Voice Liveness Detection for Voice Assistants through Ear Canal Pressure Monitoring

Jiacheng Shang, *Member, IEEE,* and Jie Wu, *Fellow*

Abstract—The voice assistants are important input devices in future smart homes. Thanks to the great performance that is provided by current voice recognition systems, current voice assistants can understand various commands in different language from users and connect with other devices in the same local network to perform corresponding actions. However, the voices are not secure due to its nature. Even if we secure voice assistant using the voiceprint, attackers can still steal the victim's voices and replay them to voice assistants for attacking purpose. In this paper, we propose a new voice liveness detection system that is specifically designed for voice assistants. The key insight behind our system is that users will open their mouth when they say some phonemes. Such opening mouth activities will impact the air pressure in the ear canal if the ear canal is an enclosed space. Therefore, we can detect the liveness of the voices on the side of voice assistants by cauterizing the correlations between each sentence and the air pressure. Experiments with ten volunteers show that our system can accurately accept voice commands from legitimate users with accuracy of 94.8% and 97%. Moreover, our system can effectively defend current voice assistant devices from replay attacks with accuracy of 99.25% and 99.5%.

Index Terms—Voice replay attack, liveness detection, ear canal pressure.

1 Introduction

The interactions and communications between users and smart devices is one of the most important issues in future smart homes. Among all interaction methods, the voices provide a natural way for users to control their devices without extra devices (such as remote controllers) and body involvement (gesture-based control). Therefore, a new technology called voice assistant is proposed to translate the voices of users to commands or actions that are going to be performed by one or more smart devices. Thanks to the great performance that is provided by current voice recognition systems, current voice assistants can understand various commands in different languages from users and connect with other devices in the same local network to perform corresponding actions.

However, the voices are not secure due to their nature. Firstly, most current voice assistants do not perform identity validation on the received voice signals. Therefore, any voice command that is produced in the same physical environment can be picked up and executed, which presents a serious threat to the security of smart homes. For example, attackers can hack any device that has a speaker to send malicious voice commands to the voice assistant. Secondly, even if we secure voice assistants using the voiceprint, attackers can still easily get the voices of the victim since voices are open to the public [5], [9], [19], [27]. Such recorded voices can be leveraged by forgery techniques [10] to bypass voiceprint check. Once attackers can break voice authentication on voice assistants, they usually can obtain more sensitive information of the victim, which would result in severe consequences harmful to the victim's safety, reputation, and property.

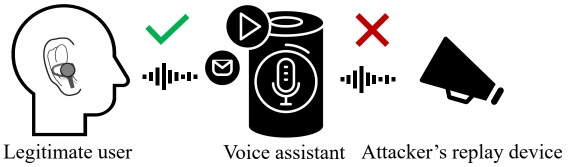


Fig. 1. The idea voice assistant should accept voice commands from legitimate users while rejecting any replayed voices from attackers.

Since it is hard for one person to perfectly impersonate the voices of another one, most attacks on voice assistants are launched by replaying malicious voice commands to voice assistants. To secure the voice assistants from such replay attacks, besides implementing voice authentication, it is also essential to determine whether the received voice is produced by human beings or replayed by replay devices. If a voice is detected from a loudspeaker, it is very likely that that voice command is from attackers. In the past few years, many researchers proposed different systems to detect the liveness of voices [3], [15], [17]. Their insight is to leverage the differences between the human vocal system and the loudspeaker. For example, Chen et al. proposed a liveness detection system that can recognize replayed voices by detecting the magnetic field of the loudspeaker. However, most of the current liveness detection systems have a short operating range since they are designed for headsets or smartphones. In the use scenario of voice assistants, the user can be a few meters away from the voice assistant, which makes most of the current liveness detection methods not applicable. To address this issue, researchers study to build new liveness detection systems for voice assistant devices with the help of extra sensors and wireless signals. [4],

- *J.Shang was with the Department of Computer Science, Montclair State University, Montclair, NJ, 07043.*
- *J. Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, 19122.*

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1618398, CNS 1651947, and CNS 1564128.

[6], [7], [25]. For example, Meng et al. proposed a system called WiVo that can detect the replayed voice by characterizing the correlation between wireless signal dynamics and mouth activities. However, WiVo requires the user's face is close enough to the antennas of the wireless receiver, which is hard to ensure in practice. Moreover, indoor wireless signal dynamics can be easily influenced by other activities in the same physical environment, which also degrades the robustness of the system.

In this paper, we propose a new voice liveness detection system that is specifically designed for voice assistants. As shown in Fig. 1, our system is designed to accept all voice commands from normal users and reject all replayed voices from attackers. The key insight behind our system is that users will open their mouths when they say some phonemes. Such opening mouth activities will impact the air pressure in the ear canal if the ear canal is an enclosed space. By embedded a tiny pressure sensor into the earbuds, we can monitor the air pressure change while the user is saying. Then, we can detect the liveness of the voices on the side of voice assistants by cauterizing the correlations between each sentence and the air pressure. If we can detect the corresponding air pressure change when the user is talking, the user is detected as a normal user. Otherwise, the voice command will be regarded as from attackers and dropped without execution.

We address three major challenges to address this goal. The first challenge is how to get the uniformly sampled air pressure data from the sensor. To address this issue, we leverage signal processing techniques to resample the raw sensor signal while still reserving useful information for detecting opening mouths. Secondly, the resampled air pressure contains much noise, which makes it hard to extract proper features directly. To solve this problem, we leverage the discrete wavelet transform-based denoising method to remove high-frequency noise and extract the significant fluctuations that are introduced by opening the mouth through calculating the short-time variance. Furthermore, we compute the power spectral density of filtered variance signal as the feature for liveness detection. Thirdly, to build a robust and accurate classifier, we propose two methods with different performance levels and resource requirements. The lightweight method can provide acceptable performance for both accepting normal users and rejecting attackers with limited training effort. The advanced method can further improve the system performance by including more features and using a neural network for classification.

Our contributions in this paper are summarized as follows:

- Our results serve as a feasibility assessment to show that air pressure changes in the ear canal can be used to detect mouth opening activity, which can be further used to validate the liveness of the voice source.
- We propose solutions to detect mouth opening activities from noisy air pressure data. We also extract useful information from the pressure data and propose two different classification methods with different computation complexity and performance levels to further enhance the detection.

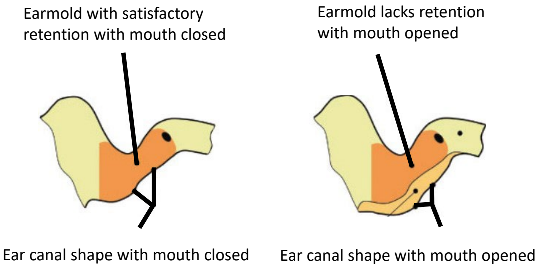

Earmold with satisfactory retention with mouth closed
Earmold lacks retention with mouth opened
Ear canal shape with mouth closed
Ear canal shape with mouth opened

Fig. 2. Changes in ear canal when the mouth is open and closed.

- We develop a prototype and conduct comprehensive evaluations.Experiments with ten volunteers show that the two methods of our system can accurately accept voice commands from legitimate users with accuracy of 94.8% and 97%. Moreover, our system can effectively defend current voice assistant devices from replay attacks with accuracy of 99.25% and 99.5%.

The remainder of this paper expands on these contributions. We first introduce voice assistant devices and related exiting attack and defense systems in Section 2. Then, we discuss background knowledge, the attack model, and feasibility experiments in Section 3. The details of our methods and solutions are presented in Sections 4 and 5, respectively. To evaluate the effectiveness and robustness of our system, we conduct various experiments in Section 6. Discussion is presented in Section 7.

2 Related work

2.1 Voice assistants in smart homes

In smart home environments, a voice assistant refers to a group of devices that can convert users' voices to text, predict users' needs, and perform corresponding actions together with other smart devices in the environment [11]. To achieve this goal, these devices are built on voice recognition, neural language processing, and speech synthesis technologies. In the past few years, many voice assistant devices have been designed and released. For example, Apple announced its HomePod in June 2017, and Amazon has also released its voice assistant AI technology called Amazon Alexa. Based on a recent report by voicebot.ai, more than 3 billion voice assistants were in use in 2019 [13]. Therefore, the security of voice assistants is very important.

2.2 Attacks on voice service

The voice service on voice assistants can be divided into two major categories: voice recognition and speaker verification. Voice recognition focuses on translating voice into text, and speaker verification focuses on validating the identity of the voice. However, both the voice recognition [20], [22], [28], [29] and speaker verification [9], [23] suffer from attacks. In terms of the attacks on voice recognition systems, [29] showed that it is feasible to replay malicious voice commands to the device of the victim in an inaudible channel. In terms of the attacks on speaker verification systems, a recent work shows that an attacker can break voice recognition systems by concatenating speech samples from multiple short voice segments of the victim [23]. To defend against

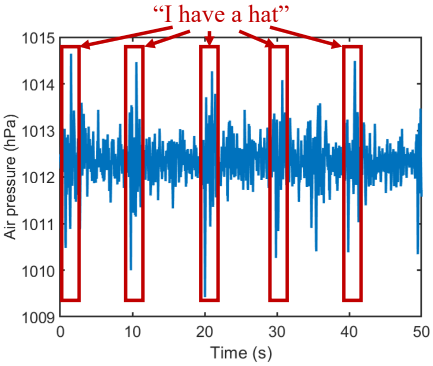


Fig. 3. Results of feasibility experiments.

these attacks, researchers have proposed various countermeasures by studying the differences between human vocal systems and loudspeakers [1], [3], [8], [15], [16], [21], [24], [26]. However, existing defense systems are all designed for smartphones and AR headsets. The significantly different usage scenarios make current defense systems hard to be implemented on voice assistants. For example, the liveness detection system proposed in [15] rejects replayed voice by measuring the relationship between mouth voice and throat voice. Apparently, this work cannot be implemented on voice assistants since voice assistants are usually far away from the user in the room.

3 Preliminary

3.1 Air pressure in ear canal

When users do not wear earphones, the air of the open ear canal is in direct contact with the atmosphere outside the body, which means the air pressure is the same as that in the environment. However, when users wear in-ear headphones, the ear canal becomes an enclosed space, so that the air pressure is largely influenced by the size of the enclosed space rather than environmental noise. Recently, research has shown that human facial activities can change the size of the enclosed space of the ear canal [2], which further introduces changes to the air pressure in it. As shown in Fig. 2, when the mouth is closed during non-speech periods, the earmold is with satisfactory retention. When the user opens the mouth, the positional relationship between the ear canal and the mandibular condyle changes correspondingly, which makes the earmold lack retention. As a result, the shape of the ear canal becomes bigger. Since the ear canal is a enclosed space when user wears the in-ear earphone, the air pressure in the ear canal also changes.

3.2 Attack model

In the attack model we consider, attackers aim to issue malicious voice commands to the voice assistant that is in the victim's smart home environment. This type of attack can be launched either remotely or in the same smart home environment. For example, the attacker can say a malicious command to the voice assistant in the same environment as the victim. Also, by using recent attack techniques, the attacker can issue these commands without the attention of the victim. However, the ability of attackers is also limited to some senses. Since earphones are private devices and always on the victim's ears, we assume that the attacker cannot get access to the victim's earphones during the procedure of attacks. This fact means that the attacker cannot forge the received air pressure signal.

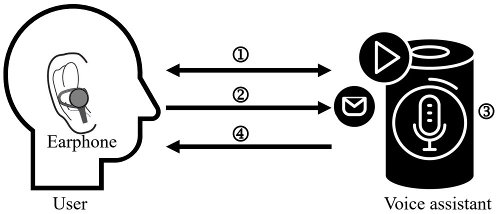


Fig. 4. Usage scenarios of our system.

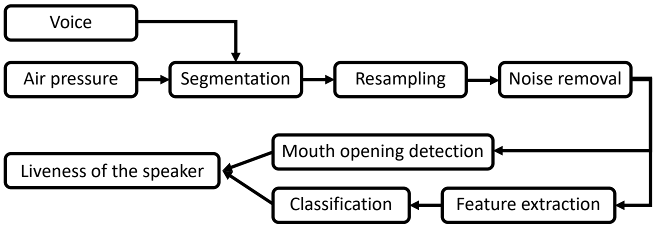


Fig. 5. System architecture.

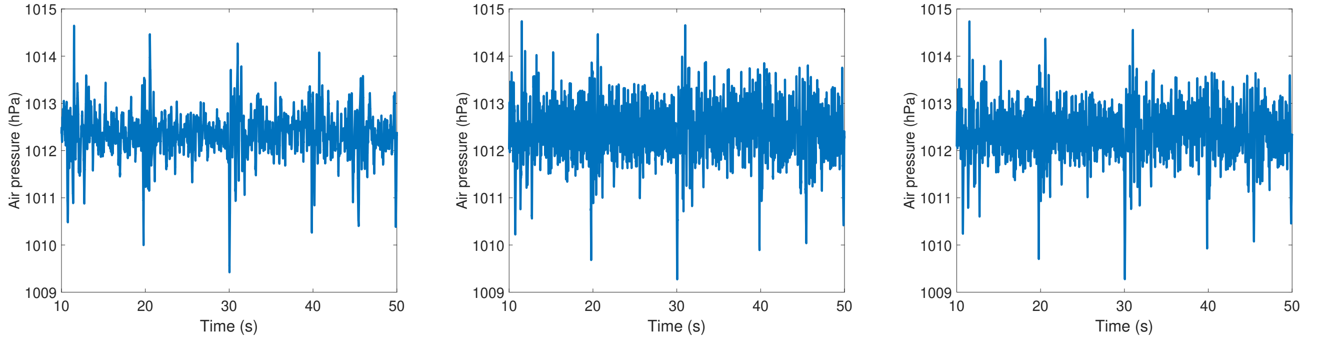
3.3 Feasibility study

Although we obtain some insights in the preliminary study, it is still not clear how sensitive the air pressure in the ear canal is to the mouth movements during the speech. Therefore, we designed a preliminary experiment to evaluate the feasibility of our idea. We built a prototype to collect the ear canal pressure with a sampling rate of about 500 Hz and record the voice at the same time. Then, we asked a user to say a short sentence, "I have a hat", every 10 seconds while using the prototype. Fig. 3 shows the collected air pressure signals. We can observe that the mouth movements during the speech generate more significant variances to the pressure signal compared with environmental noises. Moreover, for some phonemes that require users to largely open their mouths, the variances are much more significant. For example, the phoneme "e" in the word "hat" introduces the highest peaks to the air pressure signal. These facts show that the mouth movements during speeches do generate enough variances to the air pressure signal. Therefore, by monitoring whether there exist well-synchronized variances in the pressure signal, our system can determine whether the voice is from a human.

4 System Design

4.1 Usage scenarios

The objective of our system is to protect the current voice assistant devices from voice replay attacks. Fig. 4 shows the basic usage scenario of our system. In the usage scenario, we consider two major components, the user and the voice assistant. We assume that the voice assistant can exchange information with the earphones using wireless communication (e.g. WiFi and Bluetooth). The interactions between the user and the voice assistant can be divided into four steps. First, the earphones and the voice assistant device will exchange packets for several rounds so that these two



(a) Raw air pressure signal.

(b) Air pressure signal after resampling.

(c) Air pressure signal after resampling and low-pass filter.

Fig. 6. Preprocessing of the air pressure signal.

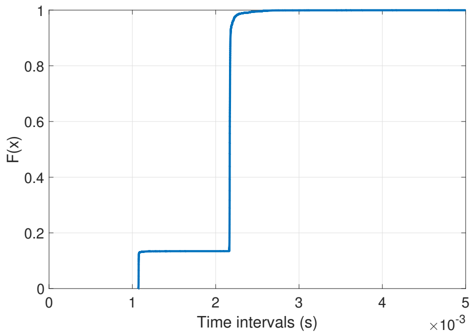


Fig. 7. The distribution of measurement intervals.

devices are using the same clock. In the second step, the user will say a voice command to the voice assistant. The voice assistant picks up the voice for further voice-to-text analysis. After the voice assistant receives the voice, it will send a message to the earphone for requesting the air pressure data. The earphones receive the message and stream the collected the air pressure data to the voice assistant for processing. In the third step, the voice assistant processes both the voice and the air pressure data either locally or remotely. Finally, the voice assistant sends a corresponding response to the user through the audio channel. If the voice and the air pressure data pass the liveness detection, the voice assistant will give the user a confirmation message of the voice command. Otherwise, the voice assistant will alert the user for a potential voice replay attack. If the voice is indeed from the user, the user can still force the assistant to follow the command using an associated smartphone.

4.2 Challenges

Although we obtain insights from preliminary experiments, it is still challenging to build such a liveness detection system. First, the sampling rate of the sensor may not be consistent during the process of data collection. Although we can write a script to read the data from the sensor, it is not always true that newly read sensor data is fresh. To address this challenge, we leverage fitting algorithms to estimate those values that are not reported by the sensor in real-time. Second, it is challenging to extract pressure signal that is under impacts of mouth movements from noisy pressure signals. As we can observe, various noises exist in the raw pressure signal. If we directly detect the movement from the raw signal, the false detection rate can be very high. To solve this problem, we leverage a series of signal processing techniques based on the features of signals in the frequency domain. Finally, it is also hard to match the air pressure signal with the voice signal in order to predict the liveness of the voice command.

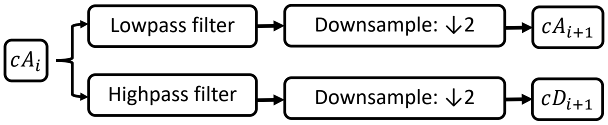


Fig. 8. Results of feasibility experiments.

4.3 System architecture

Fig. 5 shows the architecture of our system. After receiving the voice commands and air pressure signal from the user, the voice assistant first performs audio processing on the voice signal to get the starting time and ending time of the voice commands. The extract timestamps are further used to segment the air pressure signal. After that, our system resamples the air pressure signal to make sure the signal is uniformly sampled. The resampled signal is filtered by Discrete Wavelet Transform-based techniques to remove the high-frequency noise. Since mouth opening activities generate a much greater impact on the air pressure signal, we calculate the short-term variance of the filtered signal. A mouth opening activity is detected by finding whether a qualified peak exists in the short-term variance signal. To further reduce the influence of low-frequency noise, we leverage an extra classification model to enhance the system performance for both accepting legitimate user and rejecting attackers. We extract three features from the variance signal and send them to a MART-based binary classifier. A voice command is regarded from a live speaker (or legitimate user) only if the incoming signals pass both checks.

5 Solution

5.1 Preprocessing

5.1.1 Signal segmentation

To validate the liveness of the voice's source, we need to get the segments of pressure signals that are influenced by the speeches. Since we assume that the earphones are well synchronized with the voice assistant via wireless

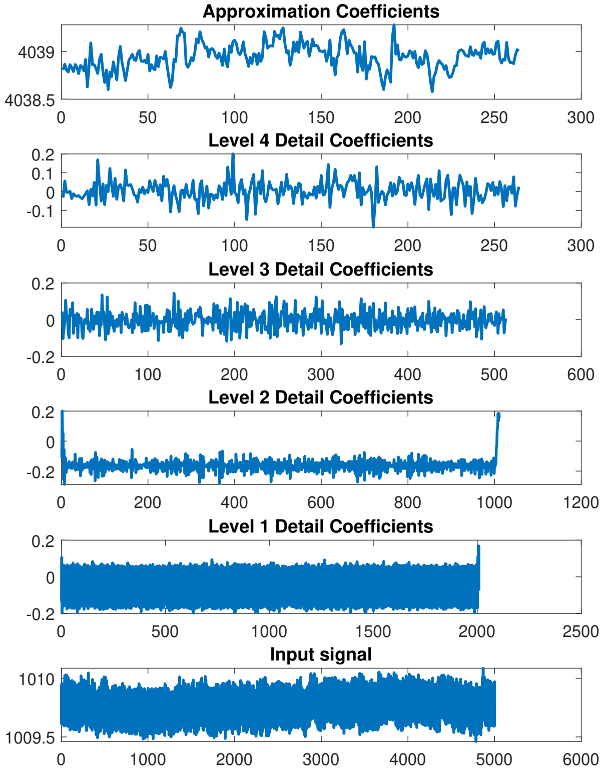


Fig. 9. DWT-based noise removal.

communication, we can accurately find the starting and ending points of each speech behavior in pressure signals by analyzing the voice signals. Therefore, we first segment the voice signals into different sentences by performing Hidden Markov Model (HMM) based word segmentation techniques [12]. Then, we use the obtained timestamps to segment air pressure signals for further analysis.

5.1.2 Resampling

However, raw air pressure signals cannot be directly used for analysis. First, although we use a fixed sampling rate by setting the control bits on the sensor hardware, the sensor may not report the sensor data uniformly. As shown in Fig. 7, the time interval between two neighboring samples can be either value, which introduces much difficulty to the signal processing procedure. To solve this problem, we first filter the raw signal using a finite impulse response (FIR) filter. The FIR filter is designed to minimize the weighted integrated squared error between an ideal piecewise linear function and the magnitude response of the filter over a set of desired frequency bands. We normalize the result to account for the processing gain of the window and then change the sampling rate using a polyphase interpolation structure. Figs. 6(a) and 6(b) show the raw and resampled pressure signals, respectively. We can see that the important information is reserved after resampling the signals. Fig. 6(c) shows the distributions of time intervals between two neighboring samples before and after resampling. We can see that the time interval can be either 0.0012 seconds or 0.002 seconds before resampling. By resampling the data, we make sure the signal is uniformly sampled with a frequency of 500 Hz.

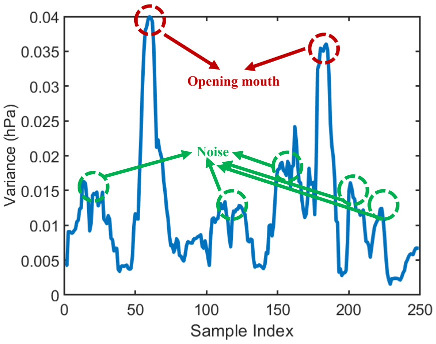


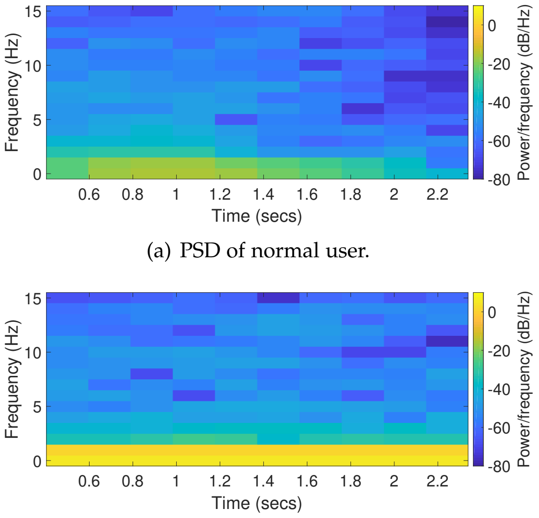
Fig. 10. Filtered variance signal.

5.1.3 Noise removal

Although we get a uniformly sampled pressure signal, it is still hard to detect mouth opening activity from the signal in Fig. 6(b). The main reason is because the pressure values are impacted by many other factors besides mouth opening activities. For example, imperfect hardware manufacture may cause small variances in pressure readings. In addition, environmental changes may also influence the air pressure in the ear canal. Therefore, we need to remove these noises in order to extract useful information for accurate detection. In our system, we leverage one-dimensional discrete wavelet decomposition-based denoising techniques. Specifically, a one-dimensional discrete wavelet transform (DWT) consists of multiple levels. The procedure in each level is shown in Fig. 8. The signal cA_i from the upper level will be filtered by a lowpass filter and a highpass filter, respectively. The filtered signal is then downsampled, which produces the two outputs cA_{i+1} and cD_{i+1}. The resulting signal cA_{i+1} reserves low-frequency features, while cD_{i+1} reserves most high-frequency features. After that, cA_{i+1} will be passed to the next level for further decomposition. In our system, we leverage a four-level DWT and let the resampled signals to be the input cA_0 of the first level. We asked a user to say two voice commands and Fig. 9 shows the calculated signals from the very first level to the last level. We can observe that most high-frequency noise in the input signal can be effectively removed in the four-level processing. Moreover, only the approximation coefficients that correspond to cA_4 have much higher variances in verbal periods than those in non-verbal periods. We further leverage the calculated approximation coefficients cA_4 at the fourth level as the features to detect mouth opening activities.

5.2 Liveness detection

After obtaining and denoising the variance signal, we need to extract proper features to detect the liveness of received voice commands. In our previous work [18], we extract features from the variance signals on the time domain and leverage a Multiple Additive Regression Tree (MART)-based classification model for detection. However, our previous method does not utilize the information on the frequency, which can impact the performance of the system on a



(a) PSD of normal user.

(b) PSD of attacker.

Fig. 11. Power spectral density (PSD) of normal user and attacker.

larger dataset. For example, some high-frequency (above 3 Hz) components can still be there even after denoising. Such high-frequency components may also introduce significant fluctuations to the time-domain signal. Therefore, we propose two new detection methods that can leverage the information on both time and frequency domains in this paper. Moreover, these two methods are with different overhead and performance levels, a lightweight method with acceptable performance and a complex method with higher performance. The user or system administrator can choose according to their requirements and available resources.

5.2.1 *Lightweight liveness detection*

First, we focus on proposing a detection method that can be executed on most voice assistants and still provide acceptable performance. There are two major challenges to achieve this goal. The first challenge is how to extract proper features from both time and frequency domains. To understand this, we first perform the Short-time Fourier transform (STFT) on the filtered variance signal. Then, we calculate the power spectral density based on the STFT results. Assuming $S(i,j)$ is the STFT result of the i^{th} time frame and the j^{th} frequency frame, the power spectral density $P(i,j)$ of the signal is

$$P(i,j) = \frac{2|S(i,j)|^2}{fs\sum_{n=1}^{L}|w(n)|^2}, \tag{1}$$

where fs is the sampling rate, L is the signal length, and $w(n)$ is window function. In our implementation, we use Hamming window as window function for calculating power spectral density. We further convert the power measurement in decibels (dB) for the following analysis. Fig. 11 shows the power densities of the signals collected from replay attackers and normal users. We can observe that opening mouth impacts the variance signal mostly at a low frequency (under 3 Hz). For the power spectral density of the attacker, the power spectral density is nearly consistent over time. Based on this insight, we only calculate the variance of power spectral density from the following three frequency bands: 0 Hz - 1 Hz, 1 Hz - 2 Hz, and 2 Hz - 3 Hz, respectively.

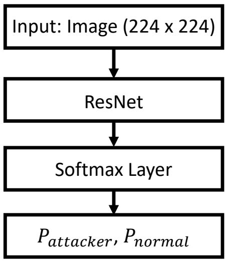


Fig. 12. System performance of using others' trained model.

After obtaining the proper features, the second challenge is how to well train a classifier with good parameters. In our previous method, we train the classifier with fixed parameters, which means the decision boundary largely depends on the training dataset. To address this issue, we train a classifier based on the support vector machine and leverage Bayesian Optimization and three-fold cross validation for finding the best training parameters within a specific time. Compared with our previous method, the proposed method can train a more robust classifier to ensure better performance on a different dataset.

5.2.2 *Transfer learning-enabled advanced liveness detection*

In the light-overhead method, we leverage the low-frequency information in the spectrogram to detect the liveness of the voice signal. However, this will ignore all the information above 3 Hz. Based on our preliminary experiments, opening mouth can also impact the variance signal at a frequency between 3 Hz and 8 Hz for some users. To include this part of information and deliver better detection performance, we propose to leverage the spectrogram under 8 Hz. Moreover, instead of only using the variance of power spectral density as features, we use all entries in the power spectral density matrix as features to get the detection results. In our system, we resize the power spectral density matrix to 224 × 224 and take the matrix as an image to fully leverage each entry. Since the input is changed to images, we leverage the deep learning models that are proven to have great performance. However, it is nearly impossible to train a deep learning model from scratch since the number of parameters that need to be trained is much larger than the data we have. In practice, it is unrealistic to collect a huge amount of data from a new user by asking the user to say a lot of sentences to the voice assistant. To address this issue, we leverage the idea of transfer learning, which is a machine learning method where a model developed for one task is reused for a new task to reduce training costs. In our system, we use a deep learning model called ResNet that is already trained on a large dataset. Fig. 12 shows the structure of our classification model. We add a new final layer that has only two outputs. To reduce the number of parameters that needs to be trained, we freeze all the trained network layers in the original ResNet so that the gradients are not computed backward. We will show in the evaluation section that our transfer-learning enabled classification can provide great performance with only ten training instances collected from a new user.

TABLE 1
Air pressure in the environment during data collection.

User	1	2	3	4	5	6	7	8	9	10
Air pressure (hPa)	1009.9	1.14.6	1011.3	995.3	1012.4	1013.8	1006.9	1014.9	1018.2	995.6

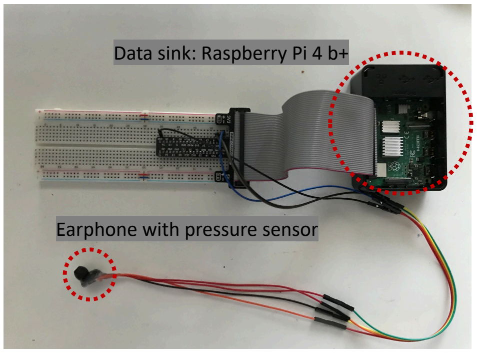


Fig. 13. Testbed that is used to collect ear carnal pressure.

6 Evaluation

6.1 Implementation

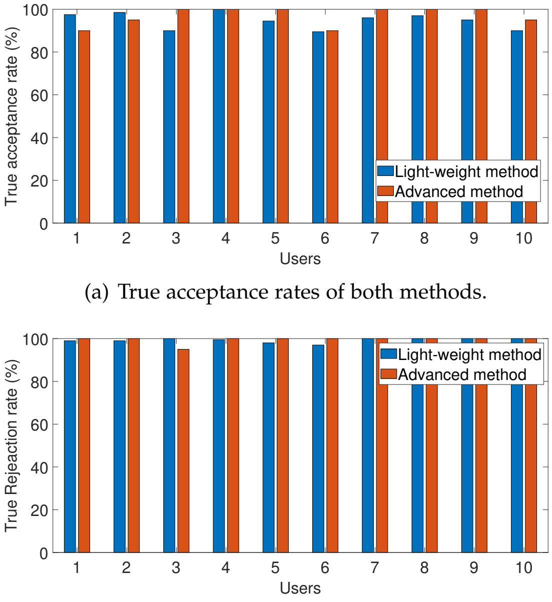
6.1.1 Hardware

To evaluate the performance of our system, we build a prototype to collect both ear canal pressure signal and the voice signal. The prototype consisted of five major components: a pressure sensor, a pair of ear phones, a mini PC to collect the pressure data, a microphone to collect voices, and a data processing center. Specifically, we selected BMP 280 as the sensor and embedded it into a Passion earphones, which are shown in Fig. 13. The pressure data is then transferred by wire to the Raspberry Pi (mini PC) and then sent to the data processing center by a wireless network. At the same time, we use a smartphone to record the voice.

6.1.2 Data collection

In our experiments, we collected data from ten participants (5 females and 5 males) who are university students and age from 25 to 30. Each participant was asked to wear the earphones with the pressure sensor in their right ear and record their voice. While using our system, each of them said a command “Alexa, turn on the light.” 50 times. In order to make sure the air pressure in their ear canals are only influenced by mouth openning activities, we use earbuds to ensure the participants wear the earphones tightly enough. Each participant attend the data collection in different rooms and at different times, so the air pressure (shown in Table. 1) in their environments can be different. For data analysis and processing, the data was then transmitted to a desktop computer with Intel(R) Core(TM) Devil’s Canyon Quad-Core i7-8700K @ 4.00 GHz CPU and 16 GB of RAM. In our experiments, we use the following performance metrics to evaluate the validation performance of our system. True acceptance rate (TAR) is defined as the rate at which a normal user is correctly accepted, and true rejection rate (TRR) refers to the probability that an attacker is successfully rejected by the system.

6.2 Overall performance



(a) True acceptance rates of both methods.

(b) True rejection rates of both methods.

Fig. 14. Overall performance.

In terms of the overall performance of our system across different users, we first evaluate how accurately our two methods can accept normal users. In these experiments, we randomly pick 20 instances from each user and the attacker respectively as the training dataset and evaluate the system performance on another 40 instances that are equally collected from the user and the attacker. We repeat the evaluation ten times to get the average true acceptance rate. For the lightweight method, we run the optimization five times, and we train the advanced model 25 times. As shown in Fig. 14(a), both methods can provide good performance on accepting normal users. For example, the lightweight method can already provide an average true acceptance rate of 94.8% for all users. Even if in the worst case (user 6), the lightweight method can still accept the normal user with an average accuracy of 89.5%. Also, our advanced method can achieve better performance with an average true acceptance rate of 97% for all users. For example, the advanced method can raise the true acceptance rate from 94.5% to nearly 100% for user 5.

Similarly, both of our methods achieve good performance on rejecting attackers. More specifically, our lightweight method can reject attackers with an average accuracy of 97% for all users, and the advanced method can further raise this accuracy to 99.5%. Also, compared with true acceptance rates, the true rejection rates are more steady with the lowest average rate of at least 95%.

6.3 Impacts of the size of training dataset

Then, we study how many training instances we need to collect from a new user to ensure good system performance. Fig. 15 shows the true acceptance and true rejection rates

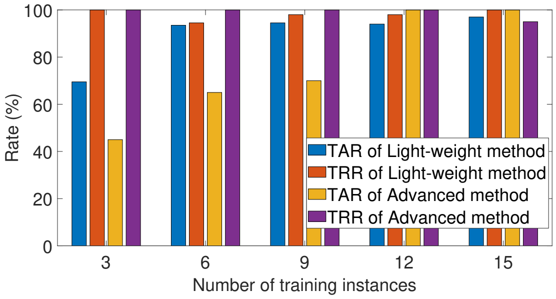


Fig. 15. Impacts of the size of training dataset.

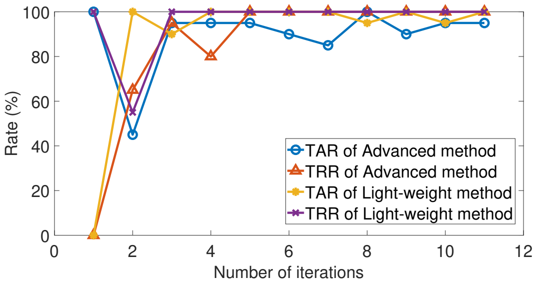


Fig. 16. Impacts of the size of training and optimization iterations.

of both methods with a different number of instances from the user. Besides the number of training instances, we use the same experimental setting as the one used in evaluating overall performance. It is clear that more training instances can significantly improve true acceptance rates. When we only have three training instances from the user, we can only accept the user with an accuracy of at most 69.5%. When the number of instances rises to nine, both systems can accurately accept the user with an average accuracy of at least 94.5%. Also, we notice that the true rejection rates are not significantly impacted. This is because the features or the power spectral density of the attackers' signals are more consistent than those of the normal users. Due to the differences in behaviors and pronouncing different phonemes, even the features from the same user can be slightly different. Moreover, we observe that the lightweight method has better performance when only training instances are available. The reason behind this is that the few training instances are not enough for determining the parameters in the final layer of the neural network.

6.4 Impacts of training iterations

We also evaluate the impacts of the number of training iterations. Moreover, training and optimization iterations can bring us less loss and a more accurate classification model, but more overhead will be introduced. To understand what is the minimal number of iterations required for the two methods, we fix the training dataset and adjust the number of iteration from 1 to 21, and the results are shown in Fig. 16. It is clear that both methods benefit from higher iterations. With a low iteration, the detection accuracy can be as low as 0%. Compared with the advanced method, the lightweight method needs a smaller number of iterations to provide good performance. For example, when the number of iteration is five, the lightweight methods can achieve a true detection rate of at least 95%. Since the advanced method has more parameters to train, it needs more iterations to provide at least the same performance.

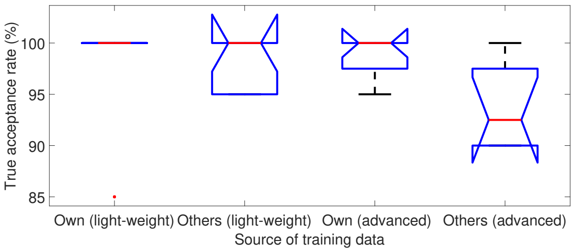


Fig. 17. System performance of using others' trained model.

6.5 Performance with other's trained model

Although we leverage different techniques, such as transfer learning, to reduce the training effort, it would be better if a new user can directly use a classifier that has been already trained using the data of other users. Therefore, we conduct experiments to evaluate if the trained model can be used on a new user. We trained two classifiers using two methods with the training data from one user and evaluate the true acceptance rates of the system on the dataset of another user. The experimental results are shown in 17. In terms of the lightweight method, using other's trained classifiers does not impact the average performance too much, but the robustness of the system will degrade. We can see a similar factor for the advanced method. By using other's trained model, the average true acceptance rate drops by about 6%. Also, the robustness of the system is also negatively impacted. Although the performance is not as good as using the own data for training, another's trained model can still provide acceptable performance for a new user.

6.6 Impacts of head movements

Recent research [2] indicates that head movements can also generate impacts to the air pressure in the ear canal. Such impacts are also at low-frequency bands and can potentially impact how the system rejects the voices from attackers. Therefore, we conduct experiments to evaluate the robustness of our system against head movements. Evaluation results show that the true rejection rates only drop by 4% under the impact of head movements. These results show that our system still has enough robustness against such head movements.

7 Discussion

7.1 Hardware availability and use scenario

The key of our system is to obtain the air pressure signal from the ear canal of the user. Currently, the available commercial earbuds are not equipped with an air pressure sensor. However, the air pressure sensor is expected to be embedded into the earbuds in the near future. For example, Apple Inc. was recently granted a patent that an air pressure sensor is embedded into future Powerbeats and/or Airpods pro earphones. Apple plans to insert the air pressure sensor into the ear canal of the user, which is exactly the same as our system setup. Moreover, thanks to the low power assumption of current air pressure sensors, this extra sensor will not significantly impact the working time of the wireless earphones. For example, the BMP 280 sensor only has

1120 μA at for peak current consumption [14]. Even this is the worst case, it is still lower than the current consumption of many other sensors in the smart earbuds. These facts make us believe that our system can be quickly deployed on the next-generation smart earbuds.

7.2 Device synchronization and data fitting accuracy

To detect the liveness of the voices, our system needs to perform analysis on well-synchronized air pressure signals. In the use scenarios we considered, we assume that the earbuds have Bluetooth components, such as Apple Airpods. Then, the voice assistant and the earbuds can synchronize with each other by exchanging Bluetooth packages. Also, we use signal fitting to address the issue of nonuniform sampling. The accuracy of the fitted signal will largely impact the accuracy of liveness detection. For example, if the impacts of the opening mouth are discarded in the fitted air pressure signal, our system is not able to leverage this part of the information to detect the liveness. However, it is hard to directly measure the accuracy of signal fitting since we are not able to directly measure the ground-truth air pressure while the user is talking. Therefore, we roughly measure the accuracy of the signal fitting based on the final performance of our system on liveness detection. As long as we can detect the liveness with a high accuracy, it means we reserve the most important information that is related to opening the mouth in the fitted signal. Even if some details of the ground-truth signals are discarded, it does not impact our following processing and analysis.

7.3 Limitations and future work

One limitation of our system is that we require users to wear earbuds while using our system. Considering the limited battery capacity of current wireless earbuds, this is hard to ensure in all scenarios. However, we can limit the use scenarios of our system to those that require higher security protection where users will be more willing to wear the earbuds to ensure the security of the voice service. In our current system settings, we detect the liveness of the voice based on the air pressure signal of saying each sentence. However, not all phonemes come with opening mouth. Even with the phonemes that require users to open their mouth, the degrees of opening are also different. Therefore, most parts of a sentence are not impacted by opening mouth. In our future work, we will study the impact of opening mouth at the phoneme level to achieve more accurate and robust performance.

8 Conclusion

In this paper, we conduct an in-depth study on the voice replay attacks towards voice assistant and propose a new voice liveness detection system with two detection methods. The basic insight of our system is that mouth opening activities will change the space size in the ear canal, which further changes the air pressure in ear canals. More specifically, we leverage signal processing techniques to detect mouth opening activities from the noisy air pressure data. In addition, we extract features from the time-frequency domain of the signal and propose two classification methods with different performance levels and computation complexities. To evaluate the system, we develop a prototype on Raspberry Pi and conduct comprehensive evaluations. Experiments with ten volunteers show that our system can accurately accept voice commands from legitimate users with accuracy of 94.8% and 97%. Moreover, our system can effectively defend current voice assistant devices from replay attacks with accuracy of 99.25% and 99.5%.

References

[1] A. Aley-Raz, N. M. Krause, M. I. Salmon, and R. Y. Gazit. Device, system, and method of liveness detection utilizing voice biometrics, Nov. 1 2016. US Patent 9,484,037.

[2] T. Ando, Y. Kubo, B. Shizuki, and S. Takahashi. Canalsense: Face-related movement recognition system based on sensing air pressure in ear canals. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 679–689, 2017.

[3] S. Chen, K. Ren, S. Piao, C. Wang, Q. Wang, J. Weng, L. Su, and A. Mohaisen. You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In *Proc. of ICDCS*, pages 183–195. IEEE, 2017.

[4] G. Cho, J. Choi, H. Kim, S. Hyun, and J. Ryoo. Threat modeling and analysis of voice assistant applications. In *International Workshop on Information Security Applications*, pages 197–209. Springer, 2018.

[5] K. Delac and M. Grgic. A survey of biometric recognition methods. In *Proc. of IS&T*, volume 46, pages 16–18, 2004.

[6] Y. Lee, Y. Zhao, J. Zeng, K. Lee, N. Zhang, F. H. Shezan, Y. Tian, K. Chen, and X. Wang. Using sonar for liveness detection to protect smart speakers against remote attackers. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–28, 2020.

[7] Y. Meng, Z. Wang, W. Zhang, P. Wu, H. Zhu, X. Liang, and Y. Liu. Wivo: Enhancing the security of voice control system via wireless signal in iot environment. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 81–90, 2018.

[8] Y. Meng, H. Zhu, J. Li, J. Li, and Y. Liu. Liveness detection for voice user interface via wireless signals in iot environment. *IEEE Transactions on Dependable and Secure Computing*, 2020.

[9] D. Mukhopadhyay, M. Shirvanian, and N. Saxena. All your voices are belong to us: Stealing voices to fool humans and machines. In *Proc. of Esorics*, pages 599–621. Springer, 2015.

[10] J. Rodgers. Adobe voco - should we be afraid? http://www.pro-tools-expert.com/home-page/2016/11/16/adobe-voco-should-we-be-afraid.

[11] M. Rouse and M. Haughn. voice assistant.

[12] F. Schiel. Automatic phonetic transcription of non-prompted speech. 1999.

[13] E. H. Schwartz. https://voicebot.ai/2019/12/31/the-decade-of-voice-assistant-revolution/.

[14] B. Sensortec. Data sheet: Bmp280 digital pressure sensor.

[15] J. Shang, S. Chen, and J. Wu. Defending against voice spoofing: A robust software-based liveness detection system. In *Proc. of MASS*, pages 28–36. IEEE, 2018.

[16] J. Shang, S. Chen, and J. Wu. Srvoice: A robust sparse representation-based liveness detection system. In *Proc. of ICPADS*. IEEE, 2018.

[17] J. Shang and J. Wu. Secure voice input on augmented reality headsets. *IEEE Transactions on Mobile Computing*, 2020.

[18] J. Shang and J. Wu. Voice liveness detection for voice assistants using ear canal pressure. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 693–701. IEEE, 2020.

[19] M. Shirvanian and N. Saxena. Wiretapping via mimicry: Short voice imitation man-in-the-middle attacks on crypto phones. In *Proc. of CCS*, pages 868–879. ACM, 2014.

[20] T. Sugawara, B. Cyr, S. Rampazzi, D. Genkin, and K. Fu. Light commands: laser-based audio injection attacks on voice-controllable systems. *arXiv preprint arXiv:2006.11946*, 2020.

[21] E. Uzun, P. H. Chung, I. A. Essa, and W. Lee. rtcaptcha: A real-time captcha based liveness detection system, Mar. 26 2020. US Patent App. 16/580,628.

[22] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields. Cocaine noodles: exploiting the gap between human and machine speech recognition. *WOOT*, 15:10–11, 2015.

[23] J. Villalba and E. Lleida. Detecting replay attacks from far-field recordings on speaker verification systems. In *Proc. of BIOID*, pages 274–285. Springer, 2011.

[24] C. Wang, S. A. Anand, J. Liu, P. Walker, Y. Chen, and N. Saxena. Defeating hidden audio channel attacks on voice assistants via audio-induced surface vibrations. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 42–56. ACM, 2019.

[25] C. Wang, C. Shi, Y. Chen, Y. Wang, and N. Saxena. Wearid: Wearable-assisted low-effort authentication to voice assistants using cross-domain speech similarity. *arXiv preprint arXiv:2003.09083*, 2020.

[26] Y. Wang, W. Cai, T. Gu, W. Shao, Y. Li, and Y. Yu. Secure your voice: An oral airflow-based continuous liveness detection for voice assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):157, 2019.

[27] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, 66:130–153, 2015.

[28] Q. Yan, K. Liu, Q. Zhou, H. Guo, and N. Zhang. Surfingattack: Interactive hidden attack on voice assistants using ultrasonic guided waves. In *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[29] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu. Dolphinattack: Inaudible voice commands. In *Proc. of CCS*, pages 103–117. ACM, 2017.

Jiacheng Shang received his Ph.D. degree in Computer and Information Sciences from Temple University in 2020. He is currently an Assistant Professor in the Department of Computer Science, Montclair State University, Montclair, New Jersey, USA. His current research focuses on the security and privacy issues in mobile cyber-physical systems.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.