Defending Against Voice Spoofing: A Robust Software-based Liveness Detection System

Jiacheng Shang[†], Si Chen[‡], and Jie Wu[†]
[†]Center for Network Computing, Temple University, Philadelphia, PA, 19121
[‡]Computer Science Department, West Chester University of Pennsylvania, West Chester, PA, 19383

***Abstract*—The recent proliferation of smartphones has been the primary driving factor behind the booming of voice-based mobile applications. However, the human voice is often exposed to the public in many different scenarios, and an adversary can easily steal a person's voice and attack voice-based applications with the help of state-of-the-art voice synthesis/conversion softwares. In this paper, we propose a robust software-based voice liveness detection system for defending against voice spoofing attack. The proposed system is tailored for mobile platforms and can be easily integrated with existing mobile applications. We propose three approaches based on leveraging the vibration of human vocal cords, the motion of the human vocal system, and the functionality of vibration motor inside the smartphone. Experimental results show that our system can detect a live speaker with a mean accuracy of 94.38% and detect an attacker with a mean accuracy of 88.89% by combining three approaches we proposed.**

***Index Terms*—Voice authentication, liveness detection.**

I. INTRODUCTION

The recent proliferation of smartphones coupled with the demand for a convenient and non-intrusive way of communicating and controlling have been the primary driving factors behind the booming of voice-based mobile applications. In addition to traditional voice over IP (VoIP) applications, e.g., Skype, which allows users to make voice calls to contacts, voice-based mobile applications have also become mainstream. These applications all provide a voice input interface, which allows users to submit their voices and receive information from that voice. For example, WeChat provides "Voiceprint" [22] authentication interface, which allows users to log into their accounts by speaking passphrases. Besides, SayPay [14] offers a solution that fuses mobile payments with the human voice. These voice-based mobile applications can be quickly developed and implemented for existing smart devices as they require only a microphone, which is small and inexpensive [9].

However, unlike other human biometrics, the human voice is often exposed to the public in many different scenarios, e.g., people making a presentation in public, answering phone calls, talking loudly in a restaurant. As such, with the availability of high quality and low-cost handy recorders and other recording devices (e.g., smartphones), a malicious user can easily steal a person's voice without being noticed. Several security issues are therefore caused by the leakage of people's voices and pose a severe threat to voice-based applications [10, 15, 24]. For instance, with state-of-the-art speech synthesis techniques (e.g., Adobe Voco [13]), an adversary could impersonate the victim to spoof the voice-based authentication system once they acquire enough victim's voice samples. Since voice is considered as a unique biometrics of a person, and thereby, it is characterized as a basis for personal authentication [4], these voice-spoofing attacks would result in severe consequences harmful to victim's safety, reputation, and property.

The traditional technique for defending against voice-spoofing attacks is to implement an automatic speaker verification (ASV) system, which has already been deployed in many popular mobile applications like WeChat. The ASV systems employ unique vibration patterns of a user's vocal chords and the sound-based feature created by other physical components (e.g., mouth) to assign a unique fingerprint. However, spoofing techniques against these systems have also progressed drastically [7, 10, 24]. Moreover, when detecting the attack, current ASV systems require prior knowledge of specific voice spoofing techniques used by the attacker [6], which is unrepresentative of the practical scenario. Therefore, the development of a generalized defense system for voice-spoofing attacks is of the utmost importance. Recently, many liveness detection systems are proposed to fight voice-spoofing attacks by studying the differences between the human vocal system and loudspeakers. VoiceLive [26] can fight replay attack by capturing time-difference-of-arrival (TDoA) changes in a sequence of phoneme sounds to the two microphones of the phone. However, it needs the same relative location of user's mouth during authentication, which is hard to satisfy in practice. A liveness detection system is proposed in [25] and can detect a live user by leveraging the unique articulatory gestures of the user when speaking a passphrase. However, it cannot work if the attacker performs a jamming attack using high-frequency audio.

Considering the limitations of current solutions, we propose a robust software-based voice spoofing defense system which is tailored for mobile platforms and can be easily integrated with existing voice-based mobile applications. Our solutions use the unique vibration of human vocal cords and the movement of throat as key differentiating factors for liveness detection. Compared with existing ASV system, our solution does not assume any prior knowledge of the attacking method and is easy to operate. Moreover, our pure software solution is ready to use and can be seamlessly deployed on off-the-shelf smartphones.

We solve two challenges in the design of our system.

First, in order to capture the vibration of vocal cords and the movement of the throat simultaneously, we need to use both the prime microphone (at the bottom) and front microphone. Since different people have different speaking habits and use different languages, it is difficult to extract a common pattern that can be used to detect the liveness of a speaker. To solve this problem, we perform spectrum subtraction of two audio signals and utilize the energy differences of different time slices and frequency band as a unique feature. Second, the sampling rate of the accelerometer-equipped on smartphones is only 50 Hz, which is not good enough to fully recover the human throat movement. To address this issue, we extract multiple features from the acceleration readings to build a robust classification model and use it to determine if the captured data is generated by human throat movement.

We summarize our contributions as follows:

- We propose a robust software-only solution for defending against voice-spoofing attacks on smartphones with high accuracy.
- We select and combine advanced acoustic signal processing, mobile sensing, and machine learning techniques and apply them in detecting the unique vibration pattern when speaking.
- We develop a prototype and conduct comprehensive evaluations. Experimental results show that our spectrum-based approach can achieve both 100% true acceptance and rejection rates. Our motion-based approach can achieve mean accuracy of 96.8% and mean true rejection rate of 88.89%. Our random vibration-based approach can detect and locate the vibration with an accuracy of 97.5%. By combining three approaches we proposed, our system can detect a live speaker with a mean accuracy of 94.38% and detect an attacker with a mean accuracy of 88.89%.

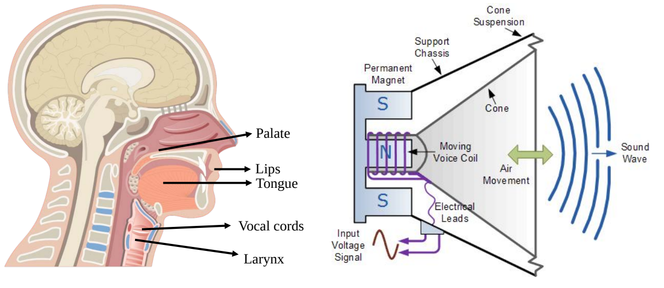
The remainder of this paper expands on above contributions. We first introduce our attack model and key insights in Section II and present our solutions in Section III. We conduct various experiments to evaluate proposed solutions in Section IV and discuss the usability and limitations of our system and related work in Sections V and VI.

II. PRELIMINARIES

A. Attack model

The voice-spoofing attacks aim to attack the biometric identification of the normal user. In our attack models, an attacker is able to access victim's smartphone and record the voice of the victim without being noticced. Also, an attacker can be equiped with one or more high-quality loudspeakers. Based on collected audio signals, an attacker can launch various attacks like replay attacks. The voice-spoofing attacks considered in our work can be divided into two categories.

A simple replay attack. In this type of attack, an attacker can use high-quality loudspeakers to replay collected victim's voice or morphed voice, so that the attacker can impersonate the victim at a high degree of similarity. We assume that an attacker can access victim's smartphone in the case of not being noticed.



(a) Human vocal structure (b) Speaker's structure

Fig. 1. The differences between the human vocal system and a loudspeaker

A strong replay attack. In this type of attack, we assume that the attacker can capture more information besides the victim's voice at the mouth. For example, the attacker can attack the database of current ASV system and fetch the voice signals at both victim's mouth and throat. An attacker can leverage multiple loudspeakers to replay two audio signals to two microphones and imitate the human vocal systems better.

B. Background knowledge

In order to achieve robust liveness detection, we need to understand the structural differences between the human vocal system and loudspeakers. As shown in Fig. 1(a), the mechanism for producing the human voice can generally be subdivided into three parts: the lungs, the vocal folds, and the articulators. The lung first produces adequate airflow and air pressure to vibrate vocal folds. The vocal cords vibrate and chop up the airflow from the lungs into audible pulses that form the laryngeal sound source. Then, the length and tension of the vocal cords are adjusted to produce 'fine-tune' pitch and tone. The articulators consisting of tongue, palate, cheek, lips further filter the sound generated from the larynx to strengthen it or weaken it. The vocal folds (vocal cords) are the primary sound source to produce voiced phoneme in the human vocal system. Besides voiced phoneme, there exist other sound production mechanisms produced from the same general area of the body, involving the production of unvoiced consonants, clicks, whistling, and whispering. The only difference between voice and unvoiced phonemes is that there is no vibration of the vocal cords for unvoiced phonemes. This fact suggests that the audio signals collected near the throat and the mouth can be different, and this difference can only be produced by the human speaker.

Strong attackers usually use high-quality loudspeakers for spoofing attacks. As shown in Fig. 1(b), the loudspeakers usually use an electromagnet to translate an electrical signal into an audible sound. The electromagnet is a metal coil that creates a magnetic field when there is an electric current flows through it. When electrical pulses pass through the coil of the electromagnet, the direction of the magnetic field is frequently changed. Also, there is a permanent magnet fixed firmly into the loud speaker. With rapidly changed magnetic filed, the coil is attracted to and repelled from the permanent magnet. As a result, the cone attached on the coil will vibrate back and forth, pumping sound waves into surrounding air and smartphone's

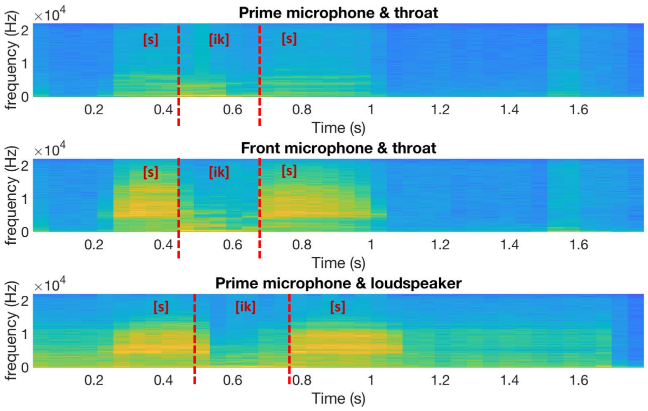


Fig. 2. The spectra of audio signals collected from two microphones near the mouth, the throat, and the loudspeaker

speaker, which means the two microphones of a smartphone around the loudspeaker will capture very similar audio signals.

C. Key insights

In order to resist two types of attacks we considered, we need to leverage the structural differences between human vocal systems and loudspeakers discussed in Section II-B. We observe that human voice can be divided into the voiced and unvoiced part. During voiced part, the vocal cords keep vibrating and generate low-frequency audio signals at the throat. The vocal cords stop vibrating during unvoiced part, while the other parts of the human vocal system generate different sounds. We collect voice signals when a user says "Six" at two locations (the throat and the mouth) using two microphones, and the results are illustrated in Fig. 2. It is clear that the audio signal collected near mouth reserves the information of unvoiced parts, but most information of the unvoiced part is lost in the audio signal collected near the human throat. Also, both audio signals reserve the information of voiced part, while the audio signal collected near throat only contains the information at the low-frequency part. Different from human vocal systems, the cone keeps vibrating for both voiced and unvoiced parts in order to generate sounds. We use a loudspeaker to replay the voice of the user and collect the audio signals in the same way. Fig. 2 also shows the spectrum of the same audio signal played by a loudspeaker and captured by the prime microphone. We can observe that the spectrum contains much more information of unvoiced parts than that collected near the human throat.

When a person is speaking, the vocal cords vibrate at a relatively high frequency, and the throat also moves up and down in a relatively low frequency. Opposite to this, loudspeakers do not have the same movement pattern. If we put a motion sensor next to a human throat, the vibration of vocal cords and movement of a throat generate two different influences on the readings. Based on this observation, we argue that the influences generated by vocal cords and throat are hard to be imitated by loudspeakers. We will discuss the liveness detection using acceleration signals in Section III-D.

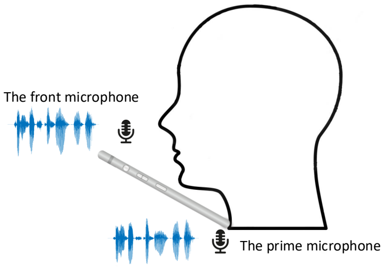


Fig. 3. The use case of our liveness detection system

D. Use case

In order to successfully defend users from spoofing attacks, our system requires users to put the bottom side of the smartphone on the throat while using the normal voice authorization systems, as shown in Fig. 3. We leverage two microphones that are available on most current smartphones. The prime microphone is used to capture the low-frequency voice cased only by the human throat, and the front microphone is used to record human voice on the whole frequency band. Two audio signals are well synchronized by smartphones operating systems. The distance between the human throat and the prime microphone must be zero, and the distance between human lips and the front microphone is about $6cm$. Since the distance is pretty short, the time delay between two audio signals is less than 8 samples when the sampling rate is 44,100 samples per second. While speaking the passphrase, the user should put the bottom side of the smartphone on the throat. During this process, the user should be in stationary postures, like sitting and standing.

E. Challenges

Although we get insights in Section II-C, it is still challenging to perform liveness detection on a smartphone using only audio signals and accelerations readings. The first challenge is how to extract useful information from audio signals in two channels. Since different people have different speaking habits and use different languages, it is extremely hard to extract a common pattern that can be used to determine if the source is a real person. To solve this problem, we compute the STFT of two audio signals and get their spectrum subtraction. The spectrum subtraction is then treated as a picture, and the color represents the energy in corresponding time frequency band. We use an image classification algorithm to determine the liveness of the speaker.

The second challenge is that current smartphones only provide acceleration reading at a sampling rate of 50 Hz. Since voice-based authentication only lasts for about 3 seconds, it is hard to extract human throat movement from limited acceleration readings (about 150 samples). To address this issue, we extract multiple features that describe acceleration signal in different aspects. The features are then used to build a robust classification model and determine if the acceleration reading is affected by human throat movement.

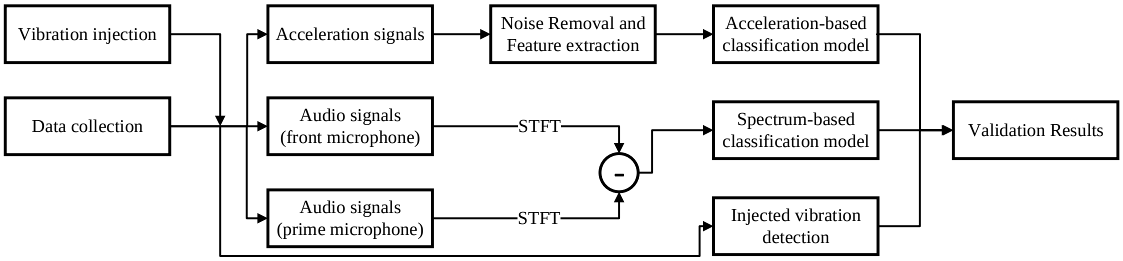

Vibration injection
Data collection
Acceleration signals
Audio signals (front microphone)
Audio signals (prime microphone)
Noise Removal and Feature extraction
STFT
STFT
-
Acceleration-based classification model
Spectrum-based classification model
Injected vibration detection
Validation Results

Fig. 4. System pipeline

III. SYSTEM DESIGN

A. Approach overview

The key idea underlying our liveness detection system is to fully leverage the nature of human vocal system in order to detect the liveness of the speaker. When a live speaker speaks a passphrase as we asked in Section II-D, the primary microphone only records the voice produced by the vibration of the vocal cords, while the front microphone records the voice produced by the whole vocal system. Based on this observation, we study the spectrum property of audio signals and propose a spectrum-based approach to determine if the input audio is from a real user in Section III-C. Moreover, human throat will move up and down, and vocal cords will vibrate in high frequency. Both movements generate different influences on the accelerator embedded in the smartphone. A motion-based approach is designed in Section III-D to find proper features and classification model to determine if an acceleration sequence is from a normal user. An attacker, who wants to perform replay attack, cannot imitate the human vocal system well and cannot get the same pattern on the audio spectrum and acceleration sequence. Furthermore, in case that an strong attacker can steal victim's raw audio files from database, we design a random vibration-based approach to inject a random noise in the collected audio signals. By analyzing the number of injected vibrations, our system can recognize if the input audio signal is new or stolen from the victim.

B. System pipeline

The pipeline of data collection and processing is shown in Fig. 4. Our system captures audio signals in two channels and collects acceleration reading at the same time. The acceleration reading is further processed and analyzed to validate if the smartphone is touching human throat during data collection. A classification model is trained based on support vector machine (SVM) using proper features. The raw audio signals are processed by short-time Fourier transform (STFT) to get the spectra. We compute the subtraction of two spectra and use it as an input to match existing patterns. If the spectrum subtraction matches the existing patterns, the spectrum-based classification model will regard the user as a real person. In case that the attacker steals user's voice recording from other databases, we inject a random and short vibration during data collection. The random vibration is then used to evaluate if the input is a new recording or a stolen recording. Then, three results are combined together to get the final validation results. A user is recognized as a real person if and only if all three decision components give positive results.

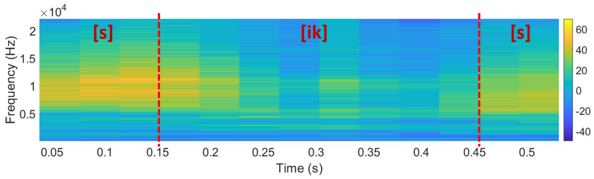


Fig. 5. The spectra difference

C. Spectrum-based approach

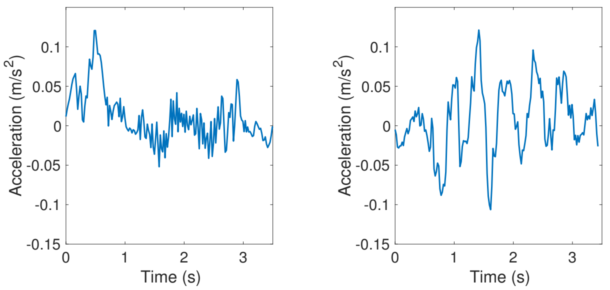
To distinguish if the voice is from a live speaker or a loudspeaker, we need to find features to represent the relationship and differences between two audio samples collected from two microphones. In order to capture features on both frequency domain and time domain, we perform Short-Time Fourier Transform (STFT) on two audio samples with a window size of 46ms based on:

$$X(\tau, \omega) = \sum_{n=-\infty}^{n=+\infty} x[n]w[n-\tau]e^{-j\omega n} \tag{1}$$

where τ is the time axis, ω is the frequency axis, $x[n]$ is the an audio sample, and $X(\tau, \omega)$ is a complex function representing the phase and magnitude of the signal over time and frequency. Then, the spectrogram of the complex function $X(\tau, \omega)$ is computed based on:

$$spectrogram\{x[n]\}(\tau, \omega) \equiv |X(\tau, \omega)|^2 \tag{2}$$

Fig. 2 illustrates the spectra of two audio samples when a user speaks "Six" to a smartphone, and we can find following observations that can help us detect the liveness of the speaker: 1) Since the vocal cords do not vibrate during producing unvoiced voice, the prime microphone loses most information for unvoiced part, while the front microphone can capture this information; 2) For the voiced part, the prime microphone can only capture voice information at low frequency band. If the voice is from a live speaker, the differences of two spectra should contain most information of the voice except that in the low frequency band of voiced part, as shown in Fig. 5. Based on these observations, we compute the difference between two spectra and leverage its energy distribution as the



(a) User's acceleration waveform (b) Attacker's acceleration waveform

Fig. 6. Filtered acceleration waveforms from a normal user and an attacker

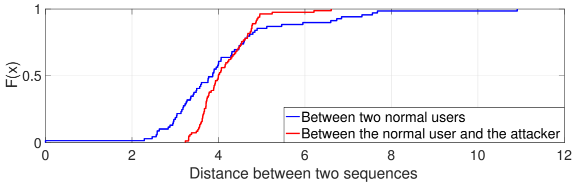


Fig. 7. CDF of distances between acceleration sequences of normal users and the attacker

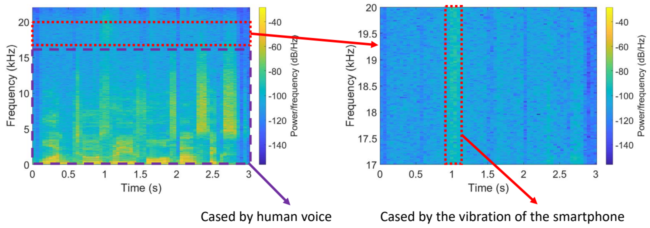


Fig. 8. The spectrum of the audio signal with random vibration injected

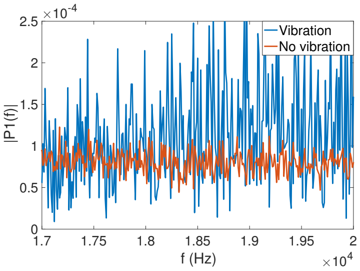


Fig. 9. Single-Sided Amplitude Spectrum w and w/o injected vibration

feature to detect the liveness of a speaker. Due to unpredicted noise and speaking volumes of the speakers, it is hard to robustly extract the shape of energy distribution. To solve this problem, we treated the spectra difference as an image, and its energy represents the color. Considering the diversity of energy distribution due to various speaking manners of different people, all energy values (pixels in the image) are used to build the classifier. To eliminate the influence of different speaking time, we resize the spectra difference (the image) and convert them to vectors. The resulted vectors are used to build a binary Support Vector Machine (SVM) with nonlinear kernel function to determine if the input spectra difference satisfies the two observations we find.

D. Motion-based approach

When a user speaks a passphrase to the smartphone in our system, there are two kinds of movements involved. First, the throat will move up and down in a low frequency. In addition, the vocal cords will vibrate in high frequency for voiced phonemes. These two movements will generate different influences on the acceleration readings in the smartphone. To understand the influences of human speaking activity on the acceleration readings, we first collect the acceleration waveforms from normal users. Then, raw acceleration data is smoothed using a moving average filter with window size of 10. Fig. 6(a) illustrates the filtered acceleration waveforms under the influences of the human speaking activity. We can see that low-frequency throat movements generate 7 significant pulses by moving up and down. Also, vocal cords vibration affects the acceleration reading in high frequency, which is shown as small spikes across the whole waveform. We further study the influence of a loudspeaker on the embedded accelerator and find that it is hard for an attacker to perform attacks using a loudspeaker. Fig. 6(b) shows the filtered acceleration waveforms under the influences of a loud speaker. We can see that the waveform contains much more significant spikes whose magnitudes are mainly within $[-0.05, 0.05]$. Dynamic Time Wrapping (DTW) is an efficient way to measure the similarity between two temporal sequences. However, it is hard to determine if an acceleration sequence is collected from a loudspeaker using only DTW algorithm. Fig. 7 shows the distributions of distances of acceleration sequence calculated by DTW between normal users and between a normal user and an attacker. We can see that two distributions are very similar. The distances between a user and and an attacker are even smaller than those between normal users in some cases. To address this issue, we select 7 features to represent an acceleration sequence: (1) Variance; (2) Minimum; (3) Maximum; (4) Mean; (5) Skewness; (6) Kurtosis; (7) Standard deviation. We select the features based on Principal component analysis (PCA) and use selected features to train a SVM-based classification model. The model is then used to determine if an acceleration signal is from a live speaker or not.

E. Random vibration-based approach

Even if our spectrum-based approach and motion-based approach can fight spoofing attacks effectively, we argue that there are stronger attackers who can hack the database and steal the voice at victim's throat. Also, we assume that the strong attacker can leverage multiple speakers and imitate human vocal system perfectly with a high cost. In this case, our spectrum-based approach and motion-based approach cannot ensure good performance. To address this problem, we further introduce a random vibration strategy so that the strong attacker cannot fool our system even if the attacker can steal the raw audio file and imitate victim's vocal system perfectly. Current smartphone operating system provides us the privilege to operate the vibration motor and define the vibration pattern. We fully leverage the vibration motor embedded in most smartphones. While recording, our system will randomly trigger the vibration moto for a given constant time t. Then, our system will detect the number of random vibrations in the received audio signals. If the number is larger than 1, the audio signal is classified as “stolen audio file” and the validation is rejected.

To effectively detect this attack, we need to locate the vibration accurately and determine what value of t. Here is a trade-off of determining the value of t. If t is too small, the intensity of the vibration may not be strong enough to be detected. If t is too large, the noise generated by the vibration will influence the original validation process and our system. Based on our experiment, $t = 100\ ms$ gives us the best performance on two smartphones. Due to the high sampling rate provided by the current microphone, we can design a robust algorithm to detect the vibration of smartphone based on the audio signal. Fig. 8 shows the spectrum of the audio signal with injected vibration with the length of 100 ms at 1 second. We can see that it is hard to detect the vibration under 15 KHz on the spectrum since the influence caused by vibration is buried by that of the human voice and background noise. However, the influence caused by smartphone’s vibration dominates the high-frequency part of the spectrum (17 KHz $\sim$ 20 KHz). Fig. 9 shows the single-sided amplitude spectrum from 17 KHz $\sim$ 20 KHz. It is clear that much more energy is in the given frequency band if there is a vibration.

Based on this insight, we design a vibration detection algorithm to locate the vibration at the frequency domain and validate the duration of each vibration. After getting the raw audio signal from the front microphone, we cut the audio sequence into frames with the equal size of 50 ms. Within each time frame, we perform STFT and calculate the sum of energy in the selected frequency band (17 KHz $\sim$ 20 KHz). If the sum of the energy is higher than a threshold τ, a vibration is detected at the current time frame. After vibration detection on all time frames, we group the frames that contain a vibration as long as they are neighbors with each other. Then, we check the length of each group. The audio is recognized as collected from a normal user if and only if there only exists one group with the length of N. Otherwise, the sequence is recognized as stolen. In our experiment, we find that in some cases the vibration motor vibrates a little bit earlier than the random starting time we generate, and the pre-start will generate a vibration in the previous vibration. So, we set the $N = 3$ and $\tau = -15300$ in our system. Since people need at least 2.2 seconds to finish a 6-digit passphrase, the possibility that an attacker can get the same vibration location of the original audio signal is less than 4.3%.

IV. Evaluation

A. Experiment methodology

Experiment setup In order to evaluate the effectiveness of our system, we build a prototype on two smartphones with different sizes (LG Nexus 5 and MOTO Nexus 6). Both of the smartphones run on Android. The smartphones are used to capture audio signals in two channels. We design a simple graphical user interface (GUI), as shown in Fig. 10, to help users collect audio signals. The application starts capturing user’s voice in two channels as soon as the user presses the button and stops data collection immediately when the user releases the button. After data collection on smartphones, audio signals are sent to a local server for further validation. The server runs on a MacBook Pro with 2.9 GHz Intel Core i5 processor and 8GB 1867 MHz DDR3 memory.

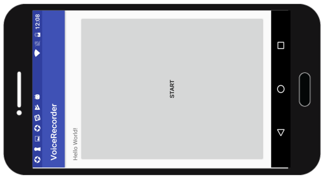


Fig. 10. A simple graphical user interface

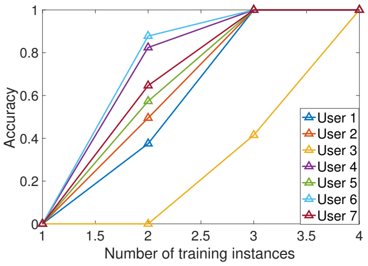


Fig. 11. Performance of spectrum-based approach

Performance Metrics 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 live speaker is correctly accepted by the system and considered as a real person. True rejection rate is defined as the rate at which an attacker is correctly rejected by the system.

TABLE I
Types of loudspeakers

Maker	Model	Number of trumpets
Willnorn	SoundPlus	2
Amazon	Echo	2

TABLE II
Users’ information

Sex	Age	Height (cm)	Average validation time (s)
Female	28	162	2.2616
Male	27	172	2.9977
Male	22	180	3.3551
Male	27	185	4.7149
Female	25	165	2.7279
Male	24	187	3.6396
Female	23	175	3.9321

B. Performance of spectrum-based approach

To evaluate the performance of our spectrum-based approach, we collect 350 raw audio waveforms from 7 different users. These 7 users include 4 males and 3 females. Each user is asked to speak to the smartphone using the same 6-digit password as we ask in Section II-D for 50 times. For each

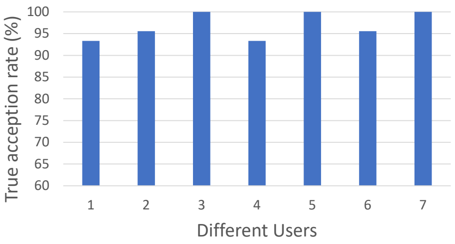


Fig. 12. Performance of motion-based approach

user, 5 audio waveforms are used as training data, and the remaining audio waveforms are used as validation data. Also, an attacker uses two loudspeakers to replay victims' voice. The speakers we use are listed in Table I. During replay attack, the relative location between the loudspeaker and the smartphone should remain the same as we do for normal users.

We can observe that spectrum-based approach can achieve 100% true acceptance rate and true rejection rate for all users. We further evaluate how many training instances we need to build a strong classification model and if we can provide good validation accuracy without collecting training instances from the new user. Therefore, we only use the audio instances collected from one user as training data and perform evaluation on all users. Fig. 11 shows the evaluation results. We can observe that, with no less than 4 training instances, our system can accurately detect both live speakers and attackers with a accuracy of 100%. Also, our spectrum-based approach does not need to collect much training data from a new user, which makes our system more practical.

C. Performance of motion-based approach

In this subsection, we evaluate the validation performance of our motion-based approach. Similarly, we collect 350 raw acceleration sequences from 7 different users. For each user, 5 acceleration sequences are used as training data, and the remaining are used as validation data. Also, 20 acceleration sequences collected from the attacker are used as negative instances. Fig. 12 illustrates the true acceptance of our motion-based approach. We can see that our system can achieve high true acceptance rate of at least 93.33% for most users and provides true rejection rate of 88.89%. To further improve the true acceptance rate, we can add more instances only collected from the new user. We argue that user can manually label wrongly predicted results, and our classification model can leverage new labeled data to build a better classification model for user 1. Experiment results show that the true acceptance rate can be improved to at least 95% after each user adds 5 more instances to the training set.

TABLE III
PERFORMANCE OF VIBRATION DETECTION

Locations	Number of TAV	Number of FAV
1	40	0
2	40	0
3	40	0
4	39	0

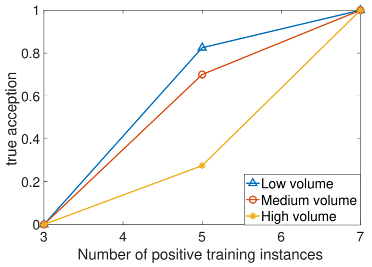


Fig. 13. Influence of background noise

D. Performance of random vibration-based approach

In this study, we investigate the performance of our random vibration-based approach when a strong adversary tries to fool our system by using the collected audio profile of the victim and imitating natural human voice using multiple speakers. First, we examine how accurately our system can detect the number of vibration in the audio signal. We let a user speak in front of our system for 20 times. During each recording process, our system generates two non-overlapped vibrations and record the ground truth. We repeat the experiment in 4 different rooms, and the results are illustrated in Table. III. The *true accepted vibration (TAV)* is the vibration generated by the human vocal and correctly detected by our algorithm. The *false accepted vibration (FAV)* is the vibration generated by the background noise but wrongly detected. We can see that our vibration detection algorithm can achieve an accuracy of 100% on detecting non-overlapped vibration for the first three rooms. The fourth location is in a kitchen where there may exist high-frequency noise produced by electrical appliances. Several time frames could be wrongly recognized as containing vibration due to the high-frequency noise, which makes the duration of 4 vibrations longer than $150ms$ and be rejected by our system. In this scenario, our system can still identify all the vibrations with an accuracy of 97.5%.

E. Influence of ambient noise

To evaluate the influence of ambient noise on spectrum-based approach, we place a loudspeaker at a distance of about 1 meter. We let the loudspeaker keep playing audio from a talk show with different volumes. For each volume, we collect 40 audio waveforms from a user. We use the same classification model used in Section IV-B. We change the number of positive instances to evaluate the true acceptance rate, and the results are shown in Fig. 13. We can see that we cannot perform validation with three positive instances when there is background noise. When we increase the number of positive instances to 5, we can get true acceptance rate of 82.5% in a low background noise environment. However, the validation performance is deficient in a noisy environment with a true acceptance rate of only 27.5%. This problem can be solved by involving more positive instances or increasing the weight of positive instances in the classification model. We can see that our system can achieve a true acceptance rate of 100% when seven positive instances are involved.

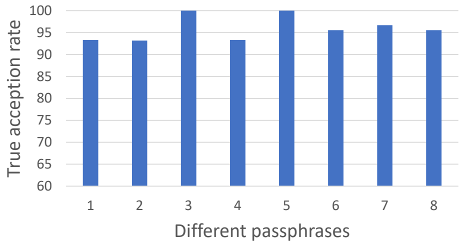


Fig. 14. Influence of different passphrases

F. Influence of different passphrases

We also conduct an experiment to show the performance for different passphrases. In our system, we select 8 different passphrases, and a user is asked to repeat each passphrase for at least 45 times. For each passphrase, 15 measurements including audio and acceleration are used for training, and the others are used for validation. We also let the attacker perform replay attack for 45 times for each passphrase using recorded victim's voice and use them as negative training and validation data. Fig. 14 shows the true acceptance rates for 8 passphrases. We can see that our system can achieve a true acceptance rate of at least 93.2% for all 8 passphrases. Also, we examine the true rejection rate of our system on the selected 8 passphrases. Experimental results show that our system can provide true rejection rate for at least 86.7%.

G. Influence of different phones

To show that our system can be implemented to any smartphone equipped with two microphones, we evaluate our system on LG Nexus 5 and LG Nexus 6. The reason we choose these two smartphones models is that the sizes of these two smartphones differ a lot. We ask a user to speak to these two smartphones for 45 times, respectively. Similarly, five measurements on each smartphone are added to the pre-trained model in Sections IV-B and IV-C, and the remaining are used as validation data. Experimental results show that our system can achieve a true acceptance rate of at least 95% on the two smartphones and get an acceptable true rejection rate of at least 88.75%.

V. DISCUSSION

A. Usability

Except for accuracy, validation time is also critical and determines the usability. We further test the time our system needs to process the raw signal and get the final validation results. Experiment results show that our method can finish the work within 500ms in all cases, which means our system can respond to the user right after the user stops recording and does not introduce too much overhead. Compared with existing works, our system does not need user's extra effort in operating the smartphone, e.g., moving the smartphone around the audio source. To further strengthen the usability of our system, we adopt the same human-computer interaction methods used by Wechat for recording, so that users can quickly get used to using our system.

B. Limitations

Our system involves a limited number of participants, and all users are university students. To better understand the performance of our system, it will be necessary to engage more participants with a more diverse background. Also, the experiments are conducted within one month. Considering that human behavior and habits may change, a long-term evaluation can be conducted. Moreover, in our system, the duration of each random vibration is set to $100ms$ to get enough vibration intensity. However, the longer the random vibration is, the more likely the attacker can get the similar vibration location. The current Android operating system does not allow for changing the power of vibration. If smartphone operation system can release the permission on adjusting the power of vibration in the future, the duration of each random vibration can be further shortened to a significant degree in our system, so that it is much harder for the attacker to get the same vibration location.

VI. RELATED WORK

A. Voice-based Mobile Applications

With advances in modern smartphones, voice-based mobile applications, i.e., mobile apps, have grown in popularity as these applications provide an intrinsically efficient, comfortable interaction interface to users. These existing voice-based mobile applications can be divided into two categories based on its functionalities: i) voice communication ii) voice control. For the first category, we have voice over IP (VoIP) apps, by which people can make a voice call to anyone using the Internet (e.g., Skype, Google Voice). In addition, many voice instant messenger mobile apps have been developed in recent years, such as WeChat, WhatsApp, TalkBox, Skout, and iMessage. These apps allow users to record short voice messages and directly send them to others. Hence, this offers opportunities to attackers who are able to launch a voice-spoofing attack by imitating a victims voice, tone, and speaking style. This attack could harm victims reputation, safety, and property. The attacker could scam victims friends and family through fake phone calls and leave fake voice messages, etc.

B. Automatic Speaker Verification (ASV) System

An automatic speaker verification system is able to accept or reject a speech sample submitted by a user for claiming certain identity [18]. Recently, the development of ASV systems has made a major progress as they are widely adopted by smartphones and online commerces [8, 11]. Existing ASV systems are divided into two types: *text-dependent* and *text-independent*. Text-independent ASV systems are able to accept arbitrary utterances, i.e., different speaking habits and languages from speakers [3]. As a matter of fact, the text-dependent ASV is widely selected for authentication applications since it provides higher recognition accuracy with fewer required utterances. The current practice of building an ASV system involves two processes: offline training and runtime verification. During the offline training phase, the ASV system uses several speech samples provided by the genuine speaker to extract certain spectral, prosodic [1, 16], or other high-level features [5, 12] and uses them to create a speaker model. Then, in the runtime verification phase, the ASV system uses the trained speaker model to verify the incoming voice.

C. Voice-Spoofing Attacks

The voice-spoofing attacks aim to break the biometric identification of the victim. It can be divided into two categories: voice replay attack and voice synthesis or conversion attack. [19] shows that an attacker can overcome text-dependent ASV systems by concatenating speech samples from multiple short voice segments of the target speaker. Due to the simplicity of voice replay attacks, a few research papers have been published in developing relay attack countermeasures [19–21]. However, all these countermeasure systems suffer high false acceptance rate (FAR) compared to respective baselines. In [2], the authors demonstrate the vulnerabilities of ASV systems for voice synthesis attack (generate artificial speech from text input). [17] proposes the voice conversion attack in which the attacker converts the spectral and prosody features of his or her own speech and makes it resembles to the victim's speech. To detect voice synthesis and voice conversion attack, [23] exploits artifacts introduced by the vocoder to discriminate converted speech from original speech.

VII. Conclusion

In this paper, we propose a robust software-based voice spoofing defense system, which is tailored for mobile platforms and can be easily integrated with existing mobile applications. We propose three approaches based on leveraging the audio spectrum pattern, motion of the human vocal system, and the functionality of vibration motor. Experimental results show that our spectrum-based approach can achieve a 100% true acceptance and rejection rates. Our motion-based approach can achieve mean accuracy of 96.8% and mean true rejection rate of 88.89%. Our random vibration-based approach can detection and location the vibration with an accuracy of at least 97.5%. By combining the three approaches we proposed, our system can detect a live speaker with a mean accuracy of 94.38% and detect an attacker with a mean accuracy of 88.89%.

Acknowledgement

This research was supported in part by NSF grants CNS 1757533, CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, and IIP 1439672.

References

[1] A. G. Adami, R. Mihaescu, D. A. Reynolds, and J. J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *Proc. of ICASSP*, volume 4, pages IV–788. IEEE, 2003.

[2] F. Alegre, R. Vipperla, N. Evans, and B. Fauve. On the vulnerability of automatic speaker recognition to spoofing attacks with artificial signals. In *Proc. of EUSIPCO*, 2012.

[3] J. P. Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

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

[5] G. Doddington. Speaker recognition based on idiolectal differences between speakers. In *Proc. of EUROSPEECH*, 2001.

[6] N. Evans, J. Yamagishi, and T. Kinnunen. Spoofing and countermeasures for speaker verification: a need for standard corpora, protocols and metrics. *IEEE Signal Processing Society Speech and Language Technical Committee Newsletter*, pages 2013–05, 2013.

[7] A. Janicki, F. Alegre, and N. Evans. An assessment of automatic speaker verification vulnerabilities to replay spoofing attacks. *Security and Communication Networks*, 9(15):3030–3044, 2016.

[8] K. A. Lee, B. Ma, and H. Li. Speaker verification makes its debut in smartphone. *IEEE signal processing society speech and language technical committee newsletter*, 2013.

[9] K. B. Lee and R. A. Grice. The design and development of user interfaces for voice application in mobile devices. In *Proc. of ProComm*, pages 308–320. IEEE, 2006.

[10] 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.

[11] Nuance. Nuance vocal password. http://www.nuance.com/, 2013.

[12] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, et al. The supersid project: Exploiting high-level information for high-accuracy speaker recognition. In *Proc. of ICASSP*, volume 4, pages IV–784. IEEE, 2003.

[13] 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.

[14] SayPay. http://saypaytechnologies.com/.

[15] 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.

[16] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke. Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, 46(3-4):455–472, 2005.

[17] Y. Stylianou. Voice transformation: a survey. In *Proc. of ICASSP 2009*, pages 3585–3588. IEEE, 2009.

[18] R. Togneri and D. Pullella. An overview of speaker identification: Accuracy and robustness issues. *IEEE circuits and systems magazine*, 11(2):23–61, 2011.

[19] 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.

[20] J. Villalba and E. Lleida. Preventing replay attacks on speaker verification systems. In *Proc. of ICCST*, pages 1–8. IEEE, 2011.

[21] Z.-F. Wang, G. Wei, and Q.-H. He. Channel pattern noise based playback attack detection algorithm for speaker recognition. In *Proc. of ICMLC*, volume 4, pages 1708–1713. IEEE, 2011.

[22] WeChat. Voiceprint. http://thenextweb.com/apps/2015/03/25/wechat-on-ios-now-lets-you-log-in-using-just-your-voice/.

[23] Z. Wu, E. S. Chng, and H. Li. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In *Proc. of INTERSPEECH*, 2012.

[24] 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.

[25] L. Zhang, S. Tan, and J. Yang. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proc. of CCS*, pages 57–71. ACM, 2017.

[26] L. Zhang, S. Tan, J. Yang, and Y. Chen. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proc. of CCS*, pages 1080–1091. ACM, 2016.