Defeating Hidden Audio Channel Attacks on Voice Assistants via Audio-Induced Surface Vibrations

Chen Wang
WINLAB, Rutgers University
New Brunswick, NJ, USA
chenwang@winlab.rutgers.edu

S Abhishek Anand
University of Alabama at Birmingham
Birmingham, AL, USA
anandab@uab.edu

Jian Liu
WINLAB, Rutgers University
New Brunswick, NJ, USA
jianliu@winlab.rutgers.edu

Payton Walker
University of Alabama at Birmingham
Birmingham, AL, USA
prw0007@uab.edu

Yingying Chen
WINLAB, Rutgers University
New Brunswick, NJ, USA
yingche@scarletmail.rutgers.edu

Nitesh Saxena
University of Alabama at Birmingham
Birmingham, AL, USA
saxena@uab.edu

ABSTRACT

Voice access technologies are widely adopted in mobile devices and voice assistant systems as a convenient way of user interaction. Recent studies have demonstrated a potentially serious vulnerability of the existing voice interfaces on these systems to "hidden voice commands". This attack uses synthetically rendered adversarial sounds embedded within a voice command to trick the speech recognition process into executing malicious commands, without being noticed by legitimate users.

In this paper, we employ low-cost motion sensors, in a novel way, to detect these hidden voice commands. In particular, our proposed system extracts and examines the unique audio signatures of the issued voice commands in the vibration domain. We show that such signatures of normal commands vs. synthetic hidden voice commands are distinctive, leading to the detection of the attacks. The proposed system, which benefits from a speaker-motion sensor setup, can be easily deployed on smartphones by reusing existing on-board motion sensors or utilizing a cloud service that provides the relevant setup environment. The system is based on the premise that while the crafted audio features of the hidden voice commands may fool an authentication system in the audio domain, their unique audio-induced surface vibrations captured by the motion sensor are hard to forge. Our proposed system creates a harder challenge for the attacker as now it has to forge the acoustic features in both the audio and vibration domains, simultaneously. We extract the time and frequency domain statistical features, and the acoustic features (e.g., chroma vectors and MFCCs) from the motion sensor data and use learning-based methods for uniquely determining both normal commands and hidden voice commands. The results show that our system can detect hidden voice commands vs. normal commands with 99.9% accuracy by simply using the low-cost motion sensors that have very low sampling frequencies.

ACSAC '19, December 9–13, 2019, San Juan, PR, USA

ACM ISBN 978-1-4503-7628-0/19/12…$15.00
https://doi.org/10.1145/3359789.3359830

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; **Systems security**.

KEYWORDS

voice access; hidden voice command detection; motion sensor; surface vibrations

ACM Reference Format:
Chen Wang, S Abhishek Anand, Jian Liu, Payton Walker, Yingying Chen, and Nitesh Saxena. 2019. Defeating Hidden Audio Channel Attacks on Voice Assistants via Audio-Induced Surface Vibrations. In *2019 Annual Computer Security Applications Conference (ACSAC '19), December 9–13, 2019, San Juan, PR, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3359789.3359830

1 INTRODUCTION

Voice controllable systems (VCS) have become increasingly popular in recent years. They provide a convenient way of meeting a user's various daily needs through voice commands and taking actions when necessary, such as access

control, personal schedule/memo inquiry, smart home appliance control, online purchases, etc. Due to their convenience, these systems have already been integrated into various platforms including mobile phones (e.g., Siri and Google Now), stand-alone assistants (e.g., Amazon Echo Dot, Google Home and HomePod), and even smart appliances (e.g., smart TVs and smart refrigerators). A market research report suggests that the global voice assistant market is expected to grow at approximately $7.8 billion by 2023, at 39.27% of Compound Annual Growth Rate (CAGR) between 2017 and 2023 [3]. With the ever-growing deployment, the VCSs' security vulnerabilities become an increasing public concern.

Due to the open propagation properties of sound, voice commands as an input solution have some fundamental vulnerabilities. An emerging class of potentially devastating attacks against VCSs is known as *hidden command attacks* [9, 28], which are recognizable to the VCS devices but are incomprehensible to humans. These hidden voice commands are generated by iteratively shaping their audio features to meet the requirements for being understandable to VCSs, but hard to be perceived by the users [10]. The malicious hidden voice commands could be played covertly by an adversary, in the vicinity of the victim, to make the victim's device inadvertently obey the adversary's command. For example, the adversary could play hidden voice commands (e.g., *"browse evil.com"*, *"call 911"*) via a loudspeaker to trigger actions on the victim's VCS devices to browse phishing sites or make spam calls to the 911 emergency center. The hidden voice commands could also be embedded in the audio tracks of regular media (e.g., Youtube videos, radios or TVs), potentially controlling many VCS devices exposed to that media [31].To ensure the successful deployment of VCS, it is thus paramount to combat the *hidden voice command* attack.

In this paper, we develop a defense system that could be integrated with VCSs by utilizing the vibration signatures of the voice command to uniquely determine if the issued command is provided by a human user or falls in the category of hidden voice commands. We notice that many VCS devices such as smartphones and standalone voice assistant systems (e.g., HomePod[1]) are already equipped with motion sensors, which could capture vibration signals caused by voices. It has been shown that the features of speech, captured in the vibration domain, have enough information to perform speaker identification [22]. However, Anand et al. [5] showed that live human speech is unable to significantly impact the on-board motion sensors of a smartphone (especially accelerometer) via aerial medium, and in order to have a noticeable impact, the speech generating device need to share a common solid surface with the motion sensors. We thus design two modes for our defense mechanism. First, we let the user's own device play back the received voice command and use the device's built-in sensors to measure the vibration signatures for verifying the voice commands, which is referred as the *frontend mode*. Alternatively, the user can also choose to play back the voice commands in the *backend mode* via cloud services, when the user's VCS device does not have motion sensors or playing back the command may be disturbing to the user. In this mode, a low-cost device (e.g., a loudspeaker with an on-board motion sensor) in the cloud plays back the user's voice command. Moreover, our system is able to verify voice commands with *partial playback* (e.g., 0.5 second and even less than a single word) and is easy to integrate with the VCS without causing additional delay.

Realizing such a system that seeks to use vibration signals, measured by the low-sampling-rate and low-fidelity motion sensors, to discriminate live human vs. hidden voice commands is challenging. To address this problem, our system derives unique features from the vibration signals to capture the inherent vibration signatures of voice commands. In particular, we derive the temporal and frequency statistical features aiming to achieve a higher tolerance of the errors induced by low-fidelity motion sensors. We further seek to obtain acoustic features in the low-frequency vibration domain relying on MFCC and chroma vectors so as to capture the speech characteristics embedded in the voice commands. To distinguish hidden voice commands from normal commands, a feature selection strategy is developed to find a set of features that are more discriminative to the differences between the two types of commands and are also relatively more independent from various people's voices and command contexts. This process is aided by empirically analyzing a set of pre-collected voice commands that are mixed with various hidden voice commands. Based on the selected vibration features, our system could effectively detect the existence of hidden voice commands using machine-learning-based methods with only limited training efforts. In addition, motion sensors are affected by ambient vibration noises (e.g., surface vibrations caused by environmental noises and people walking around). Our system calibrates the input motion sensor data by removing the mechanical noises with a high-pass filter and identifies the data segment containing the commands by adaptively examining the energy levels of motion data variance.

Why Vibration? Existing studies usually defend against these hidden voice command attacks using *audio domain* features, such as speech vocal features [12, 25, 30]. Carlini et al. [9] proposed audio-feature-based classification as a possible countermeasure to detect the hidden voice commands. However, features in the audio domain have been shown to be susceptible to duplication by voice synthesis

[1]The on-board motion sensor is to detect when HomePod is moved to start its sound quality re-calibration [17].

attacks [23]. An adversary could iteratively modify a voice command to exhibit all of the required features desired for recognition by a voice controllable system, while remaining undetectable to human listeners. This vulnerability of the audio domain features prompted us to look for the features in the vibration domain. Different from audio, the vibration domain features are hard to imitate for the following reasons: (1) the vibration signatures of an audio captured by the motion sensor is unique and new, shown in the form of distinct amplitudes, frequencies and aliased signals, which are hard to forge or imitate from an audio signal in software (shown in Section 4). (2) any two sounds having similar audio features could result in distinct vibration features because the relationship between audio and vibration is non-linear (shown in Section 5.2); and (3) the resulting vibration response is also associated to the physical vibration properties of the device and the specifications of the motion sensor. Thus, the vibration domain approach can work in conjunction with the audio domain approach to more effectively detect the hidden voice commands. The adversary would have to mimic both audio features and vibration features to maintain the "hidden voice command" characteristics of the generated voice command.

We summarize our contributions as follows.

- **Detection of Hidden Voice Commands using Vibration Domain Features of Speech:** We propose a solution in the *vibration domain* to detect the existence of hidden voice commands that are incomprehensible to a human listener but recognized by the VCSs. Our method uses unique speech features, found in the vibration domain including statistical time/frequency features and acoustic features, to distinguish normal commands from hidden voice commands. These vibration features are able to verify the voice commands with even partial voice commands (e.g., 0.5 second).
- **Design and Implementation of the Proposed Defense System:** We design a novel classification-based defense system, as the core of our proposed detection approach for distinguishing between normal commands and hidden voice commands. We implement the proposed defense system for the scenarios where an attacker may launch hidden voice commands externally via a loudspeaker or internally on the victim's smartphone. We use the inbuilt accelerometer of the victim's device (e.g., smartphone or stand-alone assistant) or a low-cost device in the cloud to measure the vibrations generated by these voice commands, when played back completely or partially via the speaker in the victim's device or the cloud device, respectively. We then extract and select highly discriminative vibration features and perform machine learning-based classification to detect the hidden voice commands.
- **Evaluation of the Proposed Defense:** We evaluate the proposed defense system by classifying normal command samples and corresponding hidden voice commands. We perform the experiment under both *frontend playback* and *backend playback*. Our results show that the proposed defense system is able to detect these hidden voice commands with 99.9% accuracy. These accuracies can be deemed significant enough to believe that the proposed defense has the potential to be successfully deployed against the hidden voice commands.

2 RELATED WORK

Security of Voice Controllable Systems. Due to the popularity of recent VCS services, such as Siri, Cortana, Google Now and Alexa, many studies have focused on the security issues of these systems [9, 15, 21, 23, 28, 32]. For instance, researchers have shown that the intentional electromagnetic interference on headphone cables can be used to inject commands into voice assistants [21]. The attacker could spoof the system by using either voice morphing techniques [23] or the permission bypassing attack method [15]. More recently, DolphinAttack [32] used the non-linearity of microphones to modulate voice commands on ultrasonic carriers to launch inaudible voice command attacks. CommanderSong [31] stealthily embedded voice commands into songs to launch attacks. Additionally, existing work [9, 28] demonstrates *hidden voice commands* by using the mangled audio commands which are incomprehensible to humans but can be recognized by the VCS to launch attacks. The authors also proposed an audio-based human/machine classifier to detect such an attack [9]. However, the voice synthesis attacks are able to forge the similar audio domain features to pass the system [23].

Voice Authentication for Virtual Voice Assistants. To defend against various attacks (e.g., replay and impersonation attacks), most voice authentication schemes mainly use advanced speaker models (e.g., Gaussian mixture model-universal background model (GMM-UBM) [4], i-vector models [18, 19]), and various speech vocal features [12, 25, 30]. However, the aforementioned audio-based approaches are still vulnerable if an adversary has the full knowledge of the system's model as they are solely based on the properties of speech itself. Thus, a multi-modality authentication framework to provide enhanced security is highly desirable. Additionally, VoiceLive [35] and VoiceGesture[34] exploited the geometrical information and dynamic acoustic characteristics derived from the received sound to perform liveness detection. 2MA [8] took advantage of the presence of multiple microphones to localize and authenticate the source of a command. Moreover, Feng *et al.* [16] developed a user verification system on wearable devices (e.g., eyeglasses) based

on the collected body-surface vibrations to defend against various speech attacks. However, these approaches either require the phone to be held closely to the speaker's mouth or require the user to wear eyeglasses while operating the systems, which largely restricts their application scenarios.

Speech Effect on Motion Sensor. Existing studies have shown that the MEMS sensor attributes and structures could be easily interfered by ambient sound and noise [11, 13, 14]. WALNUT [27] modeled the physics of acoustic injection attacks on MEMS accelerometers, and showed that the outputs of sensors are subjected to the acoustic interference. In addition, researchers showed that embedded MEMS motion sensors have the possibility of detecting *hotwords* [33] or even recognizing speech information [5, 22]. Moreover, Gyrophone [22] showed that gyroscope can be used to measure acoustic signals from a loudspeaker to reveal speaker information (e.g., gender and identity). A more recent work, Speechless [5] went a step further to evaluate the necessary conditions and scenarios for the speaker information leakage problem. It showed that the recorded effect on the motion sensors is from conductive vibrations through a shared surface between the speaker and sensor.

Since an attacker could easily tamper the human speech pattern in the *audio domain* to fool the system, in this paper, we take a different approach by using the unique speech features captured by motion sensors to defend against hidden voice commands [9, 28]. Particularly, we use the device's inbuilt motion sensor to pick up minute device-body vibrations to record the unique imprint of acoustic vibrations. Unlike the features in the audio domain, vibration features of the user's speech that are associated with both speech and device's airborne structure, are unique and hard to be imitated or synthesized by a spoofing attack. Our approach can either work as a stand-alone authentication mechanism or be seamlessly integrated with existing voice authentication systems, forming a two-modality or multi-modality authentication protocol.

3 APPROACH OVERVIEW

3.1 Background on Hidden Voice Commands

A typical speech recognition system usually requires four steps to recognize each voice command: pre-processing, feature extraction, model-based recognition and post-processing. Pre-processing contains speech/non-speech segmentation which removes background noise causing insertions of phonemes or words into the recognition result. Feature extraction extracts acoustic observations, Mel-frequency cepstral coefficients (MFCC) [20, 29], over time frames of uniform length. In the model-based recognition phase, the system uses the acoustic models, such as Hidden Markov

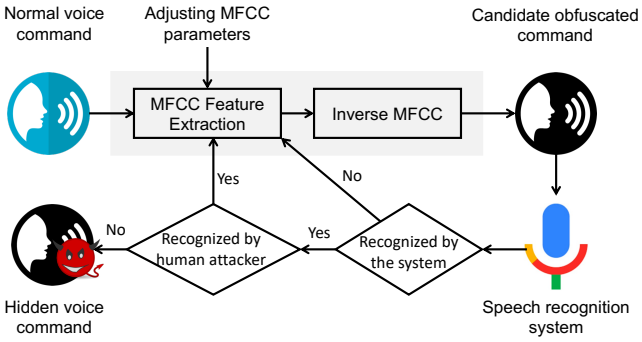


Figure 1: Workflow of generating hidden voice command [9] from a normal voice command.

Models (HMM) and recurrent neural networks (RNNs), to predict a sequence of words that are most likely to match the extracted acoustic features. In the latter, the system employs additional sources of information (e.g., grammar rules) to improve the recognition accuracy.

In order to generate hidden voice commands to spoof voice recognition system, an adversary could use general acoustic processing methods to generate obfuscated commands with acoustic features (e.g., MFCC) that can be correctly recognized by the system [9]. As shown in Figure 1, the adversary first extracts commonly used acoustic features (i.e., MFCC) from a normal command, and then performs inverse MFCC to convert the extracted MFCC features back to an audio sample. Through this step, the generated audio sample only contains audio features that are used in the speech recognition system while disregarding other features that might be helpful for human's comprehension.

The MFCC parameters in the MFCC feature extraction determine the resolution of the extracted feature, which play an important role on the generated audio sample's capability of being recognized by a human/machine. The parameters include the number of cepstral coefficients, the number of warped spectral bands, the length and stride of the sliding window. The higher dimension of the MFCC features could make the generated audio sample to have a higher probability of being recognized, while lower feature dimension could make the generated audio file more obfuscated. To use the perception gap between human and machine to create hidden voice commands, the adversary needs to iteratively adjust these MFCC parameters to check whether the generated commands could be recognized by the system but hardly recognized by the victim. Through such an iterative testing, the adversary could find an optimal set of parameters to make the generated command recognizable by machines while remaining incomprehensible to humans.

3.2 Attack Model

We target the defense against hidden voice commands. We assume the adversary does not have the capability to compromise the voice assistant systems and can only use hidden voice commands to access the system. The hidden voice commands can be embedded in the audio tracks of regular media (e.g., Youtube videos and Podcast) and played by the target device's built-in speaker to deliver the hidden voice command (*internal attack*). Moreover, the hidden voice commands can also be played by an adversary via a loudspeaker near the target device to launch the attack (*external attack*).

In the first scenario, the attacker tricks the victim into playing the audio with hidden voice commands on their own device (e.g., smartphone and standalone assistant). We term this scenario as *internal attack* because the targeted device itself plays the hidden voice commands (in contrast to an external loudspeaker). This scenario requires the attacker into either fooling the victim into playing the audio containing hidden voice commands or by using a malicious application that can play the audio with hidden voice commands, while being inconspicuous to the victim. To fool the victim into playing the audio, the attacker could send an audio or a video file to the victim under the pretense of a benign message. A malicious website could embed auto-playing videos on their web page that could play (unprompted) the hidden voice commands while navigating the web page by the victim. Similarly, a malicious application could play audio with hidden voice commands under the disguise of a gaming application. Diao et al. [15] designed an attack termed "GVS-Attack" against Google Voice Assistant, that launched the voice assistant (using VoicEmployer malware) and then played standard voice commands. These voice commands were then faithfully executed by the voice assistant.

The other scenario, where the attacker uses an external loudspeaker to propagate these hidden voice commands can be termed as *external attack* where the attacker has the capability to exploit any nearby loudspeaker that is in the vicinity of a single or multiple targeted devices. For example, an unsuspecting victim could be sitting with their smartphone, in a coffee shop where music is being played through loudspeakers. The attacker could connect to the audio system of the coffee shop to play his own curated music that has malicious voice commands embedded in it. The attacker could also be in physical proximity of the user and play an audio with hidden voice commands through his own loudspeaker device.

3.3 System Overview

The basic idea of our system is to analyze the speech features that are captured in the *vibration domain* to detect hidden voice commands. Unlike audio-based voice command authentication techniques, vibration domain features are much harder to forge as the vibrations captured by the device's motion sensors are nonlinear responses to the sound and are affected not only by the played back voice commands but also by the physical vibration properties of the device itself. Such vibration domain features can be used as a stand-alone command authentication mechanism as well as multi-modality authentication protocol in conjunction with audio-based attack defense. Particularly, our system is triggered by the "wake word" received by the microphone of the VCS device (e.g., a smartphone or a standalone voice assistant device). Then the received voice command (either from an external loudspeaker or the built-in speaker) that follows after the wake word could be played back through two alternative ways based on the user's preference. In the *frontend playback*, the system directly plays back the received voice commands using the user's own device, and the built-in motion sensors (i.e., in a smartphone) or the on-board motion sensors (i.e., on a standalone VCS device) record the sound vibrations. From the user's view, this playback is the confirmation of their voice command, while from our design perspective, this is an assurance of the VCS security in coping with hidden voice commands. In the *backend palyback*, an alternative option for the user, our system plays back the received voice command through a remote loudspeaker in the cloud service, and the on-board motion sensors of the cloud loudspeaker to record the resulted sound vibrations. The recorded motion sensor data is fed as input to the system to detect the hidden voice commands. The entire process is simultaneous with the command context processing, which is 2 seconds for Google and 4 seconds for Siri service, for instance [6]. Thus, there would be no additional delay required by our system.

The flow of our proposed system is shown in Figure 2. Particularly, our system first performs *Vibration Data Calibration* including vibration noise removal and voice command segmentation to remove the noises caused by the mechanical vibrations and obtain the precise vibration data segment of the voice command. Based on the calibrated voice command segment, *Vibration Feature Derivation* derives the unique vibration features including the time and frequency statistical features (e.g., mean, standard deviation, energy) and the speech features (e.g., MFCCs and chroma vectors). We found these vibration features are effective to capture the unique statistical traits and acoustic characteristics in the low frequency range of the motion sensor data (e.g., 200Hz). The system further performs *Vibration Feature Selection* through feature normalization and statistical analysis to identify a subset of features. The selected features exhibit more discriminative patterns between the normal commands and hidden voice commands and are relatively independent from the various people's voices and command contexts.

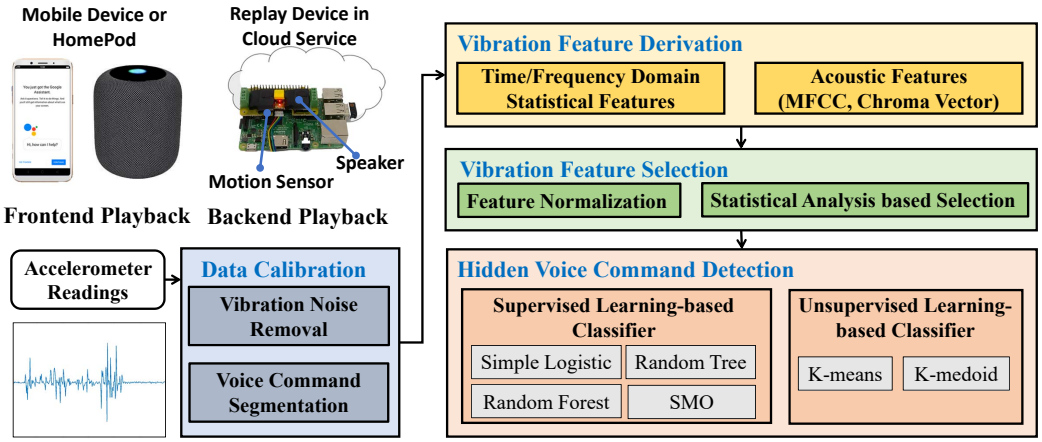


Figure 2: Overview of the proposed system.

Hidden Voice Command Detection component then leverages either a supervised learning-based classifier or an unsupervised learning-based classifier to detect the hidden voice command based on their unique vibration features. In particular, the supervised learning-based classifier, such as Simple Logistic, Sequential Minimal Optimization (SMO), Random Forest and Random Tree can effectively distinguish the two types of sounds based on the acoustic profile trained with labeled voice command samples. In comparison, the unsupervised learning methods such as K-means and K-medoid do not require much training effort and directly divide the two types of the sound into two clusters based on their inherent speech characteristics shown in the vibration domain without any labeled inputs. When verifying the input voice command, the unsupervised learning model calculates the Euclidean distance of a vibration feature vector to the normal command cluster centroid and applies a threshold-based methods to make the decision. The hidden voice command is identified if the Euclidean distance is greater than the threshold.

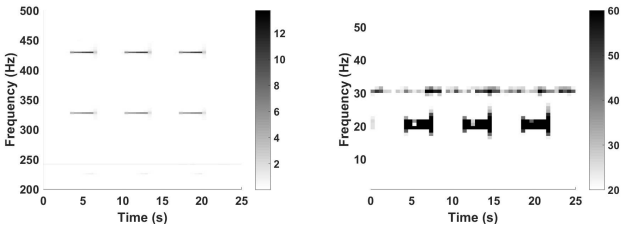
4 PREMISE & FEASIBILITY ANALYSIS

4.1 Capturing Voice Using Motion Sensors

Motion sensors are typically used to measure the movement of an object in a given direction (accelerometer) or its rotation around an axis (gyroscope). Many commodity devices (e.g., smartphones) are equipped with a miniaturized version of these motion sensors (i.e., MEMS motion sensors). An audio signal can be considered as a vibration of air molecules with a vibrating frequency that lies within the human audible range (20 Hz - 20 kHz). Anand et al. [5] showed that the vibrating air molecules, forming the human speech, were unable to affect the MEMS motion sensors, especially the accelerometer, of a smartphone. The human speech can however invoke a response in the MEMS motion sensors, if it is replayed via a loudspeaker in the vicinity of the smartphone, when they share a solid surface with it [5, 22]. This is because the vibrations of the inbuilt diaphragm of a loudspeaker propagate along the shared surface towards the motion sensors to which they transfer their vibrations. However, the motion sensors on these devices are limited to a very low sampling rate (e.g., approximately 200 Hz) and can only capture limited speech information. Since the fundamental frequency of a male voice lies between 85 to 180 Hz and that of a female voice lies between 165 to 255 Hz [7, 26], these low sampling rate motion sensors are still able to capture some features of the fundamental frequency contained in the speech signal. Recent studies [22, 33] have shown that the fundamental frequency information captured by the gyroscope and the accelerometer on smartphones could be further used to perform speaker classification and hot-word detection with a sufficiently high degree of accuracy. In this work, we leverage the speech vibrations that can be captured by the motion sensors to distinguish hidden voice commands from normal commands. We find that the mobile devices' built-in motion sensors and the standalone VCS devices' on-board motion sensors can both capture the speech vibrations of their played back speech.

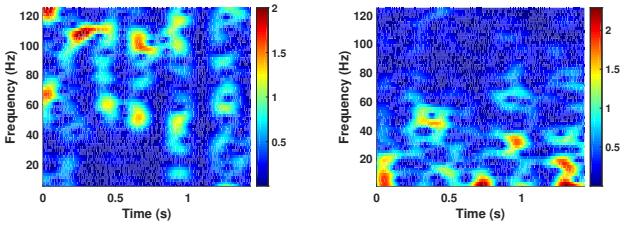
4.2 Nonlinear Vibration Responses

When using the MEMS motion sensors of the VCS devices to capture speech vibrations of an audio, their low sampling rate (e.g., 200Hz) could lead to aliased vibration signals. The signal aliasing is a phenomenon when different frequencies of an



(a) Played audio (b) Vibration responses

Figure 3: Aliased responses of a acoustic signal in vibration domain.



(a) Down-sampled mic data (b) Accelerometer data

Figure 4: Spectrogram comparison between the accelerometer data and the down-sampled microphone data (illustrated with the command "show facebook.com").

original signal are overlapped after sampling (e.g., by motion sensors). In particular, the relationship between the vibration data frequency f_{alias} and its original audio frequency f can be expressed as

$$f_{alias} = |f - Nf_s|, N \in Z, \quad (1)$$

where f_s is the sampling rate of the motion sensor. This equation indicates that the vibration responses of an audio are a nonlinear transformation of the audio. Moreover, such transformation from audio to vibration is irreversible. To illustrate the signal aliasing in the vibration responses, we conduct an experiment by using a loudspeaker to play a dual-frequency sound (i.e., 420Hz and 320Hz) for three times and an on-board motion sensor with a sampling rate of 100Hz to pick up the sound vibrations. As shown in Figure 3, the two frequency components of the audio only generate a single frequency response (around 20Hz) on the motion sensors, which complies with the nonlinear relationship of Equation 1.

4.3 Distinct Vibration Domain

It is important to note that knowing the nonlinear relationship (as Equation 1) between the vibration readings and the

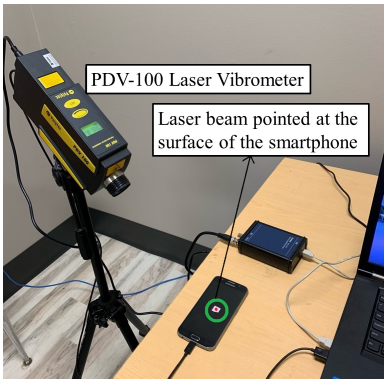
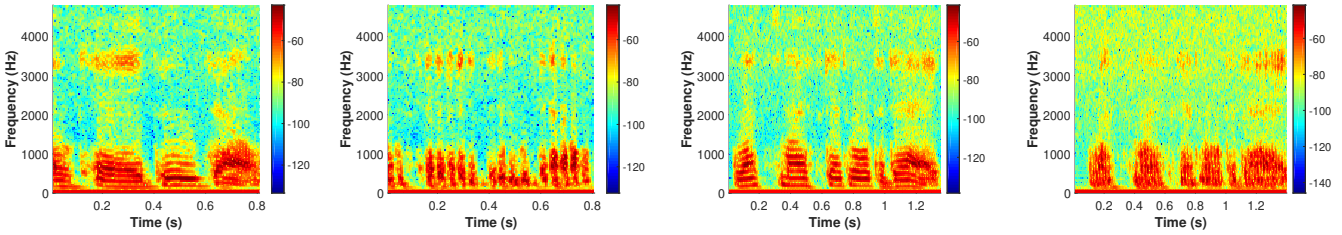


Figure 5: The experimental setup for surface vibration measurement using laser vibrometer.

original sound does not mean that one could forge or simulate the vibration responses by down-sampling the audio for generating an aliased signal. This is because besides signal aliasing, the accelerometer also generates distinct responses to the sound in the form of unique amplitudes and frequencies. As shown in Figure 4, the microphone data of the voice command "show facebook.com" is down-sampled to compare with the accelerometer data under the same sampling rate. It is clear that the descriptions of the voice command in the two domains are distinct. In particular, for the same frequency point, the two sensing modalities show very different amplitudes. Thus, we believe the accelerometer, as a different sensing modality, brings in new descriptions about the voice command, which could work with the traditional audio domain to describe the voice command's signatures in two domains. Moreover, it would become much harder for an adversary to forge the signatures of the voice command as they would need to cheat two different domains simultaneously. This subject is further explained in more detail in Section 5.2.

4.4 Observing Speech Vibrations

To further confirm the presence of the speech vibrations caused by the VCS device's speaker and their differences between the hidden voice commands and normal commands, we use a powerful laser vibrometer to capture the speech vibrations in high frequency. In particular, we use a PDV-100 Portable Digital Laser Vibrometer [1] to measure the speech vibrations generated in the body of a smartphone (Samsung Galaxy S6), as shown in Figure 5, during the playback of the voice commands by the smartphone's loudspeakers. The PDV-100 vibrometer points to the screen of the phone and can measure the vibration's frequencies up to 22 kHz with a high vibrational velocity resolution of 0.02μm/s. In Figure 6, we compare the power spectrum of the vibrations generated in the smartphone's surface, when a normal command and a corresponding hidden voice command are played through its



(a) Normal command "Ok, Google"

(b) Hidden voice command "Ok, Google"

(c) Normal command "What's my schedule today?"

(d) Hidden voice command "What's my schedule today?"

Figure 6: The power spectrum of the surface vibrations of normal commands and the corresponding hidden voice commands, as measured by a laser vibrometer pointed at the smartphone.

loudspeakers. The high sampling rate of the laser vibrometer allows us to identify the differences between the features of the normal commands and the hidden voice commands that will later help in distinguishing them. When we compare the power spectrum of Figure 6(a) and (b), we notice that the human voice spectrum's shape is not preserved in the corresponding hidden voice command even though it is successful at preserving individual word boundaries and approximating the frequency distribution. Similar behavior is observed for a different voice command in Figure 6(c) and (d). This observation supports that hidden voice commands fail to carry over the speech features in the vibration domain, when they attempt to mimic the speech features of a normal command in the audio domain.

5 SYSTEM DESIGN

5.1 Vibration Data Calibration

As introduced in Section 4, the accelerometer is capable of capturing a large portion of human voice frequencies, but they show the responses in a low frequency range due to their low sampling rate (e.g., 200Hz). In order to extract more precise vibration responses to the voice commands using the limited motion sensor information, we need to remove the vibration noises that come from the surrounding environment, such as the mechanical vibrations. In particular, we apply a high-pass filter with a cutoff frequency of 20Hz to process the accelerometer data, which removes most of the background mechanical vibrations. Moreover, before extracting the unique vibration features, we need to identify the precise segment containing the voice command. We apply a sliding window-based variance analysis method to find the starting point and ending point of the voice command to precisely segment it from the accelerometer reading. We further normalize the sound amplitudes to remove the differences caused by various sound volumes. We next extract unique vibration features based on the calibrated vibration data to analyze the hidden voice commands and normal commands in vibration domain.

5.2 Extracting Unique Vibration Features from Voice Commands

In this work, we derive the statistical features of captured vibrations in both the time and frequency domains and extract the acoustic features such as MFCCs and chroma vectors.

Statistical Features in Time and Frequency Domains. As the accelerometer data is usually used for analyzing people's various activities such as walking, running and sitting, we start by examining the activity related features as the candidate features, which have been shown to be highly correlated with the human behaviors [24]. Moreover, the accelerometer records the vibrations in three axes, which further provides the spatial information to describe the received acoustic signals by considering the vibration directions. We thus derive the features for each axis respectively. In particular, in the time domain, we derive the *maximum*, *minimum*, *variance*, *standard deviation*, *range*, *skewness*, *first quantile*, *second quantile*, *third quantile* and *kurtosis*. Moreover, we derive *absolute area* (i.e., the area under the absolute values of the accelerometer readings), *mean crossing rate* (i.e., the ratio of the number of times the signal crosses the mean value over the command segment length), signal dispersion (i.e., distance between the third quantile and the first quantile) and *absolute area sum* and *signal magnitude sum* over the three axes. In the frequency domain, we calculate the *energy*, *entropy* and the ratio of the highest magnitude FFT coefficient over the FFT coefficient sum.

Deriving Acoustic Features from Motion Sensor Data. Besides extracting the statistical features, we also derive the acoustic features from the accelerometer data. In particular, we derive the *Mel-Frequency Cepstral Coefficient*

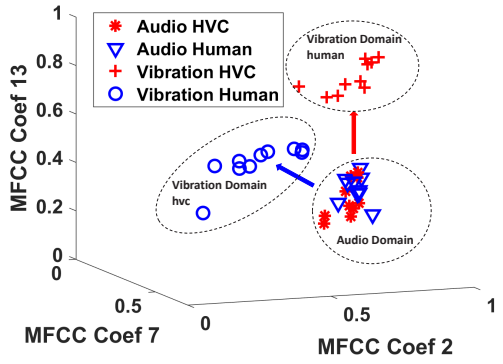


Figure 7: Nonlinear relationship between audio features and vibration features. (Illustrated with the command "Show facebook.com")

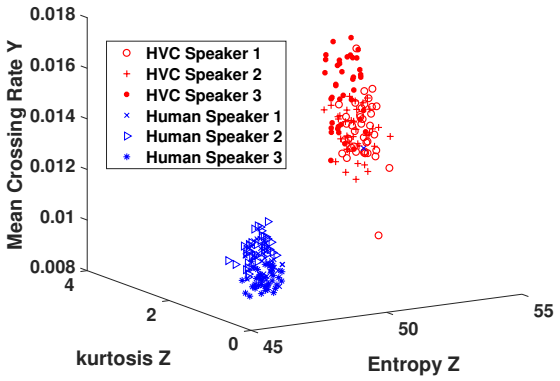


Figure 8: Illustration of the vibration features to distinguish hidden voice commands and normal commands.

(MFCC), which is widely used to describe the short-term power spectrum of acoustic signals and can reflect both the linear and nonlinear properties of the speech signal's dynamic features. While the MFCCs are able to distinguish people's voice differences in the audio domain, we find that they also capture the vibration characteristics. Moreover, we calculate the *chroma vector*, which describes twelve different pitch classes, and the *spectral centroid* and *spectral entropy*.

Unique and Hard to Forge Vibration Features. As introduced in Section 4.2, the vibration signals are nonlinear and are aliased responses of the audio signal. Such nonlinearity means that similar audio domain features may result in very different vibration features. Figure 7 illustrates three normalized MFCC coefficients in the audio and the vibration domain, when the normal commands and the hidden voice commands of "Show facebook.com" are replayed 10 times respectively. We observe that while the audio features could not differentiate the two types of voice commands, they are easily distinguished in the vibration domain in two separate clusters. Thus, audio feature modification does not aid in replicating the vibration features as the vibration features reflect an additional signature of the voice commands. To further illustrate how the vibration features can describe the physical characteristics of hidden voice commands and normal commands, we asked three participants (i.e., two male and one female) to speak multiple commands, and their audio clips were utilized to generate the hidden voice commands using the method introduced in Section 3.1. Figure 8 illustrates an example of using three features to differentiate the two types of voice commands. We find that the features "kurtosis" on the Z axis, "entropy" derived from the accelerations on the Z axis and "mean crossing rate" on the Y axis can distinguish the two types of voice commands in two well separated clusters. Moreover, the normal commands of different human speakers exhibit similar vibration features in a much smaller cluster indicating that a voice command, if mapped away from this cluster, may not be a normal command.

5.3 Feature Selection Based on Statistical Analysis

Based on our experiments, we observe that not all of the extracted statistical features and acoustic features are unique enough to distinguish the hidden voice commands and normal commands. We resort to statistical analysis to identify a subset of features from the above candidate features, which are discriminative for the different types of voice commands and maintain relatively independent to various command contexts and people's different voices. In particular, we first normalize the values of the features between 0 and 1 and then calculate the score s of a feature based on Equation 2.

$$s = \frac{\bar{F}_{hid} - \bar{F}_{hum}}{max(\frac{\sqrt{\sum(F_{hid}(i)-\bar{F}_{hid})^2}}{n}, \frac{\sqrt{\sum(F_{hum}(j)-\bar{F}_{hum})^2}}{n})}, \quad (2)$$

where $F_{hid(i)}$ and $F_{hum(j)}$ represent the feature value of each hidden voice command sample i and normal command sample j and $\bar{F}$ is the mean of the feature value. The calculated score reflects how well the two types of voice commands are separated regarding their distribution. By using a small set of the hidden voice command and normal command samples, we calculated the score for all of the candidate features and selected the more distinguishable features based on a threshold. Figure 10 illustrates the distributions of the selected features by our method when using a Samsung Note 4's built-in speaker to play and its motion sensor to record the voice commands, including different command words

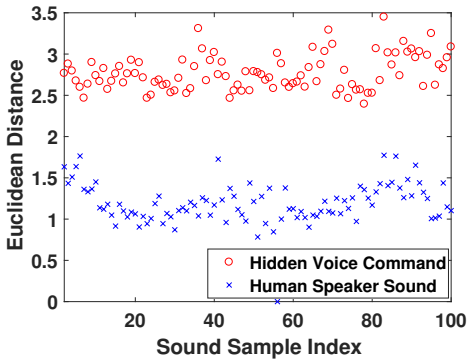


Figure 9: Euclidean-distances of the voice command samples to the human speaker sound cluster centroid based on vibration features.

and multiple people's voices. We find that all of the identified vibration features such as mean crossing rate, entropy and MFCCs show very different distributions of the two types of voice commands. Thus, we leverage the selected vibration features, which are more discriminative to the two types of voices commands, to defend against the hidden voice commands.

5.4 Supervised Learning-based Method

Based on the unique vibration features, we apply learning-based binary classifiers to determine the voice command types, i.e., from the human speakers or the hidden voice command adversary. In particular, several machine learning algorithms like Simple Logistic, Support Vector Machine, Random Forest and Random Tree can be used. Simple Logistic is a logistic regression-based classifier which predicts the types of voice commands from the vibration features using a logistic function. Support Vector Machine relies on a hyperplane to divide the input command sample space into two categories. The hyperplane is determined during the training phase with the labeled voice commands from both types. We apply Sequential Minimal Optimization (SMO) as the optimization algorithm in SVM. Random Tree and Random Forest are classifiers based on decision tree. Random Forest further corrects the over-fitting issue that exists in decision-tree based classifiers.

5.5 Unsupervised Learning-based Method

While the supervised learning-based methods are relatively more complex and require large training data sets to build the model, we utilized the unsupervised learning-based methods, which are able to learn the inherent physical differences between the hidden voice commands and normal commands, without using explicitly-provided labels. Particularly, we used the k-means based and k-medoids based methods.

Table 1: List of voice commands being used.

1	What's my current location?	6	Call 911.
2	Open Bank of America.	7	Open youtube.com.
3	Turn on airplane mode.	8	Show facebook.com.
4	Play country music.	9	Open the door please.
5	What's my schedule today?	10	Ok Google.

During the training phase, the proposed methods map the voice command samples (including both hidden voice commands and normal commands) into two clusters in the multi-dimensional feature space. Only the cluster centroid of the normal commands is calculated. We then compute the Euclidean distance of the training voice command samples to the cluster centroid. As shown in Figure 9, the normal command samples show small Euclidean distances to the normal command cluster centroid, while the hidden voice commands are far from this cluster and thus can be distinguished. A threshold is determined based on these Euclidean distances calculated from the training voice commands.

During the testing phase, an input voice command's vibration features are utilized to compute the Euclidean distance to the normal command cluster centroid and is rejected if the distance is larger than the threshold. We believe that all human voices show similar physical characteristics in the vibration domain, while the hidden voice commands exhibit different feature patterns and hence can be distinguished. As we will show in Section 6, our unsupervised learning-based method does not require much training effort and is adequate to differentiate between the hidden voice commands generated with unseen command words and voices.

6 PERFORMANCE EVALUATION

6.1 Experimental Setup

6.1.1 Devices and Experimental Setup. We evaluate our system in both frontend and backend setups. In the frontend setup, we conduct our experiments using four different smartphone models in a university office with ambient noise (e.g., heating, ventilation and air conditioning noise). Particularly, Samsung Note 4, LG G3, Motorola Nexus 6 and Samsung Galaxy S6 are used due to their different physical body designs, which may result in slightly different structure-borne propagations. Moreover, the sensor specs of the four devices are different. LG G3's motion sensor works at 120Hz while the other three work at 250Hz. Figure 11 (a) shows the frontend setup with a smartphone placed on a table. The voice commands are played by the smartphone's built-in speaker in maximum volume and recorded by its own motion sensor. We also conduct experiments in the frontend setup with the smartphone held by hand or placed on the soft surface, which are presented in Section 6.2.4.

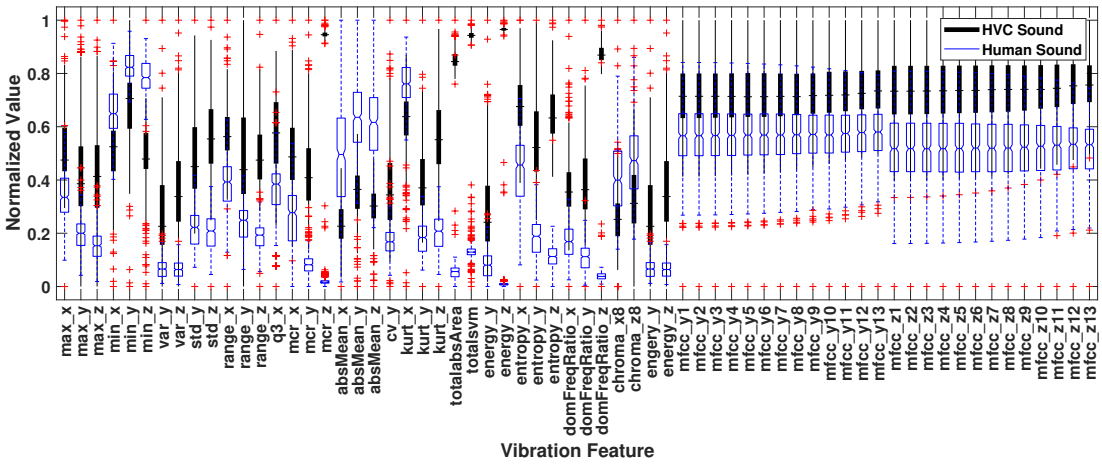


Figure 10: Distribution of the selected features (illustrated with the Note 4 placed on the table).

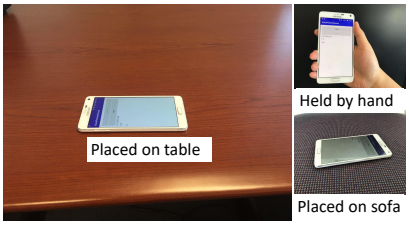
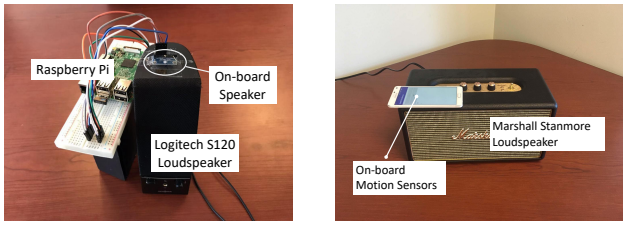


Figure 11: Frontend playback setups.



(a) Prototype on Raspberry Pi (b) Cloud device imitation

Figure 12: Backend playback setups.

In the backend setup, we utilize Raspberry Pi (Model 3B plus) to build a prototype of the cloud service device as shown in Figure 12(a). The device has a three-axis accelerometer, SunFounder Digital ADXL345, attached to the top of a common loudspeaker (Logitech S120). This accelerometer's sampling rate is set to 200Hz. The voice commands received by the user's VCS device would be sent to this cloud service device to play back, and the on-board motion sensor records the corresponding vibration signatures to detect the hidden voice commands. Furthermore, we also imitate the cloud service device by placing four types of smartphones on a Marshall Stanmore loudspeaker, and the phone's motion sensor is used to imitate the on-board motion sensor of the cloud service speaker. Note that this setup can also imitate the frontend playback on the standalone VCS device, which has an onboard motion sensor.

6.1.2 Data Collection. We used a typical Text To Speech (TTS) service [2] with 5 speaker models (i.e., 3 females and 2 males) to generate a set of benign commonly used voice commands as shown in Table 1. The sampling rate for the audio samples is 16kHz. In order to best generate hidden voice commands, we discussed with the authors of the work [9, 28]. Particularly, we gradually adjust the MFCC parameters and try to convert the low resolution MFCC features into the obfuscated commands. To ensure that they can be correctly recognized by Google Now, we use Google's Cloud Speech-to-Text service to test all of the generated hidden voice commands. For each mobile device in each setup, each benign voice command and hidden voice command are repeatedly played ten times. In total we collected 13, 000 motion sensor data traces (i.e., 6, 500 from benign commands and 6, 500 from hidden voice commands) for our experimental evaluation.

6.2 Performance of Recognizing Hidden Voice Commands

6.2.1 Supervised-learning. We first examine the performance of our system using supervised learning-based methods. Table 2 shows the accuracy of the binary classification (10-fold cross validation) of the voice commands in the frontend setup on four different smartphones placed on a table. We observe that all four supervised learning-based methods can efficiently differentiate the hidden voice commands and the normal commands on all the four mobile devices.

Table 2: Performance of supervised learning in the frontend setup (Device on a table).

	Note 4	G3	Nexus 6	S6
SimpleLogistic	100%	99.8%	100%	88.3%
SMO	100%	99.9%	99.9%	85.4%
Random Forest	100%	99.5%	100%	93.1%
Random Tree	99.9%	98.1%	100%	87.4%

Table 3: Performance of supervised learning in the backend setup.

	Note 4	G3	Nexus 6	S6
SimpleLogistic	99.9%	99.8%	99.8%	95.3%
SMO	99.9%	99.9%	99.3%	95.0%
Random Forest	99.9%	100%	99.8%	95.3%
Random Tree	99.9%	98.9%	97.9%	89.7%

In particular, Samsung Note 4, LG G3 and Nexus 6 achieve up to 100% accuracy by using the four supervised learning classifiers. Samsung S6 has the lowest accuracy, which still reaches 93.1% when using Random Forest as the classifier. Table 3 presents the 10-fold cross-validation performance of our system in the backup playback setup with all the four mobile devices used to imitate a cloud device's on-board motion sensor. We find that the Samsung Note 4, LG G3 and Nexus 6 show similar performance which is over 99.9% when using Simple Logistic and Random Forest. Samsung Galaxy S6 shows a lower accuracy of 95.3% when using Random Forest. Moreover, the performances of our system for frontend and backend playback setups are similar, which indicates that our system is flexible and deployable in both setups with different on-board motion sensor. We also test our Raspberry Pi prototype with the four supervised-learning methods, which achieve over 88% accuracy.

We further evaluate the system's performance to recognize the hidden voice commands when the command words and the human voices are not included in the training model. We train the supervised learning model with two speakers' voices and five command words. Table 4 and Table 5 presents the system performances in both frontend setups and backend setups. We find that our system recognizes the hidden voice commands from the unknown command words and unknown speaker voice with high accuracy in both the frontend and backend playback setups. In particular, Samsung Note 4 and Nexus 6 achieve over 99.6% for both setups. The LG G3's performed well in the frontend setup with 99.7% when using Simple Logistic. Its accuracy is 92.4% in the backend setup when using Random Forest. The results indicate that the training size has a small influence on our supervised learning methods in identifying hidden voice commands.

Table 4: Performance of supervised learning in frontend setup (trained with 5/10 words and 2/5 participants).

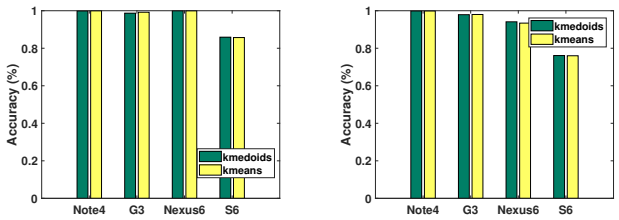
	Note 4	G3	Nexus 6	S6
SimpleLogistic	100%	99.7%	100%	87.3%
SMO	100%	99.4%	100%	86.4%
Random Forest	100%	97.9%	100%	90.6%
Random Tree	100%	94.4%	99.6%	87.8%

Table 5: Performance of supervised learning in backend speaker setup (trained with 5/10 words and 2/5 participants).

	Note 4	G3	Nexus 6	S6
SimpleLogistic	100%	86.4%	99.6%	89.7%
SMO	100%	86.6%	99.2%	90.0%
Random Forest	100%	92.4%	98.8%	84.8%
Random Tree	99.4%	85.8%	98.1%	78.7%

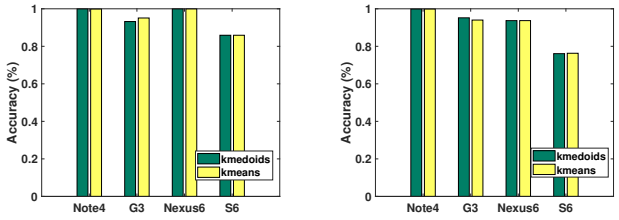
6.2.2 Unsupervised-learning. We now present the results of using our unsupervised learning-based methods to distinguish between the hidden voice commands and normal commands, which capture the inherent physical differences between the normal commands and the hidden voice commands, without requiring much training effort. Figure 13 shows the performance of our k-means and k-medoids based methods when using four different mobile devices in the frontend and backend setups. We observe that both K-means and K-medoids based methods accurately identify the hidden voice commands for both setups. Moreover, both unsupervised methods perform with similar accuracy among the four devices. In particular, Samsung Note 4, LG G3 and Nexus 6 achieve over 99.2% in the frontend playback setup using the K-means method. The accuracies are 100%, 98% and 93% for the three devices in the backend playback setup. Samsung S6 obtains 85.7% and 79% in both the frontend and backend setups. The results are comparable with our supervised learning-based methods. We also evaluate our Raspberry Pi cloud device prototype with the unsupervised learning methods, which achieves 82% accuracy.

We further evaluate the influence of the training data size to the performance of our unsupervised learning-based methods, especially when an adversary plays the hidden voice commands using words and a speaker voice that are not in the unsupervised training model. We only train the k-means and k-medoids model with two speakers' five voice command words and evaluate our system's capability of identifying hidden voice commands. Figure 14 shows the performance of our system under the limited training set. We find that our system still identifies hidden voice commands with very



(a) Frontend setup (b) Backend setup

Figure 13: Performance of unsupervised learning methods.



(a) Frontend setup (b) Backend setup

Figure 14: Performance of unsupervised learning methods trained with 5/10 words and 2/5 participants.

high accuracy in both setups. In particular, Samsung Note 4, LG G3 and Nexus 6 obtain 99.9%, 95.1% and 99.9% in the frontend setup with the K-means method and that for the backend setup obtained 99.9%, 94% and 93.7%. Moreover, by comparing Figure 14 with Figure 13, we observe that the performances obtained based on the different training data sizes are similar. This is because our unsupervised learning-based methods differentiate between the hidden voice command and normal commands by capturing their inherent physical differences in the vibration domain. As a result, the hidden command sounds are mapped far away from the normal command cluster based on the vibration features, even if the hidden voice commands are generated from unseen words and voices to the system.

6.2.3 Partial Playback to Reduce Delay. We further evaluate the performance our system with playback of partial voice commands rather than the complete commands to speed up the playback process and reduce the delay. In particular, we set the system to playback only 1 second or 0.5 second of the recorded voice commands (around one word) and apply the unsupervised learning method K-means on the collected accelerometer readings to detect the hidden voice command. The performances of such partial replay in the frontend setup and the backend setup are shown in Table 6

Table 6: Performance of partial replay (i.e., 1s and 0.5s) in frontend speaker setup using the unsupervised learning method K-means.

	Note 4	G3	Nexus 6	S6
Replay all	100%	99.10%	100%	85.70%
Replay 1s	100%	89.10%	99.90%	85.60%
Replay 0.5s	99.90%	85.20%	95.90%	85%

Table 7: Performance of partial replay (i.e., 1s and 0.5s) in backend speaker setup using the unsupervised learning method K-means.

	Note 4	G3	Nexus 6	S6
Replay all	99.90%	97.90%	93.40%	76%
Replay 1s	92.9	99.10%	92.40%	75.90%
Replay 0.5s	88.5	90.20%	90.50%	73.80%

Table 8: Performance of frontend setup with various smartphone usage scenarios (unsupervised).

	Table	Held in hand	Placed on sofa	80%vol. on table	2x speed on table
Kmed	100%	87.30%	100%	100%	88.30%
Kmea	100%	87.30%	100%	100%	85.20%

and 7. We observe that our system can achieve similar performance in both the frontend mode and the backend mode with the partial playback, compared to the complete voice command playback. In particular, our system can achieve up to 99.9% accuracy for both 1-second and 0.5-second playback for the frontend mode. For the backend mode, we can achieve up to 99.1% for the 1-second playback and 90.5% for the 0.5-second playback. This is because our unique vibration features capture the inherent signatures of the user's voice that is dependent from the length of voice commands. We also find the K-medoid performance is similar to K-means and the supervised learning methods perform slightly better. The results are promising, which facilitate deploying our system without causing additional delay.

6.2.4 Various Mobile Device Usage Scenarios. We next evaluate the frontend setup with various practical mobile devices usage scenarios. We only report the unsupervised learning method results because the supervised learning performs better. Table 8 shows the comparison of our system under different scenarios, when the mobile device is on the table, held by hand, and placed on a soft surface such as a sofa. We also evaluate our system with lower volume (e.g., 80%) and fast forwarding (e.g.,×2 speed). We find our system achieves 100% accuracy of distinguishing hidden voice commands

and normal commands when the mobile device is on the hard table surface or soft sofa surface. When being held by hand, the accuracy degrades to 87.3%, because the hand suppresses the speech vibrations on the mobile device. When replaying with 80% volume, our system still achieves 100% accuracy, which indicates the robustness of our system to capture the speech vibration differences of the two types of voice commands under different playback volumes. In addition, in the fast forwarding scenario with ×2 speed, the accuracy degrades to 88.3% and 85.2% for K-medoids and K-means methods. This is because fast forwarding causes some distortions and information loss to the playback sound, compared to the original audio.

7 CONCLUSION & DISCUSSION

In this work, we show that hidden voice commands that mimic the voice features of normal commands, while remaining incomprehensible to humans, can be detected by comparing their speech features in the vibration domain. We showed that by using the vibration features, including the statistical features in the time and frequency domains and speech features (e.g., MFCCs, chroma vector), it is possible to distinguish hidden voice commands (created as per [9]) from normal commands with a sufficient degree of accuracy. Our supervised and unsupervised learning classification results from observing speech vibration on multiple smartphones, indicate that the idea of using the on-board accelerometer (found universally on all smartphones) to log these speech vibrations, can help the voice assistant technology of these smartphones to detect hidden voice commands. In conclusion, we believe that the ability of the vibration domain features to detect hidden voice commands, as discussed in this work, opens up a new discussion to secure the voice assistants on current smartphones, in the context of synthesized voice commands. Their capability as a standalone mechanism or in conjunction with audio domain features offers additional exciting defense approaches in the domain of voice security.

We recognize that the playback process of our system may cause some playback delay and the intrusion of frontend mode. However, we show that our system can achieve high accuracies with the partial command playback and n-times speed playback, which speed up the playback process (e.g., to 0.5 second). Moreover, the existing VCSs usually take several seconds to process the audio for understanding the command context (e.g., 2 seconds for Google Now and 4 seconds for Siri [6]), and our system is designed in a manner that works simultaneously with this process to avoid causing additional delay. Furthermore, such short time playback incur less intrusion to the users, who can still choose the backend mode of our system to defend against the hidden command attacks.

In our future work, we will study modulating the frontend playback sound to an inaudible frequency (e.g., greater than $16KHz$) to achieve zero intrusion. Meanwhile, a more practical backend playback setup consisting of a tiny low-cost device that is equipped with an on-board speaker and motion sensors will be explored. Moreover, it is potential to extend our unsupervised learning-based method to also defend other attacks, such as ultrasound attacks, without requiring much training effort. It is also worth exploring whether the replay sounds could be distinguished from the live human voice in the vibration domain.

8 ACKNOWLEDGMENTS

This work was supported partially by the National Science Foundation Grants CNS1820624, CNS1814590, CNS1801630, CNS1714807, CNS1526524, DOE1662762 and ARO W911NF-18-1-0221.

REFERENCES

[1] [n.d.]. PDV-100 PORTABLE DIGITAL VIBROMETER: Vibration sensor for mobile use. https://www.polytec.com/us/vibrometry/products/single-point-vibrometers/pdv-100-portable-digital-vibrometer/. Accessed: 12/13/2018.

[2] 2018. Text To Speech (TTS) service. http://www.fromtexttospeech.com/.

[3] 2018. Voice Assistant Market Research Report- Global Forecast 2023. https://www.marketresearchfuture.com/reports/voice-assistant-market-4003.

[4] Talal B Amin, James S German, and Pina Marziliano. 2013. Detecting voice disguise from speech variability: Analysis of three glottal and vocal tract measures. In *Proceedings of Meetings on Acoustics 166ASA*, Vol. 20. ASA, 060005.

[5] S Abhishek Anand and Nitesh Saxena. 2018. Speechless: Analyzing the Threat to Speech Privacy from Smartphone Motion Sensors. In *Proceedings of the 39th IEEE Symposium on Security and Privacy (S&P)*. 116–133.

[6] Mehdi Assefi, Guangchi Liu, Mike P Wittie, and Clemente Izurieta. 2015. An experimental evaluation of apple siri and google speech recognition. *Proccedings of the 2015 ISCA SEDE* (2015).

[7] Ronald J. Baken and Robert F. Orlikoff. 1987. *Clinical Measurement of Speech and Voice.* London: Taylor and Francis Ltd.

[8] Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2018. 2MA: Verifying Voice Commands via Two Microphone Authentication. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM, 89–100.

[9] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden Voice Commands.. In *USENIX Security Symposium*. 513–530.

[10] Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 1–7.

[11] Simon Castro, Robert Dean, Grant Roth, George T Flowers, and Brian Grantham. 2007. Influence of acoustic noise on the dynamic performance of MEMS gyroscopes. In *ASME 2007 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical

Engineers, 1825–1831.

[12] Phillip L De Leon, Michael Pucher, Junichi Yamagishi, Inma Hernaez, and Ibon Saratxaga. 2012. Evaluation of speaker verification security and detection of HMM-based synthetic speech. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 8 (2012), 2280–2290.

[13] Robert Neal Dean, Simon Thomas Castro, George T Flowers, Grant Roth, Anwar Ahmed, Alan Scottedward Hodel, Brian Eugene Grantham, David Allen Bittle, and James P Brunsch. 2011. A characterization of the performance of a MEMS gyroscope in acoustically harsh environments. *IEEE Transactions on Industrial Electronics* 58, 7 (2011), 2591–2596.

[14] Robert N Dean, George T Flowers, A Scotte Hodel, Grant Roth, Simon Castro, Ran Zhou, Alfonso Moreira, Anwar Ahmed, Rifki Rifki, Brian E Grantham, et al. 2007. On the degradation of MEMS gyroscope performance in the presence of high power acoustic noise. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*. IEEE, 1435–1440.

[15] Wenrui Diao, Xiangyu Liu, Zhe Zhou, and Kehuan Zhang. 2014. Your voice assistant is mine: How to abuse speakers to steal information and control your phone. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. ACM, 63–74.

[16] Huan Feng, Kassem Fawaz, and Kang G Shin. 2017. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 343–355.

[17] Jeff Gamet. 2018. Need Your HomePod to Recalibrate its Sound for Your Room? Give it a Shake. https://www.macobserver.com/tips/quick-tip/homepod-recalibrate-shake/.

[18] Rosa González Hautamäki, Tomi Kinnunen, Ville Hautamäki, and Anne-Maria Laukkanen. 2015. Automatic versus human speaker verification: The case of voice mimicry. *Speech Communication* 72 (2015), 13–31.

[19] Rosa González Hautamäki, Tomi Kinnunen, Ville Hautamäki, Timo Leino, and Anne-Maria Laukkanen. 2013. I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry.. In *Interspeech*. Citeseer, 930–934.

[20] Chadawan Ittichaichareon, Siwat Suksri, and Thaweesak Yingthawornsuk. 2012. Speech recognition using MFCC. In *International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) July*. 28–29.

[21] Chaouki Kasmi and Jose Lopes Esteves. 2015. IEMI threats for information security: Remote command injection on modern smartphones. *IEEE Transactions on Electromagnetic Compatibility* 57, 6 (2015), 1752–1755.

[22] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing Speech from Gyroscope Signals.. In *USENIX Security Symposium*. 1053–1067.

[23] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. 2015. All your voices are belong to us: Stealing voices to fool humans and machines. In *European Symposium on Research in Computer Security*. Springer, 599–621.

[24] Emmanuel Munguia Tapia. 2008. *Using machine learning for real-time activity recognition and estimation of energy expenditure*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[25] K Sri Rama Murty and Bayya Yegnanarayana. 2006. Combining evidence from residual phase and MFCC features for speaker recognition. *IEEE signal processing letters* 13, 1 (2006), 52–55.

[26] Ingo R. Titze and Daniel W. Martin. 1994. *Principles of Voice Production*. Vol. 104. Prentice Hall (currently published by NCVS.org). https://doi.org/10.1121/1.424266

[27] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. 2017. WALNUT: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*. IEEE, 3–18.

[28] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. 2015. Cocaine noodles: exploiting the gap between human and machine speech recognition. *WOOT* 15 (2015), 10–11.

[29] Olli Viikki and Kari Laurila. 1998. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication* 25, 1-3 (1998), 133–147.

[30] Zhizheng Wu, Xiong Xiao, Eng Siong Chng, and Haizhou Li. 2013. Synthetic speech detection using temporal modulation feature. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7234–7238.

[31] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. 2018. CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition. In *USENIX Security Symposium*.

[32] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. DolphinAttack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 103–117.

[33] Li Zhang, Parth H Pathak, Muchen Wu, Yixin Zhao, and Prasant Mohapatra. 2015. Accelword: Energy efficient hotword detection through accelerometer. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 301–315.

[34] Linghan Zhang, Sheng Tan, and Jie Yang. 2017. Hearing Your Voice is Not Enough: An Articulatory Gesture Based Liveness Detection for Voice Authentication. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 57–71.

[35] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 2016. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1080–1091.