16354.