EchoHand: High Accuracy and Presentation Attack Resistant Hand Authentication on Commodity Mobile Devices

Cong Wu
Wuhan University, China
cnacwu@whu.edu.cn

Jing Chen
Wuhan University, China
chenjing@whu.edu.cn

Kun He
Wuhan University, China
hekun@whu.edu.cn

Ziming Zhao
University at Buffalo, USA
zimingzh@buffalo.edu

Ruiying Du
Wuhan University, China
duraying@whu.edu.cn

Chen Zhang
Wuhan University, China
whuhdc@whu.edu.cn

ABSTRACT

Biometric authentication schemes, i.e., fingerprint and face authentication, raise serious privacy concerns. To alleviate such concerns, hand authentication has been proposed recently. However, existing hand authentication schemes use dedicated hardware, such as infrared or depth cameras, which are not available on commodity mobile devices. In this paper, we present EchoHand, a high accuracy and presentation attack resistant authentication scheme that complements camera-based 2-dimensional hand geometry recognition of one hand with active acoustic sensing of the other hand. EchoHand plays an inaudible acoustic signal using the speaker to actively sense the holding hand and collects the echoes using the microphone. EchoHand does not rely on any specialized hardware but uses the built-in speaker, microphone and camera. EchoHand does not place more burdens on users than existing hand authentication methods. We conduct comprehensive experiments to evaluate the reliability and security of EchoHand. The results show that EchoHand has a low equal error rate of 2.45% with as few as 10 training data points and it defeats presentation attacks.

CCS CONCEPTS

• **Security and privacy** → Usability in security and privacy; **Multi-factor authentication**; *Biometrics*; *Mobile and wireless security*.

KEYWORDS

Hand Authentication, Presentation Attack, Acoustic Sensing, Hand Geometry.

ACM Reference Format:
Cong Wu, Jing Chen, Kun He, Ziming Zhao, Ruiying Du, and Chen Zhang. 2022. EchoHand: High Accuracy and Presentation Attack Resistant Hand Authentication on Commodity Mobile Devices. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3548606.3560553

CCS '22, November 7–11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9450-5/22/11…$15.00
https://doi.org/10.1145/3548606.3560553

1 INTRODUCTION

Biometric authentication methods have become ubiquitous on mobile devices. However, the two widely deployed schemes, namely fingerprint and face authentication, utilize users' sensitive biological traits and have raised privacy concerns [8, 22]. To alleviate the concerns, hand authentication has been proposed as a promising method [22, 37] for the following reasons: i) hands have rich intrinsic biometric features, e.g., palmprint, hand geometry, etc., that can provide reliable authentication [66]; ii) users are less concerned with hand privacy [22], while the fingerprint is widely used for law enforcement in many countries. The face is the most sensitive information for a person, and facial recognition is even prohibited in some regions. Since 2019, Amazon has launched a contactless hand authentication payment system based on palm vein and palmprint [7, 9]. Other hand authentication solutions, such as LG Hand ID [10] and PalmID [6], have also been proposed and deployed.

Simple hand authentication can be implemented with a camera. For instance, palmprint authentications use high-resolution cameras to catch the skin patterns of a palm, such as lines, points, and texture [16, 28]. However, this approach is less accurate and vulnerable to presentation attacks, e.g., the attacker can easily spoof the system using images [2]. Other systems use specialized hardware, which is not available on commodity mobile devices, to capture sophisticated traits, such as 3-dimensional hand geometry or palm vein information. For example, GesID [60] uses a depth camera to collect hand thickness. PalmID [6] and Hand ID [10] use time-of-flight cameras [36] and infrared sensors to map out the veins under the hand skin.

In this paper, we present EchoHand, a high accuracy and presentation attack resistant hand authentication scheme for commodity mobile devices. EchoHand complements camera-based hand geometry recognition of one hand with active acoustic sensing of the other holding hand. To this end, EchoHand plays an inaudible acoustic signal using the speaker to actively sense the holding hand and collects the echoes using the microphone. EchoHand is based on the observation that the way a user holds the phone affects the phone's vibration, which results in delay and attenuation of the signal propagating through structure-borne and air-borne paths [57]. EchoHand does not rely on any specialized hardware but uses the built-in speaker, microphone and camera. Moreover, EchoHand does not place more burdens on users than existing hand authentication methods [6, 10, 50, 60] since it senses the holding hand.

Even though acoustic sensing has been used to detect and recognize motions [20, 39, 54, 70] and faces [74], existing approaches all

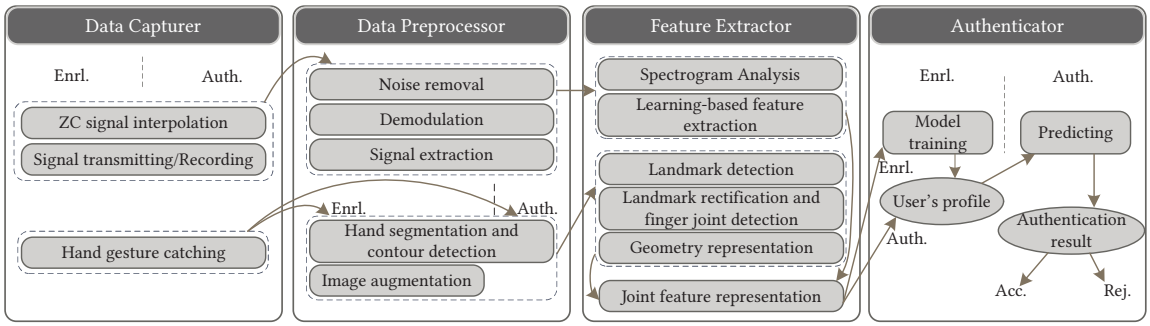


Figure 1: The workflow of EchoHand

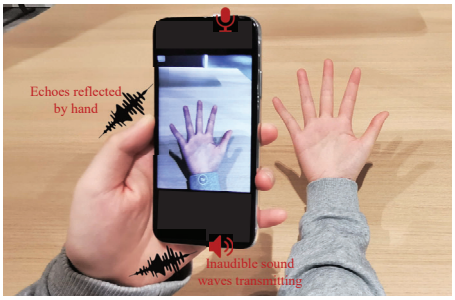


Figure 2: Illustration of EchoHand

require the sensed objects to be at least centimeters away from the sensor for better accuracy. For instance, VSkin [54] uses acoustic sensing to identify finger motions and touch gestures at the back of the device. Chaperone [20] detects whether a user is leaving a phone to prevent phone losses. VoiceGesture [70] and Lippass [39] rely on the Doppler shifts resulting from articulator motions for liveness detection to complement voiceprint authentication. EchoPrint [74] characterizes a user's face and combines acoustic features and facial landmarks to authenticate users.

However, when authenticating a holding hand, the distance between the hand and sensor is short, making it difficult to extract the echoes reflected by the holding hand and distinguish its features. To overcome these challenges, EchoHand uses the Zadoff-Chu (ZC) sequence [68] as the base signal and modulates it to an inaudible frequency band. EchoHand distinguishes the echoes reflected by the holding hand since the signals from different paths arrive at the microphone with different delays due to different propagation speeds and path distances. To extract salient acoustic features from the separated signals, we transfer a pre-trained neural network as the generalized learning-based feature extractor.

Attack Models. We consider adversaries who can conduct three types of attacks to bypass EchoHand: i) gesture spoofing attack, where adversaries know the victim's registered hand gesture and try to spoof the system by performing the same gesture; ii) presentation attack, in which adversaries have a picture of the victim's registered hand gesture and attempt to spoof the system using the picture; iii) mimicry attack, in which adversaries have a picture of the victim's hand gesture and also mimic the holding style of the victim. The contributions of this paper are summarized as follows:

- We present EchoHand, which characterizes the holding hand using acoustic sensing to complement hand geometry features from the other hand. To extract acoustic features, we design a learning-based feature extractor. We also implement a hand geometry feature extraction approach, extends state-of-art camera-based hand authentication;
- We conducted comprehensive experiments to evaluate the effectiveness of EchoHand under different settings and real environments, e.g., low light, audible noise, different devices, periods, and hardware settings. The experiment results show that EchoHand can achieve a low equal error rate (EER) of 2.45% with as few as 10 training data points;
- We evaluated EchoHand's ability to defeat the aforementioned three types of attacks. The experiment results show that attack success rates are below 1.35%. The overhead evaluation shows that EchoHand is efficient with a low authentication latency of 0.59 seconds and memory usage of 83MB. We also performed a user study to understand users' acceptance of EchoHand.

2 OVERVIEW OF EchoHand

As shown in Figure 2, EchoHand uses the speaker and microphone for acoustic sensing to complement geometry-based hand authentication. Similar to other authentication schemes, EchoHand consists of two phases: enrollment and authentication. In enrollment, EchoHand builds a legitimate user's profile, which includes acoustic sensed and the camera captured data. The user is required to hold the device in hand and perform a hand gesture facing the camera for registration. In the authentication phase, EchoHand compares both acoustic sensed and the camera captured biometrics.

As shown in Figure 1, EchoHand consists of four modules: data capturer, data preprocessor, feature extractor, and authenticator. The data capturer applies interpolation to the ZC sequence and modulates it to an inaudible frequency band. EchoHand transmits the inaudible acoustic signal using speaker and captures the echoes using microphone. It captures the hand gesture image using the camera. For the received echoes, the data preprocessor removes noise with a band-pass filter and applies signal demodulation. EchoHand performs signal extraction using cross-correlation to derive the target signal shaped by the holding hand. For the hand gesture image, EchoHand performs segmentation to remove background and contour detection to derive the hand contour.

The feature extractor applies continuous wavelet transform (CWT) for the extracted signal to generate its time-frequency spectrogram and uses a learning-based approach to extract acoustic features. With the hand gesture image and its contour, EchoHand performs landmark detection and rectification to find accurate key points, and offers the hand geometry representation based on these key points. It marks the acoustic and hand geometry features as a joint feature representation. To profile the legitimate user, the authenticator uses the combined feature set to train a machine learning model based on a one-class classifier, which is later used for authentication.

3 ACOUSTIC SIGNAL PROPAGATION

When using the speaker and microphone on the same device for acoustic sensing, the received acoustic signals are a collection of the transmitted signal propagating through different paths, which is subject to the multi-path effect [45]. These paths have different time delays, phase delays, and energy attenuation thanks to the characteristics of different propagation media and distances. Multi-path propagation can be modeled as a linear time-invariant system [45], where the received signal ($y[n]$) can be described as a convolution of the input ($x[n]$) and the impulse response (IR, $h[n]$) of the multi-path propagation as shown in Eq. 1.

$$y[n] = x[n] * h[n] = \sum_{i=0}^{M-1} a_i e^{-j\theta_i} x[n-\tau_i] \tag{1}$$

where $n \in \mathbb{N}$, $y[n]$ is the n_{th} sample in the sequence, $*$ is the convolution operator, M is the number of propagation paths, a_i is attenuation coefficient of the i_{th} path, θ_i is phase delay, and τ_i is time delay. The IR can be formulated as Eq. 2.

$$h[n] = \sum_{i=0}^{M-1} a_i e^{-j\theta_i} \delta[n-\tau_i] \tag{2}$$

where $\delta[n]$ is the Dirac's delta function ($\delta[n] = 1$ for $n = 0$, otherwise $\delta[n] = 0$). The IR of multi-path propagation depends on many factors, such as device layout, material, the holding hand, etc. Given a linear time-invariant system, the system output $y[n]$ can be formulated as Eq. 3.

$$y[n] = x[n] * h[n] = \sum_{k=0}^{N-1} x[k]h[n-k] \tag{3}$$

where N is the maximum length between $x[n]$ and $y[n]$. In particular, the cross-correlation of the output $y[n]$ and input $x[n]$ can be formulated as Eq. 4.

$$\begin{aligned} r_{yx}[\tau] &= \sum_{k=\tau}^{N-1} y[k]x^{\star}[k-\tau] = \sum_{k=\tau}^{N-1}(\sum_{j=0}^{N-1} x[j]h[k-j])x^{\star}[k-\tau] \\ &= \sum_{j=0}^{N-1}\sum_{k=\tau}^{N-1} x[j]x^{\star}[k-\tau]h[k-j] \end{aligned} \tag{4}$$

In particular, if $y[n]$ is same as $x[n]$, $r_{yx}[n]$ is also namely auto-correlation of $x[n]$, i.e., $r_x[\tau]$. Its formulation is given as Eq. 5.

$$r_x[\tau] = \sum_{k=\tau}^{N-1} x[k]x^{\star}[k-\tau] = \sigma_x^2\delta[\tau] \tag{5}$$

where σ_x is the standard deviation of the input signal $x[n]$. As a result, the cross-correlation $r_{yx}[\tau]$ can be expressed as Eq. 6.

$$r_{yx}[\tau] = \sum_{j=0}^{N-1} \sigma_x^2\delta[\tau]h[\tau-j] = \sigma_x^2 h[\tau] \tag{6}$$

Thus, given that the transmitting signal $x[n]$ is constant, the $r_{yx}[\tau]$ is linearly dependent on $h[\tau]$ (Eq. 6). To characterize the holding hand, we estimate the IR using the received signal and transmitted signal. The estimation of IR $\hat{h}[n]$ can be formulated as Eq. 7, which is based on the cross-correlation of the received signal $y[n]$ and the transmitted signal $x[n]$.

$$\hat{h}[n] = \frac{1}{\sigma_x^2} r_{yx}[n] = \frac{1}{\sigma_x^2} \sum_{k=n}^{N-1} y[k]x^{\star}[k-n] \tag{7}$$

where $x^{\star}[n]$ is the complex conjugation of $x[n]$. σ_x is the standard deviation of $x[n]$.

We use cross-correlation to measure the displacement of a signal relative to another [52]. Figure 3(a) shows an example cross-correlation of the received multi-path signal and the transmitted signal. Because the signal propagates through the device and air, the received signal consists of the structure-borne propagation through the device and air-borne propagation. The structure-borne propagation (Path 1) arrives first due to higher speed (>3,000m/s). The air-borne propagations, which consist of the direct transmission and the reflection of the holding hand (Path 2) and the reflections from other surrounding objects (Path 3), arrive later.

Figure 4 compares the IR estimation results with two subjects in our study. Since the IR estimation is a complex-valued signal, we calculate its magnitude with real and imaginary parts as shown in Figure 4(a) and (b), and show the trace of real and imaginary parts in Figure 4(c) and (d). As the figures show, the IR estimations of two subjects are much different. But, the two different executions from the same user are similar. Also, it shows that the trace of real and imaginary parts of two subjects are significantly different, whereas the two traces of the same user are consistent. This observation clearly shows that acoustic sensing can be exploited for characterizing hand to distinguish different subjects.

4 CHARACTERIZING ACOUSTIC ECHOES

In this section, we present how EchoHand extracts acoustic features that can effectively characterize the holding hand.

4.1 Acoustic Data Capturer

We have the following design considerations for the transmitted signal: i) to better locate and separate the target signal reflected by hand from the received signal, the transmitted signal should have a narrow main lobe in its auto-correlation [51]; ii) the transmitted signal should be inaudible to avoid annoying users; iii) the frequency range of the transmitted signal should be supported by the frequency response of audio hardware on commodity mobile devices, i.e., less than 24kHz.

Base signal selection. Existing acoustic sensing-based authentications [39, 65, 74] only consider magnitude characteristics by using a chirp signal, whose frequency changes with time. Chirp signal is usually used for ranging [32, 35]. We choose ZC sequence

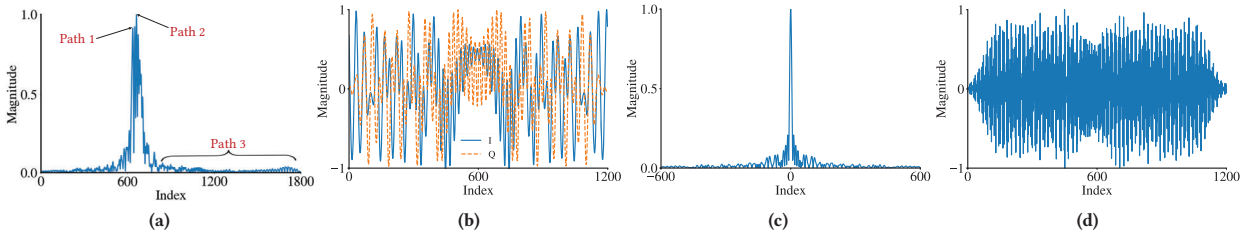


Figure 3: An example of cross-correlation of received multi-path signal and the transmitted signal (a), real (I) and imaginary (Q) part of interpolated ZC sequence (b), auto-correlation of interpolated ZC sequence (c), and modulated sequence (d)

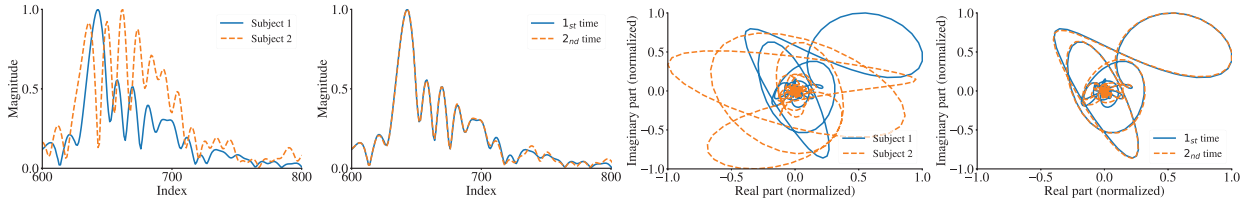


Figure 4: IR estimations using cross-correlation of the received signal and the transmitted signal for two subjects: the magnitude of IR from two subjects (a); the magnitude of IR from the same subject at two times with 48kHz sampling rate (b); trace of the real/imaginary parts from two subjects (c); trace of the real/imaginary parts from the same subject at two times (d)

as the base signal [54, 68], which has a narrower main lobe of its auto-correlation. As shown in Figure 3(c), the auto-correlation of ZC sequence has an extremely narrow main lobe. The auto-correlation of ZC sequence is close to zero when delay τ is not zero, whereas its auto-correlation is maximized only with a delay $\tau = 0$. To separate the multi-path signal arriving at different delays, we can perform cross-correlation of the received multi-path signal and the transmitted signal. Also, ZC sequence is a complex-valued signal with a constant amplitude, which contains not only the time-varied magnitude information but also phase information, which can help distinguish signals. A ZC sequence $x[n]$ is formulated as $x[n] = \exp(-j\pi Rn(n+1)/N)$, where N is the length of ZC sequence and $n \in [0, N-1]$, R is a constant with $R < N$, N and R are odd-valued positive integer and coprime, i.e. $gcd(R, N) = 1$.

Interpolation. Since the raw ZC sequence covers the entire frequency band, we need to fit its bandwidth to a narrow frequency band for signal transmitting. We employ Fourier interpolation [3] to increase the sequence length, which pads zeros in the frequency domain. After interpolation, we obtain the interpolated ZC sequence $x'[n]$ as shown in Figure 3(b).

Modulation. The interpolated ZC sequence lies in an audible low-frequency band. We modulate the interpolated sequence to an inaudible high-frequency band. Considering that most smartphones only support the sampling rate up to 48kHz, we set the sampling rate f_s as 48kHz, and the center frequency of carrier f_c as 20kHz. We modulate the real and imaginary probabilities of the complex-valued sequence to a single real sequence. The modulated sequence $y[n]$ is formulated as Eq. 8.

$$y[n] = \cos\left(\frac{2\pi f_c n}{f_s}\right) x'_I[n] - \sin\left(\frac{2\pi f_c n}{f_s}\right) x'_Q[n] \tag{8}$$

where $x'_I[n]$ and $x'_Q[n]$ is the real and imaginary part of $x'[n]$, respectively; $n \in [0, N'-1]$; and N' is the length of interpolated sequence $x'[n]$. Figure 3(d) shows an example modulated signal. A Hamming window is applied to the first and last points to reduce the audible noise caused by spectral leakage.

EchoHand first plays a modulated signal using the speaker and at the same time starts recording with the microphone. The playing takes N'/f_s seconds, while the recording takes $2N'/f_s$ seconds. The recorded signal is passed to the acoustic data processor module.

4.2 Acoustic Data Processor

The acoustic data processor performs signal preprocessing and extraction to derive the signal shaped by the holding hand.

4.2.1 Signal Preprocessor. It performs noise removal and demodulation for the captured signal to reconstruct the complex-valued baseband signal. EchoHand first performs noise removal to remove out-band interference. To separate high-frequency echoes from low-frequency ambient noise, we use a Butterworth band-pass filter (BPF) to filter the target signal in the transmission band. To reconstruct the baseband complex-valued signal, EchoHand demodulates the filtered high-frequency signal. It derives real and imaginary components by multiplying the signal and two orthogonal subcarriers that are used for modulation. We also use a Butterworth low-pass filter (LPF) to eliminate the high-frequency interference incurred by multiplication.

Table 1: Propagation speed, distance, delay, and energy level of different propagation paths

Path	Speed (m/s)	Distance (cm) †	Delay (ms) / Points	Energy
1	>3,000	15.2	0.05/-19	Medium
2	~343	[15.2, 15.2×2]	[0.44/0, 0.89/22]	High
3	~343	[15.2×2, ∞]	[0.87/22, ∞]	Low

†: As an example, we use the distance of Pixel 3A in which the microphone and bottom speaker are 15.2cm apart.

4.2.2 Signal Extractor. Table 1 presents the propagation speed, distance, delay, and energy of Path 1, 2, and 3. We aim to extract the Path 1 and 2 propagation from the received signal since they are shaped by the holding hand. We first estimate IR using cross-correlation of the demodulated signal and the transmitted signal. Then we leverage the magnitude of IR estimation to locate the candidate propagation path with the highest energy, and identify other propagation paths based on the known relative delays. Finally, we segment the target signal based on the different arrival delays of different propagation paths.

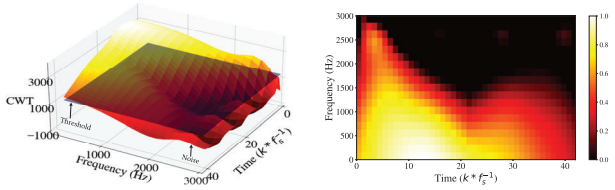
As shown in Table 1, the air-borne propagation via direct transmission from the speaker to the microphone has the most energy compared with Path 1 and 3, because acoustic signal energy attenuates fast through solid device body and is absorbed more by environmental objects through the air [57]. Figure 3(a) illustrates the energy and arrival time of three paths, the highest peak represents the air-borne propagation through direct transmission. We determine the arrival time of Path 2 by detecting the highest peak.

Given the distance between the microphone and speaker as d, Path 1 arrives with $(\frac{df_s}{v_a} - \frac{df_s}{v_s})$-point ahead of direct transmission through the air, where v_a and v_s are the propagation speed of air-borne and structure-borne, respectively. The target signal of Path 1 is within the range of $[-(\frac{df_s}{v_a} - \frac{df_s}{v_s}), 0]$, where 0 denotes the detected arrival time of air-borne propagation through direct transmission. To separate air-borne propagation signal shaped by the holding hand, we focus on Path 2 with a propagation distance ranged from d to $2d$, where d is the propagation distance of direct transmission. The target signal of Path 2 ranges from of $[0, \frac{df_s}{v_a}]$-point after Path 2. Therefore, the structure-borne and air-borne signal shaped by the holding hand is within a range of $[-(\frac{df_s}{v_a} - \frac{df_s}{v_s}), \frac{df_s}{v_a}]$ relative to Path 2. For example, when v_a = 343 m/s, v_s = 3000 m/s, f_s = 48kHz, and d = 15.2cm (Pixel 3A), the signal shaped by the holding hand is in the range of [-19, 22] as shown in Table 1. Finally, the output of acoustic data processor is the separated structure-borne and air-borne complex-valued signal, which is with the length of $(\frac{2df_s}{v_a} - \frac{df_s}{v_s})$-point.

4.3 Acoustic Feature Extractor

We adopt time-frequency spectrogram analysis and learning-based feature extraction to characterize the holding hand.

4.3.1 Spectrogram Analyzer. For the complex-valued signal shaped by the holding hand, we calculate the magnitude and phase information. Then we employ continuous wavelet transform (CWT) [42, 64] to construct a time-frequency representation of the magnitude and phase. We remove noise with low energy and perform normalization. CWT has better time and frequency resolution to perform time-frequency analysis than other approaches, such as short-time Fourier transform [42].



(a) (b)

Figure 5: An example CWT result of the magnitude: the raw CWT result (a); the CWT result after applying threshold (b)

Table 2: The size, number of parameters, and mean/standard deviation of inference time transferring different neutral networks as a feature extractor

Base model	Size (MB)	# Of parameters	Inference time (ms)
VGG16	512.23	134,259,392	15.92/0.01
ResNet50	90.43	23,581,440	38.47/0.48
InceptionV3	83.97	21,802,208	59.53/1.37
DenseNet121	27.90	7,031,232	68.61/3.27

As an example, Figure 5(a) illustrates the CWT result of the magnitude. We observe that the major components lie in a low-frequency band of 0-500Hz, which is shaped by the user's device holding style and the physiological characteristics of the hand. The time-frequency spectrogram also shows a significant decrease after the delay of over 30 points, which are noise components. We set the threshold as the standard deviation of the input signal to remove noise components. Figure 5(b) shows the CWT result after applying the threshold and normalization, where the noise components have little influence on the time-frequency spectrogram.

4.3.2 Learning-based Feature Extractor. To analyze the distinguishable features from the time-frequency spectrogram, we build a learning-based extractor based on transfer learning [61]. We train a base model using acoustic sensing data from different subjects, then transfer the pre-trained base model to a generic feature extractor.

Base model structure. We use a lightweight convolution neural network, DenseNet [25], as the base model. DenseNet has four dense blocks, in which the feature maps of all preceding layers and the current layer are concatenated and then passed on as the input to the subsequent layer. DenseNet recognizes input with 3 channels, whereas the time-frequency spectrogram has 2 channels, i.e., magnitude and phase spectrograms. To make it compatible with the output of spectrogram analyzer, we add a 3x3 convolutional layer before the input layer of DenseNet. Also, we add the fully-connected layer and the SoftMax layer after the output of the model to distinguish different subjects. Thanks to the lightweight base model with few parameters, its size is 27.90 MB and the average inference time is 68.61ms, which is available on mobile devices. We have also considered other CNN structures, including VGG [49], ResNet[24], and Inception [56]. Table 2 presents the comparison of using other models. Although it has the largest inference time, this delay is short enough to support user authentication.

Base model training. We train the base model using collected data points from 15 subjects, where each subject contributes 500 data

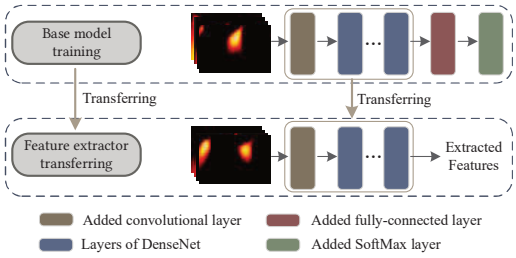


Figure 6: Transferring the base model as the feature extractor

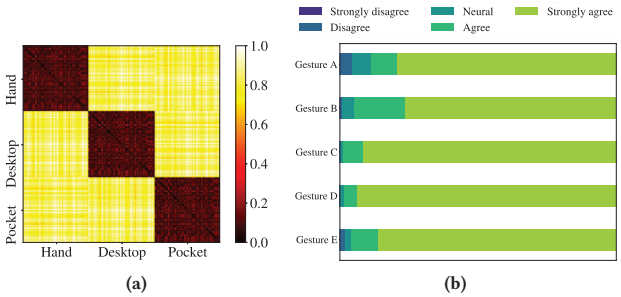


Figure 7: L2 distance of acoustic features under three scenarios (a), participants' perception of whether the hand gesture is easy to perform (b)

points. We use Adam optimizer for parameters optimization [31] and categorical cross-entropy as the loss function. The training batch size and epochs are set as 100 and 5,000. We used DenseNet121 in Keras [5] as the base model and TensorFlow as the backend. Training the base model takes around two hours using 1x Tesla P40 GPU. The base model only requires to be trained only once, and then can be transferred to unseen subjects for feature extraction [61].

Transferring the base model as a feature extractor. The basic idea of transfer learning is to transfer the knowledge from a pre-trained teacher model (base model) to a new student model (feature extractor). Since the shallow layers, i.e. forward layers, have already learned representative features for the student task during base model training, the output of these shallow layers can be used as the extracted features [58, 74]. Figure 6 presents the illustration of transferring the base model as a feature extractor. We build the feature extractor by dropping the fully-connected layer and SoftMax layer, and saving the former layers as the feature extractor. The output is a 1024-dimensional acoustic feature vector.

We also investigated the distinguishability of using acoustic sensing to sense three typical scenarios, where the device is in the user's hand, pocket, and on a table. Figure 7(a) shows the L2 distance of acoustic features under three scenarios, where 50 data points of each scenario are used to extract acoustic features. We observe that the features of the same scenario present a higher correlation than that of different scenarios.

5 HAND GEOMETRY FEATURE EXTRACTION

In this section, we present our hand geometry feature extraction method to help validate the effectiveness of EchoHand. Compared

Table 3: Parameters of image augmentation

Operation	Parameters
Scaling	Scaling factor ∈ [0.8, 1.2]
Rotation	Rotation angles ∈ [-60°, 60°]
Translation	Translation percent ∈ [-0.1, 0.1]
Shearing	Shearing factor ∈ [-30, 30]

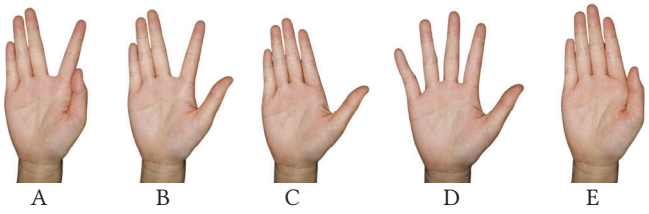


Figure 8: Five hand gestures, where fingers and palm should be roughly in the same plane, and fingers should be straight

Figure 9: An example of the original image (a), hand segmentation (b), and detected hand contour (c)

with the existing methods [13, 15, 33], our implementation contributes new methods in hand segmentation, landmark rectification, hand joint detection, and geometry representation.

5.1 Hand Gesture Image Processing

Hand gesture. To capture 2-dimensional hand geometry features from an image, our system imposes the following rules: i) the fingers and palm should be approximately in the same plane; ii) the fingers should be straight and not overlap with each other. These rules can be seen in most mature hand authentication systems, such as Amazon One [9] and Hand ID [10]. Figure 8 presents five example hand gestures in our experiments. These hand gestures are easy to perform, and these restrictions will not undermine the user experience. Figure 7(b) presents participants' perception of the difficulty of performing five hand gestures. In enrollment, the user is required to choose a hand gesture to register, and then perform the registered hand gesture in the authentication phase.

Hand segmentation and contour detection. To eliminate the impact of cluttered image backgrounds, we utilize the DeepLabv3+ model [1] to identify the location of the hand in the image. We transform RGB to HSV color space and apply a color range to obtain the clean hand image [34]. After deriving the clean hand image, we conduct hand contour detection [55]. The detected contour is the collection of pixel location of hand edges. Figure 9 presents an example of hand segmentation and contour detection.

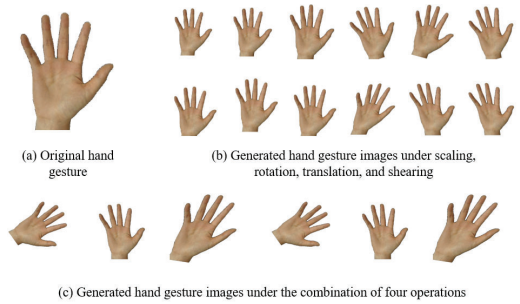


Figure 10: Generated hand gesture images under image augmentation

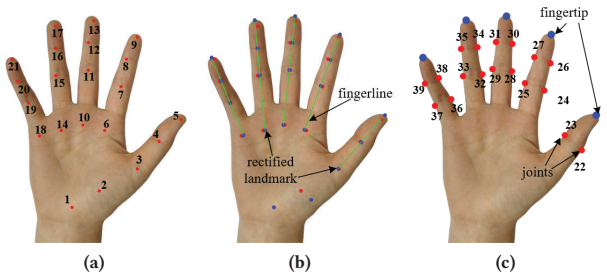


Figure 11: The labeled red points #1-21 are the detected hand landmarks using Openpose (a), the labeled blue points are rectified landmarks based on the finger line (b), the labeled blue points are rectified fingertip landmarks and red points are detected finger joints (c).

Table 4: List of extracted hand geometry features

Feature	Description	# Of features
Finger length	Length of each finger, including 3-5, 6-9, 10-13, 14-7, and 18-21	5
Finger width	Distance between pairs of finger joints, including 22-23, 24-25, 26-27, 28-29, 30-31, 32-33, 34-35, 36-37, 38-39	9
Palm size	Area and length of polygons consisting with lines 1-3-6-10-14-18. Distance of 1-3, 1-6, 1-10, 1-14, 1-18	7
Finger distance	Distance between 2 adjacent fingers, including 2-6, 3-7, 4-8, 5-9, 6-10, 7-11, 8-12, 9-13, 10-14, 11-15, 12-16, etc.	16

Image augmentation. To generate more training data, we use image augmentation for each captured image in enrollment [44]. The idea is to generate new images that are similar to the registered hand gesture image. We consider four image augmentation operations: scaling, rotation, translation, and shearing. Specifically, we perform image augmentation using the combination of these operations. Table 3 presents the parameter range for these operations. As an example, Figure 10 shows the captured raw hand gesture image, and generated images under different image augmentation operations.

5.2 Hand Geometry Representation

To extract geometry features from hand gesture images, we first detect the hand landmarks. Next, we rectify the biased landmarks and detect the finger joints. Finally, we extract hand geometry features using the rectified landmarks.

Landmark detection. To analyze the hand landmarks, we employ a released hand landmark detection model in Openpose [4, 48], which was trained to detect 21 hand landmarks. As shown in Figure 11(a), the labeled red points (#1-21) are detected hand landmarks, i.e., hand skeleton nodes.

Landmark rectification. Although the model has achieved great performance in detecting the hand landmarks, the estimated landmarks may not be accurate. To rectify the biased points, we consider that each finger is straight, i.e., the key points on the same finger are on the same straight line. Thus, the detected point of #6, 7, 8, and 9 should be on the same straight line, which we term as *finger line*. As shown in Figure 11(b), we fit a straight line using the landmarks on the finger line, e.g., #6, 7, 8, and 9, based on the least squares method. Then, for each point on the finger, we find its perpendicular foot on the finger line as the rectified point, as shown as blue points in Figure 11(b). We also rectify the fingertip point as the nearest intersection point of the finger line and hand contour. To locate the finger joints, we draw a line perpendicular to the finger line based on the rectified key point (blue points in Figure 11(b)). The two nearest intersection points of the perpendicular line and hand contour are detected as a pair of finger joints (red points in Figure 11(c)).

Geometry representation. With rectified hand landmarks and detected finger joints, we extract the following hand geometry features: i) *finger length*, which is defined as the distance between the fingertip point and the metacarpophalangeal joint, e.g., #3, 6, 10, 14, or 18. There are 5 features for finger length; ii) *finger width*, which is defined as the distance of a pair of finger joints with the hand landmark in the middle. We calculate 9 widths for 5 fingers; iii) *palm size*, which is defined as the area and length of polygons consisting of 5 landmarks, e.g., #1, 3, 6, 10, 14, 18. To describe the shape, we also calculate the distance between point #1 and each of #3, 6, 10, 14, 18. By doing so, 7 features of the palm size are calculated; iv) *finger distance* implies how users perform the hand gesture, which is relevant to the user's hand gesture behavior. We define the finger distance as the length between 2 fingers, such as #2-6 and 5-9. We calculate 16 features of finger distance.

In total, we extract 37 features, which are summarized in Table 4. Figure 12(a) shows the two-dimensional feature space of three users' raw and augmented hand images, which can be easily classified. Figure 12(b) presents the differences in hand geometry features extracted from three users.

6 ONE-CLASS CLASSIFIERS

Because only the legitimate user's data is available in enrollment, EchoHand uses one-class classifiers, including centroid classifier (CC), local outlier factor (LOF), and one-class support vector machine (OCSVM): i) CC [38] is a distance-based classifier, which computes the distance between the test data point and the centroid of training points; ii) LOF [17] is a density-based method to recognize the outlier data point by computing the local density deviation between the test data point and its neighbors. The data point will be considered as the outlier if it has a substantially lower density than its neighbors, where the local density deviation is estimated based on the L2 distance of its k neighbors; iii) OCSVM [46] is a distance-based classifier. It works by first mapping input data points

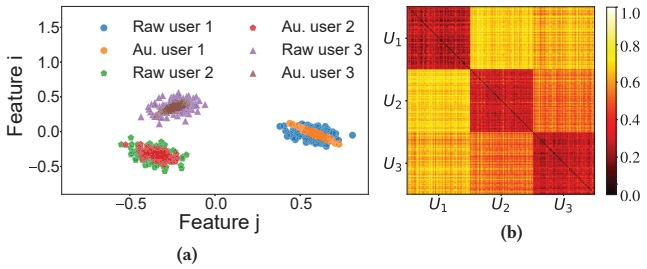


Figure 12: Visualized feature space of raw and augmented (au.) hand gesture data under PCA (a), L2 distance of three users' hand geometry features (b)

into another feature space with the kernel function and optimizing a hypersphere with minimal volume that best includes the training data points. The model parameters are determined by a centroid and a radius. The data point will be regarded as the outlier if it is outside the hypersphere.

7 DATA COLLECTION

We developed an Android application with the sampling rate of audio transmitting and recording as 48kHz for data collection. We used the bottom speaker to play the signal and the top microphone to record echoes. Figure 2 shows the position of the used audio hardware. After receiving the IRB approval from our institute, we started our data collection in March 2020. We recruited 45 subjects (Age from 18 to 38; 19 females and 26 males). We also recruited another six subjects to role-play the attacker to carry out gesture spoofing, presentation, and mimicry attacks. Before data collection, we explained to each participant the purpose of this research and the data we collect. Each subject was asked to use the smartphone for about one minute to get familiar with the device and find a comfortable and relaxing device-holding style. We compiled the following datasets.

1) *Dataset-1.* We collected only acoustic signals from 15 subjects to train the base model on a Pixel 3A. The app transmits the inaudible and saves the recorded signal continuously. We collected 1,200 data points for each subject, and it took about three minutes for a subject to complete. As a result, we collected 15×1,200 = 18,000 acoustic signals for *dataset-1*.

2) *Dataset-2.* We collected hand gesture images and acoustic sensing data from 30 subjects to compile *dataset-2*. Besides holding the device in a comfortable style, a subject also used the smartphone camera to catch the hand gesture image. A subject needed to perform each hand gesture in Figure 8 50 times. For acoustic sensing, the app collected 500 data points for each subject. During image and acoustic data collection, subjects were asked to place down the device, pick it up again, and hold the device in a familiar holding style to help get more different data samples. A subject spent ~12 minutes to complete this task. As a result, we collected 30×500 = 15,000 acoustic signals and 30×250 = 7,500 hand gesture images.

3) *Dataset-3.* To evaluate the effectiveness of EchoHand in noisy environments, we compiled *dataset-3* with the same 30 subjects. We created a noisy environment by playing the song 'Sugar-Maroon 5' at ~ 62-65dB using the speaker of another smartphone, and the app collected 500 data points for each subject. As a result, we collected 30×500 = 15,000 acoustic signals in total for *dataset-3*.

4) *Dataset-4.* To evaluate the generalization of EchoHand on different devices, we compiled the *dataset-4* on two more smartphones: Xiaomi6 (5.15 inches) and Redmi Note7 (6.3 inches). We divided 30 subjects into two groups that are assigned two devices for collecting acoustic data and hand images. Similar to the procedure of *dataset-2*, we collected 500 acoustic signals and 250 hand images for each subject. As a result, we collected 30×500 = 15,000 acoustic signals and 30×250 = 7,500 hand gesture images in total for *dataset-4*. Besides, we also used Samsung GALAXY On5 (a small device with 5 inches) to collect five subjects' acoustic echoes. For each user we collected 500 acoustic signals.

5) *Dataset-5.* To evaluate the effectiveness under real settings, we compiled *dataset-5* in four different environments: *Env-1* is a quiet meeting room with a big table and chairs. *Env-2* is a noisy but empty meeting room with music playing at ~ 42dB. *Env-3* is a noisy and crowded room with people walking and talking. *Env-4* is a quiet room but with the inaudible high-frequency sound (same as the acoustic signal for sensing) continuously playing nearby. We called back 20 subjects and divided them into four groups to collect acoustic signals in four environments respectively. Each subject is required to collect 500 acoustic signals. As a result, we collected 15×500 = 7,500 acoustic signals in total for *dataset-5*.

6) *Dataset-6.* To evaluate the authentication consistency, we compiled *dataset-6* with five subjects in four periods with an interval of one week. In the data collection of each period, we collected 500 acoustic signals and 250 hand images for each subject. As a result, we collected 5×500×4 = 10,000 acoustic signals and 5×250×4 = 5,000 hand gesture images in total for *dataset-6*.

7) *Dataset-7.* To evaluate the effectiveness under the low light, we compiled *dataset-7* with five subjects. To build the low light environment, we turned off the light sources and closed the curtains in a meeting room in the evening. Each subject was required to perform each hand gesture 50 times and catch the hand gesture using the smartphone camera and the flash. As a result, we collected 5×50×5 = 1,250 hand gesture images in total for *dataset-7*.

8) *Dataset-8.* To evaluate the impact of using the adjacent microphone and speaker, and covering the speaker, we collected the previous five subjects' acoustic data in the two hardware settings on Pixel 3A. The five subjects were first required to perform hand sensing using adjacent bottom microphone and speaker (the distance is 2cm) for 500 times, then using the bottom speaker and top microphone while covering the bottom speaker for 500 times. We collected 2×500×5 = 5,000 acoustic echoes.

8) *Dataset-9.* We want to evaluate if EchoHand can defeat the aforementioned attacks. To increase the possibility of successful attacks, we collected the acoustic sensing data and hand gesture images of six attackers. We calculate the L2 distance between the feature vector of the attackers and the previous 30 subjects. Then, we assigned each attacker five subjects as his/her targets based on the similarity. We collected the following attack datasets: i) *dataset-9a: gesture spoofing attack.* Similar to the procedure of *dataset-2*, each attacker was asked to hold the device to collect 500 acoustic signals and 50 hand gesture images per gesture. We collected 6×250 = 1,500

hand gesture images and 6×500= 3,000 acoustic signals for *dataset-9a*; ii) *dataset-9b: presentation attack.* The attacker was asked to present the previously recorded hand images to the data collection device using the tablet screen, and each image was only used to present only once. As a result, we collected 30×250 = 7,500 hand images for *dataset-9b*; iii) *dataset-9c: mimicry attack.* We asked each attacker to carefully observe how the attack target holds the device at a close distance (~1.5m) and mimic the device-holding behavior. After the attacker was confident about what they observed, she/he would mimic the subject's holding behavior to collect acoustic sensing data. The attacker is allowed to perform mimicry attacks for unlimited times, but at least 50 times, where for each attack the app actively senses the holding hand 10 times. We select 50 data points with a higher probability to be accepted by the authentication model. We collected 30×500 = 15,000 data points for the *dataset-9c*.

8 EVALUATION

In this section, we report the evaluation results of the proposed system. In EchoHand, R of ZC sequence is 63, and the length is 127. We apply 1200-point interpolation to the 127-point ZC sequence. Thus, the sequence is 25ms, i.e. $\frac{1200}{48000}$, and the frequency band of interpolated sequence is 0-2.54kHz, i.e. $\frac{24k}{1200}$×127. The modulated sequence $s[k]$ has an inaudible frequency band of 17.46 - 22.54kHz. The first and last 150 points of the modulated sequence are applied to a Hamming window. For the data processor, the cutoff frequency of LPF is 3kHz, and the passband of BPF is 17 - 23kHz.

8.1 Evaluation Metrics

False acceptance rate (FAR) is defined as the ratio between the number of falsely accepted data points and illegal data points. It indicates the probability of an unauthorized user being falsely accepted as the authorized ones. False rejection rate (FRR) is the ratio between the number of falsely rejected data points and legitimate data points. It represents the probability of the authorized user being falsely rejected. Receiver operating characteristics (ROC) curve is a dynamic depiction of FRR against FAR at a varying decision threshold. The area under the ROC curve (AUC) represents the probability that prediction scores of legitimate users' samples are higher than illegal users' samples. Equal error rate (EER) is the point on the ROC curve, where FAR is equal to FRR, i.e., EER = FAR = FRR. Lower EER indicates that the authentication system is more reliable. We use the frequency count of scores (FCS) [53] to show the frequency count of all test data points' prediction scores. FAR (i.e., the attack success rate) is used as the attack resistance evaluation criteria, which is defined as the ratio between the number of incorrectly identified data points and the number of all attack data points.

8.2 Reliability Analysis

To find out how distinguishable of acoustic features, we split each user's data points into training and test sets randomly, and trained the authentication model for each subject. We used the rest data points and other users' data points to evaluate the model. We performed 5-fold cross-validation to evaluate performance, and conducted grid search to find the best parameter combination for each classifier. The best parameter of LOF was `n_estimators`=3. For OC-SVM, we used radial basis function as the kernel function, and optimal parameters γ and ν were 0.19 and 0.03.

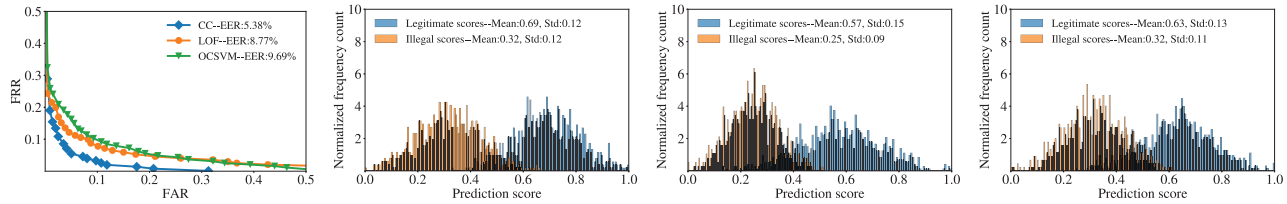
Table 5: The average EERs of gesture A, B, C, D, E (Figure 8)

	Classifier	A	B	C	D	E
W. IA	CC	7.38%	6.90%	7.52%	6.48%	7.70%
	LOF	11.15%	10.88%	11.80%	10.13%	12.15%
	OCSVM	9.31%	8.83%	8.96%	8.85%	9.37%
W/o. IA	CC	6.36%	6.16%	6.38%	6.06%	6.39%
	LOF	6.89%	5.70%	6.05%	5.91%	7.24%
	OCSVM	7.49%	6.97%	8.17%	7.10%	8.78%

8.2.1 Performance of Acoustic Sensing. A. Finding the optimal middle layer to transfer the base model as a feature extractor. The base model has four dense blocks (DB) and a fully-connected (FC) layer. We used the output of each block and FC layer as the features to investigate how the choice of these layers influences the distinguishability. The base model was trained using 18,000 acoustic data points of *dataset-1*. We used the acoustic data points in *dataset-2* to evaluate the reliability of extracted features, where the user is profiled using 10 training data points. Figure 14(a) reports the EERs under the feature sets extracted from different layers. The results show that using the output of 4_{th} dense block as the acoustic features achieves the lowest EER. Figure 14(b) presents the comparison of different classifiers using the features extracted from 4_{th} dense block. CC achieves an average EER of 5.60%, which is much lower than LOF or OCSVM. For later experiments, we use the output of the 4_{th} dense block as the extracted acoustic features.

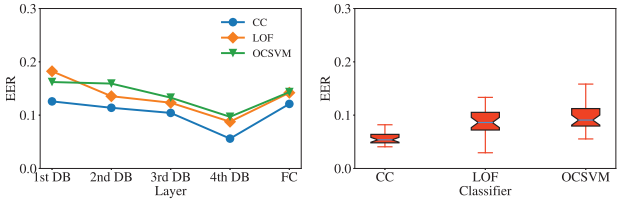
B. Performance of extracted acoustic features To evaluate the effectiveness of only applying acoustic sensing to distinguish between legitimate and illegal users, we used all collected acoustic data points in *dataset-2*. Each user was profiled with 10 data points. Figure 13(a) shows ROC curves under different classifiers. The results indicate that CC achieves a lower EER. CC, LOF, and OCSVM achieve an average EER of 5.60%, 8.77%, and 9.69%, respectively. Figure 13(b)(c)(d) present FCS of legitimate and illegal data points, showing that prediction scores under three different classifiers have similar distributions, while the overlapping region between the scores of legitimate and illegal data points under CC is smaller than the region of the other two classifiers. This implies that the CC outperforms LOF and OCSVM in terms of the error rate.

8.2.2 Effectiveness of Hand Geometry Features. To evaluate the effectiveness of only applying hand geometry features to distinguish between legitimate and illegal users, we used all collected hand image data points in *dataset-2*. For each image, we applied image augmentation (IA) and generated 100 hand gesture images used for model training. We evaluated performance for five hand gestures respectively. Each user was profiled with only 10 raw data points when not using IA, and with 1,010 data points (1,000 images were generated) when using IA. Table 5 reports EERs of five hand gestures under different classifiers. The results show that image augmentation enhances the generalization ability of the authentication model or template, especially LOF and OCSVM. For example, the average EER of gesture A under CC, LOF, and OCSVM are 7.38%, 11.15%, and 9.31% without IA. While using image augmentation, the average EER of gesture A under CC, LOF, and OCSVM can be improved to 6.36%, 6.89%, and 7.49%.



(a) ROC curves (b) FCS under CC (c) FCS under LOF (d) FCS under OCSVM

Figure 13: ROC curves (a) and normalized FCS (b, c, d) when using only acoustic features to authenticate users



(a) (b)

Figure 14: Transferring different middle layers as the acoustic feature extractor: EERs under the feature sets extracted from different middle layers (a), EERs under the features extracted from the 4_{th} dense block (b)

8.2.3 Effectiveness of EchoHand to Authenticate Users. To evaluate the effectiveness of EchoHand, we used the collected acoustic data points and hand images in *dataset-2*. In this step, each raw image was used to generate 100 new images. Each user was profiled using 10 raw data points, i.e., 10 acoustic data points and 1,010 hand images. Figure 15(a) shows ROC curves when using acoustic features to complement hand geometry features. CC, LOF, and OCSVM achieve an average EER of 2.45%, 5.96%, and 6.82%, respectively. Figure 15(b)(c)(d) report FCS of legitimate and illegal data points. The results show that the overlapping region under CC is smaller than LOF and OCSVM. Besides, there exist a few illegal data points with high prediction scores under the two classifiers. The results suggest that CC outperforms LOF and OCSVM, and we used CC as the classifier due to its lower EER in later experiments.

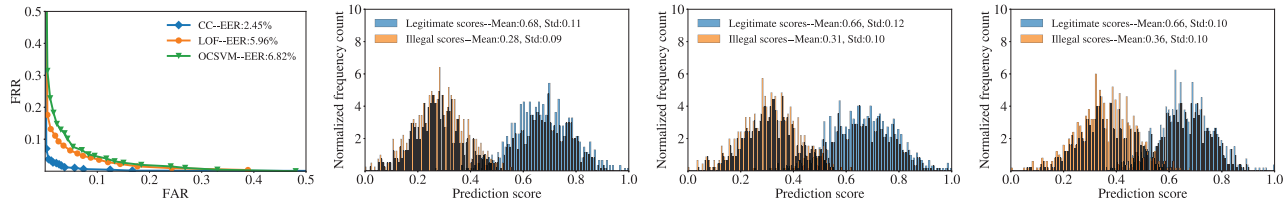
8.2.4 Impact of Audible Noise. To evaluate the impact of audible noise on acoustic sensing, we profiled the legitimate user with 10 data points from *dataset-2*, and evaluated EER using acoustic data in *dataset-3* and image data in *dataset-2*, where CC was used as the classifier. Figure 16(a) shows ROC curves under the environments with and without noise. If using acoustic features to complement hand geometry features, it achieves an average EER of 2.66% and 2.45% under the environment with and without audible noise, where the error rate is almost approximate. If only using acoustic features, it achieves an average EER of 6.12% and 5.38% under the environment with and without audible noise. Results suggest that the ambient noise has little impact on EchoHand for acoustic sensing.

8.2.5 Consistency Over Time. To evaluate the consistency of EchoHand over different periods, we used *dataset-6*. The 10 data points in week 1 were used to train the authentication model, while the rest data in week 1, 2, 3, and 4 were used to evaluate the EER. Figure 16(b) shows the results under different periods of 4 weeks. The average EERs under acoustic features are 5.14%, 6.07%, 9.15%, and 15.98% in Week 1, 2, 3, and 4, respectively, while the EERs under the hand geometry and acoustic features are 2.10%, 2.13%, 3.03%, and 4.53%. We observed that hand geometry features are more consistent than acoustic features under different weeks. Combining these two kinds of features significantly improves authentication accuracy and consistency. We also found that the increasing EER of week 4 was due to a few subjects' device-holding behaviors changing dramatically.

8.2.6 Performance on Different Devices. To evaluate the performance of EchoHand on different devices, we used *dataset-1* and *4*. We trained authentication models and evaluated performance using the data points from the same device. CC was used as the classifier and the legitimate user was profiled using 10 data points. Figure 17(a) shows results under the different devices. The average EERs on Pixel 3A, Xiaomi 6, Redmi Note7, GALAXY On5 are 2.45%, 7.24%, 3.69%, and 10.33%, respectively. The results suggest that the acoustic features on Xiaomi 6 and GALAXY On5 may undermine the distinguishability of hand features. This may be due to the short distance between the top microphone and bottom speaker.

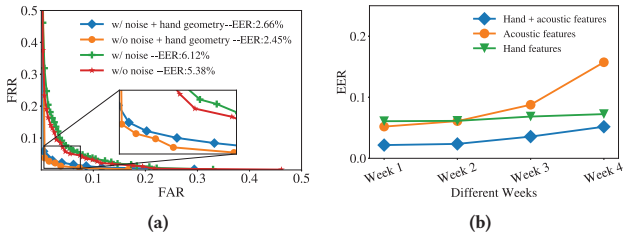
8.2.7 Performance Under Real Environments. We also used *dataset-5* to evaluate the performance of EchoHand under real environments. We trained the authentication model using 10 data points in *Env-1*, and tested the model using the rest data in *Env-1*, *2*, *3*, and *4*. Figure 17(b) shows results under different environments, where we regard the results under *dataset-2* as *lab* environment. The average EERs under lab and five real settings are 2.45%, 4.95%, 4.79%, 5.55%, and 6.53% respectively. The performance of five real settings is approximate to the lab setting. The results suggest that EchoHand is reliable in real environments.

8.2.8 Effectiveness of Landmark Rectification. To evaluate the effectiveness of our designed landmark rectification, we used *dataset-2*. Each user was profiled with 10 data points and CC as the classifier. Figure 18(a) shows ROC curves under landmark rectification. If only using geometry features, the average EERs under rectified and unrectified geometry are 6.27% and 9.25%. If using acoustic sensing features to complement hand geometry features, it achieves an average EER of 2.45% and 4.61% under rectified and unrectified



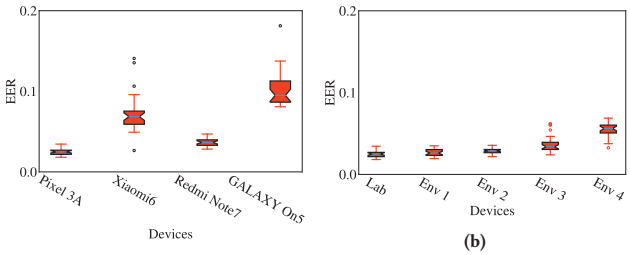
(a) ROC curves (b) FCS under CC (c) FCS under LOF (d) FCS under OCSVM

Figure 15: ROC curves (a) and normalized FCS (b, c, d) when using acoustic features to complement hand geometry features



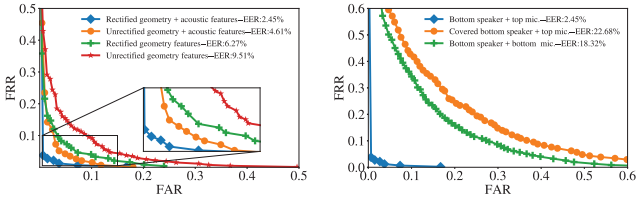
(a) (b)

Figure 16: ROC curves under environments with (w/) and without (w/o) audible noise (a), EERs under different weeks (b)



(a) (b)

Figure 17: EERs under different devices (a), and environments (b)



(a) (b)

Figure 18: ROC curves under landmark rectification (a), and different hardware settings (b)

geometry. The result suggests our designed hand landmark rectification improves the robustness of hand geometry features in authenticating users.

Table 6: EERs and AUC under low light

Setting	Features	EER	AUC
Low light	Hand geometry	8.13%	0.9761
	Sensing + hand geometry	2.92%	0.9955
Normal light	Hand geometry	6.27%	0.9864
	Sensing + hand geometry	2.45%	0.9977

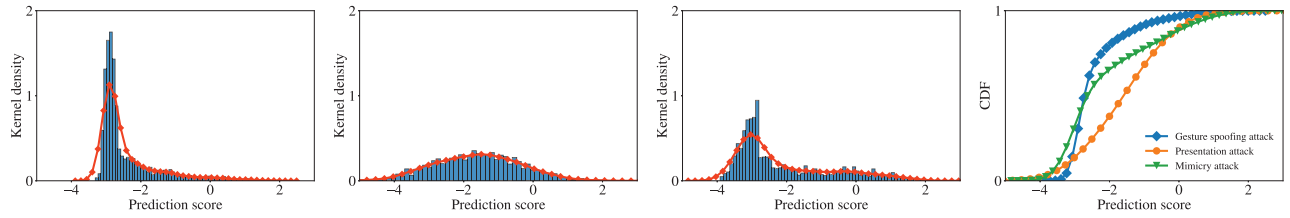
8.2.9 Performance Under Low Light Setting. To evaluate the robustness under low light setting, we trained the authentication model using *dataset-2*, and evaluate the authentication model using *dataset-2* and *7* respectively. Table 6 shows EERs and AUC under low light. If using only hand geometry features, the EERs are 8.13% and 6.27% under low and normal light. While using acoustic features and hand geometry features, EchoHand achieves an average EER of 2.92% and 2.45% under low and normal light. This shows that acoustic sensing enhances the robustness of hand geometry-based authentication under low light.

8.2.10 Impact of Different Audio Hardware Settings. To evaluate the impact of different audio hardware settings, we used *dataset-2* and *dataset-8*. 10 data points in *dataset-2* were used to train the authentication model, while the combinations of acoustic data in *dataset-8* and hand images in *dataset-2* were used to evaluate EER. Figure 18 presents ROC curves under different audio hardware settings. If using the covered bottom speaker and top microphone, EchoHand achieves an average EER of 22.68%. While using the bottom speaker and bottom microphone, the average EER is 18.32%.

8.3 Evaluation of Attack Resistance

To evaluate the resistance against three different attacks, we used *dataset-9a, b*, and *c* to test the authentication model trained using *dataset-2* from the previous 30 subjects. We normalized the decision threshold (where FAR is equal to FRR) to 0, and then investigated the distribution of the attack dataset's prediction scores. We reported FAR, i.e., attack success rate, distribution of the attack dataset's prediction scores, the kernel density of prediction scores evaluated under Gaussian kernel [61], and the cumulative distribution function (CDF).

Figure 19(a) shows the prediction scores' distribution under gesture spoofing attack using *dataset-9a*. The mean prediction score of attack data points is -2.42. The kernel density shows a narrower range but consistently low prediction scores. EchoHand can defend against gesture spoofing attack with a FAR of 0.21%. Figure 19(b) shows the prediction scores' distribution under presentation attack using hand images of *dataset-9b* and acoustic data of *dataset-9a*.



(a) Gesture spoofing attack (b) Presentation attack (c) Mimicry attack (d) CDF of attack data points' score

Figure 19: PDF/FCS (a, b, c) and CDF (d) under gesture spoofing, presentation, and mimicry attack

Table 7: FAR and mean/standard deviation of attack dataset's prediction scores

Attack type	FAR	Prediction scores
Gesture spoofing attack	0.21%	-2.42/ 0.86
Presentation attack	0.62%	-1.60/ 1.21
Mimicry attack	1.35%	-2.11/ 1.37

Table 8: The latency of different mobile authentication schemes [61, 62]

Scheme	Latency	Motion behavior[1]
PIN	1.25s	✓
Pattern lock	3.14s	✓
Fingerprint authentication	0.29s	✗
Facial authentication	1.48s	✗
EchoHand	0.59s	✗

[1] Require users to perform motion behavior, e.g., typing, and drawing.

The mean prediction score is -1.60. The kernel density shows a wide range but a higher score than the scores under the gesture spoofing attack. Presentation attack is with a FAR of 0.62%. Figure 19(c) shows the prediction scores' distribution under mimicry attack using hand images of *dataset-9b* and acoustic data of *dataset-9c*. The mean prediction score is -2.11. The kernel density shows a wide range but is with consistently low prediction scores, where there exist many attack data points with the scores ranging from -4 to -2. Mimicry attack is with a FAR of 1.35%.

The results suggest that EchoHand can defeat gesture spoofing, presentation, and mimicry attacks. Table 7 reports FAR and prediction scores of attack data points. Figure 19(d) presents the CDF of prediction scores, where the scores less than zero are with a high probability.

8.4 Latency and Memory Usage

The authentication latency is composed of the time required for data processing, feature extraction, and model inference. We evaluated the latency of these modules respectively and monitored the total memory usage. We developed a prototype system of EchoHand on Android, and evaluated the average latency for 50 authentications. It takes about an average latency of 0.37, 0.14, and 0.08 seconds for the 3 modules respectively on Pixel 3A (2x2.0 GHz). In total, EchoHand requires 0.59 seconds to complete authentication. Table 8 compares the latency between EchoHand and different mobile authentication schemes. We also used Android Profiler to monitor the memory usage of EchoHand, and the average memory usage on Pixel 3A is 83MB.

9 RELATED WORK

Hand authentication. Hand authentication schemes distinguish the legitimate user and impostors based on the intrinsic hand traits. Palmprint-based methods rely on the high-resolution camera to capture palmprint (i.e., a complex set of skin lines), and sophisticated image processing methods to extract the texture features [14, 28]. Nevertheless, it is vulnerable to presentation attacks [2, 16]. Palm and finger vein-based methods rely on the dedicated hardware, e.g., an infrared camera, to scan the pattern of blood vessels [9–11, 29, 36, 63]. The dedicated hardware is not available on most commodity mobile devices.

Hand geometry-based methods identify the legitimate user by analyzing the hand geometry features [15, 27, 33, 50, 60], such as finger lengths, and widths. Simple methods relying on a monocular RGB camera to analyze 2D hand geometry are usually vulnerable to presentation attacks [13, 15, 33]. To enhance security, these methods are dependent on comprehensive camera systems, such as depth camera [60]. Some behavior-based hand authentications also characterize the hand/finger movements to identify different subjects [21, 27, 50, 60, 71, 72]. However, these methods require the user to perform predefined hand gesture movements.

Compared with hand authentications that require users to perform hand motion hand movement [27, 50, 60], EchoHand does not require users to perform any hand gesture movement but make a stationary hand gesture. EchoHand can also resist presentation attacks. Table 9 presents the comparison of existing mature commercial hand authentications and the latest research work.

Acoustic sensing-based authentications. Acoustic sensing exploits speakers and microphones to design fancy applications [20, 26, 40, 43, 54, 59, 67, 73, 75, 76], e.g., usable and secure authentications [18, 19, 30, 39, 69, 70, 74]. Sound-proof [30] and Proximity-Echo [41] were proposed to validate whether the user's phone is near the device used to log in via sensing ambient noise. Echoprint [74] enhanced the security of face authentication against presentation attacks by transmitting an inaudible acoustic signal and receiving the echo to sense the facial geometry. To enhance voiceprint against replay attacks, VoiceGesture [70] and Lippass [39] leveraged active acoustic sensing to detect mouth movements to validate the presence of the legitimate user. To maintain the security of pattern locks and PINs, TouchPrint [19, 75] leveraged acoustic sensing to characterize finger tapping/sweeping events. To provide hand authentication using acoustic sensing, Echolock [65] considered only the structure-borne signal for sensing and analyzed acoustic features in the time and frequency domain. To protect smartphone

Table 9: Comparison of existing mature commercial hand authentications, the latest related research work

Method	Required hardware	Description of hand features	EER	PAR[1]	Hand motion[2]
		Commercial product			
Amazon One [9]	Unknown customized hardware (Maybe infrared camera, RGB camera)	Palm vein and palmprint patterns	N/A	✓	✗
Hand ID [10]	Infrared illuminator, TOF sensor[3]	Palm vein patterns	N/A	✓	✗
PalmID [6]	Infrared camera	Palm vein patterns	N/A	✓	✗
PalmID [6]	RGB camera	Palmprint patterns	N/A	✗	✗
PalmSecure [12]	Near-infrared imaging camera	Palm vein patterns	N/A	✓	✗
Vein ID [11]	Near-infrared illuminator, common RGB camera	Finger vein patterns	N/A	✓	✗
		Research paper			
[60]	Leap motion controller[4]	3D motion depth features of gesture movement	∼ 2%	✓	✓
[27]	Leap motion controller	3D motion characteristics of fingertips and finger joints	< 4%	✓	✓
[50]	Multi-touch screen	Hand geometry and motion characteristics of swiping on a multi-touch touchscreen	5.84%	✓	✓
[33]	Optical scanner	Hand geometry features, including finger width and length	0.59%	✗	✗
[15]	Optical scanner	Hand geometry graph topology	3.05%	✗	✗
[23]	RGB camera, infrared lamp	Palm dorsal veins and hand geometry features	1.87%	✓	✗
[47]	IntelRealSense[5]	Palm vein patterns	< 1%	✓	✗
[13]	RGB camera	Hand images features extracted from different layers of a neural network	∼ 5.2%	✗	✗
[65]	Speaker, microphone	Time-domain, frequencey-domain, MFCC[6], and chromagram features of structure-borne echos when holding a device (Without solid hand features)	∼ 6%	✓	✗
[26]	Speaker, microphone, accelerometer	Spectrogram of microphone and accelerometer incurred by notification tones when holding a device (Without solid hand features)	∼ 5%	✓	✗
EchoHand	RGB camera, speaker, microphone	Learning-based acoustic features of structure-borne and air-borne echos while sensing the hand holding device, hand geometry features including finger length, width, palm size and finger distance	∼ 2.45%	✓	✗

[1] Presentation attack resistant. [2] Require users to perform hand motion. [3] A type of depth camera with a range imaging camera system. [4] An infrared-based depth camera used for tracking motions. [5] A high quality LiDAR-based depth cameras. [6] Mel-frequency cepstral coefficients, a kind of typical acoustic features.

notification privacy, [26] leveraged vibration response from microphone and accelerometer spectrogram to identify the user's hand gripping device, where it relies on audible message tones and cannot work well in motion scenarios. EchoHand is different in that it silently senses and profiles a user's hand using the sound propagating through the device and air to provide reliable and secure hand authentication on off-the-shelf devices.

10 LIMITATIONS

Although we took great efforts to maintain our studies' validity, there are limitations in our studies and experiments. For example, EchoHand may not work on devices where the distance between the microphone and the speaker is extremely short, which makes extracting the signal shaped by the holding hand difficult. Besides, the user needs to hold the device in hand to perform authentication with EchoHand. If the device is placed on a desktop or device holder during a meeting, EchoHand cannot authenticate the user. EchoHand might also fail to authenticate the user who wears the gloves. Also, a user's device-holding style may change over time, which may lead to additional false rejection. This issue can be addressed by employing the model updating mechanism as in FaceID. Since EchoHand detects hand landmarks based on vision algorithm, it has the similar limitations to face recognition. For example, similar to facial recognition, hand landmarks detection is not stable under poor lighting, and off-normal shooting angles, which may cause more false rejections. A lower sampling rate may undermine the time resolution of the target signal separating, and lead to the low discernibility of extracted acoustic features among different users.

11 CONCLUSION

In this paper, we present a high accuracy and presentation attack resistant hand authentication method for commodity devices. It uses built-in hardware on off-the-shelf devices, including the microphone, speaker, and camera. To mitigate the threats of presentation attacks, it characterizes the holding hand using acoustic sensing techniques to complement the hand geometry features. We compiled nine datasets to evaluate the reliability and security of EchoHand. The evaluation results demonstrate that EchoHand is robust and can identify the legitimate user and impostors with a low EER.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their constructive comments. This research of Wuhan University was supported in part by the National Key R&D Program of China under grant No. 2021YFB2700200, the Fundamental Research Funds for the Central Universities under grants No. 2042022kf1195, 2042022kf0046, and the National Natural Science Foundation of China under grants No. U1836202, 62076187, 62172303. The corresponding authors are Jing Chen and Kun He.

REFERENCES

[1] 2009. Semantic segmentation. https://github.com/PaddlePaddle/PaddleHub/tree/release/v2.1/modules/image/semantic_segmentation/deeplabv3p_xception65_humanseg.
[2] 2016. *ISO/IEC 30107-1:2016 information technology: biometric presentation attack detection - part 1: framework.* ISO/IEC.
[3] 2017. Interpft. https://ww2.mathworks.cn/help/matlab/ref/interpft.html?lang=en.
[4] 2017. Openpose. https://github.com/CMU-Perceptual-Computing-Lab/openpose.
[5] 2018. DenseNet. https://keras.io/api/applications/densenet/.
[6] 2018. PalmID. https://www.redrockbiometrics.com/.
[7] 2019. Amazon is apparently testing a system that lets you pay by scanning your hand. https://www.foxnews.com/tech/amazon-tests-whole-foods-payment-system-that-uses-hands-as-id.
[8] 2019. Data leak exposes unchangeable biometric data of over 1 million people. https://www.technologyreview.com/f/614163/data-leak-exposes-unchangeable-biometric-data-of-over-1-million-people/.

[9] 2019. One way to unlock the world, powered by your palm. https://one.amazon.com/.

[10] 2019. User guide lg g8 thinq. https://ss7.vzw.com/is/content/VerizonWireless/Catalog%20Assets/Devices/LG/LG_Alpha/lg-g8-thinq-ug.pdf.

[11] 2019. Vein id. https://www.hitachi.com.au/products/product-categories/it/veinid.html.

[12] 2020. PalmSecure: your business, easily secured. https://hyosungamericas.com/storage/app/media/Hyosung-Fujitsu-Whitepaper.pdf.

[13] Mahmoud Afifi. 2019. 11k hands: gender recognition and biometric identification using a large dataset of hand images. *Multimedia Tools and Applications*.

[14] Somaya Al Maadeed, Xudong Jiang, Imad Rida, and Ahmed Bouridane. 2019. Palmprint identification using sparse and dense hybrid representation. *Multimedia Tools and Applications*.

[15] Shanmukhappa Angadi and Sanjeevakumar Hatture. 2018. Hand geometry based user identification using minimal edge connected hand image graph. *IET Computer Vision*.

[16] Shruti Bhilare, Vivek Kanhangad, and Narendra Chaudhari. 2018. A study on vulnerability and presentation attack detection in palmprint verification system. *Pattern Analysis and Applications*.

[17] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM International Conference on Management of Data (SIGMOD)*.

[18] Jagmohan Chauhan, Yining Hu, Suranga Seneviratne, Archan Misra, Aruna Seneviratne, and Youngki Lee. 2017. BreathPrint: breathing acoustics-based user authentication. In *ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*.

[19] Huijie Chen, Fan Li, Wan Du, Song Yang, Matthew Conn, and Yu Wang. 2020. Listen to your fingers: user authentication based on geometry biometrics of touch gesture. *ACM Conference on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*.

[20] Jiayi Chen, Urs Hengartner, Hassan Khan, and Mohammad Mannan. 2020. Chaperone: real-time locking and loss prevention for smartphones. In *USENIX Security Symposium*.

[21] Eunyong Cheon, Yonghwan Shin, Jun Ho Huh, Hyoungshick Kim, and Ian Oakley. 2020. Gesture authentication for smartphones: evaluation of gesture password selection policies. In *IEEE Symposium on Security and Privacy (S&P)*.

[22] Rachel L. German and K. Suzanne Barber. 2018. *Consumer attitudes about biometric authentication.* Technical Report. The University of Texas at Austin Center for Identity.

[23] Puneet Gupta, Saurabh Srivastava, and Phalguni Gupta. 2016. An accurate infrared hand geometry and vein pattern based authentication system. *Knowledge-Based Systems*.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[26] Long Huang and Chen Wang. 2021. Notification privacy protection via unobtrusive gripping hand verification using media sounds. In *ACM Conference on Mobile Computing and Networking (MobiCom)*.

[27] Satoru Imura and Hiroshi Hosobe. 2018. A hand gesture-based method for biometric authentication. In *International Conference on Human-Computer Interaction (HCII)*.

[28] Wei Jia, Bob Zhang, Jingting Lu, Yihai Zhu, Yang Zhao, Wangmeng Zuo, and Haibin Ling. 2017. Palmprint recognition based on complete direction representation. *IEEE Transactions on Image Processing (TIP)*.

[29] Wenxiong Kang and Qiuxia Wu. 2014. Contactless palm vein recognition using a mutual foreground-based local binary pattern. *IEEE Transactions on Information Forensics and Security (TIFS)*.

[30] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. Sound-proof: usable two-factor authentication based on ambient sound. In *USENIX Security Symposium*.

[31] Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *arXiv:1412.6980*.

[32] John R Klauder, AC Price, Sidney Darlington, and Walter J Albersheim. 1960. The theory and design of chirp radars. *Bell System Technical Journal*.

[33] Marek Klonowski, Marcin Plata, and Piotr Syga. 2018. User authorization based on hand geometry without special equipment. *Pattern Recognition*.

[34] Seema Kolkur, D Kalbande, P Shimpi, C Bapat, and Janvi Jatakia. 2017. Human skin detection using RGB, HSV and YCbCr color models. *arXiv:1708.02694*.

[35] Patrick Lazik and Anthony Rowe. 2012. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*.

[36] Larry Li. 2014. Time-of-flight camera–an introduction. *Technical White Paper*.

[37] Chang Liu, Yulin Yang, Xingyan Liu, Linpu Fang, and Wenxiong Kang. 2020. Dynamic-hand-gesture authentication dataset and benchmark. *IEEE Transactions on Information Forensics and Security (TIFS)*.

[38] Giulio Lovisotto, Simon Eberz, and Ivan Martinovic. 2020. Biometric backdoors: a poisoning attack against unsupervised template updating. In *IEEE European Symposium on Security and Privacy (EuroS&P)*.

[39] 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 International Conference on Computer Communications (INFOCOM)*.

[40] Li Lu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Xiangyu Xu, Guangtao Xue, and Minglu Li. 2019. Keylistener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals. In *IEEE International Conference on Computer Communications (INFOCOM)*.

[41] Xiaobo Ma, Mawei Shi, Bingyu An, Jianfeng Li, Daniel Xiapu Luo, Junjie Zhang, and Xiaohong Guan. 2021. Proximity-echo: secure two factor authentication using active sound sensing. In *IEEE International Conference on Computer Communications (INFOCOM)*.

[42] Stéphane Mallat. 1999. *A wavelet tour of signal processing*. Elsevier.

[43] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. Fingerio: using active sonar for fine-grained finger tracking. In *ACM Conference on Human Factors in Computing Systems (CHI)*.

[44] Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv:1712.04621*.

[45] Jianjun Ran. 2008. *Signal processing, channel estimation and link adaptation in mimo-ofdm systems*. Cuvillier Verlag.

[46] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural Computation*.

[47] Syed W Shah, Salil S Kanhere, Jin Zhang, and Lina Yao. 2021. VID: human identification through vein patterns captured from commodity depth cameras. *IET Biometrics*.

[48] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand keypoint detection in single images using multiview bootstrapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[49] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

[50] Yunpeng Song, Zhongmin Cai, and Zhi-Li Zhang. 2017. Multi-touch authentication using hand geometry and behavioral information. In *IEEE Symposium on Security and Privacy (S&P)*.

[51] Katamaneni SriDevi and D Elizbath Rani. 2009. Mainlobe width reduction using linear and nonlinear frequency modulation. In *International Conference on Advances in Recent Technologies in Communication and Computing*.

[52] Petre Stoica, Randolph L Moses, et al. 2005. *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ.

[53] Shridatt Sugrim, Can Liu, Meghan McLean, and Janne Lindqvist. 2019. Robust performance metrics for authentication systems. In *Network and Distributed System Security Symposium (NDSS)*.

[54] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. VSkin: sensing touch gestures on surfaces of mobile devices using acoustic signal. In *ACM Conference on Mobile Computing and Networking (MobiCom)*.

[55] Satoshi Suzuki et al. 1985. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*.

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[57] Yu-Chih Tung and Kang G Shin. 2016. Expansion of human-phone interface by sensing structure-borne sound propagation. In *ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*.

[58] Bolun Wang, Yuanshun Yao, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2018. With great training comes great vulnerability: practical attacks against transfer learning. In *USENIX Security Symposium*.

[59] Wei Wang, Alex X Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *ACM Conference on Mobile Computing and Networking (MobiCom)*.

[60] Xuan Wang and Jiro Tanaka. 2018. GesID: 3D gesture authentication based on depth camera and one-class classification. *Sensors*.

[61] Cong Wu, Kun He, Jing Chen, Ziming Zhao, and Ruiying Du. 2020. Liveness is not enough: enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks. In *USENIX Security Symposium*.

[62] Cong Wu, Kun He, Jing Chen, Ziming Zhao, and Ruiying Du. 2021. Toward robust detection of puppet attacks via characterizing fingertip-touch behaviors. *IEEE Transactions on Dependable and Secure Computing(TDSC)*.

[63] Wei Wu, Stephen John Elliott, Sen Lin, Shenshen Sun, and Yandong Tang. 2019. Review of palm vein recognition. In *IET Biometrics*.

[64] Xiangyu Xu, Jiadi Yu, Yingying Chen, Qin Hua, Yanmin Zhu, Yi-Chao Chen, and Minglu Li. 2020. TouchPass: towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations. In *ACM Conference on Mobile Computing and Networking (MobiCom)*.

[65] Yilin Yang, Yan Wang, Yingying Chen, and Chen Wang. 2020. Echolock: towards low-effort mobile user identification leveraging structure-borne echos. In *ACM ASIA Conference on Computer and Communications Security (ASIACCS)*.
[66] Erdem Yörük, Helin Dutağaci, and Bülent Sankur. 2006. Hand biometrics. *Image and Vision Computing*.
[67] Jiadi Yu, Li Lu, Yingying Chen, Yanmin Zhu, and Linghe Kong. 2019. An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing. *IEEE Transactions on Mobile Computing(TMC)*.
[68] Hans-Jurgen Zepernick and Adolf Finger. 2013. *Pseudo random signal processing: theory and application*. John Wiley & Sons.
[69] Guoming Zhang, Xiaoyu Ji, Xinfeng Li, Gang Qu, and Wenyuan Xu. 2021. EarArray: defending against DolphinAttack via Acoustic Attenuation. In *Network and Distributed System Security Symposium (NDSS)*.
[70] Linghan Zhang, Sheng Tan, and Jie Yang. 2017. Hearing your voice is not enough: an articulatory gesture based liveness detection for voice authentication. In *ACM Conference on Computer and Communications Security (CCS)*.
[71] Ziming Zhao, Gail-Joon Ahn, and Hongxin Hu. 2015. Picture gesture authentication: Empirical analysis, automated attacks, and scheme evaluation. *ACM Transactions on Information and System Security (TISSEC)*.
[72] Ziming Zhao, Gail-Joon Ahn, Jeong-Jin Seo, and Hongxin Hu. 2013. On the security of picture gesture authentication. In *USENIX Security Symposium*.
[73] Bing Zhou, Mohammed Elbadry, Ruipeng Gao, and Fan Ye. 2017. Battracker: high precision infrastructure-free mobile device tracking in indoor environments. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*.
[74] Bing Zhou, Jay Lohokare, Ruipeng Gao, and Fan Ye. 2018. EchoPrint: two-factor authentication using acoustics and vision on smartphones. In *ACM Conference on Mobile Computing and Networking (MobiCom)*.
[75] Man Zhou, Qian Wang, Xiu Lin, Yi Zhao, Peipei Jiang, Qi Li, Chao Shen, and Cong Wang. 2022. Presspin: enabling secure pin authentication on mobile devices via structure-borne sounds. *IEEE Transactions on Dependable and Secure Computing(TDSC)*.
[76] Man Zhou, Qian Wang, Jingxiao Yang, Qi Li, Feng Xiao, Zhibo Wang, and Xiaofeng Chen. 2018. Patternlistener: cracking android pattern lock using acoustic signals. In *ACM Conference on Computer and Communications Security (CCS)*.