DolphinAttack: Inaudible Voice Commands

Guoming Zhang*
Zhejiang University
realzgm@zju.edu.cn

Chen Yan*
Zhejiang University
yanchen@zju.edu.cn

Xiaoyu Ji†
Zhejiang University
xji@zju.edu.cn

Tianchen Zhang
Zhejiang University
tianchen-zhang@zju.edu.cn

Taimin Zhang
Zhejiang University
ztm1992fly@zju.edu.cn

Wenyuan Xu†
Zhejiang University
xuwenyuan@zju.edu.cn

ABSTRACT

Speech recognition (SR) systems such as Siri or Google Now have become an increasingly popular human-computer interaction method, and have turned various systems into voice controllable systems (VCS). Prior work on attacking VCS shows that the hidden voice commands that are incomprehensible to people can control the systems. Hidden voice commands, though 'hidden', are nonetheless audible. In this work, we design a completely inaudible attack, DolphinAttack, that modulates voice commands on ultrasonic carriers (e.g., $f > 20$ kHz) to achieve inaudibility. By leveraging the nonlinearity of the microphone circuits, the modulated low-frequency audio commands can be successfully demodulated, recovered, and more importantly interpreted by the speech recognition systems. We validate DolphinAttack on popular speech recognition systems, including Siri, Google Now, Samsung S Voice, Huawei HiVoice, Cortana and Alexa. By injecting a sequence of inaudible voice commands, we show a few proof-of-concept attacks, which include activating Siri to initiate a FaceTime call on iPhone, activating Google Now to switch the phone to the airplane mode, and even manipulating the navigation system in an Audi automobile. We propose hardware and software defense solutions. We validate that it is feasible to detect DolphinAttack by classifying the audios using supported vector machine (SVM), and suggest to re-design voice controllable systems to be resilient to inaudible voice command attacks.

KEYWORDS

Voice Controllable Systems, Speech Recognition, MEMS Microphones, Security Analysis, Defense

1 INTRODUCTION

Speech recognition (SR) technologies allow machines or programs to identify spoken words and convert them into machine-readable formats. It has become an increasingly popular human-computer interaction mechanism because of its accessibility, efficiency, and recent advances in recognition accuracy. As a result, speech recognition systems have turned a wide variety of systems into voice controllable systems (VCS): Apple Siri [5] and Google Now [21] allow users to initiate phone calls by voices; Alexa [4] has enabled users to instruct an Amazon Echo to order takeouts, schedule a Uber ride, etc. As researchers devote much of their effort into improving the performance of SR systems, what is less well understood is how speech recognition and the voice controllable systems behave under intentional and sneaky attacks.

Prior work [10, 61] has shown that obfuscated voice commands which are incomprehensible to human can be understood by SR systems, and thus may control the systems without being detected. Such voice commands, though 'hidden', are nonetheless audible and remain conspicuous. This paper aims at examining the feasibility of the attacks that are difficult to detect, and the paper is driven by the following key questions: *Can voice commands be **inaudible** to human while still being audible to devices and intelligible to speech recognition systems? Can injecting a sequence of inaudible voice commands lead to unnoticed security breaches to the voice controllable systems?* To answer these questions, we designed DolphinAttack, an approach to inject inaudible voice commands at VCS by exploiting the ultrasound channel (i.e., $f > 20$ kHz) and the vulnerability of the underlying audio hardware.

Inaudible voice commands may appear to be unfeasible with the following doubts. (a) *How can inaudible sounds be audible to devices?* The upper bound frequency of human voices and human hearing is 20 kHz. Thus, most audio-capable devices (e.g., phones) adopt audio sampling rates lower than 44 kHz, and apply low-pass filters to eliminate signals above 20 kHz [32]. Previous work [61] considers it impossible to receive voices above 20 kHz. (b) *How can inaudible sounds be intelligible to SR systems?* Even if the ultrasound is received and correctly sampled by hardware, SR systems will not recognize signals that do not match human tonal features, and therefore unable to interpret commands. (c) *How can inaudible sounds cause unnoticed security breach to VCS?* The first step towards controlling VCSs is to activate them. Many VCSs (e.g., smartphones and smart home devices) implement the always-on feature that allows them to be activated by speaker-dependent wake words, i.e., such systems utilize voice recognition to authenticate a user. A random voice command will not pass the voice recognition. We solved all these problems, and we show that the DolphinAttack voice commands, though totally inaudible and therefore imperceptible

*Guoming and Chen are co-first authors.
†Corresponding faculty authors.

CCS '17, October 30-November 3, 2017, Dallas, TX, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4946-8/17/10…$15.00
https://doi.org/10.1145/3133956.3134052

to human, can be received by the audio hardware of devices, and correctly understood by speech recognition systems. We validated DolphinAttack on major speech recognition systems, including Siri, Google Now, Samsung S Voice [43], Huawei HiVoice [65], Cortana [37], and Alexa.

Inaudible voice commands question the common design assumption that adversaries may at most try to manipulate a VCS vocally and can be detected by an alert user. Furthermore, we characterize the security consequences of such an assumption by asking the following: to what extent a sequence of inaudible voice commands can compromise the security of VCSs. To illustrate, we show that DolphinAttack can achieve the following sneaky attacks purely by a sequence of inaudible voice commands:

(1) *Visiting a malicious website.* The device can open a malicious website, which can launch a drive-by-download attack or exploit a device with 0-day vulnerabilities.
(2) *Spying.* An adversary can make the victim device initiate outgoing video/phone calls, therefore getting access to the image/sound of device surroundings.
(3) *Injecting fake information.* An adversary may instruct the victim device to send fake text messages and emails, to publish fake online posts, to add fake events to a calendar, etc.
(4) *Denial of service.* An adversary may inject commands to turn on the airplane mode, disconnecting all wireless communications.
(5) *Concealing attacks.* The screen display and voice feedback may expose the attacks. The adversary may decrease the odds by dimming the screen and lowering the volume.

We have tested these attacks on 16 VCS models including Apple iPhone, Google Nexus, Amazon Echo, and automobiles. Each attack is successful on at least one SR system. We believe this list is by far not comprehensive. Nevertheless, it serves as a wake-up call to reconsider what functionality and levels of human interaction shall be supported in voice controllable systems.

In summary, our contributions are listed as follows.

- We present DolphinAttack that can inject covert voice commands at state-of-the-art speech recognition systems by exploiting inaudible sounds and the property of audio circuits. We validate DolphinAttack on 7 popular speech recognition systems (e.g., Siri, Google Now, Alexa) across 16 common voice controllable system platforms.
- We show that adversaries can inject a sequence of inaudible voice commands to activate always-on system and achieve various malicious attacks. Tested attacks include launching Facetime on iPhones, playing music on an Amazon Echo and manipulating the navigation system in an Audi automobile.
- We suggest both hardware-based and software-based defense strategies to alleviate the attacks, and we provide suggestions to enhance the security of voice controllable systems.

2 BACKGROUND AND THREAT MODEL

In this section, we introduce popular voice controllable systems, and discuss their architecture with a focus on MEMS microphones.

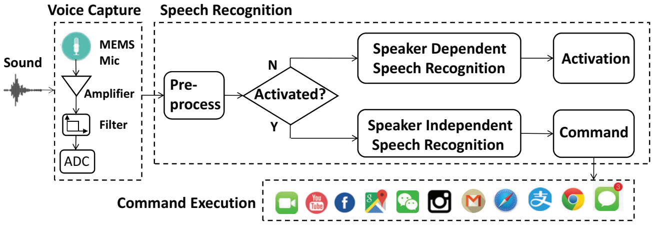


Figure 1: The architecture of a state-of-the-art VCS that can take voice commands as inputs and execute commands.

2.1 Voice Controllable System

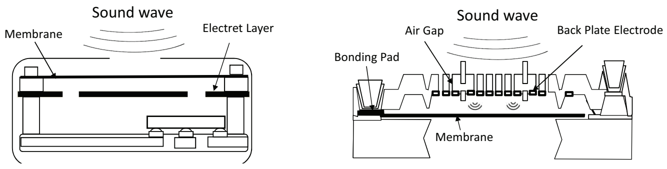
A typical voice controllable system consists of three main subsystems: *voice capture, speech recognition*, and *command execution*, as illustrated in Fig. 1. The voice capture subsystem records ambient voices, which are amplified, filtered, and digitized, before being passed into the speech recognition subsystem. Then, the raw captured digital signals are first pre-processed to remove frequencies that are beyond the audible sound range and to discard signal segments that contain sounds too weak to be identified. Next, the processed signals enter the speech recognition system.

Typically, a speech recognition system works in two phases: activation and recognition. During the activation phase, the system cannot accept arbitrary voice inputs, but it waits to be activated. To activate the system, a user has to either say pre-defined wake words or press a special key. For instance, Amazon echo takes "Alexa" as the activation wake word. Apple Siri can be activated by pressing and holding the home button for about one second or by "Hey Siri" if the *"Allow Hey Siri"* feature is enabled [1]. To recognize the wake words, the microphones continue recording ambient sounds until a voice is collected. Then, the systems will use either speaker-dependent or speaker-independent speech recognition algorithm to recognize the voice. For instance, the Amazon Echo exploits speaker-independent algorithms and accepts 'Alexa' spoken by any one as long as the voice is clear and loud. In comparison, Apple Siri is speaker dependent. Siri requires to be trained by a user and only accepts "Hey Siri" from the same person. Once activated, the SR system enters the recognition phase and will typically use speaker-independent algorithms to convert voices into texts, i.e., commands in our cases.

Note that a speaker-dependent SR is typically performed locally and a speaker-independent SR is performed via a cloud service [28]. To use the cloud service, the processed signals are sent to the servers, which will extract features (typically Mel-frequency cepstral coefficients [10, 27, 62]) and recognize commands via machine learning algorithms (e.g., the Hidden Markov Models or neural networks). Finally, the commands are sent back.

Given a recognized command, the command execution system will launch the corresponding application or execute an operation. The acceptable commands and corresponding actions are system dependent and defined beforehand. Popular voice controllable systems include smartphones, wearable devices, smart home devices,

[1] For older generation of iPhones such as iPhone 4s and iPhone 6, the "Allow Hey Siri" mode is only available when the device is charging.



(a) Structure of ECM

(b) Structure of MEMS Microphone

Figure 2: An illustration of the electret condenser microphone (ECM) and MEMS microphone structure.

and automobiles. Smartphones allow users to perform a wide range of operation via voice commands, such as dialing a phone number, sending short messages, opening a webpage, setting the phone to the airplane mode, etc. Modern automobiles accept an elaborate set of voice commands to activate and control a few in-car features, such as GPS, the entertainment system, the environmental controls, and mobile phones. For instance, if "call 1234567890" is recognized, an automobile or a smartphone may start dialing the phone number 1234567890.

Many security studies on voice controllable systems focus on attacking either the speech recognition algorithms [10] or command execution environment (e.g., malware). This paper aims at the voice capturing subsystem, which will be detailed in the next subsection.

2.2 Microphone

A voice capture subsystem records audible sounds and is mainly a microphone, which is a transducer that converts airborne acoustic waves (i.e., sounds) to electrical signals. One of the oldest and most popular types of microphones is called condenser microphones, which convert the sound waves into electrical signals via capacity changes. Both Electret Condenser Microphones (ECMs) and Micro Electro Mechanical Systems (MEMS) [2, 3, 29, 52, 53] versions are available on the market. Due to the miniature package sizes and low power consumption, MEMS microphones dominate voice controllable devices, including smartphones, wearable devices. Thus, this paper focuses mainly on MEMS microphones and will report results on ECMs briefly. Nevertheless, MEMS and ECMs work similarly. As shown in Fig. 2(b), a MEMS microphone contains membrane (a movable plate) and a complementary perforated back-plate (a fixed plate) [54]. In the presence of a sound wave, the air pressure caused by the sound wave passes through the holes on the back-plate and reaches the membrane, which is a thin solid structure that flexes in response to the change in air pressure [64]. This mechanical deformation leads to a capacitive change. Since a nearly constant charge is maintained on the capacitor, the capacitance changes will produce an AC signal. In this way, air pressure is converted into an electrical signal for further processing. Similarly, as shown in Fig. 2(a), an ECM microphone utilizes the capacity formed by a flexible membrane and a fixed plate to record sound waves.

Designed to capture audible sounds, microphones, low-pass filters (LPFs), and ADC in the voice capture subsystem are all designed to suppress signals out of the frequency range of audible sounds (i.e., 20 Hz to 20 kHz). According to datasheets, the sensitivity spectrum of microphones is between 20 Hz to 20 kHz, and ideally signals in any other frequency range shall be filtered. Even if a signal higher than 20 kHz is recorded by the microphone, it is supposed to be removed by the LPF. Finally, the sampling rate of the ADC is typically 44.1 kHz, and the digitized signal's frequency is limited below 22 kHz according to the Nyquist sampling theorem.

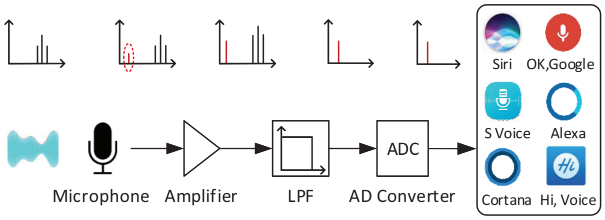


Figure 3: An illustration on the modulated tone traversing the signal pathway of a voice capture device in terms of FFT.

2.3 Threat Model

The adversary's goal is to inject voice commands into the voice controllable systems without owners' awareness, and execute unauthenticated actions. We assume that adversaries have no direct access to the targeted device, own equipment that transmits acoustic signals, and cannot ask the owner to perform any tasks.

No Target Device Access. We assume that an adversary may target at any voice controllable systems of her choices, but she has no direct access to the target devices. She cannot physically touch them, alter the device settings, or install malware. However, we assume that she is fully aware of the characteristics of the target devices. Such knowledge can be gained by first acquiring the device model and then by analyzing the device of the same model before launching attacks.

No Owner Interaction. We assume that the target devices may be in the owner's vicinity, but may not be in use and draw no attention (e.g., on the other side of a desk, with screen covered, or in a pocket). In addition, the device may be unattended, which can happen when the owner is temporarily away (e.g., leaving an Amazon Echo in a room). Alternatively, a device may be stolen, and an adversary may try every possible method to unlock the screen. Nevertheless, the adversaries cannot ask owners to perform any operation, such as pressing a button or unlocking the screen.

Inaudible. Since the goal of an adversary is to inject voice commands without being detected, she will use the sounds inaudible to human, i.e., ultrasounds (f > 20 kHz). Note that we did not use high-frequency sounds (18 kHz < f < 20 kHz) because they are still audible to kids.

Attacking Equipment. We assume that adversaries can acquire both the speakers designed for transmitting ultrasound and commodity devices for playing audible sounds. An attacking speaker is in the vicinity of the target devices. For instance, she may secretly leave a remote controllable speaker around the victim's desk or home. Alternatively, she may be carrying a portable speaker while walking by the victim.

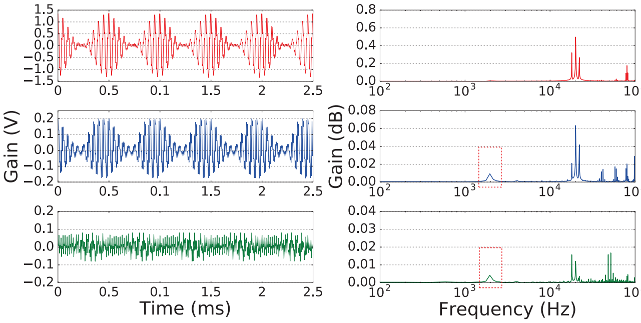


Figure 4: Evaluation of the nonlinearity effect. The time and frequency domain plots for the original signal, the output signal of the MEMS microphone, and the output signal of the ECM microphone. The presence of baseband signals at 2 kHz shows that nonlinearity can demodulate the signals.

3 FEASIBILITY ANALYSIS

The fundamental idea of DolphinAttack is (a) to modulate the low-frequency voice signal (i.e., baseband) on an ultrasonic carrier before transmitting it over the air, and (b) to demodulate the modulated voice signals with the voice capture hardware at the receiver. Since we have no control on the voice capture hardware, we have to craft the modulated signals in such a way that it can be demodulated to the baseband signal using the voice capture hardware as it is. Given that microphone modules always utilize LPF to suppress undesired high-frequency signals, the demodulation shall be accomplished prior to LPF.

Since the signal pathway of voice capture hardware starts from a microphone, one or more amplifiers, LPF, to ADC, the potential components for demodulation are microphones and amplifiers. We look into the principle of both to accomplish DolphinAttack. Although electric components such as amplifiers are designed to be linear, in reality they exhibit **nonlinearity**. With this nonlinearity property, the electric component is able to create new frequencies [25]. Although the nonlinearity for amplifier modules is reported and utilized, it remains unknown whether a microphone, including both the ECM microphone and the MEMS one possesses such a property.

To investigate, we first theoretically model the nonlinearity for a microphone module, and then show the nonlinearity effect on real microphone modules.

3.1 Nonlinearity Effect Modeling

A microphone converts mechanical sound waves into electrical signals. Essentially, a microphone can be roughly considered as a component with square-law non-linearity in the input/output signal transfer characteristics [1, 13]. Amplifiers are known to have nonlinearity, which can produce demodulated signals in the low-frequency range [20]. In this paper, we study the nonlinearity of microphones and we can model it as the following. Let the input signal be $s_{in}(t)$, the output signal $s_{out}(t)$ is:

$$s_{out}(t) = As_{in}(t) + Bs_{in}^2(t) \tag{1}$$

where A is the gain for the input signal and B is the gain for the quadratic term s_{in}^2. A linear component takes a sinusoidal input signals of frequency f and outputs a sinusoidal signal with the same frequency f. In comparison, the nonlinearity of electric devices can produce harmonics and cross-products [2]. Although they are typically considered undesirable distortions [31], the devices with nonlinearity are able to generate new frequencies and with a crafted input signal they can downconvert the signal as well as recover the baseband signal.

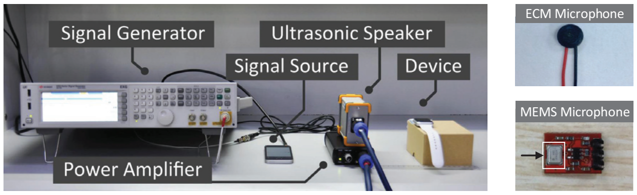


Figure 5: An illustration of the benchtop experimental setup for investigating the feasibility of receiving ultrasounds with ECM and MEMS microphones. This benchtop setup is used for validating the feasibility of attacking various VCSs as well.

Suppose the wanted voice control signal is $m(t)$, we choose the modulated signal on a carrier with central frequency f_c to be

$$s_{in}(t) = m(t)\cos(2\pi f_c t) + \cos(2\pi f_c t) \tag{2}$$

That is, amplitude modulation is used. Without loss of generality, let $m(t)$ be a simple tone, i.e., $m(t) = \cos(2\pi f_m t)$. After applying Eq. (2) to Eq. (1) and taking the Fourier transform, we can confirm that the output signal contains the intended frequency component f_m together with the fundamental frequency components of s_{in} (i.e., $f_c - f_m$, $f_c + f_m$, and f_c), harmonics, and other cross products (i.e., f_m, $2(f_c - f_m)$, $2(f_c + f_m)$, $2f_c$, $2f_c + f_m$, and $2f_c - f_m$). After a LPF, all high-frequency components will be removed and the f_m frequency component will remain, which completes the downconversion, as shown in Fig. 3.

3.2 Nonlinearity Effect Evaluation

Given the theoretical calculation of the nonlinearity effect of the microphone module and its influence on the input signal after modulation, in this section, we verify the nonlinearity effect on real microphones. We test both types of microphones: ECM and MEMS microphones.

3.2.1 Experimental Setup. The experimental setup is shown in Fig. 5. We use an iPhone SE smartphone to generate a 2 kHz voice control signal, i.e. the baseband signal. The baseband signal is then inputted to a vector signal generator [57], which modulates the baseband signal onto a carrier. After amplified by a power amplifier, the modulated signal is transmitted by a high-quality full-band ultrasonic speaker Vifa [9]. Note that we choose the carriers ranging from 9 kHz to 20 kHz, because the signal generator cannot generate signals at the frequencies lower than 9 kHz.

On the receiver side, we test an ECM microphone that was extracted from a headphone and an ADMP401 MEMS microphone [16].

[2] Harmonics are frequencies that are integer multiples of the fundamental frequency components, and cross-products are multiplicative or conjunctive combinations of harmonics and fundamental frequency components.

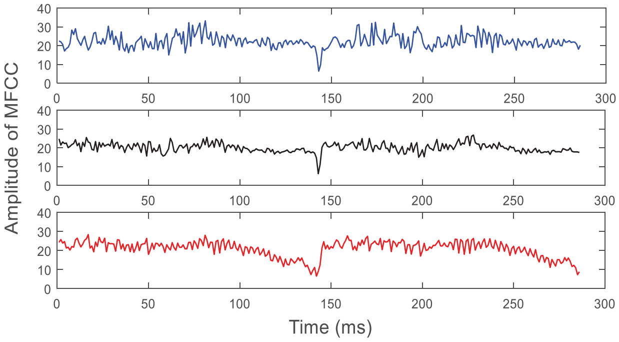


Figure 6: The MFCC for three sound clips of "Hey". From top to bottom: the TTS generated voice, the recorded voice as the TTS voice is played in audible sounds, the recorded voice as the TTS voice is modulated to 25 kHz.

As is shown in Fig. 5, the ADMP401 microphone module contains a preamplifier. To understand the characteristics of microphones, we measure the signals outputted by the microphone instead of by the preamplifier.

3.2.2 Results. We studied the nonlinearity using two types of signals: single tones and voices with multiple tones.

Single Tone. Fig. 4 shows the result when we use a 20 kHz carrier, which confirms that the nonlinearity of the microphone manages to demodulate the baseband signals. The top two figures show the original signal from the speaker in the time domain and the frequency domain, whereby the carrier frequency (20 kHz) and an upper side band as well as a lower sideband (20 ± 2 kHz) appear nicely. The two figures in the second row show the output signal from the MEMS microphone and the bottom two figures depict the output signal from the ECM microphone. Even though the signals were attenuated, especially for ECM microphones, the baseband (2 kHz) in the frequency domain for both microphones confirm the success of demodulation. Note that the frequency domain plots include several high-frequency harmonics, which will be filtered by the LPF and shall not affect the speech recognition.

Voices. Even though we can demodulate a signal tone successfully, voices are a mix of numerous tones at various frequencies and it is unknown whether a demodulated voice signal remains similar to the original one. Thus, we calculated Mel-frequency cepstral coefficients (MFCC), one of the most widely used features of sounds, of three sound clips of "Hey": (a) the original voice generated by a text-to-speech (TTS) engine, (b) the voice recorded by a Samsung Galaxy S6 Edge as an iPhone 6 plus played the original TTS voice, and (c) the voice recorded by a Samsung S6 Edge as the TTS voices are modulated and played by the full band ultrasonic speaker Vifa [9]. As Fig. 6 shows, the MFCC of all three cases are similar. To quantify the similarity, we calculate Mel-Cepstral Distortion (MCD) between the original one and the recorded ones, which is 3.1 for case (b) and 7.6 for case (c). MCD quantifies the distortion between two MFCCs, and the smaller the better. Typically, the two voices are considered to be acceptable to voice recognition systems if their MCD values are smaller than 8 [23], and thus the result encourages us to carry out further study on DolphinAttack against voice controllable systems.

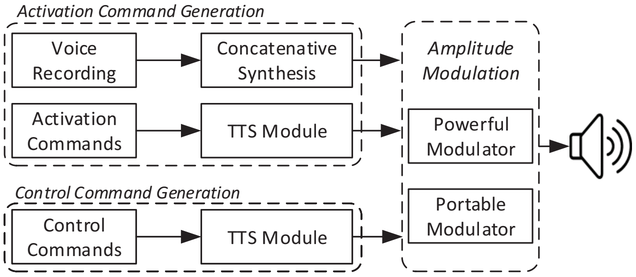


Figure 7: Architecture of the transmitter modules. The transmitter mainly includes the command generation modules and the modulation module.

4 ATTACK DESIGN

DolphinAttack utilizes inaudible voice injection to control VCSs silently. Since attackers have little control of the VCSs, the key of a successful attack is to generate inaudible voice commands at the attacking transmitter. In particular, DolphinAttack has to generate the baseband signals of voice commands for both activation and recognition phases of the VCSs, modulate the baseband signals such that they can be demodulated at the VCSs efficiently, and design a portable transmitter that can launch DolphinAttack anywhere. The basic building blocks of DolphinAttack are shown in Fig. 7, and we discuss these details in the following subsections. Without loss of generality, we discuss design details by using Siri as a case study, and the technology can be applied to other SR systems (e.g., Google Now, HiVoice) easily.

4.1 Voice Command Generation

Siri works in two phases: activation and recognition. It requires activation before accepting voice commands, and thus we generate two types of voice commands: activation commands and general control commands. To control a VCS, DolphinAttack has to generate activation commands before injecting general control commands.

4.1.1 Activation Commands Generation. A successful activation command has to satisfy two requirements: (a) containing the wake words "Hey Siri", and (b) toning to the specific voice of the user that was trained for Siri.

Creating an activation command with both requirements is challenging, unless a user happens to speak "Hey Siri" when an attacker is nearby and manages to create a clear recording. In practice, an attacker can at most record arbitrary words by chances. Generating "Hey Siri" of the specific voice using existing speech synthesis techniques [38] and features extracted from the recordings is extremely different, if ever possible, because it is unclear what set of features are utilized by Siri for voice identification. As such, we design two methods to generate activation commands for two scenarios, respectively: (a) an attacker cannot find the owner of Siri (e.g., an

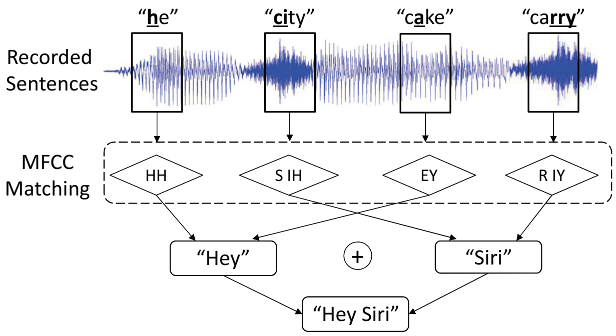


Figure 8: Concatenative synthesis of an activation command. The MFCC feature for each segment in a recorded sentence is calculated and compared with the phonemes in the activation command. After that, the matched voice segments are shuffled and concatenated in a right order.

attacker acquires a stolen smartphone), and (b) an attacker can obtain a few recordings of the owner’s voice.

(1) TTS-based Brute Force. The recent advancement in TTS technique makes it easy to convert texts to voices. Thus, even if an attacker has no chances to obtain any voice recordings from the user, she can generate a set of activation commands that contain wake words by TTS (Text to Speech) systems. This is inspired by the observation that two users with similar vocal tones can activate the other’s Siri. Thus, as long as one of the activation commands in the set has a voice that is close enough to the owner, it suffices to activate Siri. In DolphinAttack, we prepare a set of activation commands with various tone and timbre with the help of existing TTS systems (summarized in Tab. 1), which include Selvy Speech, Baidu, Google, etc. In total, we obtain 90 types of TTS voices. We choose the Google TTS voice to train Siri and the rest for attacking.

(2) Concatenative Synthesis. When an attacker can record a few words from the owner of the Siri but not necessary “Hey Siri”, we propose to synthesize a desired voice command by searching for relevant phonemes from other words in available recordings. There are roughly 44 phonemes in English, and the wake words “Hey Siri” use 6 of them (i.e., HH, EY, S, IH, R, IY). Many words pronounce the same as “Hey” or “Si” or “ri”, and it is possible to splice them together. For example, we can concatenate “he” and “cake” to obtain “Hey”. Similarly, “Siri” can be a combination of “city” and “carry”. As illustrated in Fig. 8, We first search for single or combined phonemes in a recorded sentence and then extracts the interested segments if a match is found. Finally, the matched phonemes are assembled.

To evaluate the feasibility of this scheme, we conduct the following experiments. We use the Google TTS to generate “Hey Siri” for training the SR system, and we generate two sets of candidate voices to synthesize “Hey Siri”: 1. “he”, “cake”, “city”, “carry”; 2. “he is a boy”, “eat a cake”, “in the city”, “read after me”. After synthesizing the activation commands, we test them on an iPhone 4S using the same experimental setup as shown in Fig. 5. Both of the synthesized “Hey Siri” can activate Siri successfully.

Table 1: The list of TTS systems used for attacking the Siri trained by the Google TTS system, and the evaluation results on activation and control commands.

TTS Systems	voice type #	# of successful types	
		Call 12..90	Hey Siri
Selvy Speech [51]	4	4	2
Baidu [8]	1	1	0
Sestek [45]	7	7	2
NeoSpeech [39]	8	8	2
Innoetics [59]	12	12	7
Vocalware [63]	15	15	8
CereProc [12]	22	22	9
Acapela [22]	13	13	1
Fromtexttospeech [58]	7	7	4

4.1.2 General Control Commands Generation. General control commands can be any commands that launch applications (e.g., “call 911”, “open www.google.com”) or configure the devices (e.g., “turn on airplane mode”). Unlike the activation commands, an SR system does not authenticate the identities of control commands. Thus, an attacker can choose the text of any control command and utilize TTS systems to generate the command.

4.1.3 Evaluation. We test both activation and control commands. Without loss of generality, we generate both activation and control commands by utilizing the TTS systems summarized in Tab. 1. In particular, we download two voice commands from the websites of these TTS systems: “Hey Siri” and “call 1234567890”. For activation, we use the “Hey Siri” from the Google TTS system to train Siri, and the rest for testing. We play the voice commands by an iPhone 6 Plus and the benchtop devices (shown in Fig. 5), and test on an iPhone 4S. The activation and recognition results for both commands are summarized in Tab. 1. The results show that the control commands from any of the TTS systems can be recognized by the SR system. 35 out of 89 types of activation commands can activate Siri, resulting in a success rate of 39%.

4.2 Voice Commands Modulation

After generating the baseband signal of the voice commands, we need to modulate them on ultrasonic carriers such that they are inaudible. To leverage the nonlinearity of microphones, DolphinAttack has to utilize amplitude modulation (AM).

4.2.1 AM Modulation Parameters. In AM, the amplitude of the carrier wave varies in proportion to the the baseband signal, and amplitude modulation produces a signal with its power concentrated at the carrier frequency and two adjacent sidebands, as is shown in Fig. 9. In the following, we describe how to select AM parameters in DolphinAttack.

(1) Depth. Modulation depth m is defined as $m = M/A$ where A is the carrier amplitude, and M is the modulation amplitude, i.e., M is the peak change in the amplitude from its unmodulated value. For example, if $m = 0.5$, the carrier amplitude varies by 50% above (and below) its unmodulated level. Modulation depth is directly related to the utilization of the nonlinearity effect of microphones,

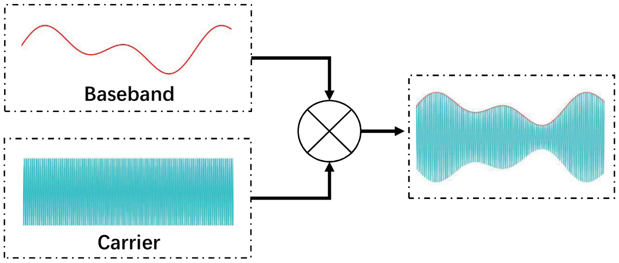


Figure 9: An illustration of modulating a voice command onto an ultrasonic carrier using AM modulation.

and our experiments show that the modulation depth is hardware dependent (detailed in Sec. 5).

(2) Carrier Frequency.

The selection of the carrier frequency depends on several factors: the frequency range of ultrasounds, the bandwidth of the baseband signal, the cut-off frequency of the low pass filter and the frequency response of the microphone on the VCS, as well as the frequency response of the attacking speaker. The lowest frequency of the modulated signal should be larger than 20 kHz to ensure inaudibility. Let the frequency range of a voice command be w, the carrier frequency f_c has to satisfy the condition $f_c - w > 20$ kHz. For instance, given that the bandwidth of the baseband is 6 kHz, the carrier frequency has to be larger than 26 kHz to ensure that the lowest frequency is larger than 20 kHz. One may consider to use the carrier that is right below 20 kHz, because these frequencies are inaudible to most people except for young kids. However, such carriers (e.g., < 20 kHz) will not be effective. This is because when the carrier frequency and lower sideband are below the cut-off frequency of the low-pass filter, they will not be filtered. Therefore, the recovered voices are different from the original signals, and the speech recognition systems will fail to recognize the commands.

Similar to many electric devices, microphones are frequency selective, e.g., the gains at various frequencies vary. For efficiency, the carrier frequency shall be the one that have the highest product of the gains at both the speaker and the VCS microphone. To discover the best carrier frequency, we measure the frequency response of the speaker and microphones, i.e., given the same stimulus, we measure the output magnitude at various frequencies. Fig. 10 shows the frequency response of the ADMP 401 MEMS microphone and the speaker on a Samsung Galaxy S6 Edge [3]. The gains of the microphones and speakers do not necessarily decrease with the increase of frequencies, and thus effective carrier frequencies may not be monotonous.

(3) Voice Selection. Various voices map to various baseband frequency ranges. For example, a female voice typically has a wider frequency band than what a male voice has, which results in a larger probability of frequency leakage over audible frequency range, i.e., the lowest frequency of the modulated signal may be smaller than 20 kHz. Thus, if possible, a voice with a small bandwidth shall be selected to create baseband voice signals.

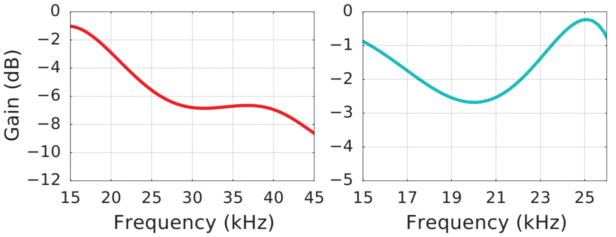


Figure 10: The frequency responses of the ADMP401 MEMS microphone (left) and the Samsung Galaxy S6 Edge speaker (right).

4.3 Voice Commands Transmitter

We design two transmitters: (a) a powerful transmitter that is driven by a dedicated signal generator (shown in Fig. 5) and (b) a portable transmitter that is driven by a smartphone (shown in Fig. 11). We utilize the first one to validate and quantify the extent to which DolphinAttack can accomplish various inaudible voice commands, and we use the second one to validate the feasibility of *a walk-by attack*. Both transmitters consist of three components: a signal source, a modulator, and a speaker. The signal source produces baseband signals of the original voice commands, and outputs to the modulator, which modulates the voice signal onto a carrier wave of frequency f_c in forms of amplitude modulation (AM). Finally, the speaker transforms the modulated signal into acoustic waves, and note that the sampling rate of the speaker has to be larger than $2(f_c + w)$ to avoid signal aliasing.

4.3.1 The Powerful Transmitter with A Signal Generator. We utilize a smartphone as the signal source and the vector signal generator described in Fig. 5 as the modulator. Note that the signal generator has a sampling range of 300 MHz, much larger than ultrasonic frequencies, and can modulate signals with predefined parameters. The speaker of the powerful transmitter is a wide-band dynamic ultrasonic speaker named Vifa [9].

4.3.2 The Portable Transmitter with a Smartphone. The portable transmitter utilizes a smartphone to transmit the modulated signals. Since we found that the best carrier frequencies for many devices are larger than 24 kHz as is depicted in Tab. 3, a majority of smartphones cannot accomplish the task. Most smartphones support at most a 48 kHz sampling rate and can only transmit a modulated narrow-band signal with the carrier frequency of at most 24 kHz. To build a portable transmitter that works for a wide range of VCSs, we acquired a Samsung Galaxy S6 Edge, which supports a sampling rate up to 192 kHz. Unfortunately, the on-board speaker of Samsung Galaxy S6 attenuates the signal with a frequency larger than 20 kHz. To alleviate the problem, we use narrow-band ultrasonic transducers [56] as the speaker and add an amplifier prior to the ultrasonic transducer as shown in Fig. 11. As such, the effective attack range is extended.

5 FEASIBILITY EXPERIMENTS ACROSS VCS

We validate DolphinAttack experimentally on 16 popular voice controllable systems and 7 speech recognition systems, and seek answers to three questions: (a) Will the attacks work against different

[3] We used a professional ultrasonic microphone and speaker to assist measurement.

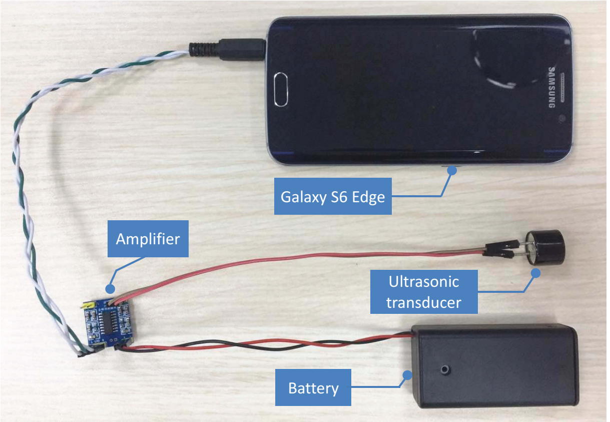


Figure 11: Portable attack implementation with a Samsung Galaxy S6 Edge smartphone, an ultrasonic transducer and a low-cost amplifier. The total price for the amplifier, the ultrasonic transducer plus the battery is less than $3.

speech recognition systems on various operation systems and hardware platforms? (b) How do different software and hardware affect the performance of attacks? (c) What are the key parameters in crafting a successful attack? This section describes the experiment design, setup, and results in detail.

5.1 System Selection

We examine our DolphinAttack attacks on various state-of-the-art speech recognition systems and off-the-shelf VCSs, which are listed in Tab. 3. The list does not intend to be exhaustive, but rather provides a representative set of VCSs that can be acquired for experiments with our best effort.

Our approach in selecting the target systems is twofold — software and hardware. First of all, we select major speech recognition systems that are publicly available, e.g., Siri, Google Now, Alexa, Cortana, etc. Unlike ordinary software, SR systems (especially proprietary ones) are highly hardware and OS dependent. For example, Siri can only be found and used on Apple products; Alexa is limited to Amazon devices; Cortana runs exclusively on Windows machines. Nevertheless, we select and experiment on the hardware whichever the SR systems are compatible with. To explore the hardware influence on the attack performance, we examine the attacks on different hardware models running the same SR system, e.g., Siri on various generations of iPhones.

In summary, we select VCS and SR systems that are popular on the consumer market with active users and cover various application areas and usage scenarios. In Tab. 3, we summarize the selected VCSs for experiments, which can be classified into three categories — personal devices (wearables, smartphones, tablets, computers), smart home devices, and vehicles.

5.2 Experiment Setup

We test our attacks on each of the selected voice controllable system and speech recognition system with the same experiment setup and equipment, and report their behavior when injecting inaudible voice commands with three goals:

Table 2: The list of systems and voice commands being tested in Tab. 3.

Attack	Device/System	Command
Recognition	Phones & Wearable	*Call 1234567890*
Recognition	iPad	*FaceTime 1234567890*
Recognition	MacBook & Nexus 7	*Open dolphinattack.com*
Recognition	Windows PC	*Turn on airplane mode*
Recognition	Amazon Echo	*Open the back door*
Recognition	Vehicle (Audi Q3)	*Navigation* *
Activation	Siri	*Hey Siri*
Activation	Google Now	*Ok Google*
Activation	Samsung S Voice	*Hi Galaxy*
Activation	Huawei HiVoice	*Hello Huawei* *
Activation	Alexa	*Alexa*

* The command is spoken in Chinese due to the lack of English support on these devices.

- Examining the feasibility of attacks.
- Quantifying the parameters in tuning a successfully attack.
- Measuring the attack performance.

Equipment. Unless specified, all experiments utilize the default experiment equipment: a powerful transmitter as shown in Fig. 5, which consists of a smartphone as the signal source, a signal generator as the modulator, and a wide-band dynamic ultrasonic speaker named Vifa [9] as the speaker to play inaudible voice commands. Since the powerful transmitter is able to transmit signals with a wide range of carriers (from 9 kHz to 50 kHz), we use it for feasibility study. In comparison, the portable transmitter utilizes narrow-band speakers, and its transmission frequencies are limited by the available narrow-band speakers. In our case, our portable transmitter can transmit signals at the frequencies of 23 kHz, 25 kHz, 33 kHz, 40 kHz, and 48 kHz.

Setup. Unless constrained by the device size, we position the selected device in front of our benchtop attack equipment at varying distances on a table, with the device microphone facing right toward the speaker. Both the device and the speaker are elevated to the same heights (i.e., 10 cm above the table) to avoid mechanical coupling. All experiments except the one with automobiles are conducted in our laboratory with an average background noise of 55 dB SPL (sound pressure level), and we confirm that no interfering sound exists within the test frequency band (20 kHz – 50 kHz). We play the inaudible voice commands through the powerful transmitter, and observe the results on the device screen or from device acoustic response.

Generally, multiple microphones are installed on a device to pick up voices from all directions. It is a common case that all the microphones are used in speech recognition. In our experiments, we specifically test the one that shows the best demodulation effect.

Voice Commands. Two categories of voice commands are prepared for two types of attacks, activation and recognition. For those systems supporting voice activation, we try to activate them with inaudible wake word commands. To examine whether the inaudible

Table 3: Experiment devices, systems, and results. The examined attacks include *recognition* (executing control commands when the SR systems are manually activated) and *activation* (when the SR systems are unactivated). The modulation parameters and maximum attack distances are acquired for recognition attacks in an office environment with a background noise of 55 dB SPL on average.

Manuf.	Model	OS/Ver.	SR System	Attacks		Modulation Parameters		Max Dist. (cm)	
				Recog.	Activ.	f_c (kHz) & [Prime f_c] ‡	Depth	Recog.	Activ.
Apple	iPhone 4s	iOS 9.3.5	Siri	√	√	20–42 [27.9]	≥ 9%	175	110
Apple	iPhone 5s	iOS 10.0.2	Siri	√	√	24.1 26.2 27 29.3 [24.1]	100%	7.5	10
Apple	iPhone SE	iOS 10.3.1	Siri	√	√	22–28 33 [22.6]	≥ 47%	30	25
			Chrome	√	N/A	22–26 28 [22.6]	≥ 37%	16	N/A
Apple	iPhone SE †	iOS 10.3.2	Siri	√	√	21–29 31 33 [22.4]	≥ 43%	21	24
Apple	iPhone 6s *	iOS 10.2.1	Siri	√	√	26 [26]	100%	4	12
Apple	iPhone 6 Plus *	iOS 10.3.1	Siri	×	√	— [24]	—	—	2
Apple	iPhone 7 Plus *	iOS 10.3.1	Siri	√	√	21 24-29 [25.3]	≥ 50%	18	12
Apple	watch	watchOS 3.1	Siri	√	√	20–37 [22.3]	≥ 5%	111	164
Apple	iPad mini 4	iOS 10.2.1	Siri	√	√	22–40 [28.8]	≥ 25%	91.6	50.5
Apple	MacBook	macOS Sierra	Siri	√	N/A	20-22 24-25 27-37 39 [22.8]	≥ 76%	31	N/A
LG	Nexus 5X	Android 7.1.1	Google Now	√	√	30.7 [30.7]	100%	6	11
Asus	Nexus 7	Android 6.0.1	Google Now	√	√	24–39 [24.1]	≥ 5%	88	87
Samsung	Galaxy S6 edge	Android 6.0.1	S Voice	√	√	20–38 [28.4]	≥ 17%	36.1	56.2
Huawei	Honor 7	Android 6.0	HiVoice	√	√	29–37 [29.5]	≥ 17%	13	14
Lenovo	ThinkPad T440p	Windows 10	Cortana	√	√	23.4–29 [23.6]	≥ 35%	58	8
Amazon	Echo *	5589	Alexa	√	√	20-21 23-31 33-34 [24]	≥ 20%	165	165
Audi	Q3	N/A	N/A	√	N/A	21–23 [22]	100%	10	N/A

‡ Prime f_c is the carrier wave frequency that exhibits highest baseband amplitude after demodulation. — No result

† Another iPhone SE with identical technical spec.

* Experimented with the front/top microphones on devices.

voice commands can be correctly recognized by the speech recognition systems, we select a few English commands that are human intelligible as listed in Tab. 2. Since no commands are supported across all devices, we prepare a set of commands to cover all devices. For each command, we try two audio sources: the synthetic voices from TTS engines, and the genuine human voices spoken by the authors.

Sound Pressure Level. Though the sound generated for attacks are inaudible to human, we nonetheless measure the sound pressure level (SPL) in decibels using a free field measurement microphone [50]. The received SPL for the ultrasound is measured at 10 cm away from the Vifa [9] speaker and is 125 dB.

Attacks. In recognition attacks, the SR systems are manually activated beforehand. While in activation attacks, physical interactions with the devices are not permitted. The attacks are only considered successful and the distances are only recorded when the recognized texts from SR systems totally match with the attack commands.

Modulation Parameters. We argue that the modulation parameters may have an influence on the attack performance. We consider two factors in the amplitude modulation: the carrier wave frequency f_c and the modulation depth. To quantify their influence, we place the devices 10 cm away from the wide-band ultrasonic speaker Vifa [9] using the Google TTS engine as the baseband audio source, and measure three values: (a) f_c *range* — the range of carrier wave frequency in which recognition attacks are successful and 100% accurate. (b) *Prime* f_c — the f_c that exhibits the highest baseband [4] amplitude after demodulation. (c) *AM depth* — the modulation depth at the prime f_c when recognition attacks are successfully and 100% accurate.

5.3 Feasibility Results

Tab. 3 summarizes the experiment results. From Tab. 3, we can conclude that DolphinAttack works with nearly all of the examined SR systems and devices. In particular, the inaudible voice commands can be correctly interpreted by the SR systems on all the tested hardware, and the activation is successful on all VCSs that require activation. The results, however, do show that devices and systems require various parameters to accomplish the same attack effect. We discuss our findings as follows.

Hardware Dependence. The basic principle of DolphinAttack is to inject inaudible voice commands before digitization components. Therefore, the feasibility of DolphinAttack depends heavily on the audio hardware rather than the speech recognition systems. For example, various devices from the same manufacturer running Siri show the great variance in the attack success rate, the maximum attack distance, and modulation parameters. This is because various models adopt different hardware (e.g., microphones, amplifiers, filters), which lead to variation in the digitized audios that are input to the same SR system. Our experiment on two identical devices (iPhone SE) exhibits similar attack parameters and results. Thus, it

[4] For simplicity, the baseband signal for finding prime f_c is a 400 Hz single tone which resides in human voice frequency.

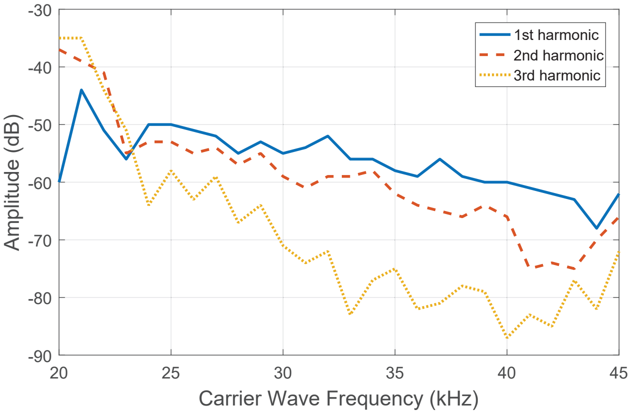


Figure 12: Amplitude of the demodulated 400 Hz baseband signal (1st harmonic) and its higher order harmonics on Nexus 7, with varying carrier wave frequency f_c.

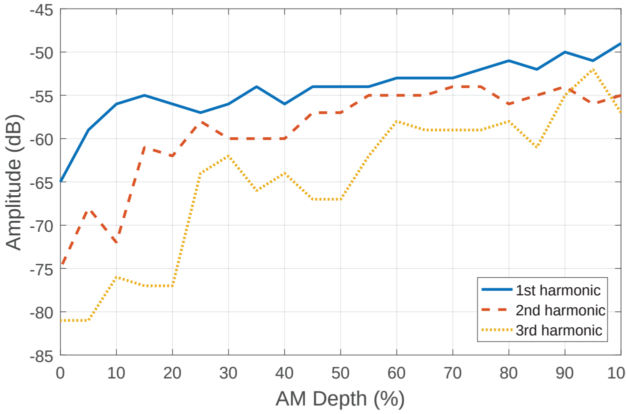


Figure 13: Amplitude of the demodulated 400 Hz baseband signal (1st harmonic) and its higher order harmonics on Nexus 7, with varying modulation depth.

is feasible for an adversary to study the hardware beforehand to achieve satisfying attack results.

SR System Dependence. We find that various SR systems may handle the same audios differently. We tested the voice search in Google Chrome running on an iPhone SE. The results in Table 3 show that the f_c range of Google Chrome overlaps with the f_c range in Siri experiment, which suggests that our attacks are hardware dependent. However, the differences in f_c, AM depth, and recognition distances are resulted from the SR systems.

Recognition versus Activation. Various devices and SR systems can react differently to recognition and activation attacks in terms of the attack distance. For some devices (8 devices), activation attacks can be achieved at a larger distance than recognition attacks, while for other devices (6 devices), the effective range of successful activation attacks is smaller than the recognition attacks. In addition, we observe that for many of the devices, appending the activation commands (e.g., "Hey Siri") before the control commands can increase the probability for correct recognition, possibly because the activation commands are trained specially by the SR systems to be recognized in the always-on mode.

Commands Matter. The length and content of a voice command can influence the success rate and the maximum distance of attacks. We are rigorous in the experiments by demanding every single word within a command to be correctly recognized, though this may be unnecessary for some commands. For instance, "Call/FaceTime 1234567890" and "Open dolphinattack.com" is harder to be recognized than "Turn on airplane mode" or "How's the weather today?". In the former scenarios, both the execution words "call", "open" and the content (number, url) have to be correctly recognized. However, for the latter scenarios, only recognizing key words such as "airplane" and "weather" shall be enough for executing the original commands. The attack performance can be improved if the attack command is short and common to SR systems.

Carrier Wave Frequency. f_c is the dominant factor that affects the attack success rate, and it also shows great variation across devices. For some devices, the f_c range within which recognition attacks are successful can be as wide as 20–42 kHz (e.g., iPhone 4s), or as narrow as a few single frequency points (e.g., iPhone 5s). We attribute this diversity to the difference of frequency response and frequency selectivity for these microphones as well as the nonlinearity of audio processing circuits.

For instance, the f_c range of Nexus 7 is from 24 to 39 kHz, which can be explained from two aspects. The f_c is no higher than 39 kHz because the frequency response of the Vifa speaker over 39 kHz is low and the one of Nexus 7 microphone is low as well. Thus, in combination, a carrier higher than 39 kHz is no longer efficient enough to inject inaudible voice commands. The f_c cannot be smaller than 24 kHz because of the nonlinearity of the microphone frequency response. We observe that the inaudible voice commands become unacceptable to SR systems when the amplitude of the harmonics of the baseband are larger than the one of baseband. For instance, given the baseband of a 400 Hz tone, we measure the demodulated signal (i.e., the 400 Hz baseband) on a Nexus 7, and observe harmonics at 800 Hz (2nd harmonic), 1200 Hz (3rd harmonic) and even higher, which are possibly caused by the nonlinearity of audio processing circuits. As shown in Fig. 12, when the f_c is less than 23 kHz, the 2nd and 3rd harmonics are stronger than the 1st harmonic, which will distort the baseband signal and make it hard for SR systems to recognize. The *Prime* f_c that leads to the best attack performance, is the frequency that exhibits both a high baseband signal and low harmonics. On Nexus 7, the *Prime* f_c is 24.1 kHz.

Modulation Depth. Modulation depth affects the amplitude of demodulated baseband signal and its harmonics, as shown in Fig. 13. As the modulation depth gradually increases from 0 to 100%, the demodulated signals become stronger, which in turn increase the SNR and the attack success rate, with a few exceptions (e.g., when the harmonics distort the baseband signal more than the cases of a lower AM depth). We report the minimum depth for successful recognition attacks on each device in Tab. 3.

Attack Distance. The attack distances vary from 2 cm to a maximum value of 175 cm and show a great variation across devices. Notably, the maximum distance that we can achieve for both attacks is 165 cm on Amazon Echo. We argue that the distance can be

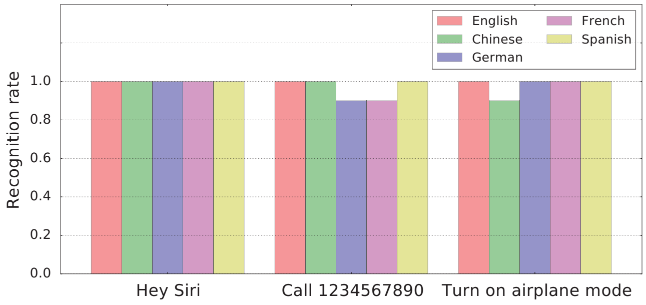


Figure 14: The recognition rates of voice commands in five languages.

increased with the equipment that can generate a sound with higher pressure levels and exhibit better acoustic directionality, or by using shorter and more recognizable commands.

Efforts and Challenges. We faced challenges in conducting the above experiments. Apart from acquiring the devices, measuring each parameter is time-consuming and labor-intensive due to the lack of audio measurement feedback interface. For example, to measure the *Prime* f_c, we analyze the demodulation results on various devices using audio spectrum analyzing software on different platforms: iOS [30], macOS [34], Android [41], and Windows [35]. For devices not supporting installing spectrum software such as Apple watch and Amazon Echo, we utilize the calling and command log playback function, and measure the audio on another relaying device.

5.4 Summary

We summarize our experiments as follows.

(1) We validated recognition and activation attacks across 16 various devices and 7 speech recognition systems, and succeeded on nearly all of them.
(2) We measured the attack performance on all devices, and some of them suffice for real attacks in daily scenarios. For instance, we can launch DolphinAttack from almost 2 meters away against an iPhone 4s and Amazon Echo.
(3) We measured, examined, and discussed the parameters involved in the attack performance, including SR systems, device hardware, voice commands, f_c, AM depth, etc.

6 IMPACT QUANTIFICATION

In this section, we evaluate the performance of DolphinAttack in terms of languages, background noises, sound pressure levels, and attack distances using the powerful transmitter (i.e., the benchtop setup shown in Fig. 5). In addition, we evaluate the effectiveness of walk-by attacks using the portable devices.

6.1 Influence of Languages

To examine the effectiveness of DolphinAttack with regard to languages, we select three voice commands in five languages. The voice commands include an activation command ("Hey Siri") and two control commands ("Call 1234567890" and "Turn on airplane mode"), which represent three attacks against SR systems: activating SR systems, initiating to spy on the user, and denial of service attacks. Each voice command is tested in English, Chinese, German, French, and Spanish, respectively.

We launch DolphinAttack against an Apple watch that is paired with an iPhone 6 Plus running iOS 10.3.1. For each voice command in each language, we repeat it for 10 times and calculate the average success rate. The distance is set to 20 cm, the measured background noise is 55 dB. We exploit a 25 kHz carrier frequency and 100% AM depth.

Fig. 14 shows the recognition results of the three voice commands in the given languages. As we can see that the recognition rate of various languages and voice commands are almost the same. In particular, the recognition rate of all the voice commands in English and Spanish is 100%, and the average recognition rate of the three voice commands across all languages are 100%, 96%, 98%, respectively. Moreover, the recognition rate for activation (i.e., "Hey Siri") is higher than the one of control commands "Call 1234567890" and "Turn on airplane mode"). This is because the length of the activation command is shorter than the control commands. In any case, the results show that our approach is effective for various languages and voice commands.

Table 4: The impact of background noises for sentence recognition evaluated with an Apple watch.

Scene	Noises (dB)	Recognition rates	
		Hey Siri	Turn on airplane mode
Office	55–65	100%	100%
Cafe	65–75	100%	80%
Street	75–85	90%	30%

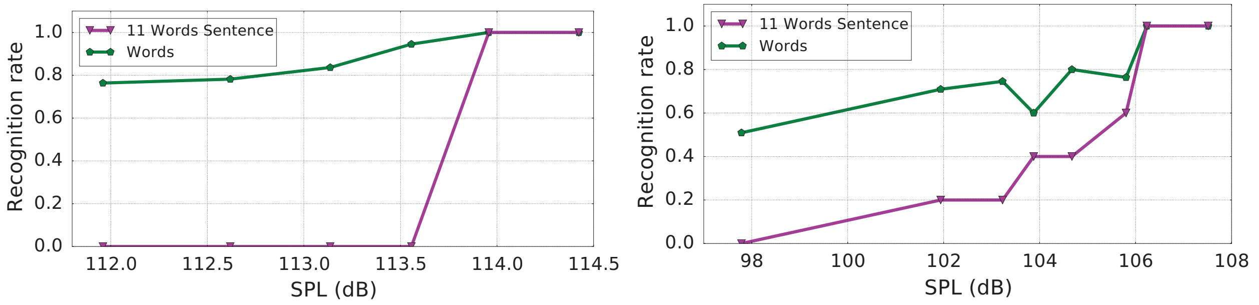
6.2 Impact of Background Noises

Speech recognition is known to be sensitive to background noises and is recommended to be used in a quiet environment. Thus, we examine inaudible voice command injection via DolphinAttack in three scenarios: at an office, in a cafe, and on the street. To ensure that the experiment can be repeatable, we simulate the three scenarios by playing background sounds at a chosen SPL and evaluate their impact on the recognition rates. We choose an Apple watch as the attack target, and measure the background noise by a mini sound meter.

From Tab. 4, we can see that recognition rates for activation command are over 90% for all the three scenes while the recognition rates of the control command ("Turn on airplane mode") decrease with the increase of ambient noise levels. That is because the activation command is shorter than the control command. With the increase of the word count for a control command, the recognition rate drops quickly because failure to recognize any word could render the command recognition unsuccessful.

6.3 Impact of Sound Pressure Levels

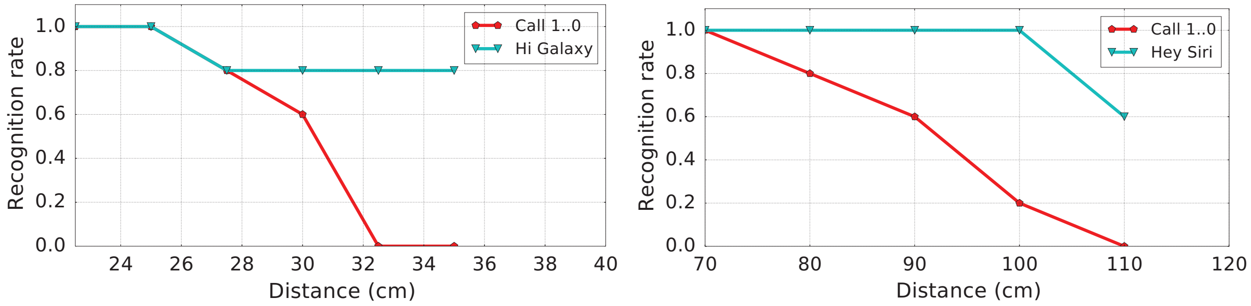
For both audible and inaudible sounds, a higher SPL leads to a better quality of recorded voices and thus a higher recognition rate. This is because a higher SPL always means a larger signal-to-noise



(a) The recognition rates of the Galaxy S6 Edge

(b) The recognition rates of the Apple watch

Figure 15: The impact of sound pressure levels on the recognition rates for two portable devices.



(a) The recognition rates of the Galaxy S6 Edge

(b) The recognition rates of the Apple watch

Figure 16: The impact of attack distances on the recognition rates for two portable devices.

ratio (SNR) for given noise levels. To explore the impact of SPLs on DolphinAttack, we test the control command ("Call 1234567890") on both the Apple watch and a Galaxy S6 Edge smartphone. In all experiments, the speaker is placed 10 cm from the target device, and the mini sound meter is placed alongside the speaker to measure the environment noise.

We quantify the impact of SPLs with two granularities: sentence recognition rates and word recognition rates. Sentence recognition rates calculate the percentage of successfully recognized commands. Only if every word in the command is recognized correctly, the command is considered to be recognized. Word recognition rates are the percentage of words that are correctly interpreted. For example, if the command "call 1234567890" is recognized as "call 1234567", the words recognition rate is 63.6% (7/11).

Fig. 15 (a) (b) show the impact of the SPLs on both types of recognition rates. Not surprisingly, given the same SPL, the word recognition rates are always larger than the sentence recognition rates until both reach 100%. For the Apple watch, both recognition rates become 100% once the SPL is larger than 106.2 dB. In comparison, the minimum SPL for the Galaxy S6 Edge to achieve a 100% recognition rate is 113.96 dB, which is higher than that of the Apple watch. This is because the Apple watch outperforms the Galaxy S6 Edge in terms of demodulating inaudible voice commands.

6.4 Impact of Attack Distances

In this section, an activation command (either "Hey Siri" or "Hi Galaxy") and a control command ("Call 1234567890") are used to test the recognition rates at various distances. We evaluate the recognition rates of two commands on an Apple watch and a Galaxy S6 Edge, and we depict the results in Fig. 16.

In general, the recognition rates of the activation command are higher than that of the control command, because the activation command contains a smaller number of words than the control command. The Apple watch can be activated with a success rate of 100% from 100 cm away, and the Galaxy S6 Edge can be activated with 100% from 25 cm. We believe that the difference between the two devices is because Apple watches are worn on the wrist and are designed to accept voice commands from a longer distance than a smartphone.

6.5 Evaluation of Portable Device Attacks

In this section, we evaluate the effectiveness of portable device attacks.

Setup. We use the Galaxy S6 Edge smartphone running Android 6.0.1 as the attack device and an Apple watch as the victim device which is paired with an iPhone 6 Plus. The attack voice command is

Table 5: Portable device attack results. Attacking an Apple watch using a Galaxy S6 Edge smartphone that is 2 cm away.

f_c (kHz)	20	21	22	23	24
Word recognition rate	80%	100%	16%	100%	0%
Sentence recognition rate	80%	100%	0%	100%	0%

"turn on airplane mode". We set f_c to be {20, 21, 22, 23, 24} kHz, respectively. The AM depth is 100%, and the sampling rate is 192 kHz. The baseband signal has a maximum frequency of 3 kHz.

Results. As shown in Tab. 5, we successfully "turned on airplane mode" on the Apple watch at the 23 kHz carrier frequency. Note that 20 kHz and 21 kHz are also successful. However, there are frequency leakages below 20 kHz and it sounds like crickets and can be heard. The word and sentence recognition rates are 100%. With the increase of f_c, the Apple watch fails to recognize the voice command because of frequency selectivity of the speaker.

To extend the attack distance, we utilize a low-power audio amplifier (3 Watt) module to drive an ultrasonic transducer, as is shown in Fig. 11. With the amplifier module, the maximum distance of effective attacks is increased to 27 cm. Note that the attack distance can be further extended with professional devices and more powerful amplifiers.

The adversary can launch a remote attack utilizing a victim's device. For example, an adversary can upload an audio or video clip in which the voice commands are embedded in a website, e.g. YouTube. When the audio or video is played by the victims' devices, the surrounding voice controllable systems such as Google Home assistant, Alexa, and mobile phones may be triggered unconsciously.

7 DEFENSES

In this section, we discuss the defense strategies to address the aforementioned attacks from both the hardware and software perspectives.

7.1 Hardware-Based Defense

We propose two hardware-based defense strategies: microphone enhancement and baseband cancellation.

Microphone Enhancement. The root cause of inaudible voice commands is that microphones can sense acoustic sounds with a frequency higher than 20 kHz while an ideal microphone should not. By default, most MEMS microphones on mobile devices nowadays allow signals above 20 kHz [2, 3, 29, 52, 53]. Thus, a microphone shall be enhanced and designed to suppress any acoustic signals whose frequencies are in the ultrasound range. For instance, the microphone of iPhone 6 Plus can resist to inaudible voice commands well.

Inaudible Voice Command Cancellation. Given the legacy microphones, we can add a module prior to LPF to detect the modulated voice commands and cancel the baseband with the modulated voice commands. In particular, we can detect the signals within the ultrasound frequency range that exhibit AM modulation characteristics, and demodulate the signals to obtain the baseband. For instance, in the presence of inaudible voice command injection, besides the demodulated baseband signals $m(t)$, the recorded analog voice signals shall include the original modulated signal: $v(t) = Am(t)\cos(2\pi f_c t) + \cos(2\pi f_c t)$, where A is the gain for the input signal $m(t)$. By down-converting $v(t)$ to obtain $Am(t)$ and adjusting the amplitude, we can subtract the baseband signal. Note that such a command cancellation procedure will not affect the normal operation of a microphone, since there will be no correlation between the captured audible voice signals and noises in the ultrasound range.

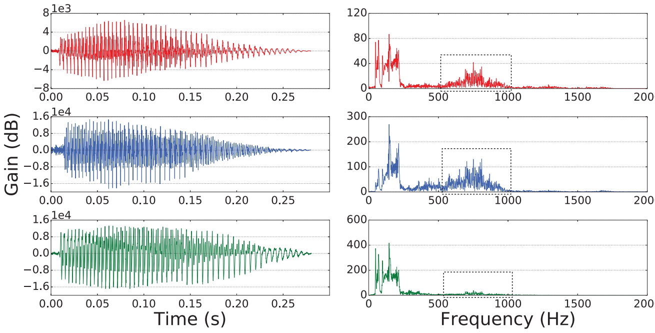


Figure 17: Original (top), recorded (middle) and recovered (bottom) voice signals. The modulated voice command differs from both the original signal and the recorded one in the frequency range between 500 and 1000 Hz.

7.2 Software-Based Defense

Software-based defense looks into the unique features of modulated voice commands which are distinctive from genuine ones.

As shown in Fig. 17, the recovered (demodulated) attack signal shows differences from both the original signal and the recorded one in the high frequency ranging from 500 to 1000 Hz. The original signal is produced by the Google TTS engine, the carrier frequency for modulation is 25 kHz. Thus, we can detect `DolphinAttack` by analyzing the signal in the frequency range from 500 to 1000 Hz. In particular, a machine learning based classifier shall detect it.

To validate the feasibility of detecting `DolphinAttack`, we utilize supported vector machine (SVM) as the classifier, and extracted 15 features in the time and frequency domains from audios. We generated 12 voice commands (i.e., "Hey Siri"): 8 types of voices from the NeoSpeech TTS engine and 4 types of voices from the Selvy TTS engine. With each type, we obtained two samples: one is recorded and the other is recovered. In total, we have 24 samples. To train a SVM classifier, we use 5 recorded audios as positive samples and 5 recovered audios as negative samples. The rest 14 samples are used for testing. The classifier can distinguish the recovered audios from recorded ones with 100% true positive rate (7/7) and 100% true negative rate (7/7). The result using a simple SVM classifier indicating that software-based defense strategy can be used to detect `DolphinAttack`.

8 RELATED WORK

Security of voice controllable systems. An increasing amount of research effort is devoted into studying the security of voice controllable systems [10, 18, 28, 38, 61]. Kasmi et al. [28] introduced a voice command injection attack against modern smartphones by applying intentional electromagnetic interference on headphone cables, while in this paper, we inject voice commands by utilizing the nonlinearity of microphones over ultrasounds. Mukhopadhyay et al. [38] demonstrated voice impersonation attacks on state-of-the-art automated speaker verification algorithms. They built a model of the victim's voice based on the samples from the victim. Diao et al. [18] designed permission bypass attacks from a zero-permission Android application through phone speakers. Hidden voice commands and Cocaine noodles [10, 61] use audible and mangled audio commands to attack speech recognition systems. Under these attacks, the victims can observe the obfuscated voice commands sometimes. `DolphinAttack` is motivated by these attacks, but is completely inaudible and imperceptible and we show it is possible to launch `DolphinAttack` using portable devices.

Security of sensor-equipped devices. Commercial devices equipped with various sensors (e.g., smartphones, wearables, and tablets) are gaining their popularity. Along with the growing trend of ubiquitous mobile devices are the security concerns. Many researchers [14, 15, 46, 60] focus on studying possible attacks against sensors on smart devices. Among which, sensor spoofing (i.e., the injection of a malicious signal into a victim sensor) has attracted much attention and is considered one of the most critical threats to sensor-equipped devices. Shin et al. investigate and classify sensor spoofing attacks [46] into three categories: regular channel attacks (replay attack) [19, 26, 66], transmission channel attacks, and side channel attacks [11, 14, 15, 47, 49]. Dean et al. [14, 15] demonstrated that MEMS gyroscopes are susceptible to high-power high-frequency acoustic noises when acoustic frequency components are close to the resonating frequency of the gyroscope's sensing mass. Utilizing on-board sensors, Gu et al. [24] designed a cryptographic key generation mechanism by using vibration motors and accelerometers. Our work focuses on microphones, which can be considered as one type of sensors.

Privacy leakage through sensors. Michalevsky et al. [36] utilized MEMS gyroscopes to measure acoustic signals, which reveal the speaker information. Schlegel et al. [44] designed a Trojan that can extract high-value data from audio sensors of smartphones. Owusu et al. [40] utilized the accelerometer readings as a side channel to extract the entire sequences of entered text on a smartphone touchscreen keyboard without requiring special privileges. Aviv et al. [6] demonstrated that accelerometer sensors can reveal user taps and gesture-based input. Dey et al. [17] studied how to fingerprint smartphones utilizing the imperfections of on-board accelerometers, and the fingerprints can act as an identifier to track the smartphone's owner. Simon et al. [48] utilized video cameras and microphones to infer PINs entered on a number-only soft keyboard on a smartphone. Li et al. [33] can verify the capture time and location of the photos with the sun position estimated based on the shadows in the photo and sensor readings of the cameras. Sun et al. [55] presented a video-assisted keystroke inference framework to infer a tablet user's inputs from surreptitious video recordings of the tablet motion. Backes et al. [7] showed it is possible to recover what a dot matrix printer processing English text is printing based on the printer's acoustic noises. Similarly, we study how to utilize microphone vulnerabilities for security and privacy breaches.

Roy et al. [42] presented BackDoor, which constructs an acoustic (but inaudible) communication channel between two speakers and a microphone over ultrasound bands. In particular, BackDoor utilizes two ultrasonic speakers to transmit two frequencies. After passing through the microphone's non-linear diaphragm and power-amplifier, the two signals create a "shadow" in the audible frequency range, which could carry data. However, the "shadow" is a single tone instead of a voice command that consists of a rich set of tones. In comparison, we show it is possible to use one speaker to inject inaudible commands to SR systems, causing various security and privacy issues.

9 CONCLUSION

In this paper, we propose `DolphinAttack`, an inaudible attack to SR systems. `DolphinAttack` leverages the AM (amplitude modulation) technique to modulate audible voice commands on ultrasonic carriers by which the command signals can not be perceived by human. With `DolphinAttack`, an adversary can attack major SR systems including Siri, Google Now, Alexa, and etc. To avoid the abuse of `DolphinAttack` in reality, we propose two defense solutions from the aspects of both hardware and software.

ACKNOWLEDGMENTS

This work has been funded in part by 360 Technology Inc., NSFC 61472358, NSFC 61702451, NSF CNS-0845671, and the Fundamental Research Funds for the Central Universities 2017QNA4017.

REFERENCES

[1] Muhammad Taher Abuelma'atti. 2003. Analysis of the effect of radio frequency interference on the DC performance of bipolar operational amplifiers. *IEEE Transactions on Electromagnetic Compatibility* 45, 2 (2003), 453–458.

[2] Akustica. 2014. AKU143 Top Port, analog silicon MEMS microphone. http://www.mouser.com/ds/2/720/DS37-1.01%20AKU143%20Datasheet-552974.pdf. (2014).

[3] Akustica. 2014. AKU242 digital silicon MEMS microphone. http://www.mouser.com/ds/2/720/PB24-1.0%20-%20AKU242%20Product%20Brief-770082.pdf. (2014).

[4] Amazon. 2017. Alexa. https://developer.amazon.com/alexa. (2017).

[5] Apple. 2017. iOS-Siri-Apple. https://www.apple.com/ios/siri/. (2017).

[6] Adam J. Aviv, Benjamin Sapp, Matt Blaze, and Jonathan M. Smith. 2012. Practicality of accelerometer side channels on smartphones. In *Proceedings of the Computer Security Applications Conference.* 41–50.

[7] Michael Backes, Markus Dürmuth, Sebastian Gerling, Manfred Pinkal, and Caroline Sporleder. 2010. Acoustic side-channel attacks on printers.. In *Proceedings of the USENIX Security Symposium.* 307–322.

[8] Baidu. 2017. Baidu Translate. http://fanyi.baidu.com/. (2017).

[9] Avisoft Bioacoustics. 2017. Ultrasonic Dynamic Speaker Vifa. http://www.avisoft.com/usg/vifa.htm. (2017).

[10] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden voice commands. In *Proceedings of the USENIX Security Symposium.*

[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 *Proceedings of the ASME International Mechanical Engineering Congress and Exposition.* American Society of Mechanical Engineers, 1825–1831.

[12] CereProc. 2017. CereProc Text-to-Speech. https://www.cereproc.com/. (2017).

[13] Gordon KC Chen and James J Whalen. 1981. Comparative RFI performance of bipolar operational amplifiers. In *Proceedings of the IEEE International Symposium on Electromagnetic Compatibility.* IEEE, 1–5.

[14] 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.

[15] 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 *Proceedings of the IEEE International Symposium on Industrial Electronics.* 1435–1440.

[16] Analog Devices. 2011. ADMP401: Omnidirectional microphone with bottom port and analog output obsolete data sheet. http://www.analog.com/media/en/technical-documentation/obsolete-data-sheets/ADMP401.pdf. (2011).

[17] Sanorita Dey, Nirupam Roy, Wenyuan Xu, Romit Roy Choudhury, and Srihari Nelakuditi. 2014. AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable.. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[18] 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 ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. ACM, 63–74.

[19] Aurélien Francillon, Boris Danev, and Srdjan Capkun. 2011. Relay attacks on passive keyless entry and start systems in modern cars. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[20] Javier Gago, Josep Balcells, David GonzÁlez, Manuel Lamich, Juan Mon, and Alfonso Santolaria. 2007. EMI susceptibility model of signal conditioning circuits based on operational amplifiers. *IEEE Transactions on Electromagnetic Compatibility* 49, 4 (2007), 849–859.

[21] Google. 2016. Google Now. http://www.androidcentral.com/google-now. (2016).

[22] Acapela Group. 2017. Acapela text to speech demo. http://www.acapela-group.com/. (2017).

[23] CMU Speech Group. 2012. Statistical parametirc sythesis and voice conversion techniques. http://festvox.org/11752/slides/lecture11a.pdf. (2012).

[24] Weixi Gu, Zheng Yang, Longfei Shangguan, Xiaoyu Ji, and Yiyang Zhao. 2014. Toauth: Towards automatic near field authentication for smartphones. In *Proceedings of the IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 229–236.

[25] Paul Horowitz and Winfield Hill. 1989. *The art of electronics*. Cambridge Univ. Press.

[26] Rob Millerb Ishtiaq Roufa, Hossen Mustafaa, Sangho Ohb Travis Taylora, Wenyuan Xua, Marco Gruteserb, Wade Trappeb, and Ivan Seskarb. 2010. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *Proceedings of the USENIX Security Symposium*. 11–13.

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

[28] 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.

[29] Knowles. 2013. SPU0410LR5H-QB Zero-Height SiSonicTM Microphone. http://www.mouser.com/ds/2/218/-532675.pdf. (2013).

[30] Dexus Pawel Krzywdzinski. 2017. Ultrasonic analyzer for iPad and iPhone. http://iaudioapps.com/page1/page1.html. (2017).

[31] Denis Foo Kune, John Backes, Shane S Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. 2013. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. IEEE, 145–159.

[32] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. 2015. Chirp signal-based aerial acoustic communication for smart devices. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2407–2415.

[33] Xiaopeng Li, Wenyuan Xu, Song Wang, and Xianshan Qu. 2017. Are You Lying: Validating the Time-Location of Outdoor Images. In *Proceedings of the International Conference on Applied Cryptography and Network Security*. Springer, 103–123.

[34] Dog Park Software Ltd. 2017. iSpectrum - Macintosh Audio Spectrum Analyzer. https://dogparksoftware.com/iSpectrum.html. (2017).

[35] Ivo Mateljan. 2017. Audio measurement and analysis software. http://www.artalabs.hr/. (2017).

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

[37] Microsoft. 2017. What is Cortana? https://support.microsoft.com/en-us/help/17214/windows-10-what-is. (2017).

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

[39] NeoSpeech. 2017. NeoSpeech Text-to-Speech. http://www.neospeech.com/. (2017).

[40] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2006. ACCessory: password inference using accelerometers on smartphones. (2006).

[41] Carl Reinke. 2017. Spectroid. https://play.google.com/store/apps/details?id=org.intoorbit.spectrum&hl=en. (2017).

[42] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. 2017. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2–14.

[43] Samsung. 2017. What is S Voice? http://www.samsung.com/global/galaxy/what-is/s-voice/. (2017).

[44] Roman Schlegel, Kehuan Zhang, Xiao-yong Zhou, Mehool Intwala, Apu Kapadia, and XiaoFeng Wang. 2011. Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, Vol. 11. 17–33.

[45] Sestek. 2017. Sestek TTS. http://www.sestek.com/. (2017).

[46] Hocheol Shin, Yunmok Son, Youngseok Park, Yujin Kwon, and Yongdae Kim. 2016. Sampling race: bypassing timing-based analog active sensor spoofing detection on analog-digital systems. In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*. USENIX Association.

[47] Yasser Shoukry, Paul Martin, Paulo Tabuada, and Mani Srivastava. 2013. Non-invasive spoofing attacks for anti-lock braking systems. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 55–72.

[48] Laurent Simon and Ross Anderson. 2013. PIN skimmer: inferring PINs through the camera and microphone. In *Proceedings of the ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. 67–78.

[49] Yunmok Son, Hocheol Shin, Dongkwan Kim, Young-Seok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, Yongdae Kim, et al. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *Proceedings of the USENIX Security Symposium*. 881–896.

[50] Cry Sound. 2017. CRY343 free field measurment microphone. http://www.crysound.com/product_info.php?4/35/63. (2017).

[51] Selvy Speech. 2017. Demo-Selvy TTS. http://speech.selvasai.com/en/text-to-speech-demonstration.php. (2017).

[52] STMicroelectronics. 2014. MP23AB02BTR MEMS audio sensor, high-performance analog bottom-port microphone. http://www.mouser.com/ds/2/389/mp23ab02b-955093.pdf. (2014).

[53] STMicroelectronics. 2016. MP34DB02 MEMS audio sensor omnidirectional digital microphone. http://www.mouser.com/ds/2/389/mp34db02-955149.pdf. (2016).

[54] STMicroelectronics. 2017. Tutorial for MEMS microphones. http://www.st.com/content/ccc/resource/technical/document/application_note/46/0b/3e/74/cf/fb/4b/13/DM00103199.pdf/files/DM00103199.pdf/jcr:content/translations/en.DM00103199.pdf. (2017).

[55] Jingchao Sun, Xiaocong Jin, Yimin Chen, Jinxue Zhang, Yanchao Zhang, and Rui Zhang. 2016. VISIBLE: Video-Assisted keystroke inference from tablet backside motion. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[56] Jinci Technologies. 2017. Open structure product review. http://www.jinci.cn/en/goods/112.html. (2017).

[57] Keysight Technologies. 2017. N5172B EXG X-Series RF Vector Signal Generator, 9 kHz to 6 GHz. http://www.keysight.com/en/pdx-x201910-pn-N5172B. (2017).

[58] From Text to Speech. 2017. Free online TTS service. http://www.fromtexttospeech.com/. (2017).

[59] Innoetics Text to Speech Technologies. 2017. Innoetics Text-to-Speech. https://www.innoetics.com/. (2017).

[60] 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 *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 3–18.

[61] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. 2015. Cocaine Noodles: Exploiting the gap between human and machine speech recognition. In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*. USENIX Association.

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

[63] Vocalware. 2017. Vocalware TTS. https://www.vocalware.com/. (2017).

[64] Xiaohui Wang, Yanjing Wu, and Wenyuan Xu. 2016. WindCompass: Determine Wind Direction Using Smartphones. In *Proceedings of the 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.

[65] Xdadevelopers. 2017. HiVoice app, what is it for? https://forum.xda-developers.com/honor-7/general/hivoice-app-t3322763. (2017).

[66] Chen Yan, Wenyuan Xu, and Jianhao Liu. 2016. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEF CON* (2016).