Voice Liveness Detection for Voice Assistants using Ear Canal Pressure

Jiacheng Shang
Department of Computer Science, Montclair State University
shanj@montclair.edu

Jie Wu
Center for Networked Computing, Temple University
jiewu@temple.edu

***Abstract*—With the success of voice recognition techniques, users can easily control any device in smart home environments by simply saying a voice command. Based on this idea, a new group of smart devices are designed and released, which are called voice assistant. However, the voice itself is not secure and can be attacked in many ways. To defend against various types of voice replay attacks, we present a new voice liveness detection system. The basic insight of our system is that mouth opening movements will change the space size in the ear canal, which further changes the air pressure in ear canals. In this paper, we propose solutions to detect mouth opening movements using the noisy air pressure data and match them with the voices to validate the liveness of the voice source. To evaluate the effectiveness of our 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 an accuracy of 91.72%. Moreover, our system can effectively defend current voice assistant devices from replay attacks with an accuracy of 97.2%.**

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

I. Introduction

Voice-related techniques are becoming more important in current smart home devices. With the success of voice recognition techniques, users can easily control any device in smart home environments by simply saying a voice command. Based on this idea, a new group of smart devices are designed and released, which are called voice assistants. In general, a voice assistant first receives a voice command from a user. Then, the voice is processed to recover the text content either locally or on the remote server. Based on the predicted text content, the voice assistant device connects associated smart home devices (e.g. smart bulb) and performs a set of actions. Due to the important role of voice assistants in the future smart home environment, the industry has presented various voice assistant products. For example, Google released its voice assistant in 2016, and Apple released its HomePod in early 2018.

However, as a major component of voice assistants, the human voice is not secure. First, most voice assistants do not validate the identity of the voice source, which means any audio signal that sounds like a voice can be accepted by the voice assistant. For example, a recent report shows that the Apple Siri can be activated by Apple's new AirPods advertisement [8]. Even if we protect the voice assistant using voice authentication techniques, they still suffer from various voice replay attacks. Since the human voice is always exposed to the public, an attacker can easily steal the victim's voices and replay it to voice assistant. Moreover, by collecting enough training data, strong attackers can generate synthetic voices of the victim even if they did not hear the victim say a certain command. For instance, with state-of-the-art speech synthesis techniques (e.g., Adobe Voco [12]), an adversary could impersonate the victim to spoof the voice-based authentication system once they acquire enough of the victim's voice samples. These recorded voices and synthetic voices can then be replayed to voice assistants through either audible or inaudible ways. Since voice is considered as a unique biometric of a person, and thereby, it is characterized as a basis for personal authentication [5], these voice-spoofing attacks would result in severe consequences harmful to the victim's safety, reputation, and property [11], [19], [28].

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

To defend against voice replay attacks, researchers have proposed various liveness detection systems in the past few years. As shown in Fig. 1, the objective of these systems is to validate whether a voice is produced by a live human being. To achieve this goal, these systems leverage key differences between human vocal systems and loudspeakers. However, most current liveness detection systems are designed for smartphones and wearable devices. Therefore, their operation ranges (less than 50 cm) are too short, which limits their implementation in voice assistant devices. To improve the range of voice liveness detection systems, researchers study to build new liveness detection systems for voice assistant devices with the help of extra sensors [4], [7], [9], [26]. For example, Wang et al. leverage WiFi signals to detect mouth movement during speech. However, their systems require that a WiFi transmitter and a receiver are always available near the user, which largely limits its deployment in real smart home environments. Lee et al. implement microphone and speaker

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

matrixes in the voice assistant and use them as a Doppler radar to associate the voice command to a moving human being [7]. However, their methods are sensitive to ambient audio noise.

Considering the limitations of current solutions, we propose a new voice liveness detection system that can achieve two major objectives. First, the liveness detection system should be able to accurately and robustly reject any voice commands from replay attackers while ensuring good user experiences of legitimate users. Second, the system should be easily deployed in the smart home environment, which means that a user can use our system at any position in the room. We notice that the average volume per capita of headphones has risen quickly worldwide. Based on the data from Statista [20], the average number of headphones per user is 0.49 in 2020. Moreover, more headphones, especially in-ear headphones, are equipped with an air pressure sensor. The original purpose of this sensor is to determine whether the earphones are worn by the user. In this paper, we reuse the air pressure sensor in in-ear headphones to validate the liveness of voices. The basic insight behind our system is that the mouth movements of a user speaking some special phonemes can influence the air pressure in ear canals. For replay attackers, they are not able to generate changes in the air pressure in the victim's ear canal. Therefore, by matching the air pressure signal with the received voice, we can determine whether the voice is from a human being.

To build such a system, we address three major challenges. First, the commercial air pressure sensor does not always report sensor data at a fixed rate. To solve this problem, we leverage signal processing techniques to resample the raw air pressure data to a uniform sampling rate while keeping useful information. Second, the air pressure data contain much noise from sensor hardware and the environment. In our system, we first leverage Discrete Wavelet Transform-based techniques to filter out high-frequency noise. Then, we study the variance features of the filtered signal to accurately locate the positions of mouth opening activities of interest. Third, a recent study shows that the air pressure in ear canals can also be influence by other facial activities like turning the head [2]. To remove the impacts of these facial activities, we further build a binary classifier based on Mulitple Additive Regression to validate whether the pressure change is indeed caused by opening of the mouth.

Our contributions are as follows:

- Our work serves 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 build a classifier to further enhance the detection.
- We develop a prototype and conduct comprehensive evaluations. Experiments with ten volunteers show that our system can accurately accept voice commands from legitimate users with an accuracy of 91.72%. Moreover, our system can effectively defend current voice assistant devices from replay attacks with an accuracy of 97.2%.

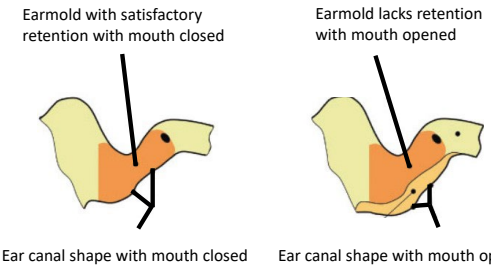


Fig. 2. Changes in ear canal when the mouth is open and closed.

II. Related work

A. 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 [13]. 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 [15]. Therefore, the security of voice assistants is very important.

B. 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 [21], [23], [29], [30] and speaker verification [11], [24] suffer from attacks. In terms of the attacks on voice recognition systems, [30] 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 [24]. To defend against these attacks, researchers have proposed various countermeasures by studying the differences between human vocal systems and loudspeakers [1], [3], [10], [17], [18], [22], [25], [27]. 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 [**?**] 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.

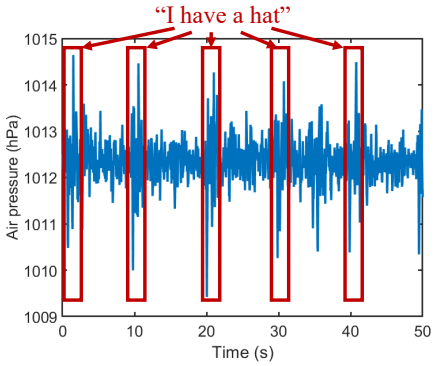


Fig. 3. Results of feasibility experiments.

III. Preliminary

A. 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. 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.

B. 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.

C. 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.

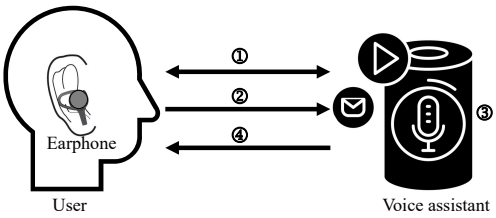


Fig. 4. Usage scenarios of our system.

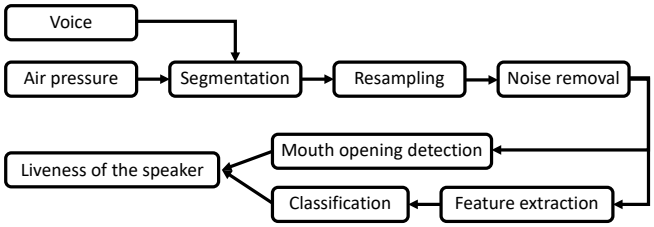
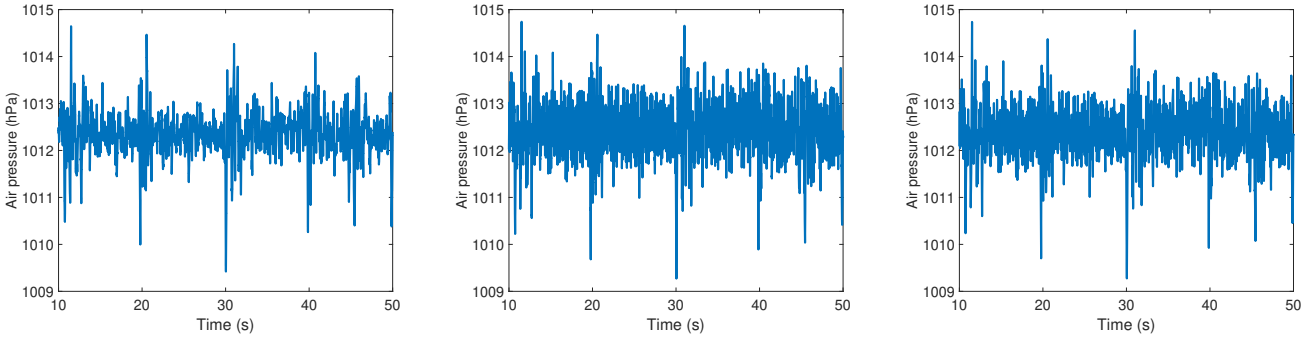


Fig. 5. System architecture.

IV. System Design

A. 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 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



(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.

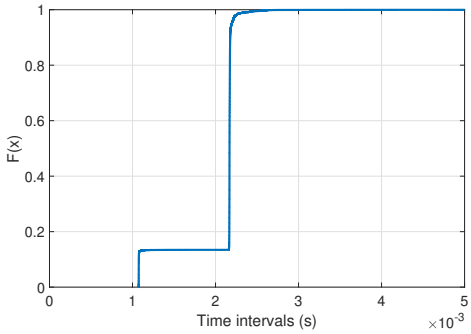


Fig. 7. The distribution of measurement intervals.

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.

B. 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 if 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.

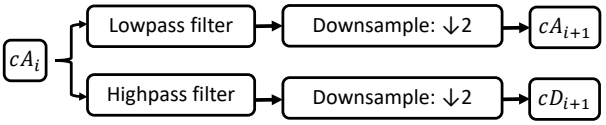


Fig. 8. Results of feasibility experiments.

C. 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.

V. Solution

A. Preprocessing

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 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

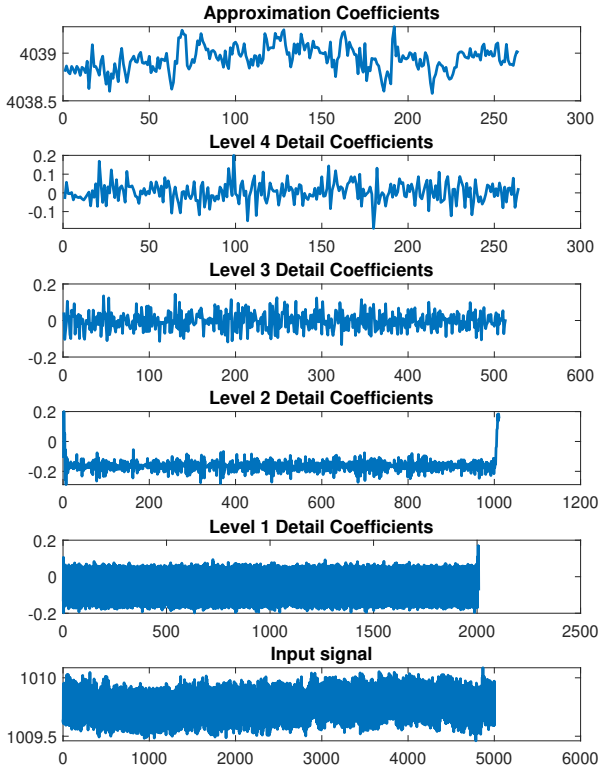


Fig. 9. DWT-based noise removal.

[14]. Then, we use the obtained timestamps to segment air pressure signals for further analysis.

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.

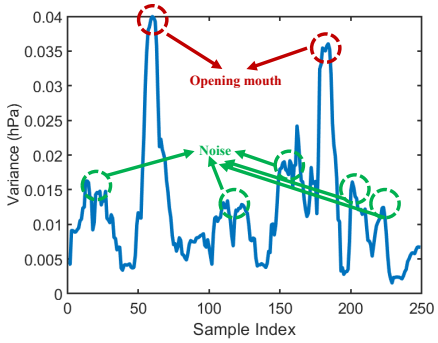


Fig. 10. Filtered variance signal.

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.

B. Mouth opening detection

By leveraging DWT-based denoising techniques, we remove most high-frequency noise in the pressure signal, but low-frequency noise still exists and negatively impacts the detection. To further eliminate the influence of low-frequency noise and accurately detect the location of mouth opening activities, we calculate the short-term variance of the signal. The basic insight is that mouth opening activities introduce much larger variances to the pressure signal. Fig. 10 shows the short-term variance signals that are calculated from the approximation coefficients cA_4 in Fig. 9 using a window of 0.4 seconds. We

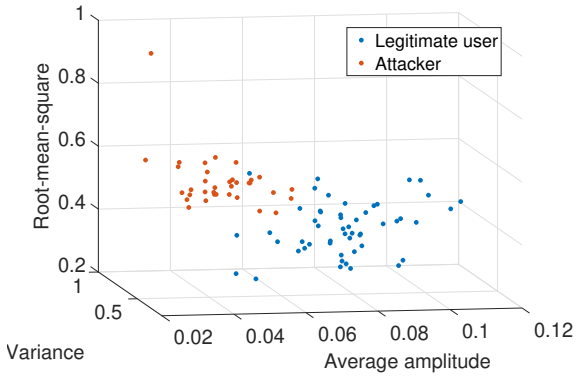


Fig. 11. Feature distribution.

can see the short-term variances reach very high values when the user opens the mouth. Although noise can also generate some peaks in the variance signals, their amplitude is much lower than those that are introduced by opening the mouth. By performing peak finding algorithms with a threshold, we can effectively detect the mouth opening activities in the variance signals. In our system, the threshold is set to the average variance value when the pressure is only influenced by the environment.

C. Enhanced detection with classification

In the last subsection, the voice is regarded as from a live speaker as long as there is a peak that exists in the air pressure signal and is greater than the threshold. However, low-frequency noise may still exist in the filtered signal, which generates variances to the short-term variance signal. These variances may also introduce some spikes that can be wrongly treated as those introduced by mouth opening activities. To further eliminate these low-frequency noises, we leverage an extra classifier to determine whether the short-term variance signal matches with those that are influenced by opening the mouth. In terms of feature extraction, there are two major challenges. First, the short-term variance value highly depends on the environment the user is in, which means the absolute values may vary even for the same user and same room. Second, the sampling rate of the short-term variance signal is low (50016=31.25 Hz), which means it is hard to perform Fast Fourier Transform on it. To address the first challenge, we normalize the segmented short-term variance signals to a range (0,1]. Considering the second challenge, we extract features from the time domain, including average amplitude, root-mean-square value, and overall variance. Fig. 11 illustrates the feature distribution for a legitimate user and an attacker on the three-dimension hyperplane. We can see that it is feasible to find a surface to split these two types of data points.

In our system, we build the classifier using the Multiple Additive Regression Tree (MART). The main reason we selected MART is the scales and units of three features are different. By using MART, the classifier can effectively deal with the colinearity of features.

A MART-based consists of a series of weak classifier where each weak classifier is a regression tree. The final

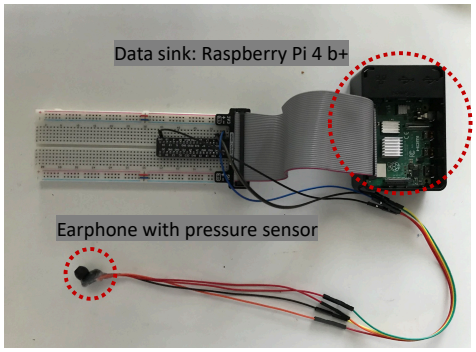


Fig. 12. Testbed that is used to collect ear carnal pressure.

classification result is produced based on the results from all weak classifiers. In this paper, we use the formulation of MART in [6]. Basically, the generation of a MART classifier consists of multiple rounds, and a new weak classifier is created every round. We use $b_i h(x; \mathbf{a})$ to represent the newly created weak classifer in i^{th} round where b_i is the coefficients and $\mathbf{a}_i$ are parameters vector. After m rounds, the estimation $F(x)$ of the strong classifier is an additive expansion of the form $F(\mathbf{x}) = \sum_{i=0}^{m} b_i h(x; \mathbf{a})$, where $h(\mathbf{x}; \mathbf{a})$ is a weak classifier with parameters $\mathbf{a} = \{a_1, a_2, \ldots, a_n\}$ and feature vector $\mathbf{x} = \{x_1, x_2, x_3\}$. Here n is the number of parameters for each weak classifier. In each round, the coefficients b_i and the parameters $\mathbf{a}_i$ are jointly fit to the training data in a forward "stage-wise" manner. Starting with an initial guess $F_0(x)$, the coefficients b_i and the parameters $\mathbf{a}_i$ in the i^{th} iteration can be found by solving the following problem:

$$(b_i, \mathbf{a}_i) = \arg\min_{b,\mathbf{a}} \sum_{j=1}^{3} L(y_j, F_{i-1}(\mathbf{x}_j) + bh(\mathbf{x}_j; \mathbf{a})), \quad (1)$$

where y_j is the diagnosis variable, and $L(y, F)$ is the loss function that is used to define lack-of-fit. Therefore, the estimation of the strong classifier after the i^{th} iteration is expressed as:

$$F_i(\mathbf{x}) = F_{i-1}(\mathbf{x}) + b_i h(\mathbf{x}; \mathbf{a}). \quad (2)$$

For example, if the labels y of all legitimate user's instances are 0, the classifier tries to find a series of parameters $\mathbf{a}$ and $\mathbf{b}$ so that their final estimations are close enough to 0. Similarly, the final estimations of attackers' data should be close enough to 1. In our system, we implemented the MART-based classifier using the library of scikit-learn [16]. Specifically, we chose the deviance function as the loss function and set the learning rate to 0.1. Since the MART-based classifier is fairly robust to over-fitting, we set the number of iterations to 5000 to achieve better performance. For each regression tree, the maximal depth is set to 4, and the number of features to consider when looking for the best split is set to 4.

VI. Evaluation

A. Implementation

1) Hardware: To evaluate the performance of our system, we build a prototype to collect both ear canal pressure signal

TABLE I
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

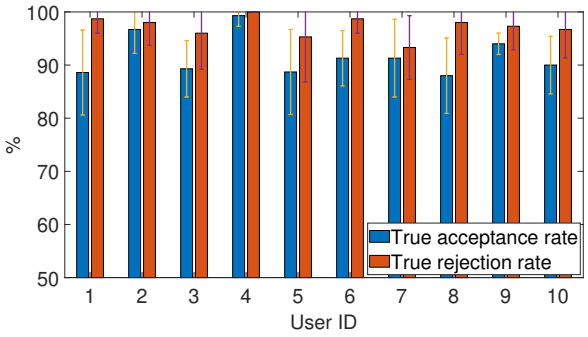


Fig. 13. Overall performance.

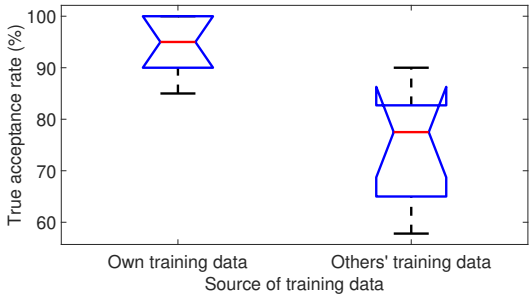


Fig. 14. Impact of the source of training data.

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. 12. 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.

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. I) 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 is defined as the rate at which a normal user is correctly accepted, and true rejection rate refers to the probability that an attacker is successfully rejected by the system.

B. Overall performance

A good liveness detection system should accurately accept voice commands from legitimate users while rejecting those from attackers. In this subsection, we evaluated the system performance in these two aspects using the data collected from eight participants. For each user, we used 15 training instances as the training data to train a MART classifier. The MART classifier contains 2,000 trees, and the maximum depth of each tree is limited to three. Fig. 13 shows the system performance for eight participants. We can see that our system can accurately accept voice commands that are from legitimate users with an average accuracy of about 91.72%. We also studied the air pressure signals of participants whose true acceptance rates are relatively low. We find that their mouth opening activity generates less influence on the pressure signal, which makes their features can be more likely to be covered by noise. Possible reasons behind this are: 1) The earbuds do not fit their ear canal well, or 2) They did not push the earbuds deep enough in their ear canal. These can be solved by giving users more choices on earbuds in practice.

We also studied how accurately our system can reject voice commands that are from replay attackers. As shown in Fig. 13, our system can provide high true rejection rates (about 97.2%) for all participants. The major reason is that the features of air pressure signals during the non-speech period are relatively stable. Therefore, the learned decision boundary can effectively distinguish attackers' data in feature hyperplane from those of legitimate users. Overall, our system provides high system performance for both accepting legitimate users and rejecting replay attackers.

C. Performance using other's training data

For commercial voice assistant devices, we would like to launch our system as quickly as possible for a new user. Ideally, we would like to reduce the training cost of the new user to zero. In this subsection, we compared the system performance for a legitimate user when using the user's own training data and other users' data, respectively. In this experiment, we trained the classifier with 2000 trees and 30 training instances. Among them, 15 instances are positive (can be from the legitimate user or other users). Fig. 14 illustrates the true acceptance rates under these two settings. We can see that the average true acceptance rate can drop to about 75.7%. In the worst case, true acceptance can be as low as 57.8%.

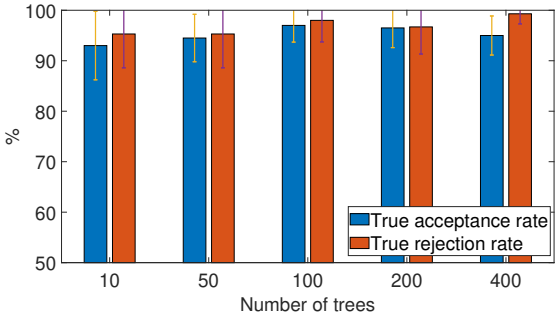


Fig. 15. Impact of number of trees in MART.

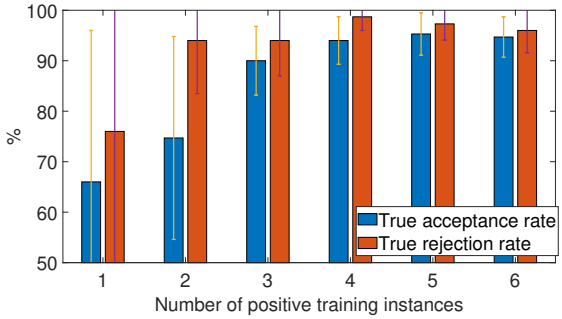


Fig. 16. Impact of training size.

The major reason is that the feature distribution of different legitimate users may vary. Therefore, the decision boundary learned for one legitimate user may not work for the other user's data.

D. Impact of number of trees in MART

Since we leverage MART as the model for classification, the computation overheard mostly depends on how many trees we use in building the classifier. To evaluate what is the minimum number of trees that are needed to ensure good system performance, we designed an experiment using a participant's data and the results are shown in Fig. 15. Even with 10 trees, Our system can already provide high system performance (over 90%) for both accepting legitimate users and rejecting attackers. By having more trees in the classification stage, the system performance can be further and slightly improved. Moreover, with a larger number of trees, the variances of both true acceptance rate and true rejection rate gradually decrease, which means that the MART classifier with more tress can provide better robustness. However, more trees mean more computation overhead not only in the training stage but also in the testing stage.

E. Impact of training size

As we discussed in Section VI-C, others' training data cannot always ensure high system performance for a new user. Therefore, we still need to collect some data from a new user to train a personalized classifier. In this subsection, we studied what number of training instances is necessary for a new user. Specifically, we leveraged a MART classifier with 200 trees and adjusted the number of new user's training instances. To ensure the balance in the training dataset, we also adjusted the number of attackers' training instances so that these two numbers are identical. Similarly, we run the evaluation ten rounds and randomly selected data for training and testing in each round. Fig. 16 shows how the system performance changes with the increase in the number of training instances from a new user. We can observe that both rates rise by introducing more training instances. For example, when only one training instance is available, our system can accept a legitimate user with an average accuracy of 66%. By introducing five more training instances from the legitimate user, the true acceptance rate can be improved to 96%. Moreover, the system's robustness can be improved with more training data. For instance, the standard deviation is 30% when only one training instances available, which means the true acceptance rate can be much lower than the average value. By asking the new user to say 6 commands for training, the standard deviation of true acceptance rates can be largely reduced to 4%.

TABLE II
COMPUTATION OVERHEAD.

Stage	Signal processing & feature extraction	Classification
Time (s)	0.091	4.0129e-04

VII. DISCUSSION

A. Impact of other facial activities

Based on recent research [2], some head movements (e.g. turning head) can also generate an impact on the air pressure in the ear canal. Usually, the variances introduced by these head movements are weaker than those introduced by opening mouth, which means we can reject them by using the threshold in peak filtering. In this subsection, we studied whether our system may wrongly accept attackers' voice commands when the legitimate user is performing head movement activities. In our experiments, we focus on two head movement activities: turning the head and nodding the head. Specifically, we asked a participant to perform these two activities while closing his mouth. We leveraged the trained system to detect these signals for liveness detection, and the results are shown in Fig. 17. We find that the introduction of these head movements does reduce system performance. For example, the average true rejection rate can drop to about 80% when the user turns the head. These facts mean that turning the head may also generate strong variances to the pressure signal. This issue can be addressed by extracting more powerful features from the pressure signal and leveraging another classifier to reject these activities. These possible solutions are our future work on this topic.

B. Usability

Except for accuracy, processing, and validation time is also critical and determines usability. We further test the time our system needs to process the raw signal and get the final validation results, and experimental results are shown in Table. II. We can see that the average time for signal processing and feature extraction is about 91 ms. Once the classifier is well trained, the system only needs about 0.4 ms to give a

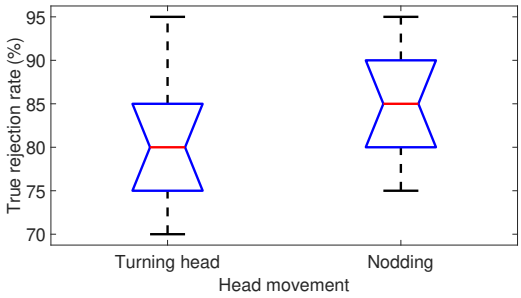


Fig. 17. Impact of head movement.

classification result. These facts mean that our system can return the results to the user within 0.1 seconds after receiving the voice command. Considering the data processing can be executed with other voice services in parallel, our system will not put obvious computation overhead to current voice assistants. In addition, compared with existing works, our system does not need users' extra effort in operating the voice assistant, e.g., moving the device around the audio source. To further strengthen the usability of our system, we adopt the same human-computer interaction methods used by current voice assistants, so that users can quickly get used to using our system.

VIII. 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. 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 detect pressure signals and match them with the features that are collected from the live person to validate the liveness of the voice source. 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 an accuracy of 91.72%. Moreover, our system can effectively defend current voice assistant devices from replay attacks with an accuracy of 97.2%.

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] J. H. Friedman and J. J. Meulman. Multiple additive regression trees with application in epidemiology. *Statistics in medicine*, 22(9):1365–1381, 2003.

[7] 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.

[8] B. Lovejoy. https://9to5mac.com/2019/07/03/hey-siri-is-being-activated/.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

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

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

[16] scikit learn.

[17] 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.

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

[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] statista.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.