IEEE INFOCOM 2023 - IEEE Conference on Computer Communications | 979-8-3503-3414-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOM53939.2023.10228943

I Can Hear You Without a Microphone: Live Speech Eavesdropping From Earphone Motion Sensors

Yetong Cao* Fan Li* Huijie Chen† Xiaochen Liu* Chunhui Duan* Yu Wang‡
* School of Computer Science and Technology, Beijing Institute of Technology, China
† Department of Computer Science, Beijing University of Technology, China
‡ Department of Computer and Information Sciences, Temple University, USA
Email: {yetongcao, fli}@bit.edu.cn, chenhuijie@bjut.edu.cn,{xiaochenliu, duanch}@bit.edu.cn, wangyu@temple.edu

***Abstract*—Recent literature advances motion sensors mounted on smartphones and AR/VR headsets to speech eavesdropping due to their sensitivity to subtle vibrations. The popularity of motion sensors in earphones has fueled a rise in their sampling rate, which enables various enhanced features. This paper investigates a new threat of eavesdropping via motion sensors of earphones by developing *EarSpy*, which builds on our observation that the earphone's accelerometer can capture bone conduction vibrations (BCVs) and ear canal dynamic motions (ECDMs) associated with speaking; they enable *EarSpy* to derive unique information about the wearer's speech. Leveraging a study on the motion sensor measurements captured from earphones, *EarSpy* gains abilities to disentangle the wearer's live speech from interference caused by body motions and vibrations generated when the earphone's speaker plays audio. To enable user-independent attacks, *EarSpy* involves novel efforts, including a trajectory instability reduction method to calibrate the waveform of ECDMs and a data augmentation method to enrich the diversity of BCVs. Moreover, *EarSpy* explores effective representations from BCVs and ECDMs, and develops a convolutional neural model with Connectionist Temporal Classification (CTC) to realize accurate speech recognition. Extensive experiments involving 14 participants demonstrate that *EarSpy* reaches a promising recognition for the wearer's speech.**

I. Introduction

Ear-wear devices are becoming more pervasive and common in our daily life, which provides unprecedented possibilities for improving our lifestyles. However, according to a report from [1], 50.2% of participants have experienced conversation surveillance on their smart devices, and more are concerned about losing conversation information on their personal devices. In view of the potential risks, microphones are typically protected by permission mechanisms in operating systems [2], [3]. As motion sensors become increasingly critical in earphones to deliver enhanced functions such as in-ear detection, touch control, and virtual assistant activating, their sampling rates have sharply increased to about 500-2,000 Hz [4], [5]. The powerful sensing capability and unrestricted access have made the earphone motion sensors attractive targets for attackers. In this paper, we raise a challenging question: *can the motion sensors embedded in earphones pick up the wearer's speech during everyday life?*

Fan Li is the corresponding author. The work of Fan Li is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No.62072040. The work of Huijie Chen is partially supported by NSFC under Grant No.62202019, and China Postdoctoral Science Foundation under Grant No.2021M700302.

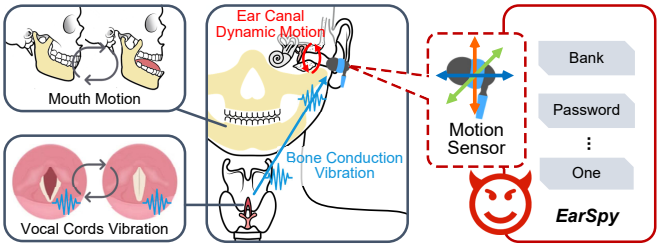


Fig. 1. An illustration of *EarSpy* scheme.

Prior studies [6]–[9] have shown that smartphone motion sensors (e.g., accelerometers and gyroscopes) are sensitive enough to measure sound vibrations replayed by an external loudspeaker placed on the same surface or the built-in loudspeakers of the same smartphone. Besides, Vibphone [10] shows the success of tracking speech-induced vibrations from the cheek using the smartphone's motion sensor during phone conversations. The recent attempt, Face-Mic [11], identifies a new security vulnerability of inferring speech based on face dynamics using the built-in motion sensor of AR/VR headsets.

While these prior works demonstrate the feasibility of using motion sensors mounted on various devices to infer speech, we find that they can not be applied to speech eavesdropping via earphones for the following reasons: (i) Different from the loudspeaker's vibrations captured by smartphone motions sensor, the wearer's speech vibrations captured by earphone motion sensors only contain a very low baseband of the voice due to the significant decay effect as sounds propagate through bone and tissue. Moreover, speech travels from the throat through face to the ear with complex rendering along multiple paths. Thereby signals captured from the ear differ greatly from those captured from the face. (ii) Existing approaches only handle scenarios that involve a single sound source (either replayed by the loudspeaker or generated by a live speaker). However, the earphone eavesdropping scenarios usually involve the wearers' speech and audio played by the earphone's speaker, leaving analysis of the wearer's speech very difficult. (iii) Besides analyzing the motion sensor data, existing works often require other data, such as labeled victim's audio, to achieve the desired accuracy. However, such information is

not available on current earphones. An effective and practical solution for eavesdropping is still needed via motion sensors on earphones.

Fig. 1 shows the scheme of *EarSpy*, which is based on our observations that (i) during speech production, vibrations from the vocal cords propagate through the mandible and tissue to the ear, affecting the earphones' motion sensors, known as Bone Conduction Vibrations (BCVs); (ii) the movements of mouth cause dynamic motions of the ear canal, thereby causing the earphones' motion sensor response, known as Ear Canal Dynamic Motions (ECDMs). The two types of ear dynamics are highly related to the speech content and can be decoded to infer the wearer's speech.

EarSpy continuously records motion sensor data at a high sampling rate to capture ear dynamics caused by speech. One may be concerned about *would continuous monitoring of the earphone's motion sensor at a high sampling rate quickly drain the battery and alert the victim?* We find that motion sensors play an important role in today's earphones and have already been commonly monitored continuously at a high sampling rate. For example, Apple describes in their patent [5] that the accelerometers in their headsets work at a sampling rate of 2,000-6,000 Hz to detect a user's voice activity. This has driven manufacturers to equip earphones with ultra-low power chips that support hours of earphone use on a single charge [12]. Thus, we believe *EarSpy* is not easily noticeable.

The novel yet plausible idea faces three major challenges: (i) Besides the widely recognized interference of body movement, motion sensors of earphones are particularly susceptible to vibrations arising from earphone's speaker playing audio [13], which brings errors in speech recognition. To address this, we design a pitch-tracking-based earphone vibration interference elimination method to disentangle the wearer's speech information. (ii) Speech-induced ear dynamics are sensitive to individual differences, which hinders the system's performance. To overcome this, we study the characteristics of BCVs and ECDMs, and make novel efforts including a data augmentation method to generate BCV features with sufficient individual variations and a Procrustes transformation-based method to reduce instability for ECDM features. (iii) It is also very challenging to devise an accurate model to infer speech based on ear dynamics since their relationship with speech content is unclear. *EarSpy* calls for a careful design to capture the nuances between different speech contents and realize accurate speech recognition. We design a deep learning model based on Convolutional Neural Networks (CNN) and Connectionist Temporal Classification (CTC), which enable accurate and user-independent word-level speech recognition. Overall, the contributions of this paper are summarized as follows:

1) We propose and implement *EarSpy*, which is the first to reveal the security vulnerability of eavesdropping wearer's live speech using the zero-permission motion sensors mounted on the earphones.
2) We propose several novel techniques to separate reliable ear dynamics from the complicated interference and incorporates an instability reduction method and a data augmentation method together to achieve user-independence. In addition, we design a CNN-CTC-based deep learning architecture that effectively fuses the two types of ear dynamics for accurate speech recognition.
3) We conduct extensive evaluations with 14 participants under various situations, which demonstrate the effectiveness of our proposed attack.

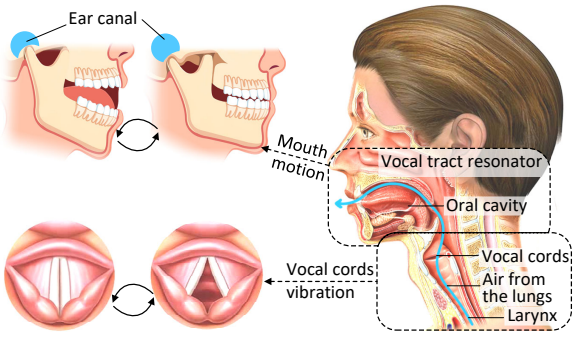


Fig. 2. Speech production process.

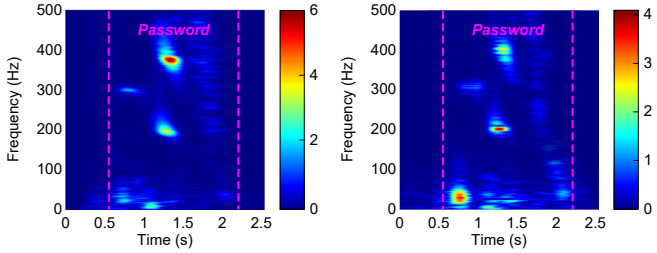
II. Primaries and Observations

A. Ear Dynamics Production

Fig. 2 shows the production of voice and the associated two types of ear dynamics. The lungs push air through the vocal cords, causing vocal cords to vibrate and produce a buzzing sound. Meanwhile, the vocal tract resonator (i.e., mouth cavity, tongue, nose, and lips) modify the buzzing sound and produce the voice. The vocal cord vibrations are highly related to the speech content and can propagate to the ears through the mandible and tissue, enabling the earphone's motion sensor to capture BCVs. Besides, ECDMs are determined by highly speech-dependent vocal tract resonator motions, further adding speech information to the earphone's motion sensor measurements. Therefore, the ear dynamics of BCVs and ECDMs present rich speech information, motivating us to exploit them to infer the earphone wearer's speech.

B. Capturing Ear Dynamics Using Motion Sensors

1) BCVs: Herein we explore the opportunity of capturing speech-related ear dynamics using earphone motion sensors. We focus on using the accelerometer since it captures both vibrations and the motion change rate; it has been identified to be more sensitive to speech-induced vibrations than the gyroscope [11]. We record acceleration from the earphones while a volunteer speaks *password*. Meanwhile, we attach an accelerometer to the volunteer's throat to obtain the reference

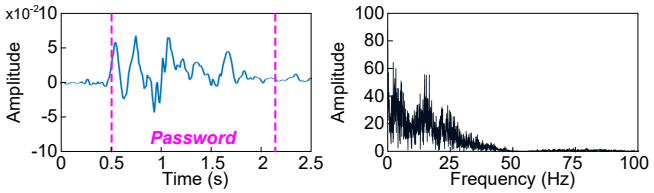


(a) Throat (b) Ear

Fig. 3. Spectrogram of vertical acceleration captured from throat and ear; note the scale difference between (a) and (b).

signal of vocal cords vibrations. Particularly, to handle the acceleration difference caused by coordinate system variations, we derive the vertical acceleration in the longitudinal axis of the body using the quaternion-based method [14]. Fig. 3 shows the spectrogram of both cases, respectively. We can observe that the earphone's accelerometer successfully captures the vocal cord vibrations (i.e., BCVs), although the bones and muscles filter out part of the high-frequency components.

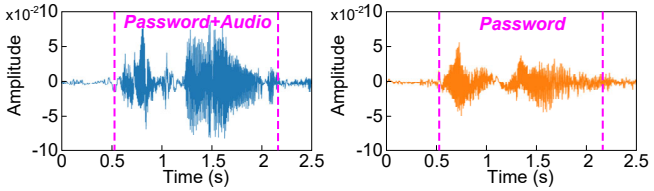
2) ECDMs: Moreover, the spectrogram shows an interesting response in low frequencies. To further study the low-frequency response, we ask the volunteer to again perform the mouth movement of speaking *password*, but without pronouncing. Fig. 4 shows the vertical acceleration captured from the ear and the corresponding frequency response. We can observe components below 50 Hz still remains. Therefore, we attribute such a low-frequency component to the motions of the ear canal wall caused by mouth movement (i.e., ECDMs).



(a) Vertical acceleration response. (b) Power spectral density.

Fig. 4. The vertical acceleration profile of mouth movements.

3) Interference: The accelerometers of earphones have been criticized for being vulnerable to body motions [15]. We, following common sense, identify body motion as one of the causes of interference. In addition, we investigate another potential interference: since the speaker and the accelerometer are in physical contact with the same board, vibration arising from the earphone's speaker playing audio inevitably affects the acceleration measurements [8]. Specifically, we record vertical accelerations using a pair of earphones when the volunteer speaks *password*, while one earphone plays audio (shown in Fig. 5(a)) and the other does not (shown in Fig. 5(b)). We can observe that earphone playing audio indeed introduces interference to the accelerometer measurements.



(a) With music playing. (b) Without music playing.

Fig. 5. Examples of vertical acceleration with and without noise arising from earphone speaker playing audio.

4) Observations: The study confirms that earphone's motion sensor sure captures BCVs (influence frequencies above 50 Hz) and ECDMs (influence frequencies below 50 Hz). Besides, we identify the interference in the accelerometer measurements, including body motions and earphone speaker vibrations. According to previous work [15], [16], earphone vibrations usually have frequencies above 50 Hz, which affects BCVs; while body motions usually have frequencies below 10 Hz, which distort ECDMs. Thus, we disentangle BCVs and ECDMs respectively to infer the wearer's speech.

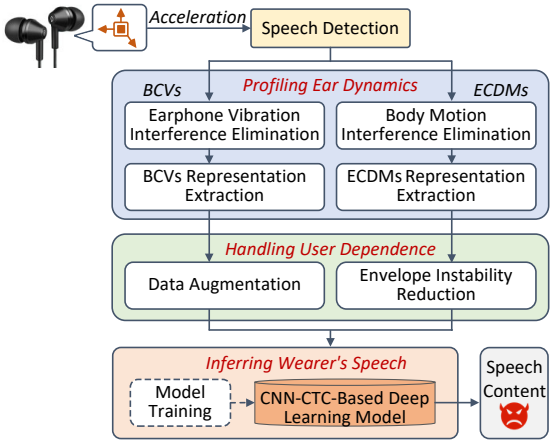


Fig. 6. The architecture of *EarSpy*.

C. Attack Model

Since access to the earphone's motion sensors requires zero permission, the malicious application can disguise itself as any application and trick the victim into installing it on their smartphones, utilizing which, the attacker gathers the accelerometer measurements of earphones. We assume that the malicious application has no access to any other information (e.g., the victim's earphone model and voice data). An attack can be launched to separate the victim's speech from the noisy sensory readings and infer the speech content. The attackers may steal important information, including 1) Private information, such as passwords to a bank account, social security number, postcode, and address. Leakage of such vital information can put the victim's security and privacy at high risk; 2) Personal preference. For example, some products frequently mentioned by the victim may be what he/she wants to buy, and advertisers can target display ads accordingly.

III. System Design

A. System Overview

We present the design of *EarSpy*, which utilizes the motion sensors mounted on earphones to infer the wearer's speech. Fig. 6 shows the architecture of *EarSpy*. The malicious application collects the accelerometer measurements of earphones and checks for the wearer's speech. During speech periods, the accelerometer measurements are separated into BCVs and ECDMs based on the frequency distribution. In *Profiling Ear Dynamics*, *EarSpy* eliminates the interference of earphone vibrations when playing audio and body motions for BCVs and ECDMs, respectively. Then, *EarSpy* extracts effective representations from BCVs and ECDMs, which capture unique speech information. After that, in *Handling User Dependence*, *EarSpy* develops two novel algorithms to inhibit representations' sensitivity to individual differences, including a trajectory difference reduction technique to calibrate the ECDMs, and a novel data augmentation method to generate a large amount of BCVs that include sufficient individual variations. Finally, in *Inferring Wearer's Speech*, *EarSpy* designs a CNN-CTC-based deep learning framework to parse BCVs and ECDMs. With special construction and training, the deep

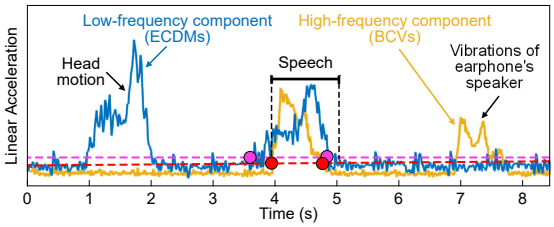


Fig. 7. An example of speech detection.

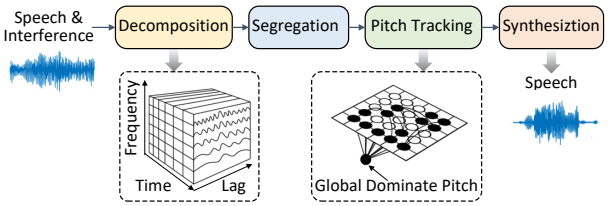


Fig. 8. Earphone vibration interference elimination pipeline.

learning model can infer the wearer's speech without needing to pre-acquiring the wearer's labeled voice data.

B. Speech Detection

We propose a lightweight method to check the presence of speech based on the fact that accelerometer measurements of speech periods have high intensity, whereas non-speech periods usually have low intensity. For speech segmentation, the double-threshold scheme is widely applied [17], [18], which respectively applies thresholds to short-time energy and zero-crossing rate to detect the starting and ending points of speech. Inspired by it, we use 50 Hz as the cut-off frequency to separate the accelerometer measurements into a high-frequency component and a low-frequency component, each corresponding to the BCVs and the ECDMs. Then we calculate the linear acceleration (LA) [19] of the two components and determine the point at which both two LAs exceed their threshold (empirically set to be 0.2 times the maximum value of LA calculated from the prior 100 ms period) as the starting point of speech. Besides, the point that LA amplitude of both components is reduced below their thresholds and the recorded speech exceeds 50 ms [20] is identified as the ending point. As shown in Fig. 7, the proposed method is effective for speech detection and robust to those body movements and earphone speaker vibrations that do not overlap with speech, as they usually affect only one of the two components without allowing both components to exceed their thresholds.

C. Profiling Ear Dynamics

Our goal is to first extract ear dynamics from interference caused by body motions and earphone vibrations when playing audio, then parse the ear dynamics to infer the wearer's speech. The past few years have seen the success of effective body motion noise elimination for ear-wear motion sensor measurements, such as deep regression [11] and sensor fusion [21]. We reduce the impact of body motion in ECDMs as suggested by [11]. However, the vibrations generated when the earphone's speaker plays audio remains a new and less studied area, which is identified as one of our major challenges. Therefore, we design a novel scheme to eliminate the impact of earphone vibrations when playing audio.

1) Earphone Vibration Interference Elimination: The pitch continuity of speech has been witnessed for decades [22], suggesting that a subject's speech has a continuous pitch trajectory over a short period. We are inspired to track the wearer's speech pitch trajectory thus separating the wearer's speech from interference caused by vibrations of earphone's speaker. The basic idea is to segment the contaminated signals into time-frequency representations, group them by pitch tracking, and recover the wearer's speech, as shown in Fig. 8.

Decomposition: We first use a bank of gammatone filters [23] with overlapping passbands to extract sound information similar to what a human would perceive. Specifically, we use 64 filters, and their center frequencies are equally distributed on the modified equivalent rectangular bandwidth (ERB) scale [24] above 50 Hz. The modified ERB scale is defined as $ERB(f) = 24.7(4.37 \times f + 1)$, where f labels the frequency. Then, we apply a 20 ms sliding window with 10 ms overlap to analyze the signal. At window w of filter channel c, the obtained signal is considered as a time-frequency element, which is denoted as $e_{c,w}$.

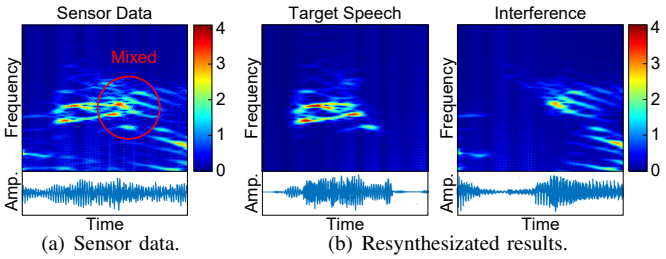
Segregation: Then we quickly merge the elements into segments of a single source based on temporal continuity and cross-channel similarity. Specifically, we calculate temporal continuity by autocorrelation [25], $A_{c,w}$, at zero lag. Elements with $A_{c,w} \geqslant \theta_S^2$ will be segregated from those with $A_{c,w} < \theta_S^2$, where $\theta_S = 50$ is approximate to the spontaneous firing rate of the auditory nerve [26]. Besides, there is strong evidence that the filter channels corresponding to the same sound component exhibit high cross-channel correlation [27]. Therefore, we calculate the cross-channel correlation between adjacent filter channels to measure their similarity as follows:

$$S_{c,w} = \frac{\sum_\tau [A_{c,w}(\tau) - \overline{A}_{c,w}][A_{c+1,w}(\tau) - \overline{A}_{c+1,w}]}{\sqrt{\sum_\tau [A_{c,w}(\tau) - \overline{A}_{c,w}]^2 [A_{c+1,w}(\tau) - \overline{A}_{c+1,w}]^2}}, \quad (1)$$

where $\overline{A}_{c,w}$ represents the averaged autocorrelation. If the neighboring elements $e_{c,w}$ and $e_{c+1,w}$ have a cross-channel correlation over 0.985 (chosen based on [20]), they are credible to belong to the same sound source, therefore we merge them into one segment. By iteratively merging the time-frequency elements, we obtain several segments.

Pitch Tracking: The basic idea is to estimate the dominant pitch of each segment and label it as wearer's speech or interference. Specifically, we determine the dominant pitch by locating the maximum peak in the range of 80-250 Hz from the summary autocorrelation $A_w(\tau) = \sum_c A_{c,w}(\tau)$. Afterward, we calculate a global dominant pitch of the captured speech. If more than half of the elements of a segment at a certain frame match the dominant pitch, the segment is considered to belong to the same source thereby labeled as a group. Otherwise, it is labeled as another group. To avoid incorrect grouping and over grouping, segments shorter than 50 ms are removed. Hence, the segments of different frames are clustered into two groups.

Synthesization: Finally, the two groups of signals are reconstructed by applying binary weights to each frame of the



(a) Sensor data. (b) Resynthesizated results.

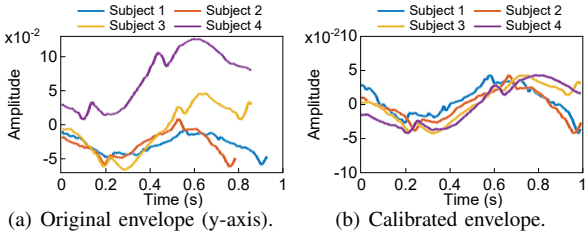
Fig. 9. An example of earphone vibration interference elimination (left ear) for BCVs. The top panel shows the spectrogram, and the bottom panel shows the acceleration response.

gammatone filterbank, then sum across all channels [28] to generate two resynthesized signals, corresponding to wearer's BCVs and interference of speaker vibrations, respectively. To distinguish between BCVs and interference, we process the acceleration measurements from the left and right ears and obtain four resynthesized signals (2 ears $\times$ 2 groups) in total. Depending on the played audio, interference in the left and right ears' data will be approximately the same (e.g., mono audio) or very different (e.g., stereo audio). Besides, BCVs in the left and right ears' data will hold slight differences due to asymmetry of the human body [29]. Therefore, we leverage this contrast to label the resynthesized signals as wearer's speech or interference by comparing cosine similarity between resynthesized signals of the two ears. Fig. 9(a) shows an example of accelerometer measurements involving the wearer's speech and audio interference, and Fig. 9(b) shows the resynthesized speech and interference. Moreover, the evaluation presented in Section IV-F1 validates the effectiveness of the proposed method.

2) BCV Representation Extraction: High-pass filter used to obtain BCVs renders the time-domain features not stable [11], thereby we extract the spectrogram as the representation of BCVs, which capture the fine-grained time-frequency difference between speech content. The BCVs representation is then processed by a data augmentation-based deep learning framework to enable user-independent speech eavesdropping.

3) Body Motion Interference Elimination: We build a deep regression model to eliminate the impact of body motion in ECDMs as suggested by [11]. Specifically, we record accelerometer measurements when performing ECDMs and representative body motions (e.g., walking), respectively. Then we mix them to generate contaminated ECDMs as training data. By chaining two fully-connected layers and a regression layer, we can train a deep regression model to separate clean ECDMs from body motion interference.

4) ECDM Representation Extraction: Although previous works [11] show the success of profiling the ear-wear accelerometer measurements using displacement trajectory, it does not work in our case because analyzing a single point's position is insufficient to obtain a comprehensive understanding of the mouth motion. Inspired by [30] that captures ear canal wall motions using envelope of acoustic signals measured from the ear, we extract the envelope of a total of six acceleration axes collected from the two ears as the representation of ECDMs.



(a) Original envelope (y-axis). (b) Calibrated envelope.

Fig. 10. An example of envelope instability reduction for ECDMs.

D. Handling User Dependence

We find through repeated experiments that ECDM representations of different subjects exhibit differences in curve shape and signal amplitude. Therefore, we incorporate a simple yet effective scheme to calibrate the ECDM representations to have similar curve shapes. Besides, BCV representations of different subjects show obvious distinctions in both time and frequency domains. We seek a data augmentation method and a deep learning framework (introduced in Section III-E) to achieve user-independent.

1) ECDMs Instability Reduction: We propose to reduce the ECDMs envelope instability based on Procrustes transformation, which changes the size and position of the signal trajectory but maintains the geometric characteristics. Procrustes transformation has proven useful in figure alignment [31], but to our knowledge has never been used in our scope.

Specifically, we first record the ECDMs from multiple subjects and obtain the data envelope $D_{\text{avg}}(t)$ over multiple subjects as the baseline. Given the currently obtained envelope $D(t)$, the Procrustes transformation process involves rotation, scaling, and translation, which can be expressed as:

$$\widetilde{D}(t) = H \cdot D(t)\alpha + \beta, \tag{2}$$

where H is the rotation matrix, α is the scaling coefficient, and β is the translation coefficient. The rotation matrix H is solved by singular value decomposition for $D(t)^\top D_{\text{avg}}(t) = U\Sigma V^\top$, where U and V are orthogonal, and Σ is diagonal. Then, α and β can be solved by Minimal Mean Square Error (MMSE) estimation [32]. Fig. 10 illustrates envelopes of original ECDM envelopes and the calibrated results of four subjects, which demonstrate the effectiveness of our proposed method.

2) BCVs Augmentation: It is very hard and entails significant time/effort to collect a large amount of BCVs with sufficient individual variations. With many open-source acoustic speech datasets but very limited BCVs, we innovatively propose a data augmentation method that decomposes the air-conduction speech audio $A(t)$ of speaker A into sub-phoneme units and matches that of BCV (denoted as $V(t)$) of speaker B then generate a new BCV signal that maintains the voice characteristics of speaker A. Our strategy is shown in Fig. 11, which consists of three steps:

Building Text-To-Vibration (TTV) and Text-To-Speech (TTS) Models: We first build a TTV model for speaker B and train a TTS model for speaker A. By doing this, we can decompose the two speakers' data into small units, and the TTS units of speaker A can then be mapped to the closest

unit in TTV of speaker B to generate "BCVs" of speaker A. Specifically, we apply a low-pass filter with 400 Hz cut-off frequency to acoustic speech of speaker A since BCVs barely have high-frequency components. Then, we extract features including Mel frequency cepstral coefficients (MFCC) and pitch as representations for each phoneme (unit). We use Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) to contextually relevant phoneme combinations, map the corresponding acoustic features, and build TTV and TTS.

Units Equalizing: To estimate the similarity between units of TTS and those of TTV, we still need to minimize the differences between $A(t)$ and $V(t)$ by bilinear spectral space warping. The bilinear function yields new units based on units of TTS and TTV, meanwhile preserving speaker A's voice characteristics and avoiding errors in automatic format extraction [33]. We define the following, which can provide better analytical property and only requires one parameter ε:

$$\phi_\varepsilon(t) = \frac{V(t)^{-1} - \varepsilon}{1 - \varepsilon V(t)^{-1}}, \ |\varepsilon| < 1. \tag{3}$$

We design a searching-based strategy to estimate ε. By aligning the power spectrogram of $A(t)$ (denoted as $P_A(f)$) and $V(t)$ (denoted as $P_V(f)$), we can derive the Log Spectral Distance (LSD) defined as follows:

$$LSD = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left[10lg\frac{P_A(f)}{P_V(f)}\right]dt}. \tag{4}$$

We first determine ε_0, the initial value of ε, by minimizing the overall mean LSD between $P_A(f)$ and $P_V(f)$. Then we alternately wrap $P_V(f)$ to be $\hat{P}_V(f)$ and update $P_A(f)$ to be $\hat{P}_A(f)$. Meanwhile, we search forward and backward around ε_0 and finally determine ε by minimizing the LSD between $\hat{P}_A(f)$ and $\hat{P}_V(f)$.

Units Mapping: We construct a matrix to describe the similarity between equalized units of TTS and TTV. Specifically, we measure the similarity between speech units from three aspects: the pitch distance d_{p}, gain distance d_{g}, and Line Spectral Pairs (LSP) distance d_{LSP}, expressed as

$$\begin{aligned}
\psi(u_{\mathrm{w}}, u_{\mathrm{m}}) &= \mathbb{M}(d_{\mathrm{p}}) + \mathbb{M}(d_{\mathrm{g}}) + \mathbb{M}(d_{\mathrm{LSP}}),\\
d_{\mathrm{p}} &= |lg(p_{\mathrm{w}}) - lg(p_{\mathrm{m}})|,\\
d_{\mathrm{g}} &= |lg(g_{\mathrm{w}}) - lg(g_{\mathrm{m}})|,\\
d_{\mathrm{LSP}} &= \sqrt{\tfrac{1}{L}\textstyle\sum_{i=1}^{L}\Theta_i(\theta_{\mathrm{w},i} - \theta_{\mathrm{m},i})^2},\\
\Theta_i &= \tfrac{1}{\theta_{\mathrm{w},i} - \theta_{\mathrm{w},i-1}} + \tfrac{1}{\theta_{\mathrm{w},i+1} - \theta_{\mathrm{w},i}},
\end{aligned} \tag{5}$$

where u_{w} represents the wrapped speech unit of speaker B and u_{m} represents the mapping candidate speech unit of speaker A, $\mathbb{M}$ is the zero-mean-unit-variance operator, which can save the trouble of adjusting the weights between the

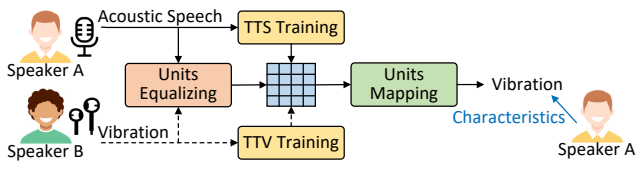


Fig. 11. Pipeline of the proposed BCVs augmentation method.

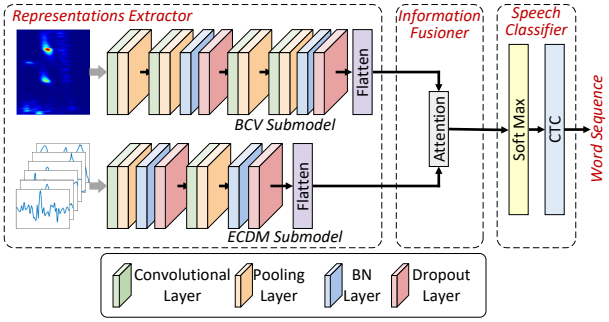


Fig. 12. The architecture of the proposed CNN-CTC model.

three components. The pitch and gain of the wrapped unit are denoted as p_{w} and g_{w}, and those of the mapped candidate unit are denoted as p_{m} and g_{m}. θ_{w}=$[\theta_{\mathrm{w},1}, ..., \theta_{\mathrm{w},L}]$ is the wrapped vector applied L_{th} order LSP. Similarly, θ_{m}=$[\theta_{\mathrm{m},1}, ..., \theta_{\mathrm{m},L}]$ is mapped candidate vector applied L_{th} order LSP. With Eq. (5), we select 5 candidate units associated with minimum $\psi(u_{\mathrm{w}}, u_{\mathrm{m}})$ to construct the similarity matrix. Then, we search for the optimal path in the constructed matrix to get the match between the units of TTS and TTV. By replacing the TTS units of speaker A with the TTV units for speaker B, we finally obtain a new TTV model for speaker A, which generates BCVs with voice characteristics of speaker A.

Using the proposed method, we can generate a large amount of "BCV" signals that contain sufficient individual differences. Specifically, we obtain air-conduction speech audio from the online dataset LibriTTS [34], which consists of 585 hours of speech records from 2,456 speakers and the corresponding texts. Besides, we collect BCVs from 14 volunteers, and manually construct the phoneme level corpora. Overall, the data augmentation method is effective, which is demonstrated through experiments in Section IV-F2.

E. Speech Recognition

Given the extracted representations of the two types of ear dynamics, we design a deep learning architecture to infer the speech in a user-independent manner.

1) CNN-CTC Model: The word-level speech recognition model is shown in Fig. 12, which is based on CNNs. Among many advanced techniques, we choose CNN since they reduce the spectral variation and model spectral correlation [35]. Besides, the recognized speech sequence usually contains multiple repetitions and blanks, resulting in poor readability. We use the CTC technique to merge and align the recognized label to realize accurate speech recognition.

Feature Extractor: BCVs and ECDMs have different physical meanings/data dimensions and convey different information. We design two submodels for them to extract features for word-level speech recognition. The two submodels consist of several cascaded convolutional blocks. Specifically, we build convolutional layers using Rectified Linear Units (ReLU) to enhance the extracted results. Besides, we use the max-pooling layer to perform subsampling, which is a nonlinear compression. The BN layer [36] further improves the model performance and stability. Due to a large number of model

(a) Prototype 1.(b) Prototype 2.(c) Prototype 3.

Fig. 13. The involved prototypes.

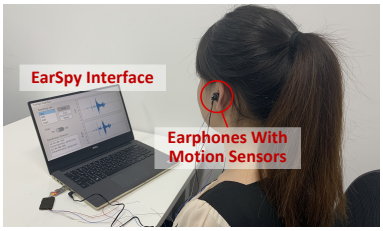


Fig. 14. Experiment setup.

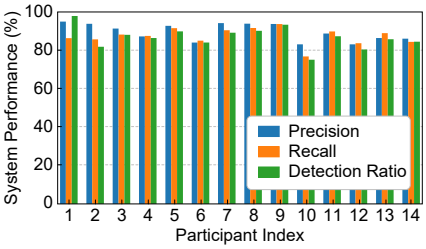


Fig. 15. System performance for each participant.

parameters, we add extra dropout layers to prevent overfitting. Finally, the flatten layer combines the learned information and generates the 1-dimensional latent feature.

The input representations of ear dynamics are derived from a 2 s sliding window with 50% overlap. The BCV representations are then resized to $128 \times 128 \times 3$ to be fed into the BCV submodel, where the chained convolutional layers use 32, 64, 128, and 128 kernels, respectively, and all kernels are 3×3. Besides, the six-dimensional ECDM representations are concatenated into one matrix and input to the ECDM submodel, which uses kernels of 1×9 and 1×3, respectively.

Feature Fusioner: Then, we propose a bilinear pooling scheme to fuse the features of the two types of ear dynamics. The bilinear pooling scheme can reduce the feature dimensions, which provides a joint representation space of the modalities with large differences and captures small differences between different speeches. Specifically, we calculate the outer product of latent representations of the ECDMs (Ψ_{ECDM}) and BCVs (Ψ_{BCV}) as $\Psi = \mathbb{VEC}(\Psi_{ECDM} \otimes \Psi_{BCV})$, where $\otimes$ is the Kronecker product operator. $\mathbb{VEC}$ represents a matrix vectorization operator that first converts a matrix into a column vector via linear transformation, then calculates the element-wise signed square-root $\Psi \leftarrow sign(\Psi)\sqrt{|\Psi|}$, and finally applies L_2 norm on Ψ.

Speech Classifier: To realize speech recognition, we first input the fused features Ψ to the *softmax* layer to obtain the probability of classifying the current time sequence as a specific word $y_t^{p_t}$. Then, we consider a CTC path ($p = p_1, p_2, ...p_T$) that allows blank (non-word) and word labels to appear repeatedly to represent the final recognized speech. The CTC loss is defined as follows:

$$Loss_{\text{CTC}} = -ln(P(Y|X)) = -\sum_{P \in \phi(Y)} \prod_{t=1}^{T} y_t^{p_t}, \qquad (6)$$

where Y is true speech content, ϕ represents the mapping function from p to Y, which can be achieved by greedy search. Overall, the parameters of the deep learning model are updated iteratively until the loss (i.e., Eq. (6)) converges. Experiments in Section IV-C validate that our proposed model is effective.

IV. Evaluation

A. Experiment Setup

1) Sensing Prototypes: To include hardware diversity, three pairs of earphones equipped with motion sensors are used to implement *EarSpy*, as shown in Fig. 13. They have different structures, diverse wearing styles, and integrate different motion sensor chips in different locations.

2) Data Collection: We recruit 14 subjects (7 males, 7 females, ages 20-53) from colleagues and friends. Among them, 5 males and 4 females are native English speakers, and the others are fluent English speakers. Each participant is asked to wear the earphone prototype and read sentences selected from a subset of the LibriTTS corpus [34], which includes 43 short English conversations consisting of 120 words. Fig. 14 illustrates a participant wearing the device to collect data. Meanwhile, we use a microphone placed 30 cm away from the participants to record their speech audio, which is used to analyze the speech decibel level and speech content (automatically extracted using the YouTube *auto-sync* engine [37]). To verify *EarSpy*'s effectiveness, we conduct leave-participant-out validation and a case study on eavesdropping passwords from phone conversations. Besides, we conduct experiments with different sampling rates, speech loudness, and hardware to evaluate *EarSpy*'s robustness against various impact factors. Furthermore, we conduct experiments to verify the effectiveness of key algorithms. Overall, we collect 2,080 sentences for evaluation.

B. Metrics

We evaluate the performance of *EarSpy* for recognizing isolated words using the **precision** (the ratio of the words correctly predicted as label *A* to all words predicted as label *A*) and **recall** (the ratio of the words correctly predicted as label *A* to all words belonging to label *A*). Moreover, as words have different frequencies in sentences, we use **detection ratio** (the ratio of the number of correctly recognized words to the total number of predicted words in a sequence) to provide word distribution-sensitive evaluation.

C. Performance of User-Independent Speech Recognition

We study the most general attack, where we can not access the victim's other information except for the motion sensor data, i.e., there are no data with speech text labels in advance to train the model. We evaluate the user independence of our system by conducting leave-one-participant-out validation, where we use data from one participant for testing and data from the remaining participants for training. Fig. 15 shows the average precision, recall, and detection ratio across all words for each participant. We can observe that 9 of the 14 participants receive precision, recall, and detection ratio above 85%. The rest of the participants have slightly worse performance, but precision,

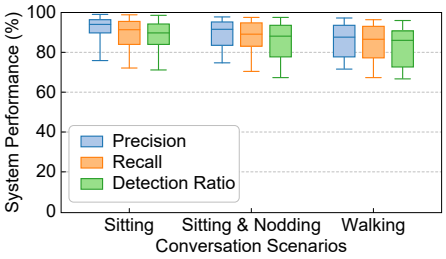


Fig. 16. Case study performance.

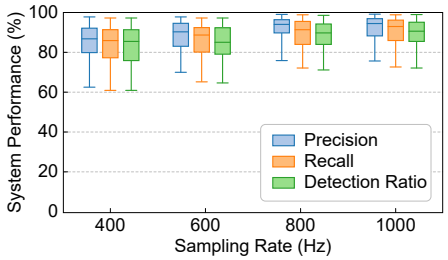


Fig. 17. Impact of different sampling rates.

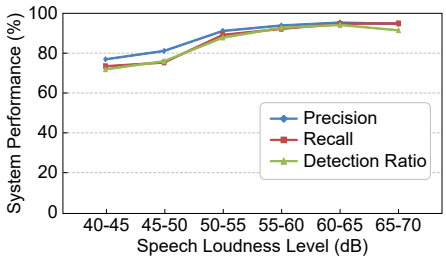


Fig. 18. Impact of different speech decibel levels.

recall, and detection ratio are all above 70%, which is still acceptable. The results outperform existing motion sensor-based speech eavesdropping in the user-independent scenario, validating the effectiveness of *EarSpy*.

Besides, we find that the most errors are reported by participants 10 and 12, both of whom are female and non-native English speakers. We find that their measured signals differ significantly in both time and frequency domains from those of others. We plan to enrich the training dataset in the future to handle the wide spectrum of human speech and diversity of ear canal dynamic motions.

D. End-to-End Case Study: Eavesdropping Passwords From Phone Conversations

We now present an end-to-end case study of eavesdropping on the wearer's speech during phone conversations. We focus on a real-world scenario where the victim wears a pair of earphones to make a phone call to a remote caller, and tell the password during the phone conversation. We first train a CNN-CTC framework with a dataset that includes sequence samples containing digits, "password", "is", and various other words. To launch this attack, we recruit three new participants (2 males and 1 female) and collect data in three cases: i) sitting, ii) sitting and nodding (light body motion), and iii) walking (intense body motion). The conversation script is designed to include "The password is" and a random 6-digits password. We ask the participants to collect 20 conversations per case and obtain $20\times3\times3$ conversations.

Fig. 16 shows the precision, recall, and detection ratio of *EarSpy* under each case. We observe that the *sitting* case has the best performance, confirming that *EarSpy*'s practical usability. Besides, the good performance of the *sitting & nodding* case and the *walking* case show that *EarSpy* successfully eavesdrops on the phone conversations even when the user involves body motions. Furthermore, we find that the frequency of different words in the sentence varies, and those with more training samples have better performance. We will study the detailed performance of each word in further work.

E. System Robustness

1) Impact of Sampling Rate: A low sampling rate allows few samples to be collected and may impair the recognition performance. We particularly evaluate *EarSpy* under four sampling rates, including 400, 600, 800, and 1,000 Hz. Fig. 17 shows a box plot regarding precision, recall, and detection ratio under the four cases. We can observe a continuous improvement in system performance as the sampling rate increased from 400 Hz to 800 Hz. We also find that the system performance is not significantly improved when the sampling rate increases to 1,000 Hz. This is due to the decay of BCVs above 400 Hz as it propagates via bone and tissue; the sampling rate of 800 Hz is sufficient to capture most of the speech information. Since today's earphones can collect acceleration hundreds of times per second, we believe the proposed attack is practical.

2) Impact of Speech Loudness: The average decibel level of human speech is estimated between 55 and 65 decibels [38]. A louder speech enables a stronger response in the motion sensor, which benefits speech analysis. Whereas a softer speech might cause insufficient profiling of the speech information, thereby impairing system performance. We study the impact of speech loudness with varying speech average decibel levels between 40 and 70 decibels. Note that the speech average decibel level is calculated from the air-conduction speech audio recorded 30 cm away from the wearer. We report the system performance in Fig. 18, where precision, recall, and detection ratio are compared according to speech average decibel levels. We can observe that as the speech loudness grows, the system performance shows an improvement. When the speech decibel level is between 55 to 65, *EarSpy* has average precision of 94.70%, recall of 93.07%, and detection ratio of 92.54%, respectively. The results suggest *EarSpy* is promising to handle speeches with various loudness.

3) Impact of Hardware Difference: We evaluate the adaptiveness of *EarSpy* by implementing our system with three different prototypes (as shown in Fig. 13) and collected over 200 sentences per device. In particular, we use data from each prototype to train the speech recognition model and test it with data from two other prototypes, denoted as cases 1, 2, and 3, respectively. Moreover, we construct a dataset with all data from the three prototypes, and use 70% for training the model while the remaining 30% to test it (denoted as *case 4*). The results are reported in Fig. 19. We notice that it is not easy for the deep-learning model trained by data from one type of hardware to generalize to other types of hardware. However, in *case 4*, we can observe a significant performance increase when the model is trained and tested with mixed data. The results indicate that *EarSpy* can be generalized to unseen devices if we regularly update the training dataset to include enough earphone models.

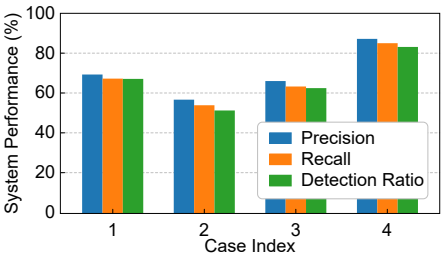


Fig. 19. Performance of different hardware.

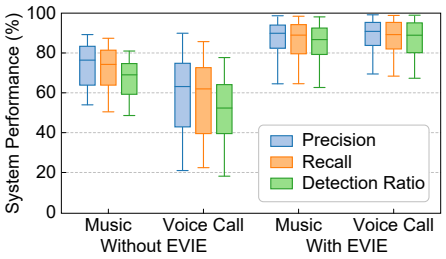


Fig. 20. Effectiveness of earphone vibration interference elimination.

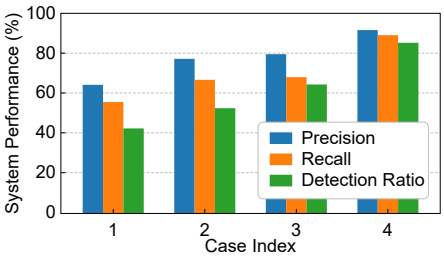


Fig. 21. Effectiveness of handling user dependence.

F. Key Algorithm Evaluation

1) Effectiveness of Earphone Vibration Interference Elimination: The motion sensor can be interfered by the earphone's speaker playing audio because they are in physical contact with the same board of the earphone. We propose to eliminate such interference by tracking the speech pitch as described in Section III-C1. In Fig. 20, we compare the system performance applying the proposed earphone vibration interference elimination (EVIE) method to the same method without applying EVIE. Apparently, EVIE can significantly increase the precision, recall, and detection ratio in both music and voice call cases (around 50 dB), confirming its effectiveness.

2) Effectiveness of Handling User Dependence: We propose two algorithms to handle the dependence of ear dynamics on individual differences. We particularly compare four cases differentiated by whether the envelope instability reduction method (i.e., EIR$\in$ 0,1) and data augmentation method are applied (i.e., DA$\in$ 0,1), namely *case 1*: EIR=0, DA=0, *case 2*: EIR=1, DA=0, *case 3*: EIR=0, DA=1, and *case 4*: EIR=1, DA=1. We conduct leave-one-participant-out-validation and report the averaged results in Fig. 21. Obviously, the two algorithms are critical and effective in improving the overall system performance.

V. Related Works

Over the past few years, active efforts have been devoted to using the zero-permission motion sensor (e.g., accelerometer and gyroscope) for side-channel attacks.

Existing studies such as TapLogger [39], TouchLogger [40], Accessory [41], and TapPrints [42] have shown that the motion sensors on mobile devices can be used to infer keystroke input on the virtual keyboard. Additionally, the motion sensors mounted on wearable devices are usually associated with a lot of hand or body movements. Such motion sensor data could be decoded to infer screen-click positions [43] and password input on real keypads (e.g., ATM keypads, real keypads) [44], [45]. Moreover, the motion sensors can track the displacement of motion, thus have been leveraged for localizing users and tracking moving trajectory [46]–[48].

More recently, new security vulnerabilities of motion sensors eavesdropping on private speech have drawn significant attention. Gyrophone [6] leverages the gyroscope in a smartphone to recognize speech replayed by external loudspeakers. Moreover, Speechless [7] analyzes various attacking scenarios and identifies the feasibility of motion sensors picking up sound vibrations propagating along the same surface on which they are placed. Furthermore, Spearphone [9], AccelEve [8] use the smartphone motion sensor to eavesdrop on the speeches from reverberations generated from the same smartphone loudspeaker. These methods focus on using motion sensors on smartphones to eavesdrop on the replayed speech generated by loudspeakers.

Recognizing users' live speech is more challenging and arouses the attackers' interest. Vibphone [10] takes advantage of the physical contact between the smartphone and the cheek during phone calls to capture cheek vibrations associated with voice using smartphone motion sensors. It then decodes the vibrations to recognize the speech content. Face-Mic [11] shows the initial success in inferring the wearer's live speech and information (e.g., gender and speaker identity) using the motion sensor embedded in AR/VR headsets. Face-Mic is very close to our work. It profiles the facial bone-conduction vibrations and facial motions associated with speech. In our work, we investigate new novel features of ear dynamics (combining bone-conduction vibrations from the ear and ear canal dynamic motions) captured via the motion sensor on earphones, which has not been explored previously in the literature. Note that the ear dynamics studied by *EarSpy* and facial dynamics studied by Face-Mic have significantly different time and frequency characteristics. Therefore, *EarSpy* develops novel algorithms to parse the sensor measurements and address unique challenges.

VI. Conclusion

In this paper, we present *EarSpy*, which is the first effort to investigate speech eavesdropping via earphone's motion sensors. *EarSpy* leverages the accelerometer to capture the speech-induced ear dynamics including BCVs and ECDMs, then parses them to infer speech content. By designing several novel signal processing algorithms and using a CNN-CTC-based deep learning framework, *EarSpy* achieves accurate and user-independent speech recognition. Extensive experiments involving 14 participants demonstrate that *EarSpy* is highly effective and robust to varying sampling rates, speech loudness, and hardware. On top of revealing potential risks, we believe *EarSpy* can draw more attention from manufacturers and the research community to enhance privacy protection on earphones.

REFERENCES

[1] N. R. Frick, K. L. Wilms, F. Brachten, T. Hetjens, S. Stieglitz, and B. Ross, "The Perceived Surveillance of Conversations through Smart Devices," *Electronic Commerce Research and Applications*, vol. 47, p. 101046, 2021.

[2] Microsoft, "Use Speech in Windows Mixed Reality," 2022. [Online]. Available: https://support.microsoft.com/en-us/windows/use-speech-in-windows-mixed-reality-af24e0a9-7e17-b542-3720-203e278e588e

[3] Oculus, "Oculus Privacy Policy," 2022. [Online]. Available: https://www.oculus.com/legal/privacy-policy-for-oculus-account-users/

[4] R. R. Choudhury, "Earable Computing: A New Area to Think About," in *Proc. of the 22nd ACM HotMobile*, 2021, pp. 147–153.

[5] S. V. Dusan, E. B. Andersen, A. Lindahl, and A. P. Bright, "System and Method of Detecting a User's Voice Activity Using an Accelerometer," Sep. 6 2016, US Patent 9,438,985.

[6] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing Speech From Gyroscope Signals," in *Proc. of USENIX Security Symposium*, 2014, pp. 1053–1067.

[7] S. A. Anand and N. Saxena, "Speechless: Analyzing the Threat to Speech Privacy From Smartphone Motion Sensors," in *Proc. of IEEE SP 2018*, 2018, pp. 1000–1017.

[8] Z. Ba, T. Zheng, X. Zhang, Z. Qin, B. Li, X. Liu, and K. Ren, "Learning-Based Practical Smartphone Eavesdropping With Built-in Accelerometer," in *NDSS 2020*, 2020, pp. 1–18.

[9] S. A. Anand, C. Wang, J. Liu, N. Saxena, and Y. Chen, "Spearphone: A Speech Privacy Exploit via Accelerometer-Sensed Reverberations From Smartphone Loudspeakers," *arXiv preprint arXiv:1907.05972*, 2019.

[10] W. Su, D. Liu, T. Zhang, and H. Jiang, "Towards Device Independent Eavesdropping on Telephone Conversations with Built-in Accelerometer," *Proce. of ACM IMWUT 2021*, vol. 5, no. 4, pp. 1–29, 2021.

[11] C. Shi, X. Xu, T. Zhang, P. Walker, Y. Wu, J. Liu, N. Saxena, Y. Chen, and J. Yu, "Face-Mic: Inferring Live Speech and Speaker Identity via Subtle Facial Dynamics Captured by AR/VR Motion Sensors," in *Proc. of the 27th ACM MobiCom*, 2021, p. 478–490.

[12] Apple, "AirPods Using Ultralow-Power Chips to Deliver an Incredible 5 Hours of Use Time on One Charge," 2022. [Online]. Available: https://www.apple.com/airpods-pro/specs/

[13] L. Zhang, P. H. Pathak, M. Wu, Y. Zhao, and P. Mohapatra, "Accelword: Energy Efficient Hotword Detection Through Accelerometer," in *Proc. of the 13th ACM MobiSys*, 2015, pp. 301–315.

[14] Y. Cao, H. Chen, F. Li, S. Yang, and Y. Wang, "Awash: Handwashing Assistance for the Elderly with Dementia via Wearables," in *Proc. of IEEE INFOCOM 2021*, 2021, pp. 1–10.

[15] J. Liu, W. Song, L. Shen, J. Han, X. Xu, and K. Ren, "MandiPass: Secure and Usable User Authentication via Earphone IMU," in *Proc. of the 41st IEEE ICDCS*, 2021, pp. 674–684.

[16] G. McCandless and D. McIntyre, *The Craft of Contemporary Commercial Music*. Routledge, 2017.

[17] Y. Cao, W. Tavanapong, K. Kim, and J. Oh, "Audio-Assisted Scene Segmentation for Story Browsing," in *International Conference on Image and Video Retrieval*. Springer, 2003, pp. 446–455.

[18] N. N. Lokhande, N. S. Nehe, and P. S. Vikhe, "Voice Activity Detection Algorithm for Speech Recognition Applications," in *Proc. of ICCIA 2012*, vol. 6, 2012, pp. 1–4.

[19] X. Guo, J. Liu, and Y. Chen, "Fitcoach: Virtual Fitness Coach Empowered by Wearable Mobile Devices," in *Proc. of IEEE INFOCOM 2017*, 2017, pp. 1–9.

[20] G. Hu and D. Wang, "Monaural Speech Segregation Based on Pitch Tracking and Amplitude Modulation," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1135–1150, 2004.

[21] J. Gong, X. Zhang, Y. Huang, J. Ren, and Y. Zhang, "Robust Inertial Motion Tracking through Deep Sensor Fusion across Smart Earbuds and Smartphone," *Proc. of ACM IMWUT 2021*, vol. 5, no. 2, pp. 1–26, 2021.

[22] Y. Shao and D. Wang, "Model-Based Sequential Organization in Cochannel Speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 289–298, 2005.

[23] V. Hohmann, "Frequency Analysis and Synthesis Using a Gammatone Filterbank," *Acta Acustica united with Acustica*, vol. 88, no. 3, pp. 433–442, 2002.

[24] B. R. Glasberg and B. C. Moore, "Derivation of Auditory Filter Shapes From Notched-Noise Data," *Hearing Research*, vol. 47, no. 1, pp. 103–138, 1990.

[25] G. Hu and D. Wang, "Monaural Speech Segregation Based on Pitch Tracking and Amplitude Modulation," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1135–1150, 2004.

[26] R. Meddis, "Simulation of Auditory–Neural Transduction: Further Studies," *The Journal of the Acoustical Society of America*, vol. 83, no. 3, pp. 1056–1063, 1988.

[27] D. L. Wang and G. J. Brown, "Separation of Speech From Interfering Sounds Based on Oscillatory Correlation," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 684–697, 1999.

[28] M. Weintraub, "A Computational Model for Separating Two Simultaneous Talkers," in *Proc. of IEEE ICASSP 1986*, vol. 11, 1986, pp. 81–84.

[29] H. A. Sackeim, "Morphologic Asymmetries of the Face: A Review," *Brain and Cognition*, vol. 4, no. 3, pp. 296–312, 1985.

[30] Y. Cao, H. Chen, F. Li, and Y. Wang, "CanalScan: Tongue-Jaw Movement Recognition via Ear Canal Deformation Sensing," in *Proc. of IEEE INFOCOM 2021*, 2021, pp. 1–10.

[31] C. Goodall, "Procrustes Methods in the Statistical Analysis of Shape," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 53, no. 2, pp. 285–321, 1991.

[32] F. J. Rohlf, "Bias and Error in Estimates of Mean Shape in Geometric Morphometrics," *Journal of Human Evolution*, vol. 44, no. 6, pp. 665–683, 2003.

[33] A. G. Nahapetyan, "Bilinear Programming," 2009.

[34] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A Corpus Derived From Librispeech for Text-To-Speech," *arXiv preprint arXiv:1904.02882*, 2019.

[35] E. Kauderer-Abrams, "Quantifying Translation-Invariance in Convolutional Neural Networks," *arXiv preprint arXiv:1801.01450*, 2017.

[36] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[37] C. Alberti and M. Bacchiani, "Automatic Captioning in YouTube," 2009. [Online]. Available: https://ai.googleblog.com/2009/12/automatic-captioning-in-youtube.html

[38] S. H. H. Al-Taai, "Noise and Its Impact on Environmental Pollution," *Materials Today: Proceedings*, 2021.

[39] Z. Xu, K. Bai, and S. Zhu, "TapLogger: Inferring User Inputs on Smartphone Touchscreens Using On-Board Motion Sensors," in *Proc. of the 5th ACM WISEC*, 2012, pp. 113–124.

[40] L. Cai and H. Chen, "TouchLogger: Inferring Keystrokes on Touch Screen From Smartphone Motion," in *Proc. of the USENIX HotSec*, 2011.

[41] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: Password Inference Using Accelerometers on Smartphones," in *Proc. of the 12th ACM HotMobile*, 2012, pp. 1–6.

[42] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "TapPrints: Your Finger Taps Have Fingerprints," in *Proc. of the 10th ACM MobiSys*, 2012, pp. 323–336.

[43] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic, "(Smart) Watch Your Taps: Side-Channel Keystroke Inference Attacks Using Smartwatches," in *Proc. of ACM ISWC 2015*, 2015, pp. 27–30.

[44] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, "Mole: Motion Leaks Through Smartwatch Sensors," in *Proc. of the 21st ACM MobiCom*, 2015, pp. 155–166.

[45] C. Wang, X. Guo, Y. Wang, Y. Chen, and B. Liu, "Friend or Foe? Your Wearable Devices Reveal Your Personal Pin," in *Proc. of the 11st ACM ASIA CCS*, 2016, pp. 189–200.

[46] J. Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang, "Accomplice: Location Inference Using Accelerometers on Smartphones," in *Proc. of IEEE COMSNETS 2012*, 2012, pp. 1–9.

[47] S. Nawaz and C. Mascolo, "Mining Users' Significant Driving Routes With Low-Power Sensors," in *Proc. of the 12th ACM SenSys*, 2014, pp. 236–250.

[48] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring User Routes and Locations Using Zero-Permission Mobile Sensors," in *Proc. of IEEE SP 2016*, 2016, pp. 397–413.