SeVI: Boosting Secure Voice Interactions with Smart Devices

Xiao Wang[1], Hongzi Zhu[1], Shan Chang[2], Xudong Wang[1]
[1]Shanghai Jiao Tong University, China
[2]Donghua University, China
{wangxiaoAlan, hongzi, wxudong}@sjtu.edu.cn, changshan@dhu.edu.cn

***Abstract*—Voice interaction, as an emerging human-computer interaction method, has gained great popularity, especially on smart devices. However, due to the open nature of voice signals, voice interaction may cause privacy leakage. In this paper, we propose a novel scheme, called *SeVI*, to protect voice interaction from being deliberately or unintentionally eavesdropped. SeVI actively generates jamming noise of superior characteristics, while a user is performing voice interaction with his/her device, so that attackers cannot obtain the voice contents of the user. Meanwhile, the device leverages the prior knowledge of the generated noise to adaptively cancel received noise, even when the device usage environment is changing due to movement, so that the user voice interactions are unaffected. SeVI relies on only normal microphone and speakers and can be implemented as light-weight software. We have implemented SeVI on a commercial off-the-shelf (COTS) smartphone and conducted extensive real-world experiments. The results demonstrate that SeVI can defend both online eavesdropping attacks and offline digital signal processing (DSP) analysis attacks.**

***Index Terms*—jamming noise, voice interaction, smart device,**

I. Introduction

With the rapid spread of mobile devices, the way of human-computer interaction is also evolving. Voice interaction as an emerging interaction method is becoming more and more mature and popular, including voice input methods like iFLYTEK [1] and voice assistants like Google Assistant [2], Siri [3], and Cortana [4]. However, due to the open nature of voice signals, these interactions may face great security threats, where attackers can obtain private information through eavesdropping or recording. With the widespread use of voice technology, the security issues have become more and more crucial. Therefore, providing strong protection for voice interactions is of great importance.

To provide secure voice interaction on smart devices, a practical scheme should meet the following four requirements: 1) *strong security:* a scheme should be able to defend eavesdropping attack with human hearing and digital signal processing (DSP) analysis attacks. 2) *high transparency:* the scheme should have negligible side effects on existing voice interaction applications. 3) *good usability:* the scheme should not rely on extra hardware and can be implemented on commercial off-the-shelf (COTS) devices as software. In addition, the scheme should also be able to use during the movement of users. 4) *low power consumption:* as most smart devices are battery powered, it is essential for such a scheme to be light-weight and to have a low power consumption.

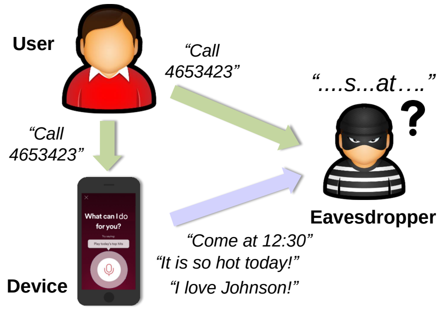

Fig. 1. The basic idea of SeVI, where user voice is deliberately jammed by the user's smartphone.

In the literature, there are some existing schemes targeting on similar problems. A category of schemes utilize the characteristics of hardware. For example, Backdoor [5] and Dolphinattack [6] leverage the nonlinearity of microphones to perform hidden interference or communication using ultrasonic signal. However, these methods will also affect the normal use of voice interaction applications and cannot protect the user voice from human hearing. Another category of schemes use noises to protect acoustic signals [7] [8]. However, these schemes are designed for data communication and acoustic signals used are quite different from human voice. Recently, convert communication is achieved with encoded audio signals [9], where the multipath effect and multiple speakers are explored to cancel noises at certain spots. Besides multiple speakers are required, it fits in static environments. Another interesting scheme [10] filters out sensitive voice contents from continuous acoustic sensing devices using an extra hardware. As a result, there exists no successful solution, to the best of our knowledge, to providing secure voice interactions on smart devices.

In this paper, we propose a novel scheme, called SeVI, to tackle the secure voice interaction problem. As illustrated in Figure 1, the core idea of SeVI is for a smart device to actively generate jamming noise to cover the voice of a user so that attackers cannot obtain the voice contents of the user. Meanwhile, the device leverages the prior knowledge of the generated noise to adaptively cancel received noise so that the user voice interactions are unaffected.

There are two main challenges in SeVI design. First, attackers can be very powerful and can conduct complex signal processing, which may separate user voice and noise. Meanwhile, human hearing is quite intelligent and hard to jam. To cope with the first challenge, we design a jamming noise generation scheme by taking advantage of the masking effect. The closer the frequency and the louder the masking sound, the stronger the masking effect. Inspired by this phenomenon, to obtain superior masking effect, we randomly select a number of pre-recorded speech records of users to generate a jamming noise. Such a jamming noise has two unique features as follows. First, it has a very similar frequency spectrum as an input voice, which makes it hard to filter out the jamming noice from the perceived sound, combined by the jamming noise and the input voice. Second, it adds an extra semantic difficulty for an eavesdropper to understand the meaning of the combined sound.

Second, jamming noise would also interfere with user voice. Though the generated jamming noise is known to the device, the timing and the contents of user voices and the time-varying channel between speakers and the microphone are unknown, making jamming noise self-cancellation very challenging. To deal with this challenge, the prior knowledge of the generated jamming signal is first used to detect user voice segments within the interfered sound. Then the time-varying channel can be constantly estimated by comparing non-user-voice segments in the generated jamming signal and the interfered sound. Finally, with estimated channel response, the recorded jamming noise can be estimated and removed from the combined sound.

We implemented a prototype system on Google Pixel3 smart phone as an app. We have conducted intensive experiments under different conditions to evaluate the performance of SeVI. The results show that the proposed noise generation scheme can indeed secure user voice interaction. When the device is 1.5m away from attackers, neither human nor machine can distinguish user voice content from the jamming noise. Moreover, the derived voice after the noise self-cancellation can achieve a recognition rate of 85% on average for voice interactive software and full comprehension for human.

II. Related work

Islam et al. proposed SoundSifter to tackle the overhearing problem of continuous acoustic sensing devices [10]. They built an independent embedded system to cover the device thus filtering out signals from unwanted sources. This work requires extra hardware and it is targeting on the untrusted interactive device itself, so that it cannot protect against eavesdropping by surrounding malicious attackers, while in our system we assume the interactive device is trustworthy.

Nowadays, there have been many active noise-canceling headphones in the market which use microphones to record ambient noise and play an inverted signal to compensate it. Thus, the most ideal case is letting smart devices cancel out user's voice in real time. However, this technique only works well on low frequency and stable sounds, plus the effect of process delay, making it unrealistic to compensate user's voice.

Roy et al. proposed Backdoor [5], which exploiting the non-linearity of microphone to conduct covert communication and jam spy microphones. Two different ultrasonic sounds will create an audible frequency range sound in the microphone after passing through the microphone's non-linear diagram which can be used to jam unknown spy microphones . However, this technique also needs extra ultrasonic transmitters. And more importantly, it cannot defend human eavesdropping attack.

In recent research, Chaman et al. [9] encoded audio signals with noises to achieve convert communication. It exploits the multi-path effect and transmits signals with multiple speakers so that noises can cancel each other out only at certain spots. The limitation of this work lies in the requirement of multi-speaker encoding of existing audios which certainly cannot be achieved by human voice. In addition, this method requires sophisticated modeling of current environment and cannot be used in dynamic scenarios.

Dhwani proposed by Nandakumar et al. also uses the jamming-based method to conduct secure acoustic near filed communication [7]. It emits a jamming signal to provide protection and uses a self-interference cancellation method to decode the acoustic signal. The main difference is that Dhwani is for data communication between devices so that the acoustic content does not need to be interpretable by human, but our system is target on human voice interaction, which is harder to effectively interference due to human auditory intelligence and rises higher requirement to the jamming noise self-cancellation. There are other researches on acoustic security using some encoding or encryption method, but they are all limited to devices and cannot be applied to actual voice interaction [8].

III. System Model and Design Goals

A. System and Attack Models

We consider to protect private voice interactions in public environments, e.g., in a cafe or an elevator, on a bus or a subway train, where people are allowed to talk and they are surrounded by untrusted individuals. There are three entities considered in the system as follows:

- **Users.** A user is an authorized person, who can operate and interact with a smart device via voice, and expects that his/her private voice messages or instructions would not be eavesdropped by unintentional strangers and deliberate attackers.

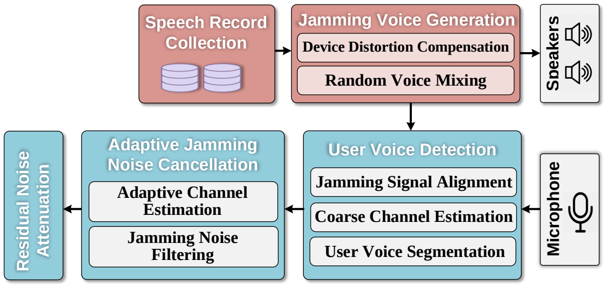

Fig. 2. System architecture of SeVI, consisting of two main parts, i.e., obscuring voice jamming (the upper half), and jamming noise self-cancellation (the lower half).

- **Smart devices.** A smart device is capable of detecting and recording a user's voice. In addition, the device has one or more speakers and moderate computation and storage capabilities.
- **Attackers.** An attacker tries to eavesdrop the voice contents of a user by human hearing or by offline digital signal processing (DSP) analysis on recorded sounds. We consider attackers use COTS mobile devices that can conduct stereo recording with two separated microphones. Moreover, he/she can can change his/her position in the scene to approach the user or to find a better place for eavesdropping.

B. Design Goals

A practical voice interaction protection scheme should meet the following goals:

- **Strong security.** The scheme should protect private voice contents of users from being obtained by unintentional strangers and deliberate attackers with high probability.
- **High transparency.** The scheme should have negligible side effects on existing voice interaction applications.
- **Good usability.** The scheme should not rely on extra hardware and can be implemented on COTS devices as software. Moreover, users should have little intervention in using the scheme.
- **Low power consumption.** The scheme should power efficient since it may run on mobile devices, powered by batteries.

IV. Overview of SeVI

The system architecture of SeVi, as illustrated in Fig. 2, consists of two main technical components as follows:

Obscuring Voice Jamming (OVJ). Human hearing is very sensitive and hard to be jammed. OVJ takes a two-fold strategy: first, users' own distorted speech records are collected (*Speech Record Collection*), restored (*Device Distortion Compensation*) and used to generate jamming noises with a very similar frequency spectrum as user voices; second, multiple speech records are randomly selected and mixed (*Random Voice Mixing*) to further confuse the semantic meanings of user voices.

Jamming Noise Self-cancellation (JNS). Jamming noise not only confuses attackers but also interferes with user voices. Though the generated jamming noise is known to the device, the timing and the contents of user voices and the time-varying channel between speakers and the microphone are unknown, making jamming noise self-cancellation very challenging. To tackle this challenge, JNS first conducts cross-correlation to align the generated jamming noise with the recorded sound (*Jamming Signal Alignment*). It then uses the beginning two-second window as a preamble to roughly estimate the speaker-to-microphone channel (*Coarse Channel Estimation*) and obtains the durations of user voices in the recorded sound (*User Voice Segmentation*). Given the separated noise segments, *Adaptive Channel Estimation* is conducted on each segment to continuously track the time-varying channel. With most up-to-date channel estimate, jamming noise is eliminated from user voice segments (*Jamming Noise Filtering*). Finally, JNS conducts *Residual Noise Attenuation* to further remove residual noise due to channel estimate errors.

SeVI only requires a microphone and speakers on smart devices. It can be implemented as a building block of voice interaction applications or as a middleware providing general voice protection APIs for upper-layer applications.

V. Obscuring Voice Jamming

The phenomenon that human ear's perception threshold of one sound is increased because of the presence of other sounds, which is referred to as the *auditory masking effect* [11] [12]. In particular, a sound of *closer frequency* and *higher intensity* has a stronger masking effect. Inspired by this phenomenon, to obtain superior masking effect, we randomly select a number of pre-recorded speech records of users to generate a jamming noise. Such a jamming noise has two unique features as follows. First, it has a very similar frequency spectrum as an input voice, which makes it hard to filter out the jamming noice from the perceived sound, combined by the jamming noise and the input voice. Second, it adds an extra semantic difficulty for an eavesdropper to understand the meaning of the combined sound.

A. Speech Record Collection

SeVI needs to collect a library of user speech records before providing protection for voice interactions. In specific, a set of sentences can be randomly picked up from pre-loaded digital books and webpages. After each sentence is displayed on the screen of a target device, the user is required to keep the device close to his/her mouth and read the sentence, which is recorded to construct the library. We do not record user conversations or voice instructions for this purpose because such voices might still contain private information even though they were recorded in the past.

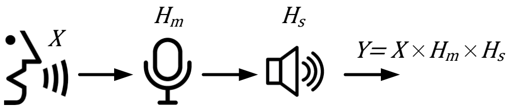

Fig. 3. User voice distorted by device frequency responses.

B. Jamming Voice Generation

1) Device Distortion Compensation: It is desirable that a jamming voice sounds exactly like an input voice of the user. However, if a speech record is directly played back, it sounds very different from the original voice of the user. As illustrated in Figure 3, the reason is that the original voice is first distorted by the microphone and then by the speaker of the device. In frequency domain, we have $Y = X \times H_m \times H_s$ [1], where Y and X are the played sound and the original voice, respectively, and H_m and H_s are the finite frequency responses of the microphone and the speaker of the device. Such distortion can severely weaken the masking effect of a jamming voice. Therefore, it is necessary to compensate this distortion by eliminating the impact of device frequency responses.

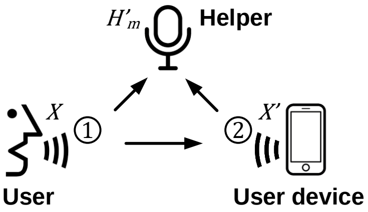

Fig. 4. Device frequency response measurement model.

It is non-trivial to acquire the frequency responses of the microphone and the speaker of the device. SeVI needs to use another device once to help measure $H_m \times H_s$ as a whole. Specifically, as illustrated in Figure 4, the user first speaks a sentence X, which is recorded by a helper device and the user device at the same time. The recorded sounds at the helper and the device are $Y_u = X \times H'_m$, where H'_m is the frequency response of the helper's microphone, and $X' = X \times H_m$, respectively. Then, the device plays back the recorded X', which is recorded by the helper as $Y_d = X' \times H_s \times H'_m$. Divide Y_d by Y_u, we get

$$\frac{Y_d}{Y_u} = \frac{X' \times H_s \times H'_m}{X_\times H'_m} = \frac{X \times H_m \times H_s}{X} = H_m \times H_s. \quad (1)$$

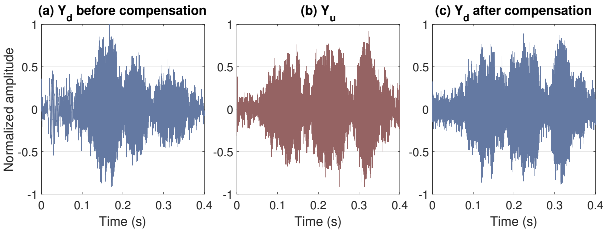

Fig. 5. An example of eliminating device distortion, where a speech record after compensation sounds more like the original user voice.

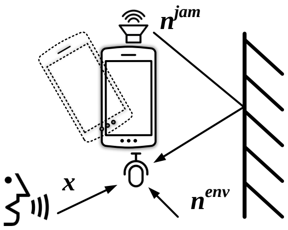

Fig. 6. A combination of user voice, generated jamming noise and environmental noise are recorded while the device might be in motion.

In above measurement, we only consider air propagation channels dominated by strong direct paths. Such a channel has equivalent attenuations among frequency components. As human ear is not sensitive to phase, therefore, $H_m \times H_s$ can be represented as $\alpha Y_d / Y_u$, where α is a constant due to channel attenuations. Note that, for a particular device, the measurement on its $H_m \times H_s$ should only be conducted once. It would be more convenient if device manufacturers could provide such parameters for the same batch of products.

With measured $H_m \times H_s$, device distortion can be compensated by dividing those speech records in the user voice library by $H_m \times H_s$ before they are played. For example, Figure 5 depicts Y_d before and after distortion compensation, and Y_u of a user speech record, respectively. It can be seen that the envelope of the signal after compensation is much more similar with that of Y_u.

2) Random Voice Mixing: Every time a user conducts a voice interaction with a device, SeVI randomly selects a few speech records from the library and mix them up to form a jamming noise. Specifically, as pauses exist in speeches, speech records are cut into small segments of 100ms. If the average power of a segment is higher than a threshold, this segment is considered as a *voice segment*; otherwise, it is considered as a *null segment*. The goal of the random voice mixing algorithm is to guarantee that the number of overlapping voice segments at any point of time is no less than two. This can be done by adding new records to the jamming voice from the current point of time if the condition is not satisfied. This process repeats until the user finishes the interaction.

VI. Jamming Noise Self-cancellation

Superior jamming noises can defend eavesdropping attacks but they also interfere with user input voices. As illustrated in Figure 6, the sound being recorded by the user device, denoted as signal $d(t)$, is a combination of user voice, denoted

[1] Comparing with device distortions, the distortion caused by the air channel between the mouth of the user and the microphone is negligible, especially when they are very close. For simplicity, we ignore the impact of channel frequency response of air channels dominated by strong direct path.

as $x(t)$, generated jamming noise, denoted as $n^{jam}(t)$, and environmental noise, denoted as $n^{env}(t)$. Specifically, we have

$$d(t) = n^{jam}(t) * h_{s,m}(t) + x(t) + n^{env}(t) \qquad (2)$$

where $h_{s,m}(t)$ denotes the impulse response of the channel between the speaker and the microphone. As the device may be in motion, $h_{s,m}(t)$ varies over time. Because the device cannot know $h_{s,m}(t)$ in advance, given $d(t)$ and $n^{jam}(t)$, it is hard to eliminate $n^{jam}(t) * h_{s,m}(t)$ from $d(t)$. Furthermore, as the device does not know $x(t)$ and $n^{env}(t)$ either, it is also hard to accurately estimate $h_{s,m}(t)$. As a result, it is of great challenge to cancel the generated jamming noise.

In SeVI, we decouple this problem in three steps. First, the prior knowledge of $n^{jam}(t)$ is used to detect user voice segments within the combined sound. Second, $h_{s,m}(t)$ can be constantly estimated to track the change of the channel, using non-user-voice segments. Finally, with estimated $h_{s,m}(t)$, the recorded jamming noise can be estimated and removed from the combined sound.

A. *User Voice Detection*

With the jamming scheme devised in SeVI, user voice is usually overwhelmed by jamming noise of similar frequency characteristics. Traditional double talk detection techniques [16] [17] cannot be used in this setting. In SeVI, we require that the user does not talk for a few seconds after the device plays a jamming noise.

1) Jamming Signal Alignment: Due to uncertain system delay, the jamming noise being played and the recorded sound are not aligned. To align both signals, we use the beginning two seconds of the jamming noise signal as a *preamble* and conduct cross-correlation on the sound being recorded. Ideally, the cross-correlation reaches the maximum when both signals are aligned. In practice, as the received signal varies from its original form, we consider that both signals are aligned if a correlation peak larger than a threshold. In our implementation, we set the threshold as the 85% of the maximum correlation value.

2) Coarse Channel Estimation: Figure 7 illustrates a jamming noise signal $n^{jam}(t)$ in subplot (a) and the aligned recorded sound $y(t)$ in subplot (b). With no user voice at the beginning of the recorded sound, we can use the preamble in $n^{jam}(t)$ and received preamble in $y(t)$ to estimate a coarse channel response, denoted as $\hat{h}_{s,m}(0)$. We then use $\hat{h}_{s,m}(0)$ to roughly estimate the received jamming noise, i.e., $\hat{h}_{s,m}(0) * n^{jam}(t)$. Although the estimated jamming noise is not accurate, it contains most energy of the received jamming noise. By subtracting the estimated jamming noise from the recorded sound, the residual power, as depicted in subplot (c) of Figure 7, mainly comes from the user voice and ambient noise.

3) User Voice Segmentation: Given the residual power signal, denoted as $r(t) = y(t) - \hat{h}_{s,m}(0) * n^{jam}(t)$, we use a sliding window of l samples to calculate the power of the window starting from the ith sample as

$$p(i) = \frac{1}{l} \sum_{k=i}^{i+l-1} |r(i)|^2 \, . \qquad (3)$$

We detect user voice by checking whether $p(\cdot)$ is larger than a threshold and divide $r(t)$ and the corresponding $y(t)$ into user voice segments and non-voice segments. In our implementation, we set $l = 256$ at a sampling rate of 16KHz and set the threshold to the 40% of the maximum $p(\cdot)$ throughout $r(t)$.

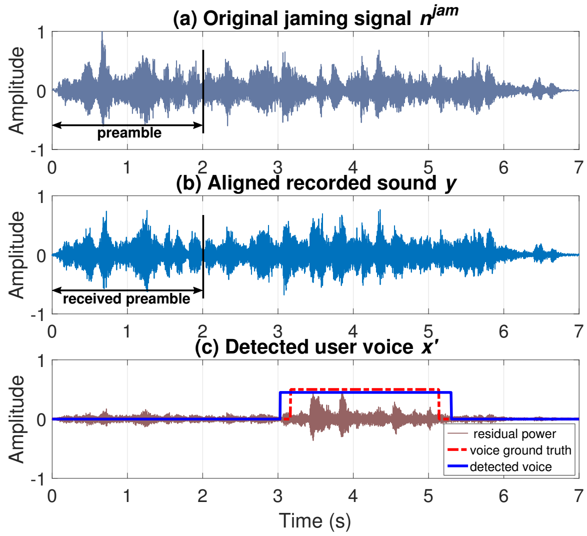

Fig. 7. The user voice can be roughly detected by subtracting the coarsely estimated jamming noise from the recorded sound.

B. *Adaptive Jamming Noise Cancellation*

With coarse channel estimation, we can detect user voice segments from the combined recored sound but the detected voice segments contain a lot of unpleasant noise. The reason is mainly because $h_{s,m}(t)$ varies over time and the initial estimate $\hat{h}_{s,m}(0)$ is out-of-date.

We use non-voice segments in the recorded sound $y(t)$ to keep the track of $h_{s,m}(t)$, adopting a frequency domain adaptive filter (FDAF) algorithm [18] for its good convergence performance and fast computation speed. Specifically, for a window of size Q, sliding Fast Fourier Transformation (FFT) with 50% overlapping is conducted on non-voice segments in the recorded sound $y(t)$ and the corresponding jamming noise signal $n^{jam}(t)$, respectively. Let $Y(i)$ and $N^{jam}(i)$ denote the Fourier transformation of the ith segment of $y(t)$ and $n^{jam}(t)$, respectively and let $\hat{H}_{s,m}(i)$ denote the channel frequency response for the ith segment.

The estimated jamming signal for the ith segment, therefore, is $N^{jam}(i) \times \hat{H}_{s,m}(i)$ and the estimated error, denoted as $E(i)$, is $E(i) = Y(i) - N^{jam}(i) \times \hat{H}_{s,m}(i)$. We define the mean square error of $E(i)$ as the cost function and try to minimize the cost function by updating the estimated $\hat{H}_{s,m}(i)$ along

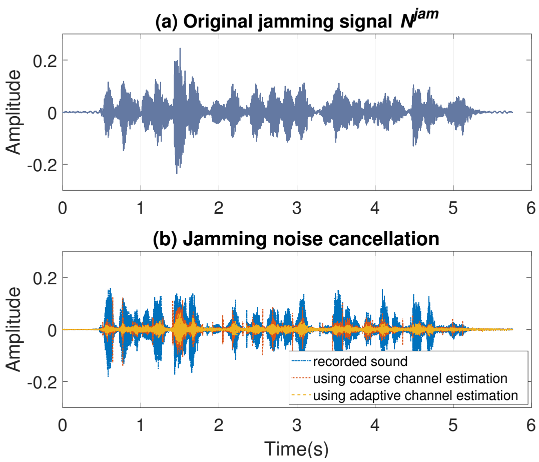

Fig. 8. Jamming noise self-cancellation using adaptive channel estimation and coarse channel estimation, respectively.

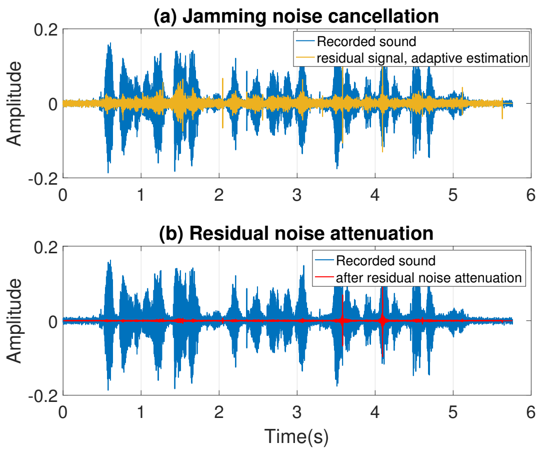

Fig. 9. Residual noise can be effectively attenuated through the spectrum subtraction algorithm.

the negative gradient direction iteratively util convergence is gradually reached.

More specifically, we first take FFT of $\hat{h}_{s,m}(0)$, i.e., resulting $\hat{H}_{s,m}(0)$, as the initial channel frequency response. We then take partial derivative about $\hat{H}_{s,m}(i)$ on the cost function and we can get the update formula of the filter represented as

$$\hat{H}_{s,m}(i+1) = \hat{H}_{s,m}(i) + \mu N^{jam}(i)^* E(i), \tag{4}$$

where μ is the step size in gradient descent. To optimize the convergence rate and obtain a more stable performance, we normalize the step size of each frequency bin according to the signal power, i.e.,

$$\hat{H}_{s,m}(i+1) = \hat{H}_{s,m}(i) + \mu N^{jam}(i)^* E(i)/P(i), \tag{5}$$

where $P(i) = [P_0(i), ..., P_{Q-1}(i)]$ and for a forgetting factor λ and $k \in [0, Q-1]$, $P_k(i) = \lambda P_k(i-1) + (1-\lambda)\left|{N^{jam}}_k(i)\right|^2$.

Consequently, the received jamming noise contained in a voice segment in $y(t)$ can be estimated using the most up-to-date channel frequency response estimate and removed from $y(t)$. Figure 8 shows an example of using coarse channel response estimation and adaptive channel response estimation to conduct self-cancellation on a jamming noise. It can be seen that the main energy of the recorded jamming noise is eliminated and using adaptive channel estimation can achieve 5dB more attenuation than using coarse channel estimation.

C. Residual Noise Attenuation

In real environment, ambient noise makes adaptive channel estimation hard to converge, which decreases the performance of adaptive jamming noise cancellation. As a result, there can still be some residual noise after cancellation.

We adopt the spectrum subtraction algorithm [19] to further mitigate residual noise. Specifically, for non-voice segments, the residual signal obtained after adaptive jamming noise cancellation has a similar energy spectrum distribution. We sample the spectrum of such residual signal and learn the noise threshold on each frequency component. Then, for user voice segments, we compare the spectrum of $E(i)$ with those thresholds. If the amplitude of certain frequency is lower than the corresponding threshold, this frequency component is regarded as noise and would be attenuated in proportion. Figure 9 shows the residual noise after adaptive jamming noise cancellation in Figure 8 is largely eliminated after this process. The overall attenuation of noise reaches 30dB.

VII. Performance Evaluation

A. Methodology

Devices. We implement SeVI as an app on a google Pixel 3 XL smartphone, which runs Android 9 Pie and has 4GB memory and a Qualcomm snapdragon 845 processor. We use the phone as the device and jamming noises are played via the main speaker of the phone.

Users. In order to ensure the repeatability of experiments and obtain accurate user voice signals for comparison, we use another Pixel 3 XL to play the role of users. Specifically, we recruit two male and two female volunteers, let each volunteer read one hundred randomly selected sentences, record with the phone, and get four speech data sets, denoted as $U1$, $U2$, $U3$, and $U4$, respectively. We then use the first fifty speech records in each speed data set to construct a speech record library for each user and use the rest speech records for testing. We use a laptop as the helper to learn the device response and compensate the device distortion for all speech record library.

Attackers. We recruit another twelve volunteers as on-site attackers, five females and seven males, aged from 21 to 43, including four undergraduate students, five graduate students, and three faculty members. In addition, we use two different devices, i.e., a ASUS T305C tablet and a Pixel 3 smartphone as stereo eavesdropping equipments. The tablet and the phone have dual stereo microphones separated at a distance of 5cm and 15cm, respectively. Attackers can conduct

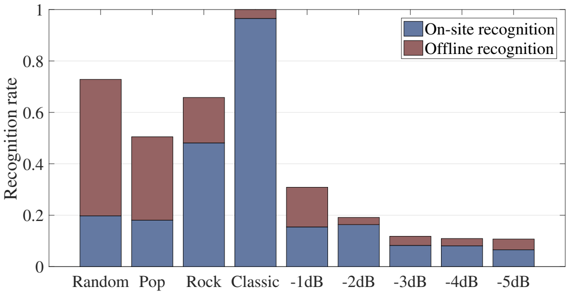

Fig. 10. Recognition rate in different jamming noise conditions.

on-site eavesdropping attacks and offline DSP attacks with stereo records. Particularly, we consider two offline blind source separation (BSS) algorithms, i.e., FastICA [20], an independent components analysis (ICA) based algorithm, and DUET [21], a binary time-frequency masking algorithm.

We consider the following three metrics to evaluate the performance of SeVI:

- **Recognition rate:** refers to the ratio of the number of words correctly recognized by human hearing to the total number of words in a sentence.
- **MFCC similarity:** is the similarity between the Mel-scale Frequency Cepstral Coefficient (MFCC) [22] of acoustic signals. MFCC is a feature representation method based on human auditory characteristics.
- **Short-Time Objective Intelligibility (STOI):** is a metric used to measure the intelligibility of speech signals. STOI algorithm [23] takes the original speech signal and the processed signal as input, and will give a value in the range from 0 to 1. A high STOI value means high intelligibility of processed signal.

B. Effect of Different Types of Jamming Noises

We first investigate the effectiveness of different types of jamming noises. We conduct the experiment in a meeting room about 36 square meters, where attackers are 1m away from the user and the distance between the device and the user is 20cm. We generate jamming noises using user speech records and vary the signal-to-noise ratio (SNR) from -1dB to -5dB with an interval of 1dB. Besides user speech records, we consider to generate jamming noise using random noise and three categories of music. For each attacker and each type of jamming noise, we randomly select ten sentences from the testing set of each user, and ask the attacker to recognize the user voice in the presence of jamming noise on site. In addition, attackers can listen to recorded contents as many times as they want.

Figure 10 plots the average recognition rate over all attackers for each jamming noise type. It can be seen that different types of jamming noise have distinct jamming effects and using user voice records to generate jamming noise can achieve superior jamming effects. Decreasing SNR will decrease the recognition rate but it would also bring difficulty for noise self-cancellation. It can be seen that SNR of -3dB, i.e., the volume of user voice is one half of the volume of jamming noise, is a good tradeoff.

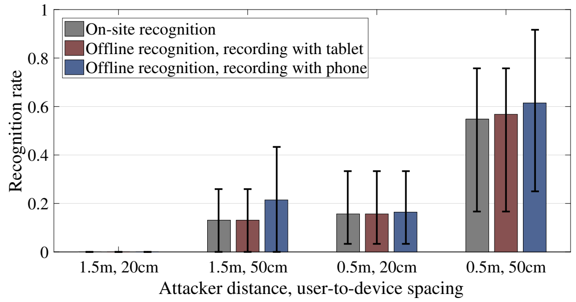

Fig. 11. Recognition rate in two attack distances and two user-to-device spacing configurations.

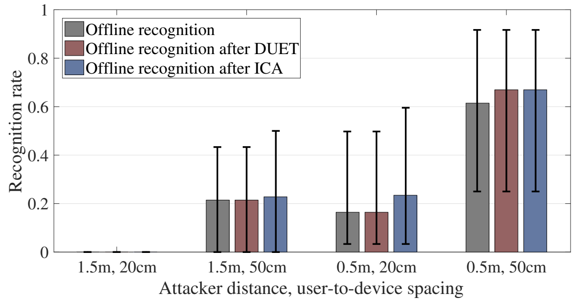

Fig. 12. Recognition rate after conducting BSS algorithms on recorded signal at attackers.

C. Impact of Attacker Positions

As an attacker deliberately approaches a victim, the probability for the attacker to perceive the content of the user's voice is increased. Similarly, as the separated distance between the user and the device increases, such probability also increases. We take the same setting as in the above experiment except that we use two attack distances, i.e., 0.5m and 1.5m, and two spacing distance between the user and the device two audio sources, i.e., 20cm and 50cm.

Figure 11 plots average recognition rate in different attack distances with two user-to-device spacing configurations. It can be seen that when the spacing between the user and device is 20cm and attack distance is beyond 1.5m, the recognition rate drops to zero. Normally, 20cm is a comfortable spacing when people talk to a phone and it is easy to keep a distance of 1.5m away from other people.

D. Impact of DSP Attacks

In this experiment, we take the same setting as in the above experiment. Especially, we conduct the FastICA and DUET algorithms to separate the jamming noise and user voice and then ask attacker to recognize the output of both algorithms.

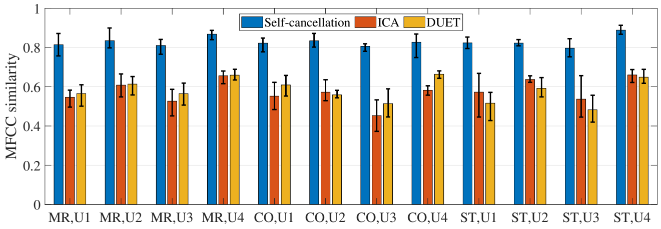

Fig. 13. MFCC similarity comparison.

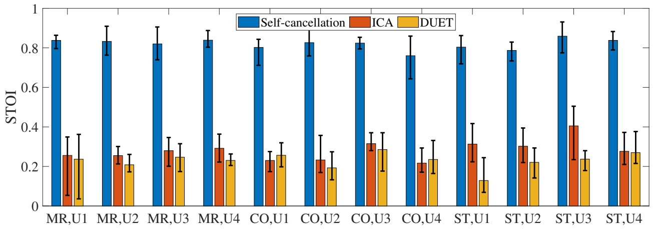

Fig. 14. STOI comparison.

Since the resolution of the phone recording is better, only the experimental results of phone recording are shown here.

Figure 12 plots the average recognition rate after conducting two BSS algorithms on recorded signal at attacker. Comparing with Figure 11, it can be seen that BSS algorithms has little effect in improving the capability of attackers. For FastICA algorithm, it can achieve a good separation effect for linearly combined and independent signals.However, in real environments, because of the spatial frequency response generated by reverberation environment, the signals of each source are no longer simply linear superposition, but convolved with the frequency response of the environment. DUET algorithm also fails in separating signals effectively. According to the principle of the algorithm, there may be two reasons. One is that in our scenario, the spatial positions of two signal sources are very close, resulting in very similar channel conditions between them. Second, and more importantly, different from sparse normal speech, in order to ensure the effectiveness of the masking, the jamming noise we played is dense and has similar frequency distribution. However, the effectiveness of the DUET algorithm is based on signal sparsity which means at any time and any frequency, there is one signal, only in this way can we get the correct information of the difference between the amplitude and phase of two channels corresponding to a certain signal.

E. Effectiveness of Jamming Noise Self-cancellation

In this experiment, the attack distance and the user-to-device spacing are set to 1.5m and 20cm, respectively. We consider three common voice interaction scenarios, corresponding to different noise levels and channel complexity, including a meeting room (MT), in a corridor (CO) and on a public street (ST). The meeting room has the lowest noise level but the most complex multipath environment. The noise level in the corridor is moderate and the echo environment is also complex. The street has the loudest ambient noise but a better echo condition than the former two.

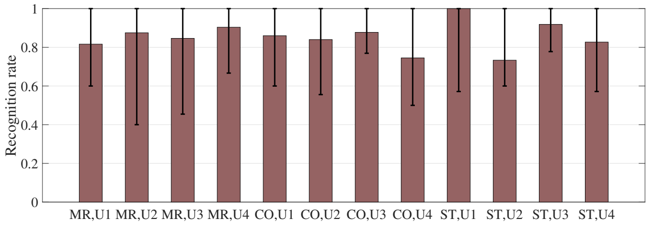

Fig. 15. Recognition rate by Google Assistant on Pixel 3 using self-cancellation results.

To verify the performance of the self-cancellation algorithm, we compare the MFCC similarity and STOI between the original voice records and the derived results after self-cancellation. We also compare MFCC similarity and STOI between the original voice records and the results of FastICA and DUET.

Figure 13 plots the MFCC similarity results. As can be seen from the figure, the MFCC similarity between our self-elimination algorithm's results and original signals is generally above 0.8, which is on average 0.2 higher than the results of FastICA and DUET. The reason why the results of the attack algorithm also have relatively high similarity is that our jamming noise is composed of the voices of the same person, and the characteristics of the same person's speech signals are very similar to each other. Figure 14 plots the STOI results. It can be seen from the figure that the intelligibility of self-cancellation results is generally between 0.8 and 0.9, while the average intelligibility of FastICA and DUET results is below 0.3.

We further play the results to Google Assistant on a Pixel 3 smartphone. The recognition rate by Google Assistant using self-cancellation results is shown in Figure 15. It can be seen that the average recognition rate is around 85%. While for FastICA and DUET results, the software cannot recognize the content at all. Moreover, we let all attackers to listen to self-cancellation results. All testing samples can be fully recognized in all three scenarios.

F. Response Time of SeVI

The device has a 2.8GHz 8-core CPU. We measure the running time of each major component of SeVI on one CPU core. We run jamming noise self-cancellation with 500 ten-second speech records sampled at 16kHz. On average, the running time of user voice detection, adaptive jamming noise cancellation, and residual noise attenuation is 0.12s, 0.06s, and 0.05s. The response time is moderate for most voice interaction applications. It is possible to consider optimization technique to further reduce the response time in our future work.

VIII. Conclusion

In this paper, we have proposed a voice interaction protection scheme, called SeVI, for smart devices. SeVI innovatively use the users own speech voice to generate jamming noises and can effectively conduct self-cancellation under time-varying channels. The advantage of SeVI is that it can be realized on COTS devices as a software component. We have implemented SeVI on a Pixel 3 XL smartphone. Our experience illustrates that SeVI is light-weight and easy to implement and use. We have conducted extensive real-world experiments and the results demonstrate that SeVI can provide superior protection for mobile voice interactions against online eavesdropping attacks and offline DSP attacks.

SeVI also has some limitations. First, jamming noise can pollute environment, especially for silent indoor scenarios. Second, since the spectrum of residue noise is also similar to that of the real interaction voice, when eliminating residual noises according to its spectrum sample, part of the spectrum of real speech is also been attenuated.

Acknowledgements

This research was supported in part by National Natural Science Foundation of China (Grants No. 61772340, 61672151, 61972081), Shanghai Rising-Star Program (Grant No.17QA1400100), and DHU Distinguished Young Professor Program.

References

[1] iFLYTEK, https://www.iflytek.com/.

[2] Google Assistant, https://assistant.google.com/.

[3] Siri, https://www.apple.com/siri/.

[4] Cortana, https://www.microsoft.com/en-us/cortana.

[5] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proceedings of ACM MobiSys*, 2017, pp. 2–14.

[6] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible Voice Commands," in *Proceedings of ACM SIGSAC*, 2017, pp. 103–117.

[7] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: Secure Peer-to-peer Acoustic NFC," in *Proceedings of the ACM SIGCOMM*, 2013, pp. 63–74.

[13] D. A. Pados and G. N. Karystinos, "An Iterative Algorithm for the Computation of the MVDR Filter," *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 290–300, Feb 2001.

[8] Z. Man, W. Qian, K. Ren, D. Koutsonikolas, and Y. Chen, "Dolphin: Real-time hidden acoustic signal capture with smartphones," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2018.

[9] A. Chaman, Y. Liu, J. Casebeer, and I. Dokmanić, "Multipath-enabled Private Audio with Noise," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 685–689.

[10] M. T. Islam, B. Islam, and S. Nirjon, "SoundSifter: Mitigating Overhearing of Continuous Listening Devices," in *Proceedings of ACM MobiSys*, 2017, pp. 29–41.

[11] P. Wang, Y. Wang, H. Liu, Y. Sheng, X. Wang, and Z. Wei, "Speech enhancement based on auditory masking properties and log-spectral distance," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, Oct 2013, pp. 1060–1064.

[12] S. A. Gelfand, *Hearing: An introduction to psychological and physiological acoustics*. CRC Press, 2017.

[14] B. R. Breed and J. Strauss, "A Short Proof of the Equivalence of LCMV and GSC Beamforming," *IEEE Signal Processing Letters*, vol. 9, no. 6, pp. 168–169, 2002.

[15] W. Liu and S. Weiss, *Wideband beamforming: concepts and techniques*. John Wiley & Sons, 2010, vol. 17.

[16] T.-A. Vu, H. Ding, and M. Bouchard, "A survey of double-talk detection schemes for echo cancellation applications," *Canadian Acoustics*, vol. 32, no. 3, pp. 144–145, 2004.

[17] J. Benesty, D. R. Morgan, and J. H. Cho, "A new class of doubletalk detectors based on cross-correlation," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 2, pp. 168–172, March 2000.

[18] S. Zhao, "Performance analysis and enhancements of adaptive algorithms and their applications," *PhD, School of computer engineering, Nanyang technological university*, 2009.

[19] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[20] H. Sawada, S. Araki, and S. Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 516–527, March 2011.

[21] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, July 2004.

[22] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *CoRR*, vol. abs/1003.4083, 2010. [Online]. Available: http://arxiv.org/abs/1003.4083

[23] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 4214–4217.