# Tampering Detection for Speech Signals Using Synchronization Code and LSF-based Watermarks

Shengbei Wang*, Weitao Yuan*, Zhen Zhang*, Jianming Wang*, and Masashi Unoki†
* Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems, Tiangong University, Tianjin, China
E-mail: {wangshengbei, weitaoyuan, zhenzhang, wangjianming}@tiangong.edu.cn
† School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Japan
E-mail: unoki@jaist.ac.jp

***Abstract*—Speech tampering has become a serious threat to speech signals. This paper proposes a method to detect horizontal and vertical speech tampering. In this method, each speech frame is divided into two non-overlapping parts to embed the synchronization code and watermarks, respectively, where the synchronization code is used to identify horizontal tampering and the watermarks are used to detect vertical tampering. To increase the robustness and security of the frame synchronization, the synchronization code (a random binary code) is embedded into the wavelet domain. Efficient line spectral frequency (LSF) information is extracted from the synchronization part and then embedded into the watermarks part using echo-hiding. Tampering detection is achieved by comparing the embedded LSF information with the extracted LSF information. Experimental results demonstrate that the proposed method provides satisfactory inaudibility, robustness, the fragility, as well as high accuracy for tampering detection with a detection precision of 0.1 s.**

***Index Terms*—Speech tampering, speech watermarking, tampering detection, robustness, fragility**

## I. Introduction

Speech signals have become an indispensable information medium. Since speech signals (in digital form) are easy to transmit, edit, and manipulate, it is necessary to detect whether or not they have been tampered with during the transmission. Most of the current tampering detection methods have been developed on the basis of digital watermarking [1] and thus are designed to satisfy several requirements related to watermarking, e.g., inaudibility, blindness, robustness, and security. When used for tampering detection, speech watermarking should satisfy another crucial requirement, namely, fragility, which ensures that the embedded watermarks are destroyed when the speech has been tampered with [2], [3].

In general, speech tampering can be divided into two categories: horizontal tampering and vertical tampering. Horizontal tampering, i.e., desynchronization attacks such as insertion and removal, changes the length of speech. An effective way to detect this kind of tampering is to embed a synchronization code into the speech [4], [5]. Vertical tampering mainly modifies the characteristics of the speech signals. To deal with this kind of tampering, it is necessary to examine whether the characteristics of the speech signals have been changed or not [2], [3], [6].

It is preferred that speech watermarking methods have the ability to deal with both types of tampering [7], [8], [9]. To achieve this, most speech watermarking methods are implemented in a frame-wise fashion, where each speech frame is divided into two parts with one used for embedding the synchronization code and the other for embedding the watermarks for tampering detection [9], [10]. However, when the synchronization code and watermarks are embedded into related speech characteristics, they are likely to be destroyed simultaneously [7], [8]. To tackle this problem, synchronization code and watermark information should be embedded into different locations (or irrelevant features) of the speech signal [9], [10]. In this case, it is also necessary to avoid a redundant embedding to ensure the inaudibility of the watermarked signal. In addition, to increase the security, the synchronization code should not be directly embedded into the time-domain speech signals.

To address the above issues, this paper proposes a more reliable tampering detection method for speech signals. In the proposed method, each speech frame is divided into two non-overlapping parts to embed the synchronization code and watermarks, respectively, where the synchronization code is used to identify horizontal tampering and the watermarks are used to detect vertical tampering. To increase the security, the synchronization code (a random binary code) is embedded into the synchronization part using discrete wavelet transform (DWT) [11], [12] and quantization index modulation (QIM) [13]. Efficient line spectral frequency (LSF) information of the synchronization part is then extracted and embedded into the watermark part by using echo-hiding [14] for tampering detection. Experimental results demonstrate that the combination of synchronization code and LSF-based watermarks enables the proposed method to deliver a satisfactory tampering detection performance.

This work was supported by grants from the National Natural Science Foundation of China (Nos. 61902280 and 61771340), the Natural Science Foundation of Tianjin (Nos. 19JCYBJC15600 and 18JCYBJC15300), the Tianjin Science and Technology Program (Nos. 20YDTPJC00870 and 19PTZWHZ00020), and the Tianjin Major Project of Science and Technology (No. 18ZXJMTG00260). It was also supported by a Grant-in-Aid for Scientific Research (B) (No. 17H01761), the I-O DATA Foundation, and the Fund for the Promotion of Joint International Research (Fostering Joint International Research (B)) (20KK0233).

Corresponding Author: Weitao Yuan, weitaoyuan@tiangong.edu.cn

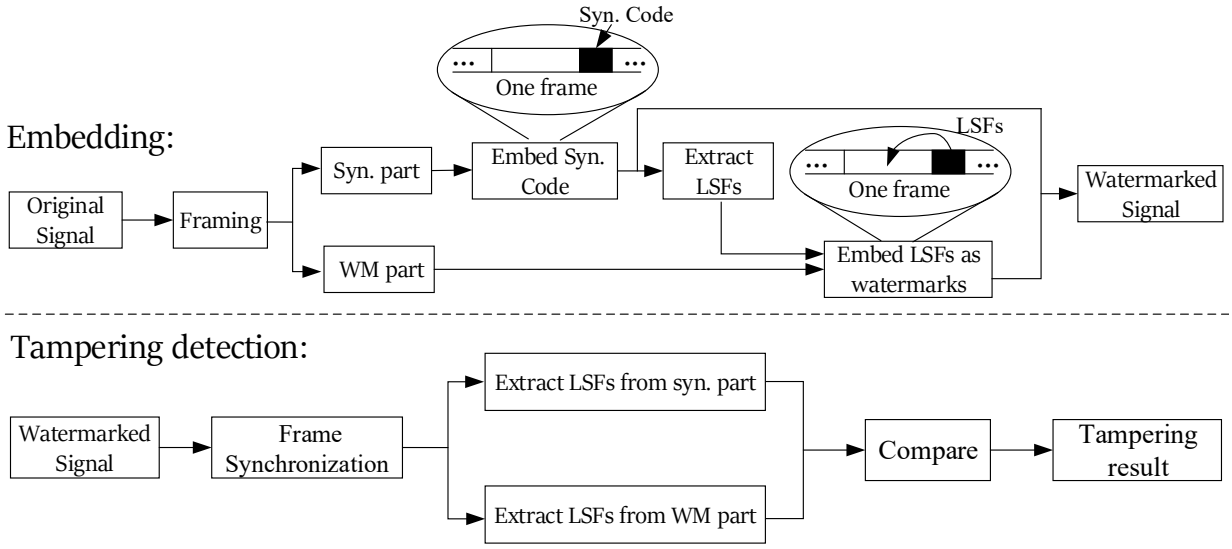


Fig. 1. Block diagram of proposed method. In the embedding process, each frame is divided into a synchronization (Syn.) part and a watermark (WM) part to embed the synchronization code (for identifying horizontal tampering) and LSF-based watermarks (for detecting vertical tampering), respectively. Each frame is synchronized and then the LSF-based watermarks are extracted and compared for detection of tampering.

## II. Proposed method

The block diagram of the proposed method is shown in Fig. 1. We first divide the speech signal into frames of length $L$. Each frame is then divided into two non-overlapping parts of different lengths, with $\lfloor \frac{1}{4}L \rfloor$ duration (synchronization part) used for embedding the synchronization code and $\lfloor \frac{3}{4}L \rfloor$ duration (watermark part) used for embedding the LSF-based watermarks. $\lfloor \cdot \rfloor$ stands for the floor function.

### A. Embedding of synchronization code

The synchronization code we use is a random binary code. Rather than directly embedding the synchronization code into the time-domain speech signal, we embed it into the transformed domain, which increases the security of the proposed method and the robustness of the frame synchronization.

We use DWT to decompose the synchronization part, as DWT has several advantageous properties including perfect reconstruction, multi-resolution capability, and a full consideration of speech characteristics [15]. Three-level DWT approximation coefficients, denoted as $C(n)$, $1 \le n \le N$, are used to carry the synchronization code, where $N$ is the length of the approximation coefficients. We generate a synchronization code $S(n)$ (composed of $-1$, $+1$) of the same length as $C(n)$. The $S(n)$ is embedded into $C(n)$ using QIM.

The basic form of QIM is formulated as

$$Q(C(n)) = \triangle \Big[ \frac{C(n)}{\triangle} \Big], \tag{1}$$

where $\Delta$ is the quantization step and $[\cdot]$ stands for the rounding function. We design the following two quantizers $Q_-$ and $Q_+$ to embed the binary code into $C(n)$, where $Q_-$ and $Q_+$ are used to embed '$-1$' and '$+1$' of $S(n)$, respectively,

$$Q_-(C(n)) = Q(C(n) - b_-) + b_-, \quad \text{if} \quad S(n) = -1, \tag{2}$$
$$Q_+(C(n)) = Q(C(n) - b_+) + b_+, \quad \text{if} \quad S(n) = +1, \tag{3}$$

where $b_- = -\frac{\triangle}{4}$ and $b_+ = \frac{\triangle}{4}$ denote the dither vector corresponding to $Q_-$ and $Q_+$. The quantized approximation coefficients $\hat{C}(n) = Q_+(C(n))$ (or $\hat{C}(n) = Q_-(C(n))$) are then used to construct the synchronization part (with synchronization code embedded) by inverse DWT.

### B. Embedding of LSF-based watermarks

Stable speech features can increase the reliability and accuracy of tampering detection. To effectively detect speech tampering, we use LSF as the tampering indicator. The LSF information is extracted from the synchronization part after embedding the synchronization code and then embedded into the watermark part for tampering detection.

The LSFs are converted from linear prediction (LP) coefficients. They differ in that the LSFs are less sensitive to noise and can provide a more accurate estimate of the formants of the speech signal. To obtain LSFs, we first use LP analysis to calculate the LP coefficients, as

$$\hat{x}(n) = \sum_{p=1}^{P} a_p x(n-p), \tag{4}$$

where $\hat{x}(n)$ is the predicted value of a speech segment $x(n)$ (i.e., the synchronization part), $P$ is the LP order, $a_p$ are the LP coefficients, and $x(n-p)$ is the $p$-th previous value. The LSFs $\varphi_p$ $(1 \le p \le P)$ converted from $a_p$ satisfy the following condition,

$$0 < \varphi_1 < \cdots < \varphi_p < \cdots < \varphi_P < \pi. \tag{5}$$

In general, the first two LSFs contain important phonetic features of the speech. Rather than directly embedding these first two as watermarks, we only embed the difference between the first two LSFs, $D_\varphi$ ($D_\varphi = \varphi_2 - \varphi_1$), for tampering detection, which not only increases the robustness of the detection but also reduces the space required for embedding (leading to less sound distortion).

The $D_\varphi$ is embedded into the watermark part by using a pseudo-noise (PN)-based echo-hiding method. This is a simple watermarking technique but provides better security and inaudibility for the proposed method.

The basic echo-hiding model is defined as

$$y(n) = x(n) \otimes h(n), \tag{6}$$

where $x(n)$ is the speech signal (i.e., the watermark part), $h(n)$ is the echo kernel, $y(n)$ is the watermarked signal, and the operation symbol $\otimes$ stands for convolution. To obtain better robustness, we use the following forward and backward PN kernel for embedding

$$h(n) = \delta(n) + \alpha \sum_{i=0}^{l-1} p(i)\big[\delta(n-d-i) + \delta(n+d+i)\big], \tag{7}$$

where $0 < \alpha < 1$ is the attenuation amplitude of the echo kernel, $p(i)$ ($p(i) \in \{-1, +1\}$) is the PN sequence of length $l = \lfloor \frac{3}{4} L \rfloor - 30$, and $d$ is the delay of the echo.

We embed $D_\varphi$ by setting different delay positions,

$$d = \begin{cases} 2 \times ([10 D_\varphi] + 1), & D_\varphi < \theta \\ 2 \times ([10\theta] + 1), & D_\varphi \geq \theta \end{cases} \tag{8}$$

where $[\cdot]$ is the rounding function.

## III. Tampering detection

The process of tampering detection is shown in the bottom panel of Fig. 1.

### A. Frame synchronization

To synchronize each frame, we first take a unit of length $\lfloor \frac{1}{4} L \rfloor$ from the beginning of the received speech signal. The DWT is performed on this unit to obtain the 3-level approximation coefficients $\tilde{C}(n)$ and then the QIM is performed on $\tilde{C}(n)$ to obtain the embedded synchronization code using

$$s_- = |Q_-(\tilde{C}(n)) - \tilde{C}(n)|, \tag{9}$$

$$s_+ = |Q_+(\tilde{C}(n)) - \tilde{C}(n)|, \tag{10}$$

$$\tilde{S}(n) = \begin{cases} -1, & s_- > s_+ \\ 1, & \text{otherwise} \end{cases} \tag{11}$$

where the operation symbol $|\cdot|$ stands for the absolute value operation. The cross-correlation between the synchronization code $S(n)$ and $\tilde{S}(n)$ is calculated by

$$T = E\big(S(n)\tilde{S}(n)\big) = \sum_{k=-\infty}^{+\infty} S(k)\tilde{S}(n-k), \tag{12}$$

where $E(\cdot)$ calculates the mathematical expectation and $0 \leq T \leq 1$. We set $T > 0.9$, a relatively high value, as the threshold to ensure a reliable synchronization result. We repeat the above procedure until the end of the speech signal to determine the frame structure.

### B. Process of tampering detection

When the frame is synchronized, we apply real cepstrum analysis to the watermark part to calculate the embedded LSF information.

The real cepstrum analysis of Eq. (6) can be written as

$$c_y(n) = c_x(n) + c_h(n), \tag{13}$$

where $c_x(n) = \mathcal{F}^{-1}\{\log|\mathcal{F}(x(n))|\}$ and $c_h(n) = \mathcal{F}^{-1}\{\log|\mathcal{F}(h(n))|\}$, $|\cdot|$ is the absolute value operation, and $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ are the Fourier transform and inverse Fourier transform, respectively. The real cepstrum $c_h(n)$ [16] of Eq. (7) can be expressed as

$$c_h(n) \approx \frac{\alpha}{2}\big(p(-n+d) + p(n-d)\big). \tag{14}$$

The PN sequence $p(i)$ is a necessary condition to obtain the embedded LSF information. The cross-correlation between $c_y(n)$ and the PN sequence $p(i)$ is derived as

$$\begin{aligned} R(\tau) &= E\big(c_y(n)p(n-\tau)\big) \\ &= E\big(c_x(n)p(n-\tau)\big) + E\big(c_h(n)p(n-\tau)\big) \\ &\approx E\big(c_x(n)p(n-\tau)\big) \\ &\quad + \frac{\alpha}{2}E\big(p(-n+d)p(n-\tau)\big) \\ &\quad + \frac{\alpha}{2}E\big(p(n-d)p(n-\tau)\big). \end{aligned} \tag{15}$$

We know from Eq. (15) that $\frac{\alpha}{2}E\big(p(-n+d)p(n-\tau)\big)$ is negligible, and when $\tau$ is equal to $d$, the term of $\frac{\alpha}{2}E\big(p(n-d)p(n-\tau)\big)$ has a maximum value of $\frac{\alpha}{2}$. Hence, we can detect the LSF information by finding the maximum value (peak value).

On the basis of Eq. (8), we calculate $R(\tau)$ at $\tau = 2, 4, \cdots, 2 \times ([10\theta] + 1)$. The $\tau$ that provides the maximum $R(\tau)$ indicates the delay position $\hat{d}$ that we used to embed the LSF information, i.e.,

$$R(\hat{d}) = \max\big(R(2), R(4), \cdots, R(2 \times ([10\theta] + 1))\big). \tag{16}$$

We then calculate the difference between the first two LSFs from the synchronization part and use Eq. (8) to determine the delay position $\bar{d}$. The current frame is judged as tampered when $\hat{d}$ and $\bar{d}$ are different.

## IV. Evaluations

### A. Evaluation conditions

The proposed method was evaluated using the ATR dataset (B set) (Japanese sentences, 8.1 seconds (s), 20 kHz, 16-bit) [17]. We resampled the speech signals from 20 kHz to 16 kHz for our evaluations. The proposed method was evaluated using different frame lengths, $L = \{4000, 2000, 1000, 500, 250\}$. In the experiments, we set the LP order as 6 and $\theta$ as 1.2. Since we use echo-hiding to embed the LSF information, the attenuation amplitude $\alpha$ (see Eq. (7)) is important for the performance of the proposed method: the bigger the $\alpha$, the better the tampering detection performance,

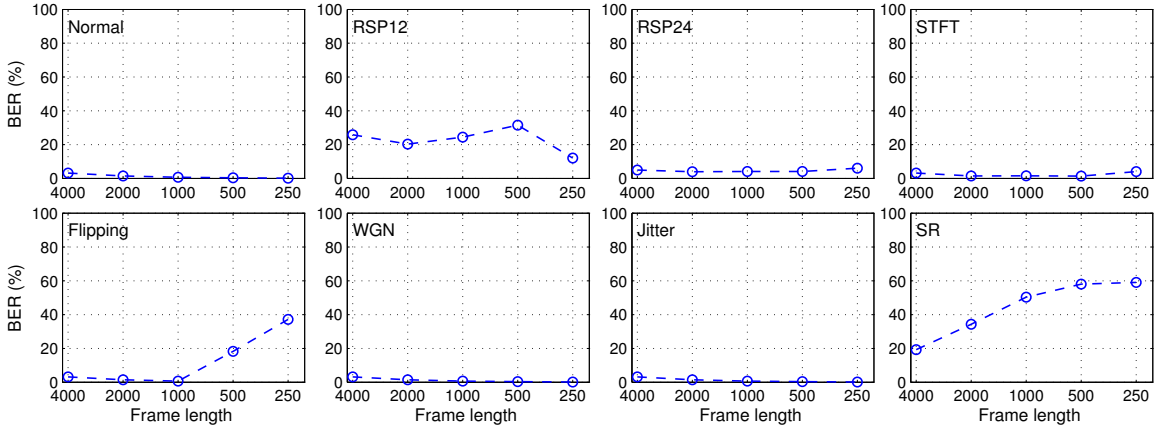


Fig. 2. Robustness of proposed method measured by BER (%).

TABLE I
INAUDIBILITY OF THE PROPOSED METHOD MEASURED BY SNR, LSD, AND PESQ.

| Metrics | Frame length | | | | |
|---|---|---|---|---|---|
| | 4000 | 2000 | 1000 | 500 | 250 |
| SNR | 20.1083 | 16.5697 | 15.1533 | 14.0590 | 16.0291 |
| LSD | 0.5343 | 0.5914 | 0.5942 | 0.6233 | 0.6162 |
| PESQ | 3.1440 | 3.0239 | 3.1786 | 3.1823 | 3.2347 |

TABLE II
TYPES OF TAMPERING. 'NT' (NO TAMPERING) IS USED FOR COMPARISON. 'CT' IS REPRESENTATIVE OF HORIZONTAL TAMPERING AND THE OTHERS OF VERTICAL TAMPERING.

| Abbr. | Tampering | Abbr. | Tampering |
|---|---|---|---|
| NT | No tampering | PSS | Pitch shift -30% |
| CT | Concatenation | HPF | High-pass filtering |
| SU | Speed up +50% | LPF | Low-pass filtering |
| SD | Speed down -50% | RB-1 | Reverberation (1.0 s) |
| PSF | Pitch shift +30% | RB-3 | Reverberation (3.0 s) |

but the speech quality will be degraded. We therefore adjusted $\alpha$ on the basis of frame length $L$, as

$$\alpha = 0.002 + \left(\frac{16000}{4 \times L} - 1\right) \times 0.003. \tag{17}$$

### B. Inaudibility results

We use the signal-to-noise ratio (SNR), the log-spectrum distortion (LSD) [18], and the Perceptual Evaluation of Speech Quality (PESQ) [19] to measure the inaudibility of the proposed method. The threshold values of SNR, LSD, and PESQ are SNR $\geq$ 15 dB, LSD $\leq$ 1.0 dB, and PESQ $\geq$ 3.0 ODG (slightly annoying), respectively.

The inaudibility results under different frame lengths are listed in Table I. We can see that the inaudibility stayed almost stable for different frame lengths and the proposed method obtained a reasonable inaudibility performance for all frame lengths.

### C. Robustness results

The bit error rate (BER (%)) was used to measure the robustness of the proposed method. We performed several processes and attacks on the proposed method to examine its robustness. These included (a) normal extraction (Normal), (b) resampling at 12 kHz (RSP12), (c) resampling at 24 kHz (RSP24), (d) speech analysis/synthesis by short-time Fourier transform (STFT), (e) signal flipping (Flipping), (f) white Gaussian noise addition (WGN), (g) signal jitter (Jitter), and (h) sample repetition (SR). The robustness results measured by BER are plotted in Fig. 2.

We can see that the proposed method exhibited good robustness against most processes and attacks. However, since resampling at 12 kHz and sample repetition led to obvious information loss for the speech signals, robustness was degraded in these cases.

### D. Fragility results

We adopted temporal tampering (horizontal tampering) and acoustic feature-based tampering (vertical tampering) to evaluate the fragility of the proposed method. The types of tampering are listed in Table II, where No tampering (NT) is included for comparison. Concatenation (CT), a typical horizontal tampering, can widely cover deletion, insertion, replacement, etc. Speed up/down (SU and SD) modifies the tempo of speech, which can be used to tamper the emotions of the speaker. Pitch shift (PSF and PSS) proportionally shifts the frequency components while preserving the duration of speech. High-pass filtering (HPF) and low-pass filtering (LPF) could remove specific frequencies from the speech. Reverberation (RB-1 and RB-3) mimics the channel distortion and can be considered as disturbing the speech.

The fragility results of the proposed method are reported in Table III. Note that we only calculated whether the embedded LSF information of the watermark part could be extracted or

TABLE III
FRAGILITY OF THE PROPOSED METHOD (BER (%)). 'NT' (NO TAMPERING) IS USED FOR COMPARISON.

| Type | Frame length | | | | |
|---|---|---|---|---|---|
| | 4000 | 2000 | 1000 | 500 | 250 |
| NT | 3.13 | 1.43 | 0.65 | 0.32 | 0.13 |
| CT | 65.12 | 54.37 | 57.28 | 60.24 | 55.22 |
| SU | 56.89 | 67.27 | 64.28 | 58.29 | 57.73 |
| SD | 57.89 | 70.35 | 68.98 | 66.35 | 71.22 |
| PSF | 45.24 | 47.28 | 40.38 | 51.39 | 50.18 |
| PSS | 40.38 | 45.48 | 43.19 | 47.23 | 52.87 |
| HPF | 49.23 | 56.31 | 58.24 | 62.34 | 65.47 |
| LPF | 38.89 | 40.24 | 39.12 | 37.57 | 42.34 |
| RB-1 | 20.43 | 35.34 | 38.42 | 43.53 | 47.85 |
| RB-3 | 40.22 | 45.28 | 46.24 | 50.49 | 56.68 |

not, i.e., the results were calculated without frame synchronization. As shown, the proposed method was fragile against all tampering attempts.

### *E. Tampering detection results*

We measured the tampering detection performance of the proposed method by False Positive Rate $P_{FP}$ (%) and False Negative Rate $P_{FN}$ (%), where $P_{FP}$ is the rate of non-tampered frames judged as tampered and $P_{FN}$ is the rate of tampered frames judged as non-tampered. This experiment differed slightly from the fragility evaluations in that each speech frame was first synchronized and then the LSF information was extracted to check whether the current frame had been tampered with or not.

Table IV lists the tampering detection results. We can see that the proposed method had a satisfactory tampering detection performance for long speech frames. When the frame length decreased, both $P_{FP}$ and $P_{FN}$ increased. Setting the frame lengths to 4000, 2000, 1000, 500, and 250 corresponded to embedding capacities of 4 bps, 8 bps, 16 bps, 32 bps, and 64 bps, which enabled the detection precision of 0.25 s, 0.125 s, 0.1 s, 0.05 s, and 0.025 s, respectively. In general, the detection precision of 0.1 s is adequate for real applications, as a shorter duration can hardly cause meaningful tampering. Therefore, these results demonstrate that the proposed method can provide a satisfactory tampering detection performance.

TABLE IV
ACCURACY OF TAMPERING DETECTION MEASURED BY $P_{FP}$ AND $P_{FN}$, WHERE FRAME LENGTHS OF 4000, 2000, 1000, 500, AND 250 CORRESPOND TO DETECTION PRECISION OF 0.25 S, 0.125 S, 0.1 S, 0.05 S, AND 0.025 S, RESPECTIVELY.

| Metrics | Method | Frame length | | | | |
|---|---|---|---|---|---|---|
| | | 4000 | 2000 | 1000 | 500 | 250 |
| $P_{FN}(\%)$ | CT | 0.00 | 0.00 | 0.00 | 0.25 | 19.17 |
| | SU | 0.00 | 0.00 | 0.00 | 0.43 | 20.09 |
| | SD | 0.00 | 0.00 | 0.00 | 0.53 | 20.15 |
| | PSF | 0.00 | 0.00 | 0.00 | 0.37 | 18.49 |
| | PSS | 0.00 | 0.00 | 0.00 | 0.40 | 19.15 |
| | HPF | 0.00 | 0.00 | 0.00 | 0.65 | 9.37 |
| | LPF | 0.00 | 0.00 | 0.00 | 0.40 | 16.51 |
| | RB-1 | 0.00 | 0.00 | 0.00 | 0.35 | 20.14 |
| | RB-3 | 0.00 | 0.00 | 0.00 | 0.33 | 19.72 |
| $P_{FP}(\%)$ | NT | 4.89 | 2.31 | 3.02 | 9.23 | 15.00 |

## V. CONCLUSIONS

In this paper, we proposed a method to detect horizontal tampering and vertical tampering for speech signals. Each speech frame is divided into two parts: a synchronization part and a watermark part. A random code is embedded into the synchronization part in the DWT domain based on QIM to improve the robustness of synchronization. To improve the embedding efficiency, we embed the difference between LSFs into the speech itself for tampering detection, which ensures the inaudibility of the proposed method. Experimental results showed that the proposed method could not only satisfy inaudibility but also provided good robustness. It was also fragile against tampering and capable of detecting tampering with adequate precision. In future work, we will further evaluate the proposed method by examining addition types of tampering. In addition, we plan to investigate the synchronization problem when the synchronization part is destroyed.

## REFERENCES


[1] Guang Hua, Jiwu Huang, Yun Q. Shi, Jonathan Goh, and Vrizlynn L. L. Thing, "Twenty years of digital audio watermarking - a comprehensive review," *Signal Processing*, vol. 128, pp. 222–242, 2016.

[2] Shengbei Wang, Weitao Yuan, Jianming Wang, and Masashi Unoki, "Detection of speech tampering using sparse representations and spectral manipulations based information hiding," *Speech Commun.*, vol. 112, pp. 1–14, 2019.

[3] Xuping Huang, "Mechanism and implementation of watermarked sample scanning method for speech data tampering detection," in *the 2nd International Workshop on Multimedia Privacy and Security, Canada*, 2018, pp. 54–60.

[4] Xiangyang Wang, Tianxiao Ma, and Pan-Pan Niu, "A pseudo-zernike moment based audio watermarking scheme robust against desynchronization attacks," *Comput. Electr. Eng.*, vol. 37, no. 4, pp. 425–443, 2011.

[5] Wenhuan Lu, Ling Li, Yuqing He, Jianguo Wei, and Neal N. Xiong, "RFPS: A robust feature points detection of audio watermarking for against desynchronization attacks in cyber security," *IEEE Access*, vol. 8, pp. 63643–63653, 2020.

[6] Jessada Karnjana, Kasorn Galajit, Pakinee Aimmanee, Chai Wutiwiwatchai, and Masashi Unoki, "Speech watermarking scheme based on singular-spectrum analysis for tampering detection and identification," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2017, pp. 193–202.

[7] Hwai-Tsu Hu, Shiow-Jyu Lin, and Ling-Yuan Hsu, "Effective blind speech watermarking via adaptive mean modulation and package synchronization in DWT domain," *EURASIP J. Audio, Speech and Music Processing*, vol. 2017, pp. 10, 2017.

[8] Weizhen Jiang, Xionghua Huang, and Yujuan Quan, "Audio watermarking algorithm against synchronization attacks using global characteristics and adaptive frame division," *Signal Process.*, vol. 162, pp. 153–160, 2019.

[9] Kanhe Aniruddha and Gnanasekaran Aghila, "A QIM-based energy modulation scheme for audio watermarking robust to synchronization attack," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3415–3423, 2019.

[10] B. M. Mushgil, W. A. W. Adnan, A. R. Al-Hadad, and S. M. S. Ahmad, "An efficient selective method for audio watermarking against de-synchronization attacks," *Journal of Electrical Engineering & Technology*, vol. 13, no. 1, pp. 476–484, 2018.

[11] Hwai-Tsu Hu, Ling-Yuan Hsu, and Hsien-Hsin Chou, "Variable-dimensional vector modulation for perceptual-based DWT blind audio watermarking with adjustable payload capacity," *Digit. Signal Process.*, vol. 31, pp. 115–123, 2014.

[12] Huda Karajeh, Tahani Khatib, Lama Rajab, and Mahmoud Maqableh, "A robust digital audio watermarking scheme based on DWT and schur decomposition," *Multim. Tools Appl.*, vol. 78, no. 13, pp. 18395–18418, 2019.

[13] Oktay Altun, Gaurav Sharma, and Mark Bocko, "Set theoretic quantization index modulation watermarking," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2006, pp. 229–232.

[14] Daniel Gruhl, Anthony Lu, and Walter Bender, "Echo hiding," in *Information Hiding, First International Workshop*, 1996, pp. 293–315.

[15] Mahmoud A. Osman and Nasser H. Ali, "Audio watermarking based on wavelet transform," *Mechanical and Electrical Technology IV*, vol. 229, pp. 2784–2788, 2012.

[16] Byeong-Seob Ko, Ryouichi Nishimura, and Yôiti Suzuki, "Time-spread echo method for digital audio watermarking," *IEEE Trans. Multim.*, vol. 7, no. 2, pp. 212–221, 2005.

[17] K. Takeda, "Speech database user's manual," *ATR Tech. Rep. TR-I-0028*, 1988.

[18] Eugen Hoffmann, Dorothea Kolossa, Bert-Uwe Köhler, and Reinhold Orglmeister, "Using information theoretic distance measures for solving the permutation problem of blind source separation of speech signals," *EURASIP J. Audio, Speech and Music Processing*, vol. 2012, pp. 14, 2012.

[19] Yi Hu and Philipos C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 16, no. 1, pp. 229–238, 2008.