HandLock: Enabling 2-FA for Smart Home Voice Assistants using Inaudible Acoustic Signal

Shaohu Zhang
szhang42@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Anupam Das
anupam.das@ncsu.edu
North Carolina State University
Raleigh, NC, USA

ABSTRACT

The use of voice-control technology has become mainstream and is growing worldwide. While voice assistants provide convenience through automation and control of home appliances, the open nature of the voice channel makes voice assistants difficult to secure. As a result voice assistants have been shown to be vulnerable to replay attacks, impersonation attacks and inaudible voice commands. Existing defenses do not provide a practical solution as they either rely on external hardware (e.g., motion sensors) or work under very constrained settings (e.g., holding the device close to a user's mouth). We introduce the concept of using a gesture-based authentication system for smart home voice assistants called *HandLock*, which uses built-in microphones and speakers to generate and sense inaudible acoustic signals to detect the presence of a known (i.e., authorized) hand gesture. Our proposed approach can act as a second-factor authentication (2-FA) for performing specific *sensitive* operations like confirming online purchases through voice assistants. Through extensive experiments involving 45 participants, we show that *HandLock* can achieve on average 96.51% true-positive-rate (TPR) at the expense of 0.82% false-acceptance-rate (FAR). We perform a comprehensive analysis of *HandLock* under various settings to showcase its accuracy, stability, resilience to attacks, and usability. Our analysis shows that *HandLock* can not only successfully thwart impersonation attacks, but can do so while incurring very low overheads and is compatible with modern voice assistants.

CCS CONCEPTS

• **Security and privacy → Usability in security and privacy**.

KEYWORDS

acoustic sensing, hand gesture, two-factor authentication, voice assistants

ACM Reference Format:
Shaohu Zhang and Anupam Das. 2021. *HandLock*: Enabling 2-FA for Smart Home Voice Assistants using Inaudible Acoustic Signal. In *24th International Symposium on Research in Attacks, Intrusionsand Defenses (RAID '21), October 6–8, 2021, San Sebastian, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3471621.3471866

RAID '21, October 6–8,2021, San Sebastian, Spain

ACM ISBN 978-1-4503-9058-3/21/10…$15.00
https://doi.org/10.1145/3471621.3471866

1 INTRODUCTION

Voice-based computer interaction thrives on the ability to enable users to interact with devices and services through voice instead of keystrokes, mouse-movements or swipes. While speech recognition has been an active field of research for many years, it has seen widespread adoption in recent years, especially with the deployment of smart home voice assistants (VAs) like Amazon Echo and Google Home. These VAs enable consumers to not only listen to music and flash briefings, but also control other smart home appliances. However, the widespread use of VAs also gives rise to both security and privacy concerns due to their always-listening capability [35] and susceptibility to audio-based attacks [32, 43, 56]. According to Edison Research, 63% of VA owners in the USA are concerned that hackers might gain access to their home or personal information through VAs [3].

One of the major security concerns with current VAs is the limited support for authentication. Other than simple customizable wake words like "Alexa" or "Hi, Google," there is not much support for authentication in VAs. VAs do provide the capability to recognize different users based on their voice profiles, however, such approach has been shown to be vulnerable to simple replay attacks [31, 53]. Other features include using voice-based PIN codes to restrict sensitive operations like voice-based online order. Again, the PIN code has to be spoken out loud and is susceptible to passive eavesdropping.

In recent years, several studies have proposed authenticating users through microphones and speakers embedded in smart devices [16, 34, 60]. BreathPrint [16] proposes utilizing the breathing sound made by a user to uniquely identify the user. BiLock [60] extracts biometric signatures from the sounds generated by a user's dental occlusion, captured through the built-in microphone of a smartphone. Lippass [34] leverages unique Doppler profiles of acoustic signals generated by a user's moving lips to authenticate the user. However, all of these schemes require the sensing device (i.e., microphones or speakers) either to be placed very close to the user's mouth or held by the user, which does not amount to a practical solution for smart home VAs.

In this paper, we introduce a hand-gesture based biometric authentication scheme called *HandLock* that can recognize an authorized user based on his/her hand movement. To this end, *HandLock* emits inaudible acoustic signals and records the reflected signals to identify a user. The underlying hypothesis for *HandLock* is that *it is possible to distinguish different users even if they perform the same hand gesture due to their differing physical biometrics.* Specifically, as shown in the Figure 1, since the length of ulna and humerus of a given user is fixed, the starting and ending positions of the hand

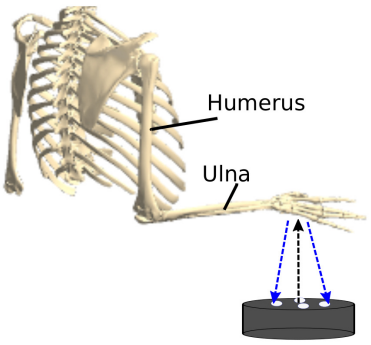


Figure 1: Anatomy of the human arm and hand.

stay the same no matter how fast the user moves the hand. Therefore, the speed profile of a given gesture from the same user should remain similar as the speeds of different parts of the hand and limb change, proportionally. We study the validity of this hypothesis through a comprehensive measurement study. Theoretically, the phase change that appears in the received acoustic signal is directly proportional to the speed at which a human hand was moved while performing a gesture. By combining our hypothesis with this theoretical result, we make the following observation: the phase shift recorded on the received acoustic signal is significantly different for different users, even if they all perform the *same* hand gesture.

HandLock, therefore, first emits inaudible acoustic signals during the authentication phase and simultaneously records audio signal through a microphone. After the user performs a hand gesture, *HandLock* extracts the phase of the received signal as a time-series data using signal processing techniques. Next, this time-series phase data is converted into a time series of speed and acceleration. The last step involves computing statistical features in the temporal and spectral domain to differentiate users via a machine learning model. We evaluate our approach by recruiting a total of 45 participants. We evaluate our approach under various settings, including multi-user enrollment, long-term stability and adversarial attempts. Through our evaluations we see that our approach can be used not only to determine the *physical presence* of a user, but also as an effective second-factor authentication method for VAs.

There are several technical challenges in building an acoustic sensing system to differentiate an individual's hand gesture. First, the time series of acoustical phase introduced by the hand is noisy mixing with ambient noise, DC offset (caused by microphone imperfection), and multi-path propagation (impacted by human body and surrounding environment). To tackle these challenges, we propose a new Quadrature-based (Q-based) phase extraction approach to remove the background noise, DC offset and multi-path propagation. Next, we need to select meaningful features to recognize a predefined gesture performed by different users. Lastly, as gestures can be performed at different speeds, we need to account for the change in speed to make all hand speeds consistent for each user. In summary, we make the following contributions:

- We introduce the idea of using gesture-based authentication mechanism for VAs, using *built-in* microphones and speakers. Our proposed approach can act as a second-factor authentication (2-FA) for sensitive operations such as making online purchases through VAs. To the best of our knowledge, we are the first to propose such a 2-FA system for VAs without requiring *additional hardware.*
- We design and implement *HandLock* using a commercial off-the-shelf (COTS) speaker (ReSpeaker Core v2.0 [8]). We also develop signal-processing techniques that are capable of extracting acoustical phase change caused by hand movements from raw acoustic signals. We, furthermore, develop machine learning models that can effectively identify users based on temporal and spectral features derived from the extracted time-series data representing gesture speed and acceleration.
- We evaluate our approach by recruiting 45 participants and by collecting over 15,000 samples that cover five different gestures. Our results show that *HandLock* can achieve on average 96.51% TPR. With three attempts, *HandLock* can achieve a TPR of 99.91%. We also evaluate our system under both benign and adversarial settings. Lastly, we thoroughly perform various sensitivity analysis to showcase the effectiveness of our proposed authentication system.

The remainder of this paper proceeds as follows. Section 2 provides background and describes related work. In Section 3, we present the detailed design of *HandLock*. Section 4 presents the comprehensive evaluation of our proposed authentication system. We analyze the usability of our approach in Section 5. We list the limitations of our approach in Section 6. Finally, we conclude in Section 7.

2 RELATED WORK

Biometric authentication has been an active field of research for a long time. Voice and facial recognition have been at the forefront of such authentication systems. In this work, we look at recognizing hand gestures through acoustic signals for authenticating voice-assistant users. In this section, we will highlight some of the relevant works in this field.

Voice-based Authentication. Voice-based authentication systems leverage unique human voice characteristics to recognize a user [36]. These voice biometrics include voice based features such as pronunciation, accent, speech speed, as well as physical characteristics of vocal tract, mouth and nasal passages. However, studies show that voice authentications are vulnerable to impersonation [28, 29] and replay attacks [31, 53]. Kinnunen et al. [31] reported that the EER of voice authentication systems can increase anywhere from 1.76% to 31.46% under replay attacks. Researchers have shown that it is easy to launch both black box (i.e., inverse MFCC) [14, 20, 49] and white box (i.e., gradient descent) [45] attacks against speech recognition systems. Recent works such as DolphinAttack [56], BackDoor [42], CommanderSong [54], SirenAttack [24] and LipRead [43] have shown that voice assistants are vulnerable to inaudible voice commands which are incomprehensible to human ear, but can be understood by speech recognition systems. Even voice processing systems such as Google, Bing, IBM and Azure speech APIs have been shown to be susceptible to hidden voice commands [11].

Many schemes have been proposed to defend against replay attacks that perform liveliness tests [19, 51, 57, 58], but such approaches are often not feasible for VAs. For instance, VocieLive [58]

captures the time-difference-of-arrival (TDoA) dynamics of an uttered passphrase to determine liveliness of an audio source. Voicepop [51] leverages the breathing noise to detect both replay and impersonation attacks. However, both of these approaches require the user to speak very closely to the microphone (in the range of 2–6 cm), which is feasible for authenticating one on a smartphone, but not realistic for VAs.

Gesture-based Authentication. Body gesture is another biometric that has been utilized to uniquely identify users. A body of studies have been conducted to authenticate users on mobile phones and wearable IoT devices, leveraging embedded or wearable sensors [25, 33] and WiFi signals [32, 37, 40, 46]. VAuth [25] collects the body-surface vibrations of the user via a wearable motion sensor and correlates the data with the speech signal recorded by the voice assistant's microphone to achieve continuous authentication. VSButton [32] utilizes the motion time-series data extracted from wristband to secure IoT devices. However, these schemes depend on external sensing hardware, and thus are not readily applicable to VAs. Recent studies [37, 40, 46] have shown that existing WiFi signals can be utilized for authentication in smart homes. WiID [46] extracts speed time-series features from WiFi Channel State Information (CSI) to infer 7 gestures such as circular arm motion, waving arm motion and kicking to identify users. REVOLT [40] leverages the WiFi and voice features to detect human presence and speaking to counter replay attacks. While sensing gestures via wireless signals has the advantages of being device-free and unobtrusive, there are two main drawbacks: 1) CSI logging requires special hardware such as an USRP or an special WiFi card (e.g., Inter 5300 NIC), and 2) accuracy can significantly degrade in environments with moving objects (e.g., pets moving) or when the position of the transceiver changes.

Acoustic Sensing. Many acoustic-based gesture recognition systems have been proposed to recognize in-air gestures [18, 27, 39, 44, 48]. SoundWave [27], AudioGest [44], and MultiWave [39] all characterize the Doppler effect to sense motion gestures. EchoTrack [18] uses two speakers and one microphone in smartphones to track hand motion. Strata [55] estimates the channel impulse response (CIR) induced by acoustical multi-path to track fine-grained finger gestures. FingerIO [38] uses OFDM modulated sound frames and enables 2-D finger tracking based on the change of the echo profiles of two consecutive frames. LLAP [52] uses Continuous Wave (CW) signal to track a moving target based on the phase information of the reflected signal. VSkin [48] characterizes the propagation of structure-borne and air-borne acoustic signals to recognize gestures performed on the back of mobile devices.

Recently, several acoustic signal based authentication systems have been proposed for smartphones [16, 34, 60]. BreathPrint [16] captures the breathing sound made by a user through an embedded microphone in close proximity to the user's nose to perform biometric authentication. BiLock [60] extracts signatures from the sounds generated by a user's occlusion activities which are recorded by the built-in microphone of a smartphone or a smartwatch placed close to the user's lips to achieve biometric authentication. Lippass [34] proposes a lip reading-based user authentication on smartphones utilizing unique Doppler profiles of acoustic signals introduced by lip movement while speaking. SpeakPrint [21] extracts MFCC features in normal voice frequency and calculates mouth movement speed derived from ultrasound signal to authenticate users. EarEcho [26] takes advantages of the unique physical and geometrical characteristics of human ear canal to authenticate users using inaudible signals. However, all of these schemes require users to *hold* the device in close proximity (within a few centimeters) of the microphones to perform authentication.

Table 1: Comparison with existing works.

Method	TPR	FAR	Extra Hardware	Device Free
WiID [46]	92.80%	-	WiFi transceiver	Yes
VAuth [25]	≤97%	0.10%	Wearable	No
P2Auth [33]	≤99.55%	2.1%	Wearable	No
HandLock	96.51%	0.82%	No	Yes

Distinction with Prior Work. To the best of our knowledge, we are the first to propose an acoustic hand gesture based 2-FA system for VAs. Our approach is *device-free* and *non-obtrusive*. As we will later on show our approach is able to not only thwart emulation attempts by an attacker, but can also allow multiple users in a household to enroll with different gestures. Table 1 highlights a comparison with other existing VA authentication systems. *HandLock* achieves similar effectiveness when compared with existing approaches. However, unlike other approaches *HandLock* does not require any additional hardware and operates device free. Thus, our approach is fully compatible with existing VAs.

3 SYSTEM DESIGN

3.1 System Overview

The key principal of *HandLock* is to derive the unique hand gesture fingerprint of an individual by analyzing the acoustics signals bouncing off from the individual's hand when he/she is making a gesture. Once the received acoustic signal is prepossessed it is then compared with known fingerprints of authorized users to complete the verification step. Figure 2 shows an overview of our proposed system, which consists of five main components: signal sensing, signal processing, feature extraction, user modeling, and verification.

In the signal sensing phase, as soon as the VA enters a *sensitive operation* (i.e., operations that a user wants to limit by an 2-FA approach), such as confirming an online purchase, the embedded speaker of the VA device prompts the user to perform a hand gesture and starts emitting *inaudible* continuous wave (CW) signal. The microphone array on the VA simultaneously starts recording the inaudible sound as the user performs a gesture over the VA. In the signal processing phase, the received signal (RF) is first multiplied with the transmitted signal $\cos 2\pi ft$ and its phase-shift version $-\sin 2\pi ft$. We then use a low pass filter to get the corresponding, In-phase (I) and Quadrature (Q) signals. Given that we can extract the acoustic phase shift from the Q signal which is less susceptible to noise (as we will show in Section 3.3.2), we use the Q trace alone to extract features. Next, we extract the phase shift from Q trace and divide phase signals into small segments that contain the hand movements. These signal segments are then passed through a feature extraction process, where *HandLock* uses an *automated* feature engineering process to select the top distinguishing features

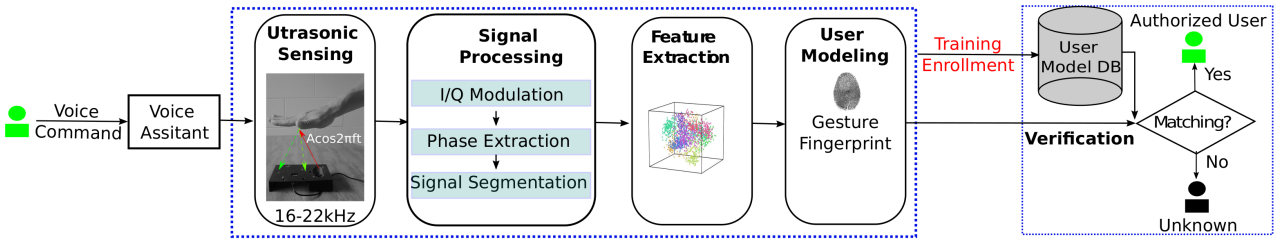


Figure 2: System overview. There are five major components: ultrasonic sensing, signal processing, feature extraction, user modeling, and verification.

across different users. Next, we use these features to train machine learning (ML) classifiers for different settings, like single-user or multi-user authentication. Lastly, *HandLock* uses the developed ML classifier to distinguish a known user from a set of unknown users. We provide more details of each step below.

3.2 Signal Sensing

In designing the signal sensing process, where *HandLock* emits audio signals we considered two factors: user experience and system performance. To make our system unobtrusive, the acoustic signal emitted by the VA is made inaudible. Therefore, *HandLock* uses sound waves with frequencies higher than 16 kHz, which are inaudible to most people and supported by Commercial-Off-The-Shelf (COTS) voice assistant devices. We transmit and record audio signals at 48 kHz. Next, to improve acoustic sensing capability, we use multiple carrier frequencies of inaudible signal of $A\cos 2\pi ft$ to improve phase detection. A gesture usually takes 1.5 $\sim$ 3 seconds and the hand trace moves up to 1 meter during this time, so the hand speed (v) is up to 0.67 m/s. The frequency shift caused by a hand movement can be measured at the microphone (f_r) using the following equations.

$$f_r = f_0 \frac{c+v}{c-v} \tag{1}$$

$$\Delta f = f_r - f_0 \tag{2}$$

The Doppler shift can be simplified as $\Delta f = 2vf_0/c$, where c is the sound speed in air and f_0 is the original transmitted frequency from the speaker, thus $\Delta f = 2 * 0.67 * 20000/343 \approx 78Hz$, when the source frequency is 20kHz. To ensure there is no interference of Doppler shifts caused by two source signals of varying frequencies, we use a frequency interval of 400 Hz. Figure 3 illustrated the recorded signal when we use 16 different source frequencies with an interval of 400 Hz, i.e., source frequencies = $\{f : f = 16000 + 400i, i = 0, 1, ..., 15\}$. We sum and normalize the 16-frequency signals as $A\sum_{f=16000}^{22000} \cos 2\pi ft$. The sound pressure was observed to be 80 dB when the signal amplitude A is set to 1. To reduce the loudness of the emitted signal to 50 dB, we set A to 0.5.

3.3 Signal Processing

Recorded acoustic signals are processed in three steps as shown in Figure 2. We provide the details of these steps here.

3.3.1 Signal I/Q Modulation. A transmitted signal arrives at the microphone from multiple paths including the structure-borne path via the body of the device, the Line-Of-Sight (LOS) propagation path via the air, and other reflection paths by surrounding objects (e.g., the user's body, the table where the device seats). Let us assume the phase of the source signal ($A\cos 2\pi ft$) changes by δ due to the Doppler effect caused by a hand movement. Let $2\pi fD(t)/c$ represents the phase delay (i.e., impact of multi-path) caused by the propagation delay of $D(t)/c$, where c is the speed of sound. The recorded inaudible signal will then be $A'\cos(2\pi ft + 2\pi fD(t)/c + \delta)$. Let ϕ represents phase shift $\frac{2\pi fD(t)}{c} + \delta$, then the received signal can be simplified using the equation shown below:

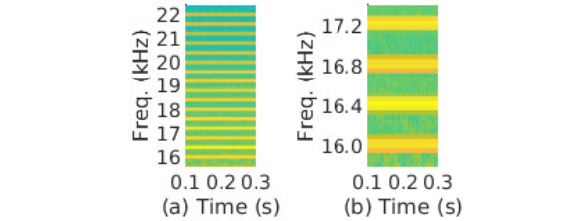


Figure 3: Received signal at 16 different frequencies.

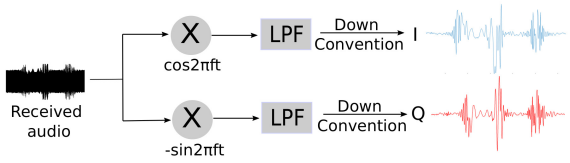


Figure 4: I/Q modulation process.

$$A'\cos(2\pi ft + \phi) = A'(\cos 2\pi ft \cos\phi - \sin 2\pi ft \sin\phi) \tag{3}$$

This equation (i.e., Eq. 3) can further be simplified by substituting the *in-phase* ($I = A'\cos\phi$) and *quadrature* ($Q = A'\sin\phi$) components of the signal as shown below:

$$A'\cos(2\pi ft + \phi) = I\cos 2\pi ft - Q\sin 2\pi ft \tag{4}$$

The general pipeline to derive I and Q signals is shown in Figure 4. The received signal is first multiplied with the transmitted signal $\cos 2\pi ft$ and its phase-shifted version $-\sin 2\pi ft$. We then use a low pass filter (LPF) to filter out frequencies greater than 24 kHz (i.e., maximum possible frequency at 48 kHz sampling rate) and get the corresponding desired I and Q traces.

$$I = LPF(2A'\cos(2\pi ft + \phi)cos(2\pi ft)) = A'\cos\phi \tag{5}$$

$$Q = LPF(-2A' \cos(2\pi f t + \phi) sin(2\pi f t)) = A' \sin\phi \quad (6)$$

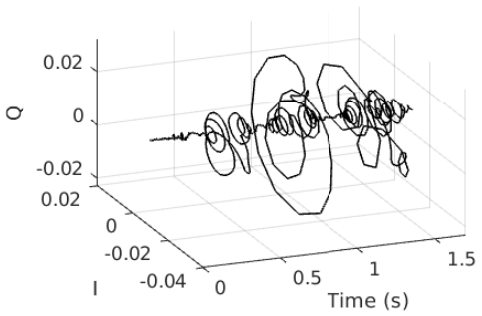


Figure 5: Corresponding I/Q signal when performing a 'Z' gesture.

3.3.2 Phase Extraction. Acoustic signal recorded by a microphone typically includes ambient noise such as human voice or environmental noise. Our data collection process took place in a lab environment that emulated a smart home living room. Hence, various smart home devices such as desktop computers, smart TVs, motion sensors and smart cameras were all present in the room while data was being collected. Also, some of the participants spoke while performing hand gestures. As we discussed in Section 3.2, the Doppler shift caused by hand movement is below 100Hz. We use a third-order Butter-worth low-pass filter [4] with a stop frequency at 100 Hz to remove undesired high-frequency noises of the modulated I/Q signal, caused by human speech and ambient noise. The filtered signal is then down sampled by a factor of 100 to reduce the system computational overhead, that is, the sampling frequency is decreased from 48 kHz to 480 Hz. Figure 5 shows the corresponding I/Q signals after the denoising and down-sampling process when a user is performing a 'Z' gesture.

However, due to hardware imperfections, the center of the recorded signal is not around 0. Figure 6 shows the offset present in I and Q trace, where I_{DC}=0.01 and Q_{DC}=-0.01, respectively. Thus, I and Q can be written as $I = I_{DC} + A' \cos\phi$ and $Q = Q_{DC} + A' \sin\phi$.

Limitations of prior works. Figure 7 highlights a short time series of IQ trace. Prior works [23, 59] approximates phase (ϕ) by considering small arcs ($\widehat{P_iP_{i+1}}$) formed by two neighbouring IQ points in a circle. Specifically, the chord length ($Chord_i$) is proportional to the angle formed by an arc when it is very small. The length of chord between two neighbouring IQ points is calculated as:

$$\begin{aligned} \widehat{P_iP_{i+1}} &= \sqrt{(I_{i+1} - I_i)^2 + (Q_{i+1} - Q_i)^2} \\ &= 2R\sin(\phi_i/2) \approx R\phi_i \end{aligned} \quad (7)$$

where R is the radius of the circle IQ points form, and ϕ is the central angle of the corresponding chord.

Using Taylor's series we know, $\sin\phi = \sum_{n=0}^{\infty} \frac{(-1)^{-n}}{(2n+1)!}\phi^{2n+1} = \phi - \frac{\phi^3}{3!} + \ldots$. Thus, $\sin\phi \approx \phi$, when ϕ is small. Assuming R is constant in a given short time, $\Delta Chord_i \approx R(\phi_{i+1} - \phi_i)$ as shown in Eq. 8, which is proportional to the phase change. We call this process Chord-based phase extraction. However, as shown in Figure 5 and Figure 7, R changes dramatically within one gesture signal, which causes an inaccurate approximation of phase change if we consider R to be constant.

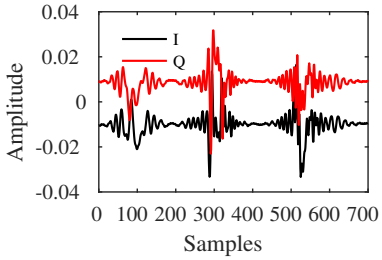


Figure 6: I/Q signal with DC offset caused by hardware imperfections.

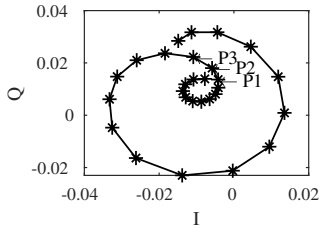


Figure 7: Sample I/Q trace.

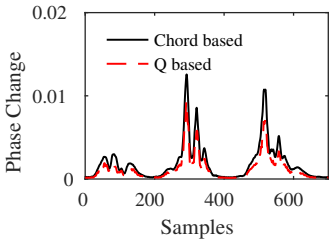
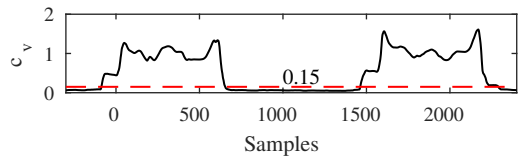


Figure 8: Phase change based on Chord-based and Q-based approach. Q-based approach does not require approximating the value of R.

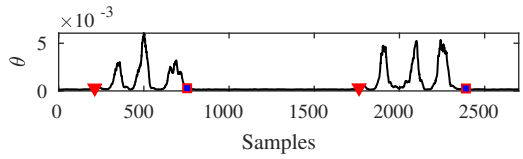
$$\Delta Chord_i = \|\widehat{P_{i+1}P_{i+2}} - \widehat{P_iP_{i+1}}\| = \|R(\phi_{i+1} - \phi_i)\| \quad (8)$$

$$\begin{aligned} Q_{i+1} - Q_i &= Q_{DC} + A' \sin\phi_{i+1} - (Q_{DC} + A' \sin\phi_i) \\ &= A'(\phi_{i+1} - \phi_i - \frac{\phi_{i+1}^3}{3!} + \frac{\phi_i^3}{3!}) = A'(\phi_{i+1} - \phi_i) \\ &= A'(\frac{2\pi f D(t)}{c} + \delta_{i+1} - \frac{2\pi f D(t)}{c} - \delta_i) \\ &= A'(\delta_{i+1} - \delta_i) \end{aligned} \quad (9)$$

Our approach. As $Q = A' \sin\phi$, the phase shift can be extracted from the Q trace alone as shown in the Eq. 9. Our approach is not dependent on approximating the value of R, and at the same time removes the DC offset and reduces the impact of multi-path propagation (eliminating both Q_{DC} and $\frac{2\pi f D(t)}{c}$). Figure 8 contrasts the Chord-based and our Q-based approach of approximating phase change. In our approach, phase change can, therefore, be represented by $\theta_i = \|\delta_{i+1} - \delta_i\| = \|(Q_{i+1} - Q_i)/A'\|$. As A' is constant, θ_i is proportional to $\|Q_{i+1} - Q_i\|$.



(a) c_v of phase change



(b) Detecting the start and end of gesture

Figure 9: Detection of the start and end point of a hand gesture within a signal.

3.3.3 Signal Segmentation. After phase extraction, we need to detect the start and end of a hand gesture to properly extract signal features. This means we need to find segments within the signal that represents hand gestures. Empirically, we found that the signal phase remains close to zero in the absence of any hand movement, whereas it is deterministically non-zero in the presence of hand movements as shown in Figure 9b. Consequently, the coefficient of variation (c_v) for any signal value is much smaller in the absence of hand movements than in the presence of hand movements. We, therefore, select a threshold T on the coefficient of variation of θ_i to detect the start and end of a gesture. *HandLock* first calculates the coefficient of variation of phase change, θ_i. After experimenting with different sliding windows, we consider a sliding window of size 200 with a step size of 10. This results in a series of coefficients of variation, where we use the moving average of 20 consecutive values to reduce the impact of outliers. We found $c_v \geq 0.15$ in the presence of hand movement. Thus, we set $T = 0.15$ to locate the start and end of a gesture movement. Figure 9a shows the c_v of θ_i for two consecutive Z gestures and Figure 9b shows the corresponding phase change for the two gestures with the start (marked as red triangle) and end (marked as blue rectangle) points marked.

3.4 Feature Extraction

The Doppler shift can be simplified as $\Delta f = 2vf_0/c$ as discussed in Section 3.2. The hand speed is proportional to the relative phase change θ_i, which can be derived from Eq. 9. Therefore, we consider θ_i as the relative speed of the hand movement. As the amplitude A' is constant, we weighted two acceleration readings to calculate the average acceleration.

$$Acc_i = \frac{\theta_{i+1} - \theta_i + \frac{\theta_{i+2} - \theta_i}{2}}{2} \tag{10}$$

Dealing with Varying Speed. Figure 10 plots the duration of performing 'Z' gestures by five users. Each box plot includes 60 samples. As we can see, different users take different amounts of time to complete the gestures. Even the same user takes different amounts of time to complete a gesture (as evident from the box plot). We must therefore handle the changes in speed from the same user. *HandLock* employs resampling so that each identified

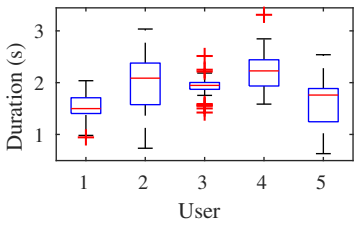


Figure 10: The distribution of duration for performing the 'Z' gesture for five users.

gesture segment contains the same number of sample points. Given a gesture segment containing M data points, *HandLock* upsamples the θ_i to N points if $M < N$, while it downsamples the θ_i to N points if $M > N$. Specifically, we apply Antialiasing Lowpass Filter [6] to resample θ_i to fixed size of N (=1000) samples as the average duration for a gesture in our dataset was around 2 seconds (with a sample frequency of 480 Hz, we set N to 1000 samples). Next, we multiply the time-series data with $\frac{M}{N}$ to normalize the value of θ_i. Thus, all gesture instances are represented by the same number of samples and are also scaled accordingly. Figure 11 illustrates the speed profiles of two users performing the Z gesture. We can see that the phase profiles are properly scaled/normalized. Similarly, we calculate the acceleration profile from phase change using Eq. 10.

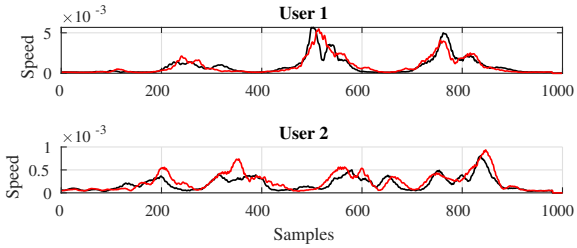


Figure 11: Speed profiles of different users performing the 'Z' gesture. Different colors represent gestures performed at different timestamp.

We observed that the phase/speed profiles of a given gesture from the same user are similar, but different from other users as shown in Figure 11. A user often performs a gesture at different speed and acceleration. To extract features, *HandLock* needs to further segment each speed and acceleration time-series data into multiple smaller chunks to capture the subtle idiosyncrasies in which the speed and acceleration changes for a given user.

Feature Vector. Table 2 lists the features we used. We extract temporal and spectral features from both the speed and acceleration time series. First, we compute 8 single-valued features for both speed and acceleration including *Mean*, *Median*, *RMS*, *STD*, *MAD*, *10th percentile*, *90th percentile*, and *median frequency*. To gain more fine-grained insights into the change in speed and acceleration, we split each gesture segment into 20 equal sized chunks and calculate *RMS*, *STD*, *MAD* and *Mean* from each chuck. To extract the spectral features like PSD from speed and acceleration, we apply FFT on each time-series data, then perform max-min normalization on the power of all frequencies. Onward, we segment the PSD values into 20 small chunks and calculate the mean value from each chunk. Similarly,

Table 2: Explored temporal and spectral features.

Domain	Feature	Description
Time	Mean	Arithmetic mean of signal strength
	Median	Arithmetic median of the signal strength
	RMS	Root mean of the squares of the signal strength
	STD	Standard deviation of the signal strength
	MAD	Mean absolute deviation of the signal strength
	10 th percentile	10th percentile value of the signal strength
	90 th percentile	90th percentile value of the signal strength
	Auto-correlation	Correlation of a signal with a delayed copy of itself
Frequency	Median Frequency	Median frequency value of the signal
	PSD	Power spectral density of the signal

we compute the signal auto-correlation coefficient of speed and acceleration, and normalize the auto-correlation coefficient; we then segment the normalized coefficient into 20 chunks and compute the mean value from each chunk. In total, we extract 256 features from both the speed and acceleration time-series data.[1]

3.5 User Modeling

HandLock adopts a supervised machine learning approach, therefore, users need to provide gestures as training data. Let us assume the number of enrolled users in a household is $\mathcal{U}$. Each enrolled user is labeled differently and is a member of the positive class set $\mathcal{U}_\mathcal{P}$. *HandLock* comes with a set of negative samples from unknown users (e.g., five users) represented as $\mathcal{U}_\mathcal{N}$. *HandLock* supports two forms of enrollment of multiple users ($1 \leq \mathcal{N} \leq \mathcal{U}$) under two different scenarios: 1) enrollment of Multiple Users Same Gesture (*MUSG*), and 2) enrollment of Multiple Users Multiple Gestures (*MUMG*).

Single-user vs. Multi-user Setting. When the user first enrolls his/her gesture, *HandLock* prompts for gesture options through the voice interface and the user provides samples of a given gesture for training. Multiple users can enroll with a gesture, where gestures can be of the same or different type. This multi-user setting is considered as a multi-class classification problem.

Feature Selection. To find the best feature for *HandLock*, we explore all the features using the FEAST toolbox [5, 13] and select the Joint Mutual Information criterion (JMI) for ranking the features.

Balanced Learning. *HandLock* uses gesture samples from unknown users as samples from the negative class. If we assume each user provides k (e.g., 20) samples for a given gesture then $\mathcal{U}_\mathcal{P}$ is the minority class while $\mathcal{U}_\mathcal{N}$ is the majority class (as we assume samples from five unknown users). The dataset size is imbalanced due to unequal size of $\mathcal{U}_\mathcal{P}$ and $\mathcal{U}_\mathcal{N}$. Therefore, we need to up sample the authentic user's data to achieve equal class representation. We test Synthetic Minority Over-sampling (SMOTE) [17] and Adaptive Synthetic Sampling (ADASYN) [30] methods which are two popular up-sampling approaches. We selected ADASYN as our up-sampling approach as it provided better performance. The results are available in Section 4.1.2.

[1]Per time-series data we compute $(8 + 20 \times 4 + 20 + 20) = 128$ features.

3.6 Verification

Once we have features extracted from the hand gesture, we use supervised learning to identify the legitimate user. *HandLock* collects training data from the authorized user to build one binary or multi-class classification model. We explore four classifiers including Random Forest (RF), Decision Tree (DT), k-nearest neighbors (KNN), and Support Vector Machine (SVM).

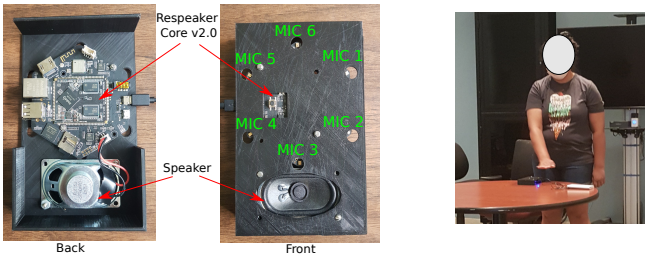
Threat Model and Attack Settings. Our threat model assumes that an adversary can interact with the victim's VA. The attacker then attempts to bypass *HandLock* by performing a hand gesture. We consider three settings, where the adversary either knows or does not know the victim's chosen gesture. Therefore, we consider the following attacks.

- *Random Gestures:* The attacker does not know the exact gesture performed by the victim, but attempts to authenticate himself/herself by performing a random gesture.
- *Gesture Mimicry:* The attacker knows the exact gesture performed by the victim, for example, by observing the victim perform a gesture during an authentication session.
- *Replay Attack:* The attacker places a nearby microphone to record the exact gesture performed by the victim and attempts to authenticate by replaying the recorded signal.

4 EVALUATION

In this section, we perform a comprehensive analysis of *HandLock* under various settings to evaluate its accuracy, stability, resiliency to attacks, and system-level performance. First, we evaluate the overall accuracy and efficiency of our system (Section 4.1), covering five gestures as shown in the Figure 13. We then examine its resilience against random gesture mimicry and replay attacks (Section 4.2). Next, we evaluate the sensitivity of our system (Section 4.3) by analyzing the impact of the following factors: number of users enrolled (Section 4.3.1), number of microphones used (Section 4.3.2), temporal stability (Section 4.3.3), distance between hand and VA (Section 4.3.4), Cross-environment stability (Section 4.3.6), and ambient noise (Section 4.3.5). Lastly, we evaluate system-level performance metrics like processing time and memory consumption in Section 4.4.

Device Setup. As current commercial VAs are not allowed to log raw audio, we implement *HandLock* using a Seeed's ReSpeaker Core V2.0 [8], which runs on GNU/Linux operating system and is designed for voice interface applications with a quad-core ARM Cortex A7, running up to 1.5GHz with 1GB RAM. Figure 12a shows the device setup of *HandLock*. The board is equipped with a six microphone array — similar to how microphones are distributed inside an Amazon Echo Dot [1]. We wire it to an external 3W speaker AS07104PO-LW152-R [41]. We use a 3D-printed casing to hold the microphone array and speaker. The device is powered by a 10000 mAh Mi Power Bank 2 [7]. We play Continuous Wave sound and record the 6-channel audio simultaneously when collecting gesture data. Participants were invited to interact with our prototype VA located inside a lab space that emulates a smart home living room equipped with a table, sofa, desktop computer, smart TV, smart lights, motion sensors and smart cameras. Figure 12b shows the lab



(a) Device setup (b) Lab setup

Figure 12: We used Seeed's ReSpeaker Core V2.0 [8], which is equipped with six microphones. Lab setup showing how participants interacted with our prototype VA.

Table 3: Demographics of participants.

Attribute	Values (count)
Age	18-24 (6), 25-34 (27), 35-44 (11), 55-64 (1)
Gender	Male (21), Female (24)
Education	High school graduate (2), Bachelor's Degree (17), Master's Degree (16), Doctorate Degree (10)
Student	Yes (28), No (17)
VAs owned	0 (23), 1 (17), 2 (1), 3 (1), more than 3 (3)

setting where a participant is performing a hand gesture with our VA system.

Participants. We obtained necessary IRB approval to collect data from participants. The total participation time was around 45 minutes (providing breaks between sessions). Participants were compensated ($15) for their time. Table 3 summarizes the participant details. In total, we recruited 45 participants, 21 identified themselves as males, while 24 identified themselves as females. Around 73.33 % (33/45) of participants were aged between 18 and 34. A majority, 95.56 % (43/45) of them reported to have earned a bachelor or higher educational degree, and 62.22 % (28/45) of participants were current students in a university, while the remaining participants were not students. 48.89 % participants reported owning one or more smart home voice assistant (VA) devices such as Google Home, Amazon Echo, or Xiaomi.

Data Collection Process. The data was collected in our lab from January, 2020 to March, 2020.[2] Before collecting any data, each participant was trained for 5 minutes, so that he/she understood how the data collection process works. We randomly split the 45 participants into 39 benign users ($\mathcal{U}$) and 6 attackers ($\mathcal{A}$). For evaluation purposes we consider five popular gestures: 'Z', 'W', 'X', '✓' and '☆' (as shown in Figure 13). We asked each participant to continuously perform a given gesture with a small pause between two subsequent gestures, where participants placed their hand anywhere in the range of 5 $\sim$ 30 cm from our prototype VA. We found that each participant typically performed 10 $\sim$ 20 gestures in a single minute. In order to make our data collection process more realistic, we asked the participants to take a rest and walk around after each one-minute data collection session. We then repeat the whole process until we get the required number of samples.

[2]We followed COVID-19 safety protocols mandated by our IRB office.

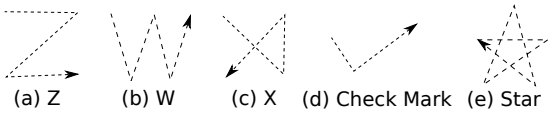


Figure 13: Different types of gestures evaluated.

Table 4: Summary of the various datasets collected.

Dataset	Participants	Gestures	Samples/gesture	Total samples
1*	39	5	60	39 × 5 × 60=11700
2†	10	5	30	10 × 5 × 30 × 2=3000
3‡	10	30	1	10 × 30 × 1=300
4◇	6	5	30	6 × 5 × 30=900

* performed in a single day; † collected after one week and one month; ‡ random gestures; ◇ attacker emulates vicitm's gesture

Datasets. We summarize the collected datasets in Table 4. In Dataset 1, for each user, we collected 60 samples for each of the five gestures illustrated in Figure 13. Dataset 2 is collected for evaluating system stability (Section 4.3.3) — 10 out of the 39 participants were requested to come back to the lab and provide new data. They came back one week and one month after their first visit. Dataset 3 is used to evaluate random gesture attacks, where we asked another 10 out of the 39 users to perform 30 random gestures, which included the numbers from 0 to 9 and the alphabets from A to T. In Dataset 4, six other participants were asked to take the role of an attacker and observe how a given victim makes a gesture. Each attacker observed a victim perform (through recorded video) the five different gestures and was asked to emulate each of the gestures 30 times. Note that participants performing additional tasks were compensated accordingly.

In addition to providing hand-gesture data, each participant was also asked to complete a *post-study* survey to provide feedback about their experience with our system and expectations for a commercial deployment. We provide more details on the post-study survey in Section 5.

Evaluation Metrics. For any decision made by a classifier, there are four possible contingencies: (1) accept a legitimate user (true positive or TP), (2) wrongly accept an illegitimate user (false positive or FP), (3) reject an illegitimate user (true negative or TN), and (4) reject a legitimate user (false negative or FN). We adopt the following well-known metrics that are typically used for assessing any authentication system [47]: a) False Reject Rate ($FRR = \frac{FN}{FN+TP}$) is the probability that the system wrongly identifies a legitimate user; b) False Accept Rate ($FAR = \frac{FP}{FP+TN}$) is the likelihood that the system wrongly accepts an illegitimate user; c) Precision ($P_r = \frac{TP}{TP+FP}$) refers to as positive predictive rate; d) Recall ($R_e = \frac{TP}{FN+TP}$) refers to as the true positive rate (TPR) or sensitivity; e) F-Score ($F_1 = \frac{2\times P_r \times R_e}{P_r+R_e}$) is the harmonic mean of the precision and recall. We perform 10 runs and report the average values.

4.1 Overall Accuracy

We use Dataset 1 to evaluate the overall performance. We consider data from all the six microphones (as shown in the Figure 12a) for our evaluation. We will evaluate the impact of the number of microphones used in Section 4.3.2.

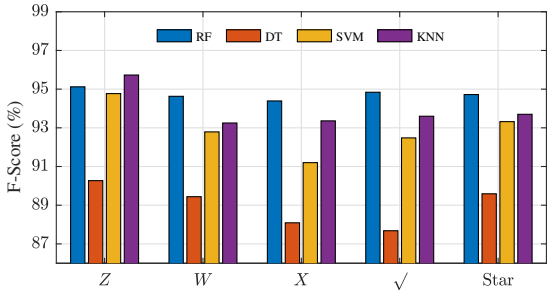


Figure 14: F-Score of different classifiers. RF has the best average F-score across all five gestures.

4.1.1 Different Classifiers. To build a binary classification model, we randomly select 30 samples out of the 60 samples per gesture from each user as training set and use the remaining 30 samples as test set. We randomly label 5 users' data as *negative class* and 34 users' data as *positive class*. We use all the features in the feature vector to train our model. For each binary classification model, we have 30 positive instances and 30×5 negative instances as training set. We use the remaining 30 positive instances and select six random negative instances from the remaining 30 instances from each user in the negative class (i.e., five users) as test set. We rerun 10 times for each user and calculate the average F-Score for 34 users.

We compare the performance of four classifiers including RF, DT, SVM and kNN. For the RF classifier, we use the Bagging algorithm and test different numbers of trees ranging from 60 to 240, and select the best number as 120. For DT, we select the maximum number of splits as 7. To generate each single binary classification model in SVM, we use the implementation of SVDE with 10-fold cross validation in libSVM [15] and chose the best complexity parameter for Radial Basis Function (RBF) through grid Search. To select the number of neighbors for kNN, we run tests with k ranging from 1 to 10 and find the best performance when k=5. Figure 14 shows the F-Score of the four classifiers covering five gestures. RF had the best average F-score across the different gestures. We, therefore, will use RF for the rest of our evaluations.

4.1.2 Imbalanced data vs. Balanced data. In Section 3.5, we discussed the challenges that a class imbalance might impose. We test two widely used upsampling algorithms including SMOTE [17] and ADASYN [30]. ADASYN shows a better performance than SMOTE on our dataset. We, therefore, adopt ADASYN to upsample the positive class instances. Table 5 presents the overall performance of *HandLock* while using RF with and without upsampling. After balancing the positive class, the FRR decreased from 5.49% to 3.49% and F-Score improved from 96.62% to 97.77%, while the average FAR decreased from 1.14% to 0.82%.

4.1.3 Varying Training Features. To evaluate the impact of the number of top features, we apply ADASYN to resample each 30 positive instances. We then use the feature selection library named FEAST toolbox [5] and use the Joint Mutual Information criterion (JMI) to determine the top features. We next vary the number of top features from 300 to 1200 in increments of 60 and compute the F-Score. We repeat the experiment 10 times for each subset of the top features, while considering data from all 34 participants

Table 5: Performance of RF classifier with imbalanced/balanced dataset (using ADASYN upsampling technique).

Gesture	FRR	FAR	Precision	Recall	F-Score
Z	4.86/3.71	1.35/0.76	98.62/99.26	95.14/96.29	96.85/97.69
W	6.19/3.14	1.05/1.05	98.90/98.95	93.81/96.86	96.29/97.83
X	4.95/4.10	2.14/1.90	97.84/98.17	95.05/95.90	96.43/96.90
✓	6.10/3.14	0.96/0.29	99.00/99.70	93.90/96.86	96.3898.22
☆	5.33/3.33	0.19/0.10	99.80/99.91	94.67/96.67	97.17/98.20
Avg.	5.49/3.49	1.14/0.82	99.82/99.20	94.51/96.51	96.62/97.77

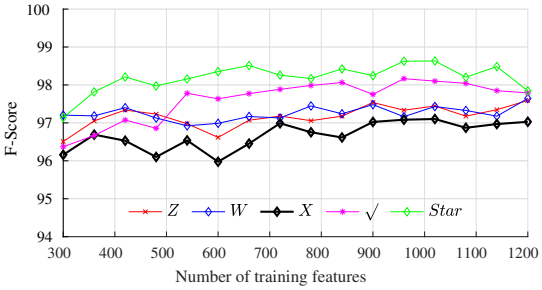


Figure 15: Impact of the number of top training features. Using the top 960 gives us the best performance.

performing the five different gestures. Figure 15 shows that the average F-Score for different number of top features. The results improve from 96.97% to 97.67% as we expand the number of top features from 300 to 960. After that, F-Score seems to plateau. We, therefore, use the top 960 features as our training features for the reminder of the evaluations.

4.1.4 Varying Training Size. To make *HandLock* user friendly, the enrollment effort for a new user is a critical factor. We consider the performance of the classifiers in the presence of limited training samples. For this experiment, we vary the training set size from 10 samples to 40 samples in increments of 5 samples, and test the remaining samples. For each training set, we also apply ADASYN to resample the positive samples so that the number of positive and negative samples are balanced. Figure 16 shows the evaluation of the TPR with the increasing training set size. The result shows that as the training set size increases the TPR also rises. However, we see that after 30 samples per class the average TPR plateaus at

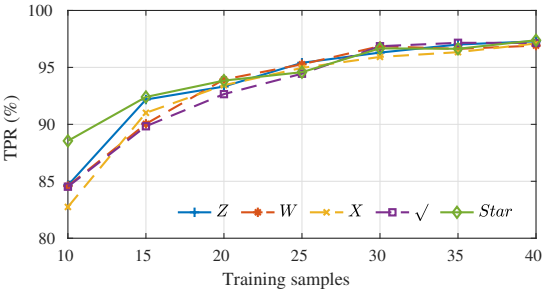


Figure 16: Impact of training set size on TPR. Using 30 training samples per class is sufficient.

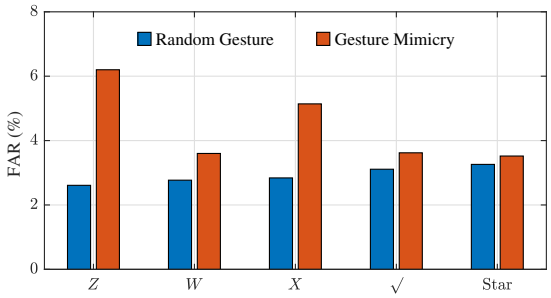


Figure 17: FAR under both random guess and mimicry attack.

around 96%. This suggests that we do not need too many training samples to construct a good predictive model.

4.1.5 Different Negative Dataset. To analyze the impact of the selection of negative samples on classification, we randomly selected 5 users out of the 39 participants as representative of the negative classes while the remaining 34 users as positive class. We test the model against 'Z' gesture and repeat the whole process 20 times. We find the average TPR to be 94.94 with a 95% confidence interval of [94.41, 95.57] (while the average the F-Score is 96.53). This result suggests that our approach is not dependent on the selection of negative samples and we can easily bootstrap our approach with any negative samples.

4.1.6 Expected Number of Attempts. The expected number of attempts required to authentic a user is another important usability factor. We record the number of attempts needed to successfully authenticate all gestures cross all users. The overall average number of attempts required is 1.05 ± 0.24. With two attempts, *HandLock* can achieve a TPR of 99.35%, while the accuracy is 99.91% with three attempts. Thus, we can limit to *three attempts* to thwart attacks and fall back on app based authentication.

4.2 Resilience to Attacks

To evaluate the resilience to attacks, we use 30 samples from each user as training set. We apply ADASYN to upsample each 30 positive instances and select the top 960 features to train a RF model (this constitutes the best configurations as described in the previous section).

4.2.1 Random Gesture Attack. We use Dataset 3 (see Table 4 for details) to assess the system's resilience to random gesture attack. We test all 300 random gestures collected from 10 users to authenticate *HandLock* against each user's binary classification model. We test 10 times for each user model and calculate the average rate of recognizing the gesture for the enrolled user. The blue bar in Figure 17 shows the average FAR of random gesture attack for each gesture, which is 2.61%, 2.77%, 2.84%, 3.11%, and 3.26% for 'Z', 'W', 'X', '✓', and star ('☆'), respectively.

4.2.2 Gesture Mimicry Attack. We use Dataset 4 (see Table 4 for details) to evaluate the system's resilience to gesture mimicry attack. Here, six attackers individually mimic the gesture of a given victim for all five gestures. Figure 17 depicts the average FAR of each gesture under this attack setting. Overall, the FAR for the 'Z' gesture is close to 6.2%, followed by the 'X' with 5.14%. However, the '☆' has the lowest FAR of 3.52%. The most probable reason for this is that simple gestures such as 'Z' and 'X' are simple and more likely to be properly emulated, whereas the '☆' gesture is more complex and less prone to emulation attack.

We also test if other users' same gestures can be used to bypass *HandLock*. We select 30 samples from each of the other 33 users (i.e., 990 test samples) excluding the user whose samples are used to train a model, to test the robustness of *HandLock*. The FAR is 7.17%, 6.68%, 5.28%, 4.13%, and 3.76% for 'Z', 'W', 'X', '✓', and '☆', respectively. Thus, we see similar level of resiliency against mimicry attacks. Furthermore, by limiting authentication attempts to three, we can drastically thwart adversarial attacks.

4.2.3 Audio Replay Attack. To emulate replay attack, we select a high performance UMA-8 USB microphone array [10], which can record audio at $48kHz$ sampling rate. We placed the microphone array close to the target VA at a distance of 20 *cm* while a user was performing 'Z' gestures. Next to we use a Sony SRS-X5 louder speaker [9] to replay the recorded gesture session at 70 dB from a very close distance of 5 *cm* from the VA to give the attacker the best chance of authentication. We collected 60 replayed gesture samples and tested the samples using our original trained model. *HandLock* achieves a FAR of 3.33%. This suggests that *HandLock* is robust against replay attacks where the attacker has the capability to record authentication sessions from a very close distance.

4.3 Sensitivity Analysis

4.3.1 Multiple users settings. To study the impact of the number of users among which *HandLock* should distinguish, we consider two enrollment scenarios: 1) different users enroll with the same single gesture (MUSG); 2) different users enroll with different gestures (MUMG). We repeat the same experiments as described in Section 3.5 for four different values of enrolled users, ranging from $\mathcal{U} = 2$ to $\mathcal{U} = 5$, in these two scenarios. We use the highest number of users as $\mathcal{U} = 5$ because more than 96% households in the US have less than 6 members according to the United States Census Bureau [2]. To calculate the overall accuracy of *HandLock* for each gesture, we randomly selected 5 users out of 39 participants as negative class, and each of the remaining $\mathcal{N} = 34$ users as positive class with *unique* positive labels. We test each user with 30 samples as positive class and 30 samples from the negative class and calculate the average TAR and FAR for $\mathcal{K} = 5$ gestures across all users.

In enrollment scenario 1, we repeat the experiment 5 times, where each experiment runs $\mathcal{K}.{}^{\mathcal{N}}C_{\mathcal{U}} = \mathcal{K}\frac{\mathcal{N}!}{\mathcal{U}!(\mathcal{N}-\mathcal{U})!}$ times. The blue bar in Figure 18 show the TPRs. We see that the average accuracy of *HandLock* reduces as the number of enrolled users increases. Nonetheless, we observe that the average TPR of *HandLock* is above 90% for 5 users. In enrollment scenario 2, the number of negative training samples vary with the number of unique gestures the enrolled users use. For example, if $\mathcal{K} = 2$ users enroll with two different gestures, the total training samples for negative class will be $2 \times 30 \times 5$, which includes 30 samples per gesture from five users representing the negative class. Consequently, we run each experiment ${}^{\mathcal{N}}P_{\mathcal{U}} = \frac{\mathcal{N}!}{(\mathcal{N}-\mathcal{U})!}$ times. We again repeat each experiment 5 times. The red bar in Figure 18 show the TPRs. The result

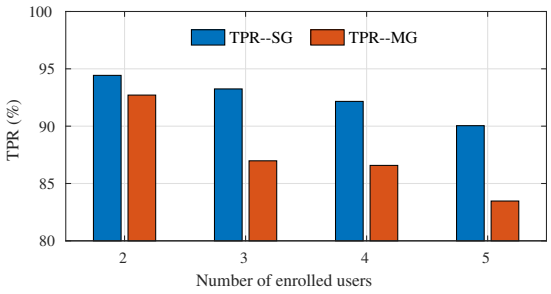


Figure 18: Impact of the number of users enrolled. As the number of enrolled user increases, the TPR slightly drops.

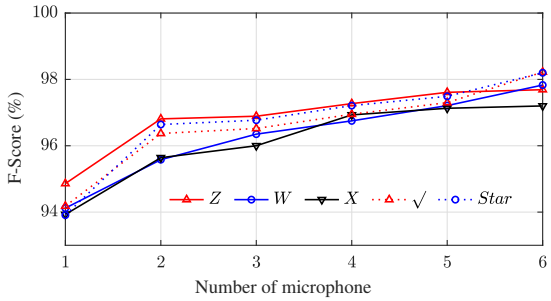
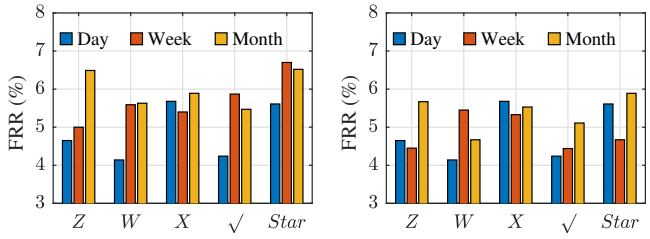


Figure 19: Impact of the number of microphones used. Combining multiple microphones results in improved accuracy, but computational overhead increases.

shows that the TPRs decrease more compared to the TPRs found in scenario 1. However, the FAR remains below 0.78% for $\mathcal{U} = 5$ users as the negative class contains more information (i.e., exposed to more negative instances).

4.3.2 Impact of the number of microphones used. To evaluate how the performance is impacted by the number of microphones used, we select n ($1 \leq n \leq 5$) out of the six microphone data. Consequently, we have ${}^6C_n$ combinations for each gesture. We run 5 experiments including any n microphones using the 960 top features. Figure 19 shows that the average F-Score, in general, improves for all gestures when increasing the number of microphones. The average F-Score using two microphones for five gestures is above 96%, which means *HandLock* still can achieve reasonable accuracy even with 2 microphones.

4.3.3 Temporal stability. The motivation behind evaluating the accuracy of *HandLock* using training and testing samples from different days is that in the real world, the user will provide training samples only on the first day when setting up *HandLock*, and then *HandLock* should be able to identify them on subsequent days. To calculate the overall accuracy of *HandLock* for each of the five gestures using training and testing samples from different days we randomly chose 10 participants out of our 39 and collected additional data after one week and one month of their first visit (i.e., Dataset 2 in Table 4). Figure 20a shows the FRR for 5 gestures for three different time-periods, where all models are trained on the data from the first visit. We can see that in general FRR slightly increases. However, we can potentially reuse high confidence samples as training data and rebuild the model periodically.



(a) No incremental learning **(b) Incremental learning**

Figure 20: FRR over time (day, week and month).

To rebuild the model periodically, we compute the ROC curve on the data collected from the first visit (i.e., Dataset 1 in Table 4) and determine the threshold (i.e., prediction probability) for the optimal operating point. We next adopted an incremental learning approach, where we divide the 30 test samples from the following week and month into 6 batches where each batch contains 5 samples. We then reuse the test samples that are correctly predicted with a probability greater than the optimal threshold (0.75) to retrain our model. Figure 20b shows the FRR when using incremental training approach. We see that FRR reduces compared to Figure 20a.

4.3.4 Impact of distance . To evaluate the impact of distance from the hand to the microphone, three participants were asked to perform 20 samples of the 'Z' gesture at varying distances of 10 *cm*, 20 *cm*, 30 *cm*, 40 *cm*, 50 *cm*. We test the model trained on Dataset 1 (see Table 4 for details) using the samples collected under different distances. After running the experiment 10 times we found the average TPR to be 93.50%, 95.10%, 94.58% and 86.28% for the distance of 10 *cm*, 20 *cm*, 30 *cm* and 40 *cm*, respectively, compared to the reference TPR of 95.17% for the 3 users. The result shows that our system has stability within the distance of $10 \sim 30$ *cm*, but the performance drops by 9% at 40 cm distance. Our segmentation method only successfully segments 20.15% of samples at the distance of 50 *cm*, as the Q traces are too weak to surpass our threshold T. Thus, our system currently performs well for any distance ≤ 30 cm.

4.3.5 Impact of ambient noise. To evaluate the impact of environmental noise, we set up our device one meter away from a TV broadcasting news with a sound pressure of 80 dB. Three participants were asked to perform 20 samples of 'Z' gesture. We observe that the background noise is below 15 kHz, confirmed by analyzing the spectrogram of the data collected. We test the model trained on Dataset 1 (see Table 4 for details) for the 'Z' gesture and test on the sample collected under ambient noise to determine robustness. After rerunning the experiment 10 times, the average TPR is 95.10%. Compared to the reference TPR of 96.29% (see Table 5) we can see that the ambient background noise typically found at homes does not significantly impact our system.

4.3.6 Cross-environment analysis. To evaluate the impact of different surroundings (e.g., rooms), we collected one additional dataset under home setting (e.g., inside a bedroom). We collected 60 samples of 'Z' gesture. We then test the model trained on Dataset 1 (see Table 4 for details) for the 'Z' gesture. The average TPR was around 89.33%. To improve the robustness, we adopted an incremental learning approach, where we divided the 60 test samples

into 12 batches with each batch containing 5 samples. We then use high confidence testing samples (i.e., test sample predicted probability ≥ 0.75) to retrain our model. We found that with incremental learning approach, *HandLock* can achieve a TPR of 98.5%.

4.4 System Performance

Next, we present the processing latency and memory consumption of our implementation of *HandLock* on a personal computer which is equipped with Intel i7-2600 3.40GHz processor and 16 GB RAM. Identifying a legitimate user for a given gesture sample (in the binary classification setting) using six audio-stream data, took on average 56 ms. When generating classification models, our implementation used an average memory of 214 MB. Respeaker Core V2 is equipped with quad-core ARM Cortex A7 (1.5GHz) and 1 GB RAM. Thus, our memory requirement can easily be fulfilled by commodity VAs.

5 USER STUDY AND FEEDBACK

In this section, we analyze the usability of *HandLock* by conducting a post-study survey, where all participants providing hand gesture data complete a survey at the end of the data collection process. We also ask participants to compare *HandLock* against an existing app-based 2-FA approach, where users authenticate by *tapping* a button on the app.

Data Collection Procedure. Each participant in our study completed a post-study survey, where they were asked about their *experience* and *expectations* using our proposed gesture-based authentication technique. We also asked participants questions about comparing our approach with a tap-based 2-FA approach on a mobile app. Table 6 list the basic questions asked. Participants also answered the SUS (System Usability Scale) questionnaire, which is a well-known standard for measuring the usability of software systems and consists of 10 standard usability questions, each with five possible answers (5-point Likert scale, where 1 represents strong disagreement and 5 represents strong agreement) [12] (the 10 questions are provided in Table 7 in Appendix A). We compute SUS score for both *HandLock* and the app-based 2-FA approach.

Takeaways. In analyzing the user responses, we filtered out one user as he/she completed the survey in 75 seconds which is below the median response time (546 seconds) by a factor of 7 (i.e., the user most likely did not pay much attention to the questions).

First, we analyze user's experience and expectations for using *HandLock*. 93.18% (41/44) participants consider *HandLock* to be either extremely easy or somewhat easy to use as shown in Table 6. 88.64% (39/44) participants said they would probably or definitely deploy *HandLock* on their own VAs. Around 77.27% of participants felt *HandLock* is better (i.e., either somewhat or much better) than using a mobile app-based 2-FA approach. Our survey also asked participants for comments about the pros and cons of our approach. Following are some examples of the comments received.

Table 6: Summary of post-study survey and responses.

Question	Response (count)
How easy was it to use our system (*HandLock*)?	Extremely easy (24), Somewhat easy (17), Neither easy nor difficult (3), Somewhat difficult (0), Extremely difficult (0)
What would be an acceptable duration for authentication?	Less than 1 sec (9), 1–4 sec (28), 4–8 sec (6), More than 8 sec (1)
At most how many authentication attempts would you be comfortable making?	1 (6), 2 (13), 3 (20), 4 (2), 5 (3)
Would you deploy *HandLock* on your voice assistant?	Definitely yes (14), Probably yes (25), Might or might not (1), Probably not (4), Definitely not (0)
Compare *HandLock* with an app-based 2-FA approach (i.e., tapping)	Much Better (16), Somewhat better(18), About the same (10), Somewhat worse (0), Much worse (0)

> P1: I think range is a very important factor in terms of user experience. If I can unlock the assistant while sitting on a bed/sofa, that would be quite impactful.
> P18: Advantage is you only need yourself (no phone on you). Disadvantage is your hands have to be free.
> P21: More convenient than other authentications. The sound could be more pleasant, if it's a tune.
> P28: Very hygienic, no need for other equipment or network. Looking forward to it, I hope it can be put into the market and applied in practice as soon as possible.
> P38: Handlock – gesture-based authentication is a very simple, efficient and fast way of getting access. The only thing I would like to suggest is it should be compact.

As we can see from the comments, in general, the participants felt that the system was easy to use. There were some concerns about being close to the device and having full hand functionality. However, we feel these are concerns that are not likely to manifest too often in reality.

The authentication duration plays a vital role in assessing usability. Vast majority of the (63.64 %; 28/44) participants report that the acceptable duration for authentication should lie within $1 \sim 4$ seconds. We computed the average duration for performing a single gesture to be around 1.89 (±0.64) seconds. Since, the average processing latency is 56 milliseconds (as we reported in the Section 4.4), the total end-to-end verification time required by *HandLock* is around 2 seconds. Thus, our approach would satisfy the expectation for the majority of the users. In terms of number of authentication attempts, 88.64 % participants (39/44) would tolerate at best three attempts. In Section 4.1.6, we show that the expected number of attempts required for a successful authentication is around 1.05 ± 0.24, which is below the tolerance level indicated by the participants.

Next, we will compare *HandLock* with other alternatives using SUS scores. A SUS score of above 68 is typically considered above average and anything below 68 is below average [12]. Based on the participants' responses the mean SUS score for *HandLock* is 71.88 ± 18.69, while the mean SUS score for an app-base 2-FA is

62.96 ± 12.25. One reason for *HandLock* obtaining a higher SUS score is that the participants consider the system as device-free and easy-to-use as evident from their free text comments. For the participants who own VAs, the mean SUS score for *HandLock* and app-based 2-FA is 74.64 ± 17.58 and 64.05 ± 15.26, respectively. For the participants who did not own VAs, the mean SUS score for *HandLock* and app-based 2-FA is 69.35 ± 19.69 and 61.96 ± 13.53, respectively. Thus, in general, participants found our approach to be more usable than an app based 2-FA approach.

6 LIMITATIONS AND DISCUSSION

Sound Generation. The majority of our participants could not hear the generated sound. However, a few participants could hear the sound. The National Institute of Occupational Safety and Health (NIOSH) defines hazardous noise as sound that exceeds 85 dB over a typical 8-hour day [22], which could cause hearing loss. In reality *HandLock* would play near-ultrasound sounds just for a few seconds at the sound pressure level of 50 dB (SPL) during the authentication process. Thus, our approach most likely will not cause any hearing issues. To further improve the usability, we can either use higher frequencies sound (i.e., above 18kHz) or use low frequency music/tone (e.g., $\leq 8kHz$), something we plan to explore in the future.

Simultaneous Movements Currently, we assume that at any given time, only a single user performs a predefined gesture. *HandLock* further assumes that while a user performs a predefined gesture, there are no background movements close to the device. If multiple users perform predefined gestures simultaneously or if there are background movements close to the device, *HandLock* may not identify the users from the gestures. *HandLock*'s accuracy, however, is not impacted significantly by the movements of people 3 meters away from the device as we observed that the signal interference is very weak from such distance.

Distance Limitation. In our setting, the user performs the gesture on top of the device from a distance of 5 to 30 cm. Millisonic [50] made a preliminary effort towards tracking gestures at room scale. In future, we plan to explore augmenting *HandLock* to enable user identification from gestures at longer distances.

7 CONCLUSION

In this paper, we present a new modality of acoustic signal based 2-FA system for smart home voice assistants, called *HandLock*. *HandLock* extracts unique movement characteristics of a user's hand gesture. We extensively evaluated *HandLock* using a large data set of over 15,000 gesture samples, covering five common gestures and showed that it can achieve an average TPR of 96.51% across 34 users while the FAR is 0.82%. However, *HandLock* can achieve a TPR of 99.91% with three gesture attempts. We also extensively evaluate *HandLock*'s accuracy, stability and resiliency to attacks under various settings. We believe this simple, yet effective 2-FA approach is a first step towards helping consumers better protect *sensitive operations* carried out by voice assistants.

ACKNOWLEDGMENTS

We thank our anonymous reviewers for their valuable feedback. We give special thanks to all the participants who took the time to participate in our data collection study. This material is based upon work supported in parts by the National Science Foundation under grant number CNS-1849997. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] 2016. *Teardown Tuesday: Amazon Echo Dot v2.* Retrieved March 25, 2020 from www.allaboutcircuits.com/news/teardown-tuesday-amazon-echo-dot-v2/
[2] 2019. *Historical Households Tables.* Retrieved March 25, 2020 from https://www.census.gov/data/tables/time-series/demo/families/households.html
[3] 2019. *The Smart Audio Report from NPR and Edison Research, Spring 2019.* Retrieved May,25 2020 from https://www.edisonresearch.com/the-smart-audio-report-from-npr-and-edison-research-spring-2019/
[4] 2020. *Butterworth filter.* Retrieved May 30, 2020 from www.mathworks.com/help/signal/ref/butter.html
[5] 2020. *FEAST: A Feature Selection Toolbox for C and Matlab.* Retrieved March 25, 2020 from http://www.cs.man.ac.uk/~gbrown/fstoolbox/
[6] 2020. *MatLab resample.* Retrieved March 26, 2020 from www.mathworks.com/help/signal/ref/resample.html
[7] 2020. *Mi Power Bank.* Retrieved March 25, 2021 from https://www.mi.com/sg/battery2/
[8] 2020. *Seeed's ReSpeaker Core v2.0.* Retrieved March 30, 2020 from https://wiki.seeedstudio.com/ReSpeaker_Core_v2.0/
[9] 2020. Sony SRS-X5 speaker. https://www.sony.com/electronics/support/speakers-wireless-speakers/srs-x5.
[10] 2020. UMA-8 USB microphone array V2.0. Retrieved March 25, 2020 from https://www.minidsp.com/products/usb-audio-interface/uma-8-microphone-array
[11] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin RB Butler, and Joseph Wilson. 2019. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. (2019). https://dx.doi.org/10.14722/ndss.2019.23362
[12] John Brooke. 1996. SUS: a 'quick and dirty' usability scale. *Usability evaluation in industry* (1996), 189.
[13] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. 2012. Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection. *J. Mach. Learn. Res.* 13, null (Jan. 2012), 27–66. https://doi.org/10.5555/2188385.2188387
[14] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden Voice Commands. In *25th USENIX Security Symposium (USENIX Security 16).* USENIX Association, Austin, TX, 513–530. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/carlini
[15] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* 2, 3, Article 27 (May 2011), 27 pages. https://doi.org/10.1145/1961189.1961199
[16] Jagmohan Chauhan, Yining Hu, Suranga Seneviratne, Archan Misra, Aruna Seneviratne, and Youngki Lee. 2017. BreathPrint: Breathing Acoustics-Based User Authentication. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (Niagara Falls, New York, USA) *(MobiSys '17).* Association for Computing Machinery, New York, NY, USA, 278–291. https://doi.org/10.1145/3081333.3081355
[17] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357. https://doi.org/10.1613/jair.953
[18] Huijie Chen, Fan Li, and Yu Wang. 2017. EchoTrack: Acoustic device-free hand tracking on smart phones. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications.* 1–9. https://doi.org/10.1109/INFOCOM.2017.8057101
[19] Si Chen, Kui Ren, Sixu Piao, Cong Wang, Qian Wang, Jian Weng, Lu Su, and Aziz Mohaisen. 2017. You Can Hear But You Cannot Steal: Defending Against Voice Impersonation Attacks on Smartphones. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS).* 183–195. https://doi.org/10.1109/ICDCS.2017.133
[20] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. 2020. Devil's Whisper: A General Approach for Physical Adversarial Attacks against Commercial Black-box Speech Recognition Devices. In *29th USENIX Security Symposium (USENIX Security 20).* USENIX Association, 2667–2684. https://www.usenix.org/conference/usenixsecurity20/presentation/chen-yuxuan

[21] Haipeng Dai, Wei Wang, Alex X. Liu, Kang Ling, and Jiajun Sun. 2019. Speech Based Human Authentication on Smartphones. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SAHCN.2019.8824958

[22] Eileen Daniel. 2007. Noise and hearing loss: a review. *Journal of School Health* 77, 5 (2007), 225–231. https://doi.org/10.1111/j.1746-1561.2007.00197.x

[23] Feng Ding, Dong Wang, Qian Zhang, and Run Zhao. 2019. ASSV: Handwritten Signature Verification Using Acoustic Signals. (2019), 274–277. https://doi.org/10.1145/3341162.3343756

[24] Tianyu Du, Shouling Ji, Jinfeng Li, Qinchen Gu, Ting Wang, and Raheem Beyah. 2020. SirenAttack: Generating Adversarial Audio for End-to-End Acoustic Systems. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security* (Taipei, Taiwan) *(ASIA CCS '20)*. Association for Computing Machinery, New York, NY, USA, 357–369. https://doi.org/10.1145/3320269.3384733

[25] 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* (Snowbird, Utah, USA) *(MobiCom '17)*. Association for Computing Machinery, New York, NY, USA, 343–355. https://doi.org/10.1145/3117811.3117823

[26] Yang Gao, Wei Wang, Vir V. Phoha, Wei Sun, and Zhanpeng Jin. 2019. EarEcho: Using Ear Canal Echo for Wearable Authentication. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 81 (Sept. 2019), 24 pages. https://doi.org/10.1145/3351239

[27] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. SoundWave: Using the Doppler Effect to Sense Gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1911–1914. https://doi.org/10.1145/2207676.2208331

[28] 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. https://doi.org/10.1016/j.specom.2015.05.002

[29] 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*. 930–934. https://www.isca-speech.org/archive/interspeech_2013/i13_0930.html

[30] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

[31] Tomi Kinnunen, Md. Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. 2017. The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection. In *Proc. Interspeech 2017*. 2–6. https://doi.org/10.21437/Interspeech.2017-1111

[32] Xinyu Lei, Guan-Hua Tu, Alex X. Liu, Chi-Yu Li, and Tian Xie. 2018. The Insecurity of Home Digital Voice Assistants - Vulnerabilities, Attacks and Countermeasures. In *2018 IEEE Conference on Communications and Network Security (CNS)*. 1–9. https://doi.org/10.1109/CNS.2018.8433167

[33] Xiaopeng Li, Fengyao Yan, Fei Zuo, Qiang Zeng, and Lannan Luo. 2019. Touch Well Before Use: Intuitive and Secure Authentication for IoT Devices. In *The 25th Annual International Conference on Mobile Computing and Networking* (Los Cabos, Mexico) *(MobiCom '19)*. Association for Computing Machinery, New York, NY, USA, Article 33, 17 pages. https://doi.org/10.1145/3300061.3345434

[34] Li Lu, Jiadi Yu, Yingying Chen, Hongbo Liu, Yanmin Zhu, Yunfei Liu, and Minglu Li. 2018. LipPass: Lip Reading-based User Authentication on Smartphones Leveraging Acoustic Signals. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. 1466–1474. https://doi.org/10.1109/INFOCOM.2018.8486283

[35] Sapna Maheshwari. 2018. *Hey, Alexa, What Can You Hear? And What Will You Do With It?* Retrieved March 25, 2020 from https://www.nytimes.com/2018/03/31/business/media/amazon-google-privacy-digital-assistants.html

[36] Weizhi Meng, Duncan S Wong, Steven Furnell, and Jianying Zhou. 2014. Surveying the development of biometric user authentication on mobile phones. *IEEE Communications Surveys & Tutorials* 17, 3 (2014), 1268–1293. https://doi.org/10.1109/COMST.2014.2386915

[37] Yan Meng, Zichang Wang, Wei Zhang, Peilin Wu, Haojin Zhu, Xiaohui Liang, and Yao Liu. 2018. WiVo: Enhancing the Security of Voice Control System via Wireless Signal in IoT Environment. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Los Angeles, CA, USA) *(Mobihoc '18)*. Association for Computing Machinery, New York, NY, USA, 81–90. https://doi.org/10.1145/3209582.3209591

[38] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. *FingerIO: Using Active Sonar for Fine-Grained Finger Tracking*. Association for Computing Machinery, New York, NY, USA, 1515–1525. https://doi.org/10.1145/2858036.2858580

[39] Corey R. Pittman and Joseph J. LaViola. 2017. Multiwave: Complex Hand Gesture Recognition Using the Doppler Effect. In *Proceedings of the 43rd Graphics Interface Conference* (Edmonton, Alberta, Canada) *(GI '17)*. Canadian Human-Computer Communications Society, Waterloo, CAN, 97–106.

[40] Swadhin Pradhan, Wei Sun, Ghufran Baig, and Lili Qiu. 2019. Combating Replay Attacks Against Voice Assistants. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 100 (Sept. 2019), 26 pages. https://doi.org/10.1145/3351258

[41] PUI speaker 2014. AS07104PO-LW152-R. Retrieved March 25, 2020 from http://www.puiaudio.com/product-detail.aspx?categoryId=6&partnumber=AS07104PO-LW152-R

[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* (Niagara Falls, New York, USA) *(MobiSys '17)*. Association for Computing Machinery, New York, NY, USA, 2–14. https://doi.org/10.1145/3081333.3081366

[43] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. 2018. Inaudible Voice Commands: The Long-Range Attack and Defense *(NSDI'18)*. USENIX Association, USA, 547–560. https://www.usenix.org/conference/nsdi18/presentation/roy

[44] Wenjie Ruan, Quan Z. Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 2016. AudioGest: Enabling Fine-Grained Hand Gesture Detection by Decoding Echo Signal. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 474–485. https://doi.org/10.1145/2971648.2971736

[45] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2019. Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*. https://dx.doi.org/10.14722/ndss.2019.23288

[46] Muhammad Shahzad and Shaohu Zhang. 2018. Augmenting User Identification with WiFi Based Gesture Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 134 (Sept. 2018), 27 pages. https://doi.org/10.1145/3264944

[47] Shridatt Sugrim, Can Liu, Meghan McLean, and Janne Lindqvist. 2019. Robust Performance Metrics for Authentication Systems. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*. https://dx.doi.org/10.14722/ndss.2019.23351

[48] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. VSkin: Sensing Touch Gestures on Surfaces of Mobile Devices Using Acoustic Signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 591–605. https://doi.org/10.1145/3241539.3241568

[49] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. 2015. Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. USENIX Association, Washington, D.C. https://www.usenix.org/conference/woot15/workshop-program/presentation/vaidya

[50] Anran Wang and Shyamnath Gollakota. 2019. Millisonic: Pushing the limits of acoustic motion tracking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI)*. 1–11. https://doi.org/10.1145/3290605.3300248

[51] Qian Wang, Xiu Lin, Man Zhou, Yanjiao Chen, Cong Wang, Qi Li, and Xiangyang Luo. 2019. VoicePop: A Pop Noise based Anti-spoofing System for Voice Authentication on Smartphones. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2062–2070. https://doi.org/10.1109/INFOCOM.2019.8737422

[52] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-Free Gesture Tracking Using Acoustic Signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (New York City, New York) *(MobiCom '16)*. Association for Computing Machinery, New York, NY, USA, 82–94. https://doi.org/10.1145/2973750.2973764

[53] Zhi-Feng Wang, Gang Wei, and Qian-Hua He. 2011. Channel pattern noise based playback attack detection algorithm for speaker recognition. In *Proceedings of the IEEE 2011 International Conference on Machine Learning and Cybernetics (ICMLC)*, Vol. 4. 1708–1713. https://doi.org/10.1109/ICMLC.2011.6016982

[54] 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 *Proceedings of the 27th USENIX Conference on Security Symposium* (Baltimore, MD, USA) *(SEC'18)*. USENIX Association, USA, 49–64. https://dl.acm.org/doi/10.5555/3277203.3277208

[55] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-Grained Acoustic-Based Device-Free Tracking. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (Niagara Falls, New York, USA) *(MobiSys '17)*. Association for Computing Machinery, New York, NY, USA, 15–28. https://doi.org/10.1145/3081333.3081356

[56] 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* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 103–117. https://doi.org/10.1145/3133956.3134052

[57] 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 (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 57–71. https://doi.org/10.1145/3133956.3133962

[58] 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* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1080–1091. https://doi.org/10.1145/2976749.2978296

[59] Man Zhou, Qian Wang, Jingxiao Yang, Qi Li, Peipei Jiang, Yanjiao Chen, and Zhibo Wang. 2021. Stealing Your Android Patterns via Acoustic Signals. *IEEE Transactions on Mobile Computing* 20, 4 (2021), 1656–1671. https://doi.org/10.1109/TMC.2019.2960778

[60] Yongpan Zou, Meng Zhao, Zimu Zhou, Jiawei Lin, Mo Li, and Kaishun Wu. 2018. BiLock: User Authentication via Dental Occlusion Biometrics. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 152 (Sept. 2018), 20 pages. https://doi.org/10.1145/3264962

A SUS QUESTIONNAIRE

Table 7: SUS questions

Item	Question
1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system