Biometric Encoding for Replay-Resistant Smartphone User Authentication Using Handgrips

Long Huang, *Student Member, IEEE*, and Chen Wang, *Member, IEEE*

***Abstract*—Biometrics have been widely applied for user authentication. However, existing biometric authentications are vulnerable to biometric spoofing, because they can be observed and forged. In addition, they rely on verifying biometric features that rarely change. To address this issue, we propose to verify the handgrip biometric that can be unobtrusively extracted by acoustic signals when the user holds the phone. This biometric is uniquely associated with the user's hand geometry, body-fat ratio, and gripping strength, which are hard to reproduce. Furthermore, we propose two biometric encoding techniques (i.e., temporal-frequential and spatial) to convert static biometrics into dynamic biometric features to prevent data reuse. In particular, we develop a biometric authentication system to work with the challenge-response protocol. We encode the ultrasonic signal according to a random challenge sequence and extract a distinct biometric code as the response. We further develop two decoding algorithms to decode the biometric code for user authentication. Additionally, we investigate multiple new attacks and explore using a latent diffusion model to solve the acoustic noise discrepancies between the training and testing data to improve system performance. Extensive experiments show our system achieves 97% accuracy in distinguishing users and rejects 100% replay attacks with 0.6 s challenge sequence.**

***Index Terms*—Challenge response, handgrip biometrics, replay resistance, user authentication.**

I. Introduction

BIOMETRICS such as faces, fingerprints, and irises have been extensively exploited to verify users because they are convenient to use [1]. However, biometric-related security issues are attracting public concerns. Due to the increasingly advanced recording technologies, 3D printing, wireless eavesdropping, and malware [2], the user's biometrics are under two major replay threats, physical forgeries and authentication data reuse. As reported by recent studies, an adversary can perform various types of replay attacks to spoof the user's face [3], [4], [5], fingerprint [5], [6], iris [7], [8] and voice [9], [10]. Addressing the replay issues has become a critical task for ensuring biometric security.

Behavioral characteristics (e.g., gaits) are a rapidly growing category of biometrics, which cannot be physically replicated like body traits and are hard to imitate. To further address the data replay issue, behavioral biometrics are increasingly integrated with Challenge-Response (CR) protocols [11], [12]. Specifically, the user is asked to respond to a random sequence challenge (e.g., letters and icons) for authentication by typing, speaking or eye-tracking. The correctly repeated sequence and the associated behavioral characteristics (e.g., keystroke dynamics, voices, and reflexive eye movements) are verified as the response. However, existing biometric CR solutions all require active participation from the user, such as cognitive activities and behavioral feedback; They are both intrusive and time-consuming, which impedes their deployment. Differently, this work, for the first time, integrates the biometric acquisition process with the CR protocol, which provides user authentication with both enhanced security and unobtrusive user experience.

In particular, this work develops a novel biometric-based CR authentication system for handheld devices, which not only solves the above replay threats but also achieves unobtrusive sensing. The hand-grip biometric inherently comes with handheld devices, and acquiring it requires no more effort compared to obtaining a fingerprint. This biometric was traditionally extracted by an array of pressure sensors that enclose the handheld device [13], [14], [15]. We propose to describe this biometric acoustically as Palm Contact Response (PCR) to facilitate dynamic biometric features. Specifically, when using an ultrasound as the stimulus signal, it interacts with the user's contacting palm and experiences damping, reflection and refraction before reaching the microphone. These signal impacts are resulted from both the user's distinctive physiological traits (e.g., hand geometry, palm size and body-fat ratio) and behavioral characteristics (e.g., gripping strength). While the hand shape can be physically replicated, the body-fat ratio and gripping strengths are more implicit and hard to imitate. Moreover, by manipulating the signal frequencies, we extract different responses from the palm to make every authentication session unique and non-repeated. In addition, the proposed biometric CR authentication can be deployed on any handheld devices (low-end or high-end) that have a speaker and a microphone. No dedicated hardware is required.

Received 26 March 2024; revised 30 September 2024; accepted 2 October 2024. Date of publication 4 October 2024; date of current version 9 January 2025. This work was supported in part by LABoR under Grant LEQSF(2020-23)-RD-A-11 and in part by NSF under Grant CNS-2155131. Recommended for acceptance by W. Xu. *(Corresponding author: Chen Wang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Southern Methodist University Institutional Review Board under Application No. 24-112.

Long Huang is with the Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: lhuan45@lsu.edu).

Chen Wang is with the Department of Computer Science, Southern Methodist University, Dallas, TX 75205 USA (e-mail: cwang6@smu.edu).

Digital Object Identifier 10.1109/TMC.2024.3474673

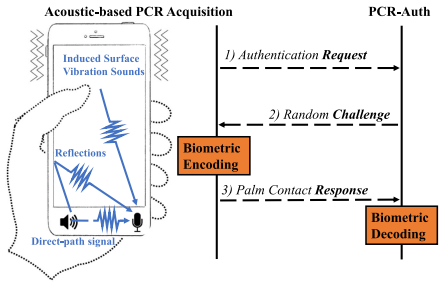


Fig. 1. The handshake process of PCR-Auth: 1) User sends an authentication request. 2) PCR-Auth generates a random challenge. 3) The device sends back the challenge-encoded PCR as a unique biometric response.

We devise two novel biometric encoding techniques (i.e., temporal-frequential biometric encoding and spatial biometric encoding) to integrate the handgrip biometric with the CR protocol. Based on that we develop the PCR-Auth system, whose three-way handshake process is shown in Fig. 1: 1) The user first sends an authentication request. 2) PCR-Auth then generates a challenge (i.e., a random sequence). The device encodes the challenge into a series of millisecond-level ultrasonic pulses on different frequencies and plays the sound to acquire the user's encoded PCR, which includes the direct-path signal, reflections, and the induced surface vibration sounds modified by the user's palm. 3) The encoded PCR is sent back to PCR-Auth and decoded. The access permission is granted only when the decoded sequence is correct and the biometric measurement matches with the profile. Our biometric encoding also enables generating a huge response universe at a minimum training overhead to support everyday authentication purposes.

The PCR-Auth consists of two components: 1) *PCR Encoder* generates a One-Time-Challenge (OTC) Code and transmits the stimulus signals through the narrow-band channels indexed by each OTC digit, which encodes the user's hand-grip biometric into a PCR code. 2) *PCR Decoder* is a per-user deep learning model trained at the registration phase, which verifies both the coding sequence and the PCR. In particular, we exploit an OTC-guided bandpass filter to extract every PCR digit from the right channels. The Signal-to-Noise Ratios (SNRs) of the PCR digits are examined to verify the code sequence, while incorrectly encoded PCR digits (i.e., on the wrong channels) are filtered out resulting in low SNRs. Next, we derive the time-frequency images to examine the user's hand-grip biometric features carried on each PCR digit. We develop two decoding algorithms, based on Convolutional Neural Networks (CNN) and a transformer respectively, to verify all PCR digits and leverage their multi-class classification capability to address human behavioral inconsistency. The model scores of each PCR digit are returned. We then apply a cluster-based method to integrate the model scores and SNRs of all PCR digits to make the authentication decision.

The main contributions are summarized as below:

- *Unobtrusive Biometric CR Authentication:* We propose a solution to address the replay issues of biometric authentication by encoding the biometric features during data acquisition. The authentication process requires neither active user participation nor additional hardware.
- *Implicit Biometric:* We extract the user's hand-grip biometric via acoustic sensing, which is a combination of the physiological and behavioral biometrics of the user's gripping hand. We show that this biometric can show dynamic features under different stimulus signals.
- *Two Biometric Encoding Approaches:* We devise two methods to encode the user's biometric features for defending against replay attacks. Temporal-frequential biometric encoding uses millisecond-level ultrasonic pulses of different frequencies to encode a user's biometric into disposable biometric codes. Spatial biometric encoding leverages smartphone's both speakers to play a dual-channel chirp sound, which at the phone's two mics to form the hard-to-forge geometric relationships. We further add artificial delays between the signals of two channels and examine their SNR relationships to validate the authentication sounds.
- *Two Alternative Decoding Algorithms:* We first derive time-frequency images to capture the user's handgrip biometric and then develop two alternative deep learning algorithms (i.e., the CNN-based algorithm and the transformer-based algorithm) to decode the biometric code of each authentication session, which not only verifies the biometric but also checks the code correctness.
- *Noise Discrepancy Between Training and Testing:* We address a long-lasting challenge in learning-based acoustic sensing: when the testing and training data are under noises of different degrees. We develop a latent diffusion model to synthesize the user's acoustic responses under different noise levels for training data augmentation, which improves our system's authentication performance in noisy scenarios without requiring additional training data collection.
- *Real-world Attacks & Experiments:* We investigate various attacks, including acoustic replays and 3D-printing attacks that can physically replicate the user's hand biometric features. We evaluate the system with different smartphones under these attacks. Results show that our system efficiently verifies users and rejects various replay attacks.

II. Background and System Models

A. Palm Contact Response

The hand-grip biometric is an extension of the hand geometry biometric in the handheld device scenarios, which describes how uniquely a user holds the device. It is traditionally extracted by the pressure sensor-enclosed device surface (e.g., piezoelectric materials) that captures not only the hand geometry but also the pressure distributions of the contacting palm [13], [14], [15]. Due to the high hardware requirement, such a biometric has not attracted much attention.

Motivated by the recent vibration studies that use vibration signals to differentiate people's palms pressing on a surface [16], [17], we find that the ordinary acoustic sounds of a handheld device can distinguish people's palm when it grips the device. Specifically, after the speaker of the handheld device generates a stimulus signal $s(t)$, a portion of the signal propagates in a direct path to reach the microphone (structure-borne or near-surface

air-borne), while other parts of the signal go through more complicated reflected paths as shown in Fig. 1. The user's gripping hand impacts these signals in their propagation paths. Moreover, the speaker's sounds induce the device surface to vibrate at the same frequencies, which serves as a second sound source and creates sounds in the same frequencies and their harmonics, though losing a few frequencies [18]. When in contact with a hand, the device surface vibrations are impeded resulting in modified sounds. All these sounds affected by the hand carry some biometric information when they are picked up by the microphone.

We model the impact of a gripping hand on the speaker sound (input) as a system response $H(f)$. The microphone signal (output) can thus be expressed as $\hat{S}(f) = H(f)S(f)$ in the frequency domain, where $S(f)$ is the original speaker sound at frequency f. To show the microphone signal as the sum of three signal components, the direct-path signal, the reflected signal and the surface vibration sound, we divide the system response into three subsystem responses $H_d(f)$, $H_r(f)$ and $H_v(f)$ accordingly and obtain (1),

$$\hat{S}(f) = H_d(f)S(f) + H_r(f)S(f) + H_v(f)S(f). \quad (1)$$

We further express each subsystem response in terms of its amplitude and phase and obtain (2),

$$\hat{S}(f) = |H_d(f)|S(f) + |H_r(f)|S(f)e^{j2\pi ft} + |H_v(f)|S(f)e^{j2\pi f\tau}, \quad (2)$$

where t and τ are the additional travel time of the reflected signal and the surface vibration sound, compared to the direct-path signal. (2) explains how the three signal components are modified by the gripping hand regarding both amplitude and phase. In particular, the three types of signals at frequency f are all damped by the gripping hand with the scale factors $|H_d(f)|$, $|H_r(f)|$ and $|H_v(f)|$ respectively, which are mainly determined by the user's gripping hand. The reflected signal and the surface vibration sound further suffer from phase changes $2\pi ft$ and $2\pi f\tau$, because they travel longer distances compared to the direct-path signal. The phase changes are more related to the user's hand geometry and holding position. As a result, the combined signal at the microphone presents individually distinctive patterns.

It is important to note that all the amplitude attenuation factors and the phase changes are also related to the signal's frequency. Such a frequency-selective nature motivates us to use the signal with richer spectral points to capture higher resolution of the user's hand-grip biometric. Furthermore, we can use the different combinations of the frequencies to extract dynamic biometric features for CR authentication. Even if an adversary eavesdrops on one authentication session, it is hard to cheat the new session by reusing the previous data. Therefore, we define *Palm Contact Response* (PCR) as

$$pcr = \langle H_d, H_r, H_v, F \rangle, \quad (3)$$

which describes the gripping hand's biometric with three signal components regarding the signal frequencies F.

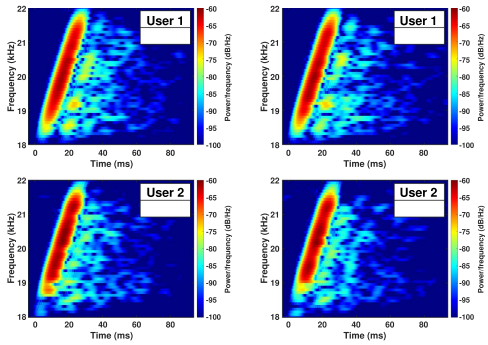


Fig. 2. Distinguishing users by palm contact responses.

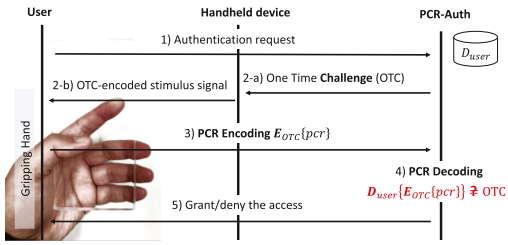


Fig. 3. Our challenge-response authentication model.

B. Motivational Study

We next study the feasibility of using PCRs to distinguish users. In particular, we play a 25 ms chirp signal sweeping from 18 kHz to 22 kHz to interact with the user's gripping hand. Fig. 2 shows the spectrograms of the received chirp signals, when two users grab a smartphone twice, respectively. It is evident that the time-frequency images are consistent for the same user but are distinctive between them. Specifically, not only the dominant direct-path chirp signal but also the sounds after it show distinct patterns between the two users. All these signals present frequency-selective features. These results indicate that we can leverage the temporal and frequential information to achieve robust user authentication. Furthermore, as mobile devices are usually embedded with two microphones for noise cancellation and stereo recording, we can use the two acoustic channels to capture more aspects of the user's PCR. When the speaker sounds travel across different routes to reach the two mics, they are impacted differently by the gripping hand. Such a spatial diversity also adds difficulties for an adversary to cheat the system.

C. Challenge-Response Model

Our system model is shown in Fig. 3, which is an integration of a CR protocol and the PCR coding/decoding modules. The handshake authentication process is between the handheld device user and the PCR-Auth. The PCR decoder $\mathbf{D}_{user}\{\}$ is created for each user, which is pre-trained with all of the user's hand-grip biometric features at the registration phase. The system works in a mechanism that each challenge expects a unique PCR code for verification. When a user sends an authentication request, PCR-Auth generates an OTC Code (i.e., nonce). The handheld

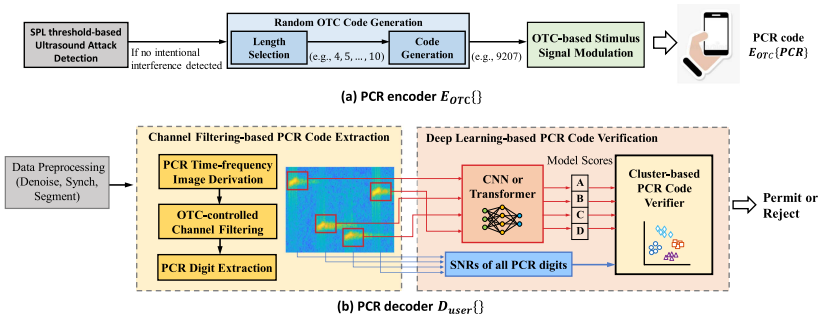


Fig. 4. The architecture of PCR-Auth.

device plays the OTC-encoded stimulus signal using its own speaker(s), and in the meanwhile, its microphones record the signals to obtain the encoded PCR $\mathbf{E}_{OTC}\{PCR\}$. Next, PCR-Auth applies the PCR decoder $\mathbf{D}_{user}\{\}$ to verify the PCR code and make the authentication decision, which examines both the biometric and the coding sequence by $\mathbf{D}_{user}\{\mathbf{E}_{OTC}\{PCR\}\}$.

D. Biometric Encoding and PCR Code.

As mentioned above, the proposed biometric CR authentication is achieved based on the encoded PCR. The authentication function can be expressed by (4),

$$\hat{K} = \mathbf{D}_{user}\{\mathbf{E}_{OTC}\{PCR\}\}. \tag{4}$$

The decoded result $\hat{K}$ matches with the OTC code, only when the presented biometric features and the coding sequence are both correct. This is more secure than the traditional methods that only rely on static biometric features. We now introduce the temporal-frequential biometric encoding, which serves as the basis of PCR-Auth and creates a huge response universe to support everyday CR authentications (the spatial biometric encoding will be introduced later in Section IX). The basic idea is to leverage the frequency-selective nature of PCR. By using the short stimulus signal pulses at different frequencies, we obtain n non-overlapped PCRs and map them to decimal and hexadecimal values (i.e., $n = 10$ or 16) as coding units, which can be used to express more complicated strings. The PCR encoder $\mathbf{E}_{OTC}\{\}$ selects the signal pulses in a sequence according to the OTC code. The *PCR Code* is then extracted by the encoded signals to be the biometric representation of an m-digit OTC as

$$\mathbf{E}_{OTC}\{PCR\} = (pcr_1, pcr_2, \ldots, pcr_m), \tag{5}$$

where pcr_i, $i = 1, 2, \ldots m$ is the ith PCR digit.

The PCR decoder $\mathbf{D}_{user}\{\}$ is trained at the registration phase with the user's all n unique PCR digits. During the authentication, the PCR decoder first examines whether the PCR digits are all at the correct frequencies indexed by the OTC code and then verifies the biometric presented by each PCR digit separately. A successfully verified PCR digit reconstructs one OTC digit. By encoding the user's hand-grip biometric with n coding units into m-digit PCR codes, our biometric encoding technique expands the biometric response universe exponentially from n to n^m based on the same training effort of the prior biometric CR method [17]. As a result, the user does not need to refill the response pool periodically with new biometric features.

E. Framework Overview

Based on the above CR model, we design the PCR-Auth framework as shown in Fig. 4. Upon each authentication request, the microphone access permission is acquired, which is revoked after authentication through auto-reset. The PCR encoder first detects whether the device is under intentional ultrasonic interference by examining the ultrasonic band against a Sound Pressure Level (SPL) threshold, which is introduced in Section VII-A2. If no dedicated ultrasound is detected, the PCR encoder generates an m-digit OTC code, where both the code length (m) and each digit of the code are randomly selected. The OTC-based Stimulus Signal Modulator then selects ultrasonic pulses according to the generated OTC code to encode the user handgrip biometric into a PCR code. It is evident that a longer PCR code can provide a better security, but it will also lead to a longer waiting time. To balance the security and usability, each time we randomly select m from the integers between 4 and 10, which results in a total response pool size of $\sum_{m=4}^{10} n^m$.

The microphone data is the input of the PCR decoder, which first performs Data Preprocessing to denoise, synchronize and segment the audio data. The preprocessed data is fed into the Channel-Filtering-Based PCR Code Extraction to pick up PCR digits from the audio. In particular, we derive time-frequency images to describe the PCR code in both the time and frequency domains. The OTC-controlled Channel Filter sets the cutoff frequencies according to the OTC-indexed channels to extract each PCR digit. Any incorrectly encoded PCR digit (i.e., not on the right channel) is filtered out at this stage, leaving a low SNR. The obtained PCR digits are sent to the Deep Learning-based PCR Decoder for verification, which is a per-user model, trained with the user and a set of other users and stored in the device.

We design the Deep Learning-based PCR Decoder with two alternative algorithms, either n CNN models with five convolutional layers for each or a single transformer model, which decode each PCR digit from its time-frequency image. The biometric verification scores (i.e., probabilities) of all PCR digits are returned. We further develop a cluster-based method to verify the PCR code by integrating the model scores and the SNRs of all PCR digits. Based on that, we compute the PCR code's

euclidean distance to the user's cluster and verify the user using a threshold, which examines both the user's biometric features and the coding sequence. The access permission is granted only when the PCR code successfully recovers the OTC code.

F. Threat Model

We investigate the potential attacks to PCR-Auth. The adversary's goal is to cheat PCR-Auth to pass the authentication. We assume the adversary can physically access the user's handheld device when it is left unattended or stolen. But the adversary can not compromise the device hardware and software, whose integrity is the minimum requirement for authentication and is protected via encryption, memory forensics and circuit security. Unfortunately, the acoustic channel eavesdropping threat is a critical issue for all acoustic systems, because the acoustic channel is open. This is the major reason that most acoustic systems suffer from replay attacks. While it would not be surprising to see our CR authentication defeat replay attacks, we take one step further to study new attacks. For example, the adversary could listen via a side-channel to obtain not only the biometric data but also the chirp signal frequencies (i.e., OTC digit). In particular, we consider the following attacks:

1) Impersonation Attack: In this scenario, the adversary uses his/her own hand to cheat PCR-Auth. The coding sequence is ensured to be correct, and the adversary expects to further imitate the victim's biometric features. Specifically, *random impersonation attacker* arbitrarily grips the victim's device to cheat PCR-Auth; *knowledgeable impersonation attacker* has the prior knowledge of how the victim grips the device, so that he/she can imitate the gripping-hand pose when in possession of the device.

2) Replay Attack: The adversary may have eavesdropped on the victim's authentication data and attempt to use the same data to cheat a new session. To attack, the adversary needs to mute the target device and use a second speaker to replay the prior sounds. This type of attack only aims to present the user's biometric features. But a challenge is to predict the precise time to start the replay, which is only a short period (e.g., 400ms for 4-digit OTC) when the mic is on. A possible solution is to turn the target device volume to be low to detect the start of the stimulus signal and then attack immediately.

3) Fake Hand Attack: The adversary can attack our system with fake hands. Based on the adversary's ability, we consider two types of fake hand attacks. In particular, we first consider a *knowledgeable fake-hand attacker*, who uses a silicone fake hand to imitate the victim's hand with more freedom. We further consider a *3D-printed biometrics attacker*, who uses current 3D scanning and printing technologies to physically replicate the user's gripping hand. The replicated fake hand reproduces some of the user's biometric features such as a similar shape/size and gripping pose as the user's hand.

4) DoS Attack: The Denial of Service (DoS) attack aims to cause authentication errors and rejections by overriding the working frequencies of PCR-Auth via dedicated ultrasounds.

III. Temporal-Frequential Biometric Encoding

A. Palm Contact Response Encoder

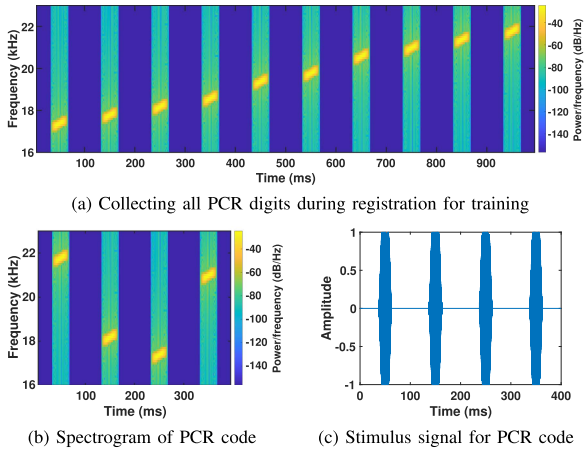
1) Stimulus Signal Design: The stimulus signal is used to interact with the user's palm and extract the PCR for authentication. In order to acoustically obtain sufficient biometric information, we exploit the upward frequency sweeping signals to capture the user's biometric in a frequency range rather than a single frequency. Intuitively, a wider frequency band enables describing more aspects of the user's biometric, and a longer time period means more audio samples and thus higher resolutions. However, to facilitate biometric encoding, we design the stimulus signals in narrow bands and short periods. The reasons are two-fold. First, the secure biometric encoding requires all PCR digits to have non-overlapping biometric information, making it necessary for us to divide the available frequency range into a number of exclusive narrow bands (i.e., channels) and extract the frequency-separable PCR digit. Second, the time period of the stimulus signal is directly related to the waiting time and must be short.

Besides the function-level requirements, a critical consideration is that the signal must be non-invasive and do little harm to humans and animals. Thus, we propose to use the ultrasounds easily generated by off-the-shelf handheld devices, whose frequency range complies with the Federal Communications Commission (FCC) Rules & Regulations Title 47 Part 18 to ensure low risks to human and animals [19]. In particular, we apply the signals within the range 17 k-22 kHz, which has been demonstrated to be hardly audible [20] and widely applied in prior ultrasonic sensing work [21], [22], [23], [24]. We further reduce disturbances by designing the stimulus signal with millisecond-level short periods, hundred-Hz-level narrow bands and the low energy (e.g., 50% volume).

To balance the above considerations, we design the stimulus signals as a number of 25 ms long and 350 Hz wide chirp pulses within the range $17\,\text{k} - 22\,\text{kHz}$. The signal frequency bands are 10 times narrower than the prior acoustic sensing work (i.e., $4-$ 6kHz wide [21], [22], [23], [24]), which means more challenge for our sensing. But we show that such narrow-band pulse signals are sufficient to distinguish people's palms. Moreover, we add a 75ms silent period after each chirp for leveraging the reflected signals and the induced surface vibration sounds in this period and reduce the inter-chirp interference. We further apply a Hamming window to both ends of each chirp to suppress the spectral leakages caused by sudden frequency changes and the hardware noises of the speaker. The complete stimulus signals used for both the registration and authentication are illustrated in Fig. 5.

2) PCR Encoding: The purpose of PCR encoding is to encode the user's hand-grip biometric into a unique PCR code based on the OTC, which can be generated by existing methods [25], [26], [27]. For simplicity of description, we select 10 exclusive narrow-band channels from the range 17 k-22 kHz to represent decimal digits. These coding channels are all 350 Hz wide and separated by a gap (e.g., 50 Hz).

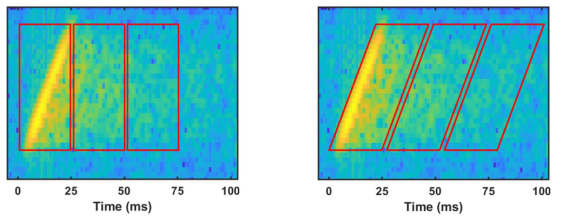
Chirp pulse is used as the basic unit to encode the user's PCR onto the corresponding channel. When training the PCR decoder,



(a) Collecting all PCR digits during registration for training

(b) Spectrogram of PCR code (c) Stimulus signal for PCR code

Fig. 5. Stimulus signals for training and authentication.



(a) Rectangular segmentation window (b) Rhomboid segmentation window

Fig. 6. Illustration of different segmentation methods.

the user's PCRs at all coding channels are collected as shown in Fig. 5(a). During the authentication, the PCR encoder scopes down to each coding channel indexed by the OTC to extract the corresponding PCR digit. Fig. 5(b) illustrates the stimulus signals for encoding a 4-digit PCR code, when the OTC code is "9207" and the selected chirp pulses are 21.6-21.95 kHz, 18.8-19.15 kHz, 21.6-21.95 kHz and 20.8-21.15 kHz in a sequence. When the stimulus signals interact with the user's contacting palm, a unique PCR code is contained in the received audio. As many as 10^4 unique 4-digit PCR codes can be generated, which are disposed of after being used.

B. Palm Contact Response Decoder

1) Denoising, Synchronization and Segmentation: The raw audio data is first preprocessed for denoising, synchronization and segmentation. In particular, a minimum-order IIR filter with the pass-band 17–22kHz is used to remove the noises out of the stimulus signals' frequency range, including the low-frequency mechanical noises caused by the gripping hand and the audible ambient noises. The frequency range is selected to include all PCR digit channels, which are in this range and also barely audible to users. Next, the synchronization is performed by leveraging the evenly spaced chirp pulses. Specifically, we use the original pulse sequence signal as the reference and calculate its cross-correlation with the received audio to find the time shift $synch\_shift$ that corresponds to the maximum correlation coefficient as expressed by

$$synch\_shift = \underset{d}{argmax}\ xcor(d). \tag{6}$$

We then use this shift to align the two signals and refer to the reference signal to localize the coding chirps in the audio for segmentation. Each resulted segment contains one 25 ms coding chirp and a 75 ms stop period to represent a PCR digit.

2) PCR Code Extraction: Because each PCR digit is encoded onto one of the predefined coding channels by the OTC, we use a bandpass filter to extract the PCR code by scoping down to each OTC-indexed channel in a sequence. For example, the upper and lower frequency bounds of the pass-band are set as 21.6-21.95 kHz, 18.8-19.15 kHz, 21.6-21.95 kHz and 20.8-21.15 kHz when the OTC code is "9207" as shown in Fig. 5(b). As a result, only the PCR digits at the right channels pass the filter, while the incorrectly encoded PCR digits are filtered out. We detect the coding errors by examining the SNRs of all coding channels both before and after filtering. The SNR examination before filtering is to make sure the coding complies with the rule: only one channel is encoded at each time slot. The examination after filtering is to detect whether there are coding errors. The SNRs of all PCR digits are used to verify the PCR code as the physical layer coding features. We next examine the biometric features.

3) PCR Time-Frequency Images Derivation: We derive the time-frequency image of each PCR digit as biometric features to describe the PCR defined by (3). The spectrogram shown in Fig. 2, as a typical type of time-frequency image, describes the temporal changes of the resulted signal at each frequency, after the original speaker signal passes a specific gripping-hand system. It is a measurement of the three subsystem responses (H_d, H_r and H_v) regarding the frequencies and the waveform patterns of the speaker signal. In addition to the spectrogram, we also derive other three types of time-frequency images for comparison, including the scalogram, the persistence spectrum, and the Hilbert spectrum. Our system can choose one type of time-frequency image as input.

4) PCR Time-Frequency Image Time Series: In order to balance the time-frequency image resolution and the decoding algorithm's complexity, we divide each PCR time-frequency image into three pieces, which separately describe three different stages of the PCR. Specifically, the first time-frequency image (0–25 ms) mainly captures the palm's impact on the dominant direct-path signal. The second (25–50 ms) and third (50–75 ms) focus on the reflected signals and the induced surface vibration sounds. All of the three time-frequency image pieces show user-distinctive patterns and are input in 2D-image time series into the PCR decoder for verification. The 75–100 ms subsegment is not utilized, because the sound degrades over 20 dB in this period. Furthermore, using time-frequency image time series also adds difficulties to the PCR digit forgery. While the direct-path signal can be synthesized, it is hard to forge the reflections and the surface vibrations that are byproducts affected by many other factors.

Advanced Segmentation Method. The above-defined time-frequency image time series of each PCR digit are segmented solely based on time, which are equivalent to the rectangular areas/windows in the time-frequency image as shown by Fig. 6(a).

However, since each PCR digit is a chirp signal with a linearly sweeping frequency, the starting time of each frequency point (i.e., the time when each frequency component starts to play) is different. Therefore, a more accurate segmentation method is needed to eliminate the biometric-unrelated signals that locate before the starting time of each frequency component of the chirp been played. In particular, we propose a new segmentation method that segments the time-frequency image time series based on both time and frequency. For each frequency component of the PCR digit, we segment the corresponding time-frequency image time series from the 75 ms duration right after the frequency component is played. This results in the equivalent rhomboid areas/windows in the time-frequency image, as shown by Fig. 6(b). The system performance with using these two different segmentation methods is compared in Section V-A3.

5) Deep Learning-Based PCR Digit Verification: When distinguishing people' hands from each PCR digit, we have the following considerations for the algorithm design: 1) The algorithm needs to be powerful to distinguish the minute differences of the acoustic signals modified by different hands; 2) The behavioral inconsistency of the user (e.g., the gripping pose changes) must be addressed; 3) The remaining ambient noises after denoising need to be tolerated; 4) The algorithm must have reasonable complexity to be usable for handheld devices. After testing multiple learning-based algorithms, we find the CNN model and the transformer model best meet the above requirements. Here we first introduce the CNN-based algorithm, which is used for performance evaluation in Section V. The transformer-based algorithm will be introduced later in Section VIII for comparison.

CNN-Based Algorithm: CNN is a deep-learning model widely used for finding patterns in images. It is thus good for capturing a gripping hand's characteristics from the 2D spectrogram images while tolerating ambient noises and behavioral inconsistency. When using PCR-Auth for the first time, the user is allowed to define a customized gripping hand pose, and a floating button on the screen marks the user's thumb location, which is displayed later for the user to recall the hand pose. But when the user grabs the device at different times, the grabbing actions may result in slightly different acoustic signal patterns. We thus leverage the CNN model's strong multi-class classification capability to cope with the potentially varying hand-grip patterns for the user, which enables the system to tolerate the user's behavioral inconsistency in practical scenarios. Specifically, when training the CNN model, the user is asked to re-grab the device multiple times, just as setting up the finger ID by pressing and lifting a finger multiple times [28]. The per-user CNN model is then created and stored in the device. Additionally, we design the CNN model with five convolutional layers and a small number of filters, which is a CNN architecture widely used for mobile devices [29]. We use the Rectified Linear Unit (ReLU) for the activation function to speed up the training, and each activation layer is followed by a 3×3 max-pooling layer to downsample the feature maps. The last max-pooling layer pools the input feature map globally over time to cope with the temporal variances of the spectrogram and reduces the parameter number in the final fully connected layer. In addition, we apply the batch normalization layers to normalize the output of each layer and a dropout layer to suppress over-fitting. The cross-entropy is used as the loss function, and the softmax layer outputs the final CNN scores of each input. The trained model has 64 202 parameters and a size less than 0.4 MB. The time and space complexity are 22.6 M FLOPs and 1.2based algorithm to verify a PCR code by integrating the MB, respectively, which can be supported by most mobile devices [29].

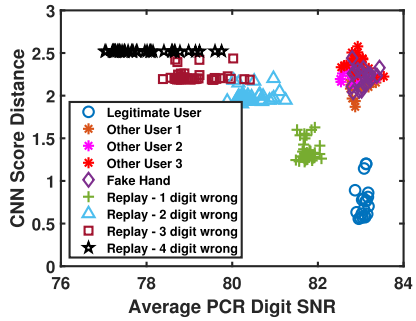


Fig. 7. Illustration of the cluster-based PCR code verifier.

6) Model Scores of the Input: We resize each spectrogram into a 98×40 time-frequency image as the input to the CNN model and the transformer model. The model scores (i.e., probabilities) are computed, which are associated with two classes, *User* and *Non-User*. A higher model score for the *User* class indicates a higher confidence to trust the biometric presented by the PCR digit. Since each PCR digit is divided into three consecutive spectrogram pieces and the smartphones have two microphone channels, a PCR digit is decoded into $12 = 2 \times 3 \times 2$ model scores. For each $m-$digit PCR code, we thus obtain $12\,m$ model scores as the biometric features for verification.

7) Cluster-Based PCR Code Verifier: We develop a cluster-based algorithm to verify a PCR code by integrating the biometric features ($12\,m$ model scores) and the coding features ($2\,m$ SNRs), which are projected into a high dimensional space for binary classification. Moreover, the proposed clustering algorithm explores the relationships among the m PCR digits to improve the decoding performance rather than treating each digit alone. The user's cluster is learned during the training phase. Specifically, we generate a large number of random m-digit PCR codes based on the user's training data and a non-user database. We also simulate diverse replay attack cases, assuming they present $1, 2, \ldots, m-1$ correct PCR digits. We then cluster these PCR codes based on their model scores and SNRs to find the user's cluster and its center and radius. During authentication, we calculate the euclidean distance of the PCR code to the user's cluster center and verify the user via a threshold-based method.

Fig. 7 illustrates the 2D clustering results of 240 random 4-digit PCR codes in the plane of CNN score distance and average SNR. We observe that the legitimate user's PCR code cluster is clearly separated from the other users (here we include three other users for illustration), the different cases of replay attacks, and a fake silicone hand. By presenting more correct digits, the replay-attack clusters are closer to the user, regarding both CNN score and SNR. For example, the replayed PCR codes

with 1 digit error have the smallest CNN score distances and the closest average digit SNR to the user. But 1-digit error is sufficient to identify them as non-valid inputs. In comparison, the inputs from the other users and the fake hand are valid as that of the user, which shows the similar digit SNRs. But their biometric features are distinguished from the user by our CNN model. Fig. 7 indicates that only breaking the coding sequence or replicating the biometric features alone is hard to attack PCR-Auth.

IV. Methodology and Evaluation Criteria

Experimental Setup: We experiment with seven different mobile device models ranging from $140 to $350, including Samsung Galaxy Note5, S8, and S20, Xiaomi10, Google Pixel2, LG K50, and Moto G8. The S8 phone is used in all scenarios. The stimulus signal is played through the phone speaker at 48kHz. Moreover, only 50% volume is used to reduce power and disturbances. The signal is recorded by the mobile device's two microphones, Mic 1 (i.e., top) and Mic 2 (i.e., bottom) with 48kHz sampling rate. We recruit 46 participants (26 males and 20 females) aged from 20 to 40 for experiments. The participants are formed by graduate students and faculties, and each is given a $10 gift card for incentive. The data are anonymized and processed offline. This study is approved by LSU with IRB4305.

Experimental Platform: We develop an experimental platform based on Android, which selects and plays the pre-generated chirp signals according to the randomly generated OTC code. In particular, the platform launches three threads to collect the authentication data, including the main thread, one thread to record the stereo sound using android.media.AudioRecord, and one thread to play the notification tone using android.media.MediaPlayer. The authentication data is processed offline.

Data Collection: The participants are asked to grip each given device for 10 minutes to get familiar with it before data collection. They are allowed to choose self-defined gripping-hand pose, and the most comfortable one is suggested. A floating button is provided to mark the thumb location on the screen, which is displayed later to recall the participant's memory of the gripping-hand pose. Each participant's data is collected in two sessions spaced by at least three weeks apart, with the first only used for training and the second for testing. A session lasts about 30 minutes. In the first session, the stimulus signal for training as shown in Fig. 5(a) is repeatedly played 20 times, and the participants are asked to re-grab the device from a table each time to present behavioral inconsistency. *re-grab-1* is collected, which contains 200 chirps at 10 frequency bands for each participant, and we respectively choose each participant as the user and the others as the non-user to train each per-user model. In the second session, the same experiment is repeated 40 times, and *re-grab-2* is collected, which contains 400 chirp signals for each participant and is used for the basic PCR analysis in Section V-A. Moreover, in the second session, the *pcr-code* data set is collected for evaluating PCR-Auth, where the stimulus signal encoded by a set of 40 different OTC codes is played similar to Fig. 5(b). 40 PCR codes are collected from each participant, when a re-grab is required each time to imitate an authentication session.

Evaluation Metrics: We first conduct the basic PCR analysis to examine the *accuracy* performance of using the biometric PCR to distinguish users, which is defined as the ratio of accurately classified test instances over all test instances. We then evaluate the authentication performance of PCR-Auth using PCR codes. In particular, we compute the *False Rejection Rate* (FRR) to examine the ratio that legitimate users are mistakenly rejected and the *False Acceptance Rate* (FAR) to show the success rate of an adversary to attack the system.

V. Performance Evaluation

A. Basic Analysis

1) PCR Coding Channels and Mic1&2: Based on our preliminary study, we choose 25 ms chirps with a 350 Hz bandwidth as coding signals. In the ultrasonic frequency range 17–22 kHz, we find 12 exclusive channels as candidates. We then evaluate the verification performance of these channels. We find that the 46 participants' PCRs are distinguished accurately on 10 channels, which achieve an average accuracy of 90.9% when two mics are used. We thus choose the 10 channels for decimal encoding. Moreover, we find that the performances of coding channels vary. For example, the accuracy achieved by Channel 2, 4, 6 and 9 is 89.2%, 88.6%, 96.1% and 92.9%, respectively, when two mics are used. The results confirm the frequency-diversity nature of PCR.

When comparing the microphones, we find that Mic 2 (bottom) close to the speaker performs better than Mic 1 (top) for most channels. The result contradicts with the intuition that the top mic-received signals should present higher accuracies because they travel across the entire smartphone body and are more heavily affected by the gripping hand. One reason is that the closer-to-speaker location makes Mic 2 more robust (or less sensitive) to the user's behavioral inconsistency and hand-grip pose changes. Moreover, some smartphones use Mic 2 as the main microphone for recording and Mic 1 for noise cancellation. Though both microphones can be used for stereo recording, Mic 2 presents higher-quality sounds and shows higher SNR when responding to our stimulus signals. The integration of the two mics makes a more robust authentication system by adding an extra sensing channel.

2) Different Time-Frequency Representations: As introduced in Section III-B3, we derive different time-frequency images to represent the user's hand-grip biometrics embedded in the PCR, which include the spectrogram derived by the short-time Fourier transform, the scalogram derived by the continuous wavelet transform, the Hilbert spectrum derived by the Hilbert-Huang transform, and the persistence spectrum. We then compare our system performance of using such time-frequency representations as the input and present the results in Fig. 8. We can observe that when using one PCR digit recorded by Mic1+2, the spectrogram achieves the best user verification performance with 90.9% average accuracy, which is higher than the 85.1% average accuaracy achieved by the scalogram. This is because compared to the short-time Fourier transform, the continuous

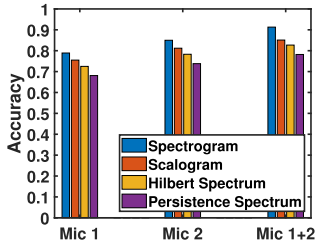


Fig. 8. Comparing time-frequency analysis methods.

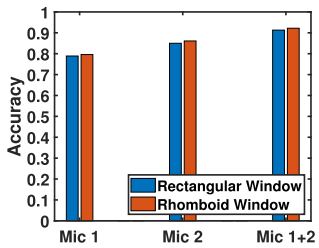
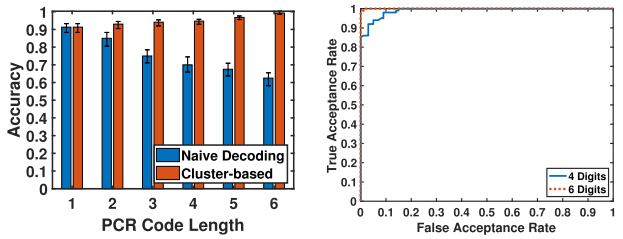
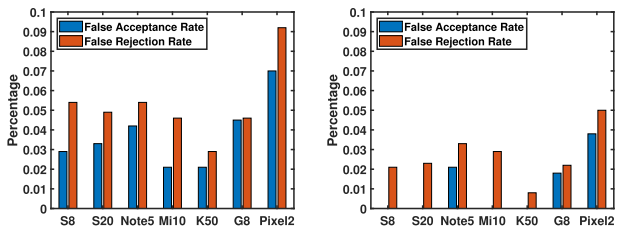


Fig. 9. Comparison of different segmentation methods.



(a) Different code lengths (b) ROC curves of 4- and 6-digit code

Fig. 10. Performance of PCR-Auth and coding gains.



(a) 4-digit PCR code (b) 6-digit PCR code

Fig. 11. Performance of different device models.

wavelet transform's time-frequency resolution is not fixed but adjusts to the signal's different frequencies, where lower frequency components are represented with finer frequency resolution and coarser time resolution, while higher frequency components are represented with coarser frequency resolution and finer time resolution. Since our ultrasonic chirp signal mainly locates in the higher frequency range, the frequency resolution of the derived scalogram is much poorer than that of the spectrogram. The Hilbert-Huang transform and the persistence spectrum also achieve lower user verification performance with 82.7% and 78.2% average accuracy, respectively. The results indicate that using the short-time Fourier transform performs better than using the other three time-frequency images. The remaining part of the paper thus uses the spectrogram for performance evaluation.

3) Segmentation: Rhomboid vs. Rectangular: We further compare the system performance when two different time-frequency image segmentation methods (i.e., with either a rhomboid or rectangular segmentation window) are used. The results are shown in Fig. 9. We can observe that with a rectangular segmentation window, our system achieves 78.9%, 85.0%, and 90.9% average user verification accuracy when using one PCR digit recorded by Mic 1, Mic 2, and Mic 1+2. While when a rhomboid segmentation window is used, the performance is slightly improved to 79.6%, 86.1%, and 91.8% for Mic 1, Mic2, and Mic 1+2, respectively. The reason is that the rhomboid segmentation window ensures the time-frequency images are exactly derived from the 75 ms duration after each frequency component of the stimulus chirp, which removes the biometric-unrealted signals that locate before the starting point of each frequency component. The results indicate that with a more advanced segmentation method, the system performance can be further improved.

B. Performance of PCR-Auth

1) Security Gains of PCR Code: We now present the performance of PCR-Auth with different code lengths and compare it with a naive decoder, which treats each PCR digit alone for decoding rather than leveraging their relationships. Fig. 10(a) shows the user verification accuracy when 1-digit to 6-digit PCR codes are used respectively. We observe that both methods achieve 90.9% average accuracy with 1 digit PCR code. But when using longer PCR codes, the performance of the naive decoder decreases drastically, because it requires all PCR digits to pass the verification independently. In comparison, the accuracy of our cluster-based PCR decoder increases. Specifically, our method achieves 94.7%, 96.8% and 99.3% average accuracy with 4-digit, 5-digit, and 6-digit PCR codes. Our cluster-based method also shows a better confidence interval (i.e., a smaller error bar range) than the naive decoding method. The reasons why PCR-Auth achieves higher performances with longer PCR codes are threefold: First, the longer PCR codes involve more coding chirps and thus have an increased temporal diversity to describe the user's biometric; Second, the PCR digits at different channels leverage the frequency diversity to capture different aspects of the biometric; Third, our cluster-based method exploits the connections and constraints among PCR digits to decode a PCR code and leverage its coding gain.

The ROC curves of the 4-digit and 6-digit PCR codes in Fig. 10(b) further confirm the high performance of PCR-Auth, and both codes achieve a high TAR and a low FAR. In particular, the 4-digit PCR code achieves 94% TAR and 4.6% FAR, while the 6-digit PCR code achieves close to 99.6% TAR and 1% FAR.

2) Performance of Different Phone Models: We next evaluate the performance of our system on seven different smartphone models, when fifteen participants are involved. Fig. 11 shows the FAR and FRR performance of PCR-Auth when 4- and 6-digit

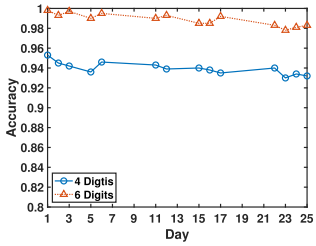


Fig. 12. Long-term study.

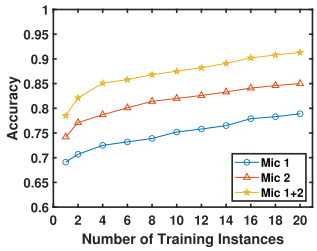


Fig. 13. Training size study.

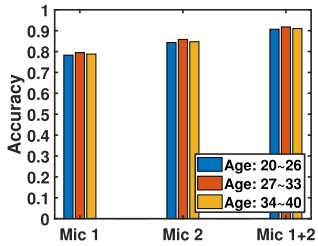


Fig. 14. Age group.

codes are used. We observe that all the seven devices achieve a low FAR and a low FRR. When using 6-digit PCR codes, S8, S20, Mi10 and K50 all achieve 0% FARs, and their FRRs are 2.1%, 2.3%, 2.9% and 0.8% respectively. Similar to the three devices, Note 5 and G8 achieve around 2% FAR and 2.5% FRR. Pixel 2 does not perform as well as the other five devices. The reason may be that Pixel 2 has the non-smooth or matte back surface, which impacts the stimulus signal propagation. But Pixel 2 still achieves 3.8% FAR and 5% FRR. When using 4-digit PCR codes, the performance degrades slightly. In particular, S8 achieves 2.9% FAR and 5.4% FRR, and that of S20 are 3.3% and 4.9% respectively. Additionally, we find the performance of PCR-Auth is not associated with the device price. For example, K50 is the cheapest, but it achieves the best performance. The results show the potential to deploy PCR-Auth generally on most handheld devices.

3) Long-Term Performance: In addition to using the two-session data, we also continuously collect data from 8 participants with S8 in 25 days for the long-term performance study. 40 PCR codes are collected for each participant for each day, which are only used for testing. Fig. 12 shows the performance changes of 4- and 6-digit PCR codes in this period. We find that both codes have a stable accuracy performance along time, which only slightly decreases. Moreover, we observe some fluctuations and two local minimums on Day 5 and Day 23. These slight performance changes are caused by many inconsistent factors on each day, including hand moisture, mood, body weight and clothes. The results reflect the robust performance of PCR-Auth over a long term.

C. Training Size Study

To learn the training efforts required in the practical deployment of our system, we conduct a training size study that evaluates the impact of training size on our system performance. In particular, we fix the testing set (where 40 PCR codes are collected from each participant in the second session for testing) and tune the size of the training set to evaluate our system. As we can observe in Fig. 13, when the number of the training instances (from each participant) is increased from 1 to 20, the average user verification accuracy of our system is improved from 69.1% to 78.9%, from 74.2% to 85.0%, and from 78.5% to 90.9% when using one PCR digit recorded by Mic1, Mic2, Mic1+2, respectively. Moreover, using only 10 instances (from each participant) for training can already achieve an accuracy of over 75%, 82%, and 87% when using one PCR digit recorded by Mic1, Mic2, and Mic1+2, respectively. The results show the potential of further reducing training efforts while maintaining a good verification performance. Our system turns out to work well with low training efforts, making itself suitable to be deployed on mobile devices.

D. Other Impact Factors

To evaluate the robustness of the proposed user authentication system, we further conduct a more comprehensive study to learn the impacts of different age groups, genders, and body fat ratios. The results are presented below.

1) Age Groups: The reason why we study this impact factor is that people from different age groups could have different gripping behaviors (e.g., gripping strength, gripping styles, etc.) that may influence the performance of our user authentication system. To validate this, we divide the participants into three different age groups (i.e., 20 to 26, 27 to 33, and 34 to 40) and evaluate our system on each group separately. As shown by Fig. 14, our system turns out to have a similar performance for the three different age groups. In particular, when using one PCR digit recorded by Mic1, our system achieves 78.3%, 79.5%, and 78.8% average accuracies in verifying the users from the three age groups. While when using one PCR digit recorded by Mic2, the average user verification accuracies are 84.3%, 85.8%, and 84.7%. After integrating one PCR digit recorded by Mic1 and Mic2, the system performance is further improved, achieving average accuracies of 90.5%, 91.4%, and 90.7%. The results indicate that our system generally works for users from different age groups who may have different gripping behaviors.

2) Genders: People of different genders usually have different hand shapes and hand sizes, where males usually turn to have larger palms and longer fingers than females. To study whether genders have an impact on the system's user verification performance, we evaluate our system separately on the male

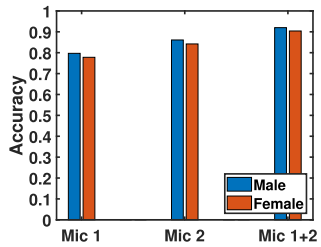


Fig. 15. Gender.

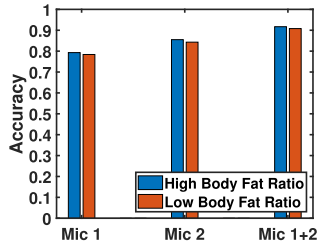


Fig. 16. Body fat ratio.

users and the female users. The results are presented in Fig. 15. We can observe that our system achieves average accuracies of 79.7%, 86.1%, and 91.7% in verifying the male users when using one PCR digit recorded by Mic1, Mic2, and Mic1+2. While the user authentication performance for female users is slightly lower, achieving average accuracies of 77.8%, 84.2%, and 90.4% when using one PCR digit recorded by Mic1, Mic2, and Mic1+2, respectively. The results indicate that genders only have a small impact on the system performance, which can be neglected.

3) Body Fat Ratios: We also study the impact of the user's body fat ratio as it also determines people's hand shapes, thus influencing the played stimulus signal. In general, people with higher body fat ratios always have thicker palms and thicker fingers, while people with lower body fat ratios usually have thiner palms and thiner fingers. We thus divide the participants into two groups based on their body fat ratios and evaluate our system on the two groups. As we can observe in Fig. 16, our system achieves 79.3%, 85.5%, and 91.4% average accuracies in verifying the users with higher body fat ratios when using a single PCR digit recorded by Mic1, Mic2, and Mic1+2. While for the users with lower body fat ratios, the user authentication performance achieved by our system is 78.4%, 84.3%, and 90.3% when using a single PCR digit recorded by Mic1, Mic2, and Mic1+2, respectively. The results show that our system generally works for people with different body fat ratios.

E. Impersonation Attack

1) Setup: We perform two types of impersonation attacks. For *random impersonation*, each of the participants is treated as the target user respectively, while the other participants' data is used for testing. For *knowledgeable impersonation*, the authors and six participants act as the skilled adversaries, who learn how each target user grips the device from videos and then imitate the gripping hand to attack. The attackers attempt 40 times for each target user's OTC.

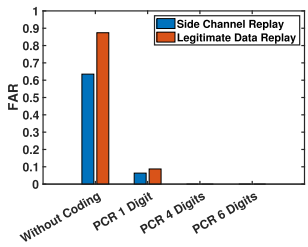


Fig. 17. Under replays.

2) Result: Table I presents the performance of the 4- and 6-digit PCR codes under the two impersonation attacks. We observe that both PCR codes achieve a low FAR and a low FRR in the two attacking scenarios. In particular, the 6-digit PCR code achieves 0.4% FRR and 1% FAR for the random impersonation, and the Equal Error Rate (EER) is 0.8%. The knowledgeable impersonation slightly degrades the performance of the 6-digit PCR code. But its EER is still low under this advanced impersonation attack, which is 2.6%. The 4-digit PCR code has a lower performance compared to the 6-digit PCR code, whose EERs are 5.7% and 6.5% in the random and knowledgeable impersonations, respectively. The results indicate the difficulty of replicating the user's PCR via impersonation attacks.

F. Replay Attack

1) Setup: For each participant, we respectively choose each of his/her 40 PCR codes for the current session and use the other 39 codes for replay attacks. As these replay codes only cover a small set of digit combinations, we further use *re-grab-1* data to construct 560 PCR codes for each participant and replay them. As a result, the replayed codes may have 1, 2, . . . , 5, 6 digit differences from the target code. For replay, we use the target user's audio data and assume the adversary precisely predicts the authentication start to launch the attack.

2) Result: Table I shows the performance of PCR-Auth under replay attacks. We find that both the 4- and 6-digit PCR codes prevent all replay attacks with 0% FAR, including the case when there is only 1-digit error. The reason is that each PCR code is only used once, and coding errors can be accurately detected based on the physical layer SNR of each digit. Fig. 17 further compares the performance of PCR-Auth with the traditional method without coding and the side-channel replay with the software-level replay. We find that without coding, the traditional biometric method suffers from 87% FAR and 63% FAR, when the software-level replay and the side-channel replay are launched, respectively. Even encoding the biometric with a single decimal digit could reduce the FAR by a factor of ten. The results confirm the security of PCR-Auth and indicate that an adversary could not attack PCR-Auth if not presenting the correct coding sequence.

G. Fake Hand Attack

1) Setup: For *knowledgeable fake hand attack*, we use a silicone fake hand [30] to replicate each user's gripping hand by placing the fake hand's fingers similar to the user's ones. For *3D-printed biometrics attack*, we randomly select two participants as

TABLE I
PERFORMANCE OF PCR-AUTH UNDER IMPERSONATION, REPLAY, AND FAKE HAND ATTACKS

Code	FRR	FAR									
		Impersonation		Replay (#Err Digit)						Fake Hand	
		Random	Knowledgeable	1	2	3	4	5	6	Knowledgeable	3D-printed
4 Digits	0.063	0.046	0.066	0	0	0	0	-	-	0.058	0.075
6 Digits	0.004	0.010	0.035	0	0	0	0	0	0	0.026	0.038

the victims and scan their hands using a commodity 3D scanner (Revopoint POP 2) to obtain the 3D hand models, which are then printed by a commodity 3D printer (Creality CR-10S) using a close-to-skin material (thermoplastic polyurethane). Each of the fake hands is used to attack our system for 40 times.

2) Result: The performance of the 4- and 6-digit PCR codes under the two types of fake hand attacks is shown in Table I. It can be observed that both PCR codes perform well in preventing the knowledgeable fake hand attack, achieving 5.8% and 2.6% FAR, respectively. For the 3D-printed biometrics attack, the two PCR codes perform slightly worse with FARs of 7.5% and 3.8%. This is because the 3D-printed hand is more close to the user hand in shape, size, and gripping pose, which may reproduces some of the user's biometric features. However, the results indicate that it is still hard to replicate the user's PCR using such fake hands.

VI. BIOMETRIC SECURITY ANALYSIS

To further evaluate the security of the proposed biometrics, we analyze its unlinkability and irreversibility according to the ISO/IEC 24745 international standard. In particular, we use the framework proposed in prior works [31], [32] and consider the high-level features extracted by the CNN model as the CNN-learned template for each user. Instead of evaluating the whole authentication system, which encodes biometrics to randomize the order of PCR digits (and their biometric templates), we test single PCR digits to focus on the unlinkability and the irreversibility of the biometric factor alone.

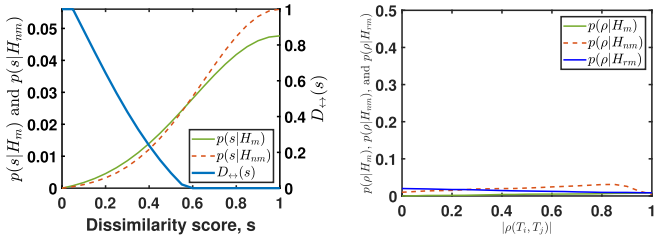
A. Unlinkability

Unlinkability of the biometrics requires templates derived from the biometric samples of the same subject (user) cannot be linked to each other. Following the linkability analysis in [31], a dissimilarity score between two templates T_1 and T_2 is defined as $s = 1 - |\rho(T_1, T_2)|$, where $\rho(T_1, T_2)$ is Pearson's correlation coefficient between T_1 and T_2. The linkability $D_{\leftrightarrow}(s)$ of the two templates is then defined as

$$D_{\leftrightarrow}(s) = \begin{cases} 0, & \text{if } LR(s) \leq 1 \\ 2[(1 + e^{-(LR(s)-1)})^{-1} - 0.5], & \text{if } LR(s) > 1 \end{cases} \quad (7)$$

where $LR(s) = p(s|H_m)/p(s|H_{nm})$ is the likelihood ratio at a specific s, and $p(s|H_m)$ and $p(s|H_{nm})$ are the dissimilarity score distribution when the templates belong to the mated (same) and nonmated (different) users. The templates are considered to be unlinkable at this specific score if $D_{\leftrightarrow}(s) = 0$. The overall linkability can be calculated by $D_{\leftrightarrow}^{sys} = \sum_{s_{min}}^{s_{max}} D_{\leftrightarrow}(s)p(s|H_m)$. To estimate $p(s|H_m)$ and $p(s|H_{nm})$, 100 instances of high-level features are extracted from the PCR digits by the CNN-models (with different configurations to simulate different applications) for each user to serve as their biometric templates. The dissimilarity scores for every possible pair of templates from the same user are then calculated. The process is repeated for all 46 users, resulting in 227700 scores for calculating $p(s|H_m)$. Similarly, the dissimilarity scores for every possible pair of templates from different users are calculated for estimating $p(s|H_{nm})$.

As we can observe in Fig. 18(a), the distribution $p(s|H_m)$ and $p(s|H_{nm})$ are similar, which results in a linkability of 0.0266. The results indicate that the similarity of any two templates is low when they belong to either the same user or different users. Therefore, it is difficult to tell if two templates are from the same user or not.



(a) Linkability (b) Irreversibility

Fig. 18. Security analysis of the proposed biometrics.

B. Irreversibility

Irreversibility requires the inability of inferring the biometric sample using its constructed templates. We follow the method in [32] to evaluate the irreversibility of our proposed biometrics. One approach to reconstruct the data samples of the user from its templates is to perform the pseudoinverse of the projection matrix used during template construction. Pearson's correlation coefficient between the original data samples and the attempted reconstructions is calculated to evaluate the accuracy of the reconstruction. The process was repeated for all 46 users, 100 templates each, so that the reconstructed mated distribution $p(\rho|H_{rm})$ could be evaluated using these 4600 correlation coefficients. The mated and nonmated distributions are also estimated.

Fig. 18(b) shows that even under the mated condition, correlation coefficients between the original and reconstructed data samples are still nearly uniformly distributed. The reconstructed mated distribution is more similar to the nonmated distribution and differs from the actual mated distribution. The result indicates that the reconstructed data samples of a given user are more likely to be identified as the data samples from another user.

TABLE II
PERFORMANCE UNDER DAILY NOISES AND DEDICATED ULTRASONIC INTERFERENCE

Noise Type (Full-band SPL)	Office (40 dB)	Parking Lot (55 dB)	AC (60 dB)	Conversation (65 dB)	In Car (70 dB)	Train Station (75 dB)	Dedicated Ultrasonic Noise		
Ultrasound SPL	10 dB	15 dB	17 dB	22 dB	25 dB	29 dB	30 dB	40 dB	50 dB
Ch.0	0.920	0.918	0.918	0.916	0.915	0.913	0.912	0.892	0.815
Ch.1	0.962	0.958	0.957	0.948	0.945	0.944	0.942	0.931	0.847
Ch.2	0.944	0.938	0.935	0.930	0.928	0.926	0.923	0.915	0.829
Ch.3	0.931	0.925	0.924	0.921	0.920	0.916	0.913	0.899	0.815
Ch.4	0.958	0.949	0.947	0.944	0.942	0.937	0.935	0.921	0.840
Ch.5	0.929	0.927	0.926	0.921	0.918	0.915	0.912	0.901	0.817
Ch.6	0.973	0.970	0.967	0.962	0.959	0.954	0.950	0.942	0.865
Ch.7	0.932	0.925	0.924	0.922	0.919	0.915	0.914	0.903	0.831
Ch.8	0.961	0.954	0.954	0.948	0.947	0.941	0.938	0.929	0.848
Ch.9	0.957	0.949	0.947	0.939	0.938	0.934	0.931	0.918	0.827
Average	0.947	0.941	0.940	0.935	0.933	0.930	0.927	0.915	0.833
4-digit PCR Code	0.979	0.970	0.969	0.964	0.959	0.956	0.954	0.940	0.865
6-digit PCR Code	1	1	1	1	0.993	0.989	0.987	0.977	0.918

VII. ADDRESSING DAILY NOISES AND DOS ATTACKS

Since our system uses acoustic signals to sense the user's gripping hand, it may be interfered by high-level ambient noises and dedicated ultrasonic interference. More particularly, when using learning-based acoustic systems in practical scenarios, the testing data showing different noise levels from the training data may severely degrade the system performance, because the system has not seen the sensing signals under such noises before. Such noise discrepancy is a critical challenge to prevent many acoustic sensing systems from being deployed practically. This section first illustrates this noise-incurred issue and demonstrates how we address this issue using longer PCR codes. We further propose a latent diffusion model for training data augmentation to improve the system's performance. To study the system performance under noise, we use eight different types of noise including the natural ambient noises and the dedicated ultrasonic interference, and play these audios using a loudspeaker. We then collect 10 users' data on the S8 phone. For each noise level, 20 instances are collected in the first session and 40 instances are collected in the second session. Testing only uses the second session data.

A. When Testing Data Has Different Noise Levels

Typically, a user enrolls the authentication system in relatively quiet indoor environments, such as office or home, with a noise level of around 40 dB. We thus use the user data collected under 40 dB noise for training (only the first session data). Because authentication may happen under any noise level, we use the second session user data collected under different noise levels for testing.

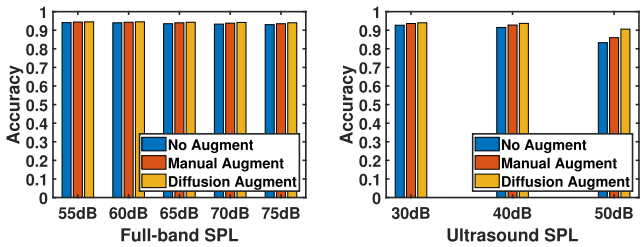
1) Impact of Daily Noises: The daily ambient noises we test include an open area scenario at a large parking lot (55 dB), a working Air Conditioner (AC) (60 dB), regular conversations (65 dB), in-car scenarios (70 dB) and a train station (75 dB). We use the Ultrasound Detector App [33] to measure the SPL at the ultrasound band 17 k–22 kHz. Table II shows the verification accuracy of using each single channel and the 4- and 6-digit PCR codes. We find that the 6-digit PCR code is slightly impacted by the daily noises for the ten participants, which has a strong capability to correct the errors caused by noises. The accuracy of each single channel and that of the 4-digit PCR code decrease under higher SPL noises. In particular, the average accuracy of all channels is 94.7%, 94.1%, 94%, 93.5%, 93.3% and 93% under the noise levels 40 dB, 55 dB, 60 dB, 65 dB, 70 dB and 75 dB. The reason is that the daily noises have limited capabilities to corrupt the coding chirps in the ultrasonic frequencies, and the structure-borne sounds are much stronger than the external ambient noises.

2) Under Ultrasounds and DoS Attacks: An adversary may use dedicated ultrasonic speakers to generate stronger interference signals to cause authentication errors or DoS. Thus, we need to know the extent of PCR-Auth to work under dedicated ultrasonic interference and exploit defense mechanisms to address this attacking scenario immediately when the ultrasonic attack exceeds a boundary. In particular, we use an external loudspeaker to continuously generate the white Gaussian noise at the frequencies from 17 kHz to 22 kHz. Table II presents the performance of PCR-Auth under three ultrasonic SPLs (17 k–22 kHz). When the SPL of the ultrasonic noise increases to 30 dB, 40 dB and 50 dB, the 4-digit PCR code's accuracy drops to 95.4%, 94%, and 86.5%, and the 6-digit PCR code's accuracy decreases to 98.7%, 97.7%, and 91.8%. The results indicate the potential of PCR-Auth to work with relatively good performance under dedicated ultrasonic interference. We choose 50dB SPL at the ultrasonic band as the threshold to detect DoS attacks before running the authentication, which is equivalent to 30 cm distance of ultrasound transmission if using normal mobile devices.

B. Addressing Train-Test Noise Discrepancy Using Diffusion

To address the impact of noise discrepancy in training and testing data for further improving system performance under practical noises, we propose to augment the training data set to cover the many testing scenarios. It is intuitive to apply manual data augmentation, which relies on human efforts to manually collect training data under different noises. We also develop a generative AI method to synthesize the authentication audios that could be acquired in practical noisy scenarios.

1) Manual Data Augmentation: We collect the user data under different levels of noise to augment the training data set. Specifically, the first-session user data under five types of daily noises and three levels of ultrasonic interference are used for training. The updated model is then tested on the second-session data under these noises. Fig. 19(a) shows that with the manual augmented training data, the system performance under daily noises is slightly improved, achieving an average accuracy of 94.4%, 94.3%, 94%, 93.8%, and 93.5% under daily noises of 55 dB, 60 dB, 65 dB, 70 dB, and 75 dB full-band SPL. This is because the system's performance under daily noises is already



(a) Under daily noises (b) Under dedicated ultrasonic noise

Fig. 19. Improving under-noise performance via augmentation.

good enough without applying the data augmentation, as there are few ultrasonic components in the daily noises that can impact the played ultrasonic chirp signals. Differently, as shown in Fig. 19(b), the system performance under dedicated ultrasonic noise is largely improved, achieving an average accuracy of 93.6%, 92.8%, and 86% under dedicated ultrasonic noise of 30dB, 40dB, and 50dB ultrasound SPL, respectively. Though promising, the manual data augmentation requires high user efforts to collect training data under different noise levels, which prevents the system to be practically deployed.

The above manual data augmentation collects the user data under different levels of noise and use them to update the trained model, which requires huge user efforts and is thus hard to be implemented. Instead, owing to the advancement of generative AI in recent years, we propose to leverage

2) Diffusion Model-Based Data Augmentation: We take a step further to address the train-test noise discrepancy challenge in acoustic sensing by using the state-of-the-art generative AI for training data augmentation. In particular, we propose a latent diffusion-based audio generation model to synthesize the user's training data under different levels of noise and then use the synthetic data for model updating. More specifically, we develop the data augmentation method based on AudioLDM [34], which is a text-to-audio system built on the latent diffusion model and trained using the contrastive language-audio pretraining (CLAP) [35] embeddings. The original model is capable of generating audio for a given text prompt and transferring the style of a given audio. To use AudioLDM to synthesize the user data under different levels of noise, we first download different types of daily noises (with captions), including the above-mentioned five types of noise (e.g., from YouTube), and also generate the dedicated ultrasonic noise spanning from 17 kHz to 22 kHz with different ultrasound SPLs (30 dB, 40 dB, 50 dB, etc.). The downloaded daily noise and the generated ultrasonic noise are then used to fine-tune the pre-trained AudioLDM model. In particular, the downloaded and generated noise audios last around 40 minutes in total, which are used to fine-tune the AudioLDM model for 200 epochs, taking around 1.5 hours on four A100 GPUs. Then, given the first-session user data with 40 dB noise (e.g., office), we use the fine-tuned AudioLDM model to transfer the real user data to generate synthetic user data under daily noises with 55 dB, 60 dB, 65 dB, 70 dB, and 75 dB full-band SPL and synthetic user data under dedicated ultrasonic noises with 30dB, 40dB, and 50dB ultrasound SPL. This is achieved by tuning the style transfer parameters of the AudioLDM model. The synthesized user data with different noise levels are then used as the augmented training data for our CNN-based system to recognize the user's hand in the unseen noisy environments.

The updated CNN model is then tested on the real user data collected under different levels of noise. Fig. 19(a) shows that with the diffusion model augmented training data, the system performance under daily noises is also slightly improved, achieving an average accuracy of 94.5%, 94.5%, 94.3%, 94.2%, and 94.0% under daily noises of 55 dB, 60 dB, 65 dB, 70 dB, and 75 dB full-band SPL. As shown in Fig. 19(b), the system performance under dedicated ultrasonic noise is largely improved, achieving an average accuracy of 94.0%, 93.7%, and 90.6% under dedicated ultrasonic noise of 30 dB, 40 dB, and 50 dB ultrasound SPL, respectively. The result indicates that our diffusion-based data augmentation solves the noise profile inconsistency challenge to improve acoustic sensing-based authentication without requiring additional user efforts.

VIII. Transformer-Based PCR Code Verification

Besides training a separate CNN model for each PCR coding channel for each user, we also use the transformer model as an alternative method to verify PCRs, where only a single model needs to be trained for each user to cover all coding channels. Transformers were initially developed for sequence transduction tasks [36], and have been applied for processing time series [37] and images [38]. Compared to CNN and RNN, transformer outperforms by processing all data samples in the input sequence in parallel and enabling each data sample to attend to all other data samples [39], [40], which enables learning the hand biometric information with both spatial and temporal characteristics. The transformer processes the whole time-frequency image at once, where positional encodings are added to make the model aware of the data sample's (time) index in the time-frequency image.

A. Model Structure

We develop the transformer-based PCR digit verification model based on the encoder structure of the Vanilla Transformer [36], [41]. The time-frequency image(s) $F \in \mathbb{R}^{\frac{T}{\tau} \times f}$ derived from the segmented PCR digit $x \in \mathbb{R}^T$ with a window length of τ is taken as the input. Linear transformation is first applied to transform the input into a vector $V \in \mathbb{R}^{\frac{T}{\tau} \times d}$ of the model dimension d. Positional encoding then adds the input data samples index information to the vector. The vector with positional information is then fed into the transformer encoder, which consists of M identical structures, where for each structure, a multi-head attention mechanism derives the input vector's self-attentions at each data sample. The transformer encoder learns hand biometric information from the time-frequency images and outputs the high-level representation of the gripping hand's characteristics $L \in \mathbb{R}^{\frac{T}{\tau} \times d}$, which describes both the spatial and the temporal location information carried by each time-frequency image. The high-level representation is finally passed through a linear transformation and a softmax function to generate the estimated user scores (probabilities).

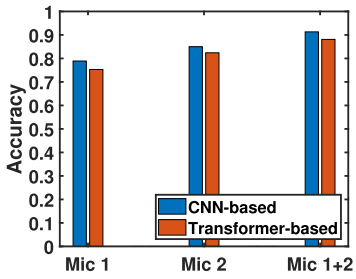


Fig. 20. Performance using different verification algorithms.

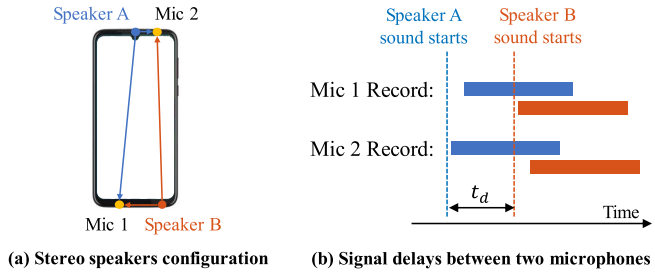


Fig. 21. Illustration of the spatial encoding mechanism.

1) Transformer Encoder & Self-Attention: The transformer encoder is formed by M identical layers. Within each layer, there is a multi-head attention sub-layer followed by a position-wise feed-forward sub-layer. A residual connection and a layer normalization are applied to each of the two sub-layers. The output of each sub-layer has the same dimension d, which facilitates the add and norm computations.

When the model is processing the frequential features at one time point, self-attention allows it to attend to all other time points in the spectrogram enabling it to learn the relationships between the frequential features at different time points. To calculate self-attentions of the spectrogram, we first create three vectors, the Query vector, the Key vector, and the Value vector, from the transformer encoder's input feature vector. The three vectors are created by multiplying the input feature vector by three matrices that are learnable during training. Then, a score is calculated by taking the dot product of the Query vector and the Key vector. For the features at time point t, its attention scores against the features at all time points in the input spectrogram are calculated by the dot product of the Query vector for time point t (q_t) and the Key vectors for all time points ($k_1, k_2, \ldots$). The scores are then passed through a softmax function to make itself add up to 1. The softmax score determines how much of each time point's feature information could be expressed by that of the currently examined time point. The value vector at each time point is then multiplied by the softmax score, and summed up to generate the self-attention for the input spectrogram at the current time point.

B. User Verification Performance

We next compare the system's user verification performance when using the CNN-based PCR digit verification algorithm and the transformer-based algorithm. As we can observe in Fig. 20, the CNN-based PCR digit verification algorithm achieves an user verification accuracy of 78.9%, 85%, and 90.9% when using one PCR digit recorded by Mic 1, Mic2, and Mic 1+2, respectively. The transformer-based algorithm performs slightly worse, achieving an accuracy of 75.3%, 82.4%, and 88.1%, when using one PCR digit recorded by Mic 1, Mic2, and Mic 1+2. This is because only a single transformer model is trained for each user to cover all the coding channels. The result shows the transformer-based algorithm achieves similar performance as the CNN-based algorithm for PCR digit verification and it is not limited by the number of coding channels or coding digits.

IX. Spatial Biometric Encoding Design

While the above focuses on encoding biometrics onto different frequencies to be compatible with challenge-response protocols, this section introduces a spatial biometric encoding mechanism. We explore encoding biometrics spatially during the data acquirement by leveraging smartphones' stereo speakers (i.e., one earphone speaker and one loudspeaker). This is the phones' inherent feature to defend against replay attacks. In particular, for each PCR digit, the stereo speakers play a chirp signal with a certain delay between the two acoustic channels. The two signals travel along different paths to arrive at the phone's mics. The geometric relationships established by the two onboard speakers and two mics can be derived based on the received sounds, which are hard to be reproduced by external speakers.

A. Spatial Encoding Mechanism

Fig. 21(a) illustrates the spatial encoding process at a smartphone with two speakers (e.g., speaker "A" and speaker "B"). During the encoding process of PCR-Auth, for each PCR digit selected by the OTC code, we use both speakers to play the corresponding chirp signal of that PCR channel, with speaker "B" being delayed for time t_d, where t_d is set to be smaller than the chirp signal's duration, enabling an overlap between the signals played by the two speakers. By doing so, each recorded PCR digit is no longer a single chirp signal played by one speaker but a combination of two chirp signals played by the stereo speakers. Moreover, since the two speakers are embedded on different locations of the smartphone, the propagation paths of the signals played by the two speakers to the smartphone microphones are thus different, which results in different signal delays at the two microphones as well as different SNR relationships and patterns of the recorded signals. The authentication system thus leverages such spatial information to prevent replay attacks. The system response under the spatial encoding can be expressed as (8), where S_A and S_B are the original sounds played by speaker "A" and speaker "B", H_A and H_B are the channel responses between the microphone and the two speakers, and t_d is the time delay experienced by speaker "B" with speaker "A" as the reference.

$$\hat{S}(f) = H_A(f)S_A(f) + H_B(f)S_B(f)e^{j2\pi f t_d} \tag{8}$$

1) Signal Delays for Validation: As we can observe in Fig. 21(b), when speaker "A" starts to play, since it is located closer to Mic2, the signal will first arrive at Mic2 and then Mic1.

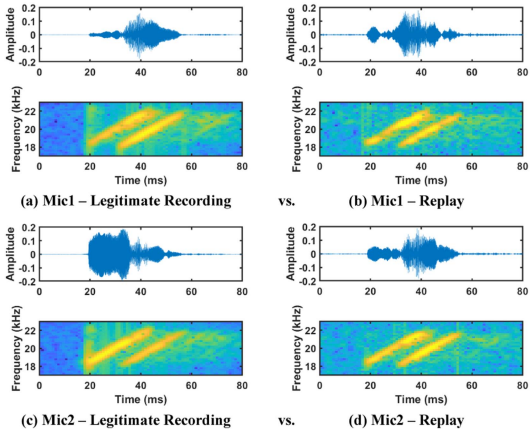


Fig. 22. Legitimate spatial encoding signal and its replay.

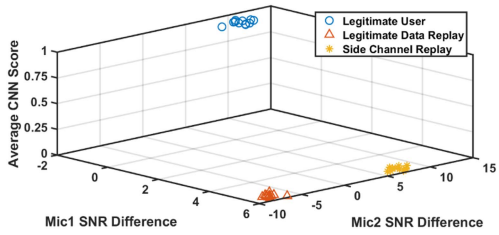


Fig. 23. Clusters of the legitimate user and replays.

TABLE III
SPATIAL ENCODING TO COUNTERACT REPLAY ATTACKS

Replay Types	FAR	
	Without Spatial Encoding	With Spatial Encoding
Side Channel Replay	0.063	0
Legitimate Data Replay	0.087	0.012

On the contrary, the signal played by speaker "B" will first arrive at Mic1 and then Mic2, as speaker "B" is located closer to Mic1. As a result, the delay between the two signals recorded at the two microphones are different, with the signal recorded at Mic2 turning out to have a longer duration.

2) SNR Relationships and Signal Patterns: In addition to different signal delays, the distance difference between the speakers and microphones also results in special SNR relationships and patterns of the signals recorded at the two microphones. As shown in Fig. 22, we play a chirp signal using the smartphone's stereo speakers at a 15 ms delay. In the signal recorded by Mic1, the former chirp (played by speaker "A") turns out to be weaker than the later chirp (played by speaker "B"), as illustrated by the spectrogram in Fig. 22(a). This also results in a lower SNR for the left half of the recorded signal and a higher SNR for the right half of the recorded signal, as shown by the waveform in Fig. 22(a). The reason is that Mic1 is located closer to speaker "B" than speaker "A". On the contrary, the spectrogram in Fig. 22(c) shows that in the signal recorded by Mic2, the former chirp (played by speaker "A") is stronger than the later chirp (played by speaker "B"). This also results in a higher SNR for the left half of the recorded signal and a lower SNR for the right half of the recorded signal, as shown by the waveform in Fig. 22(c). The reason is that Mic2 is located closer to speaker "A" than speaker "B".

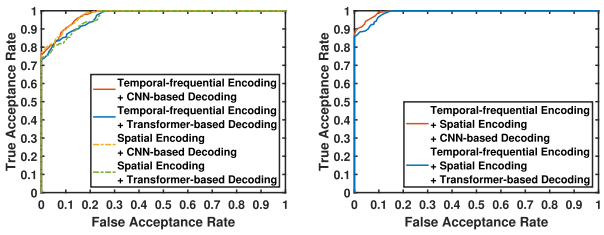
Such special signal patterns and SNR relationships can thus be leveraged to better defend against the replay attacks. As we can observe in Fig. 22(b) and (d), both the chirp patterns and the SNR relationships of the replayed spatial encoding signals are distinctive from those of the legitimate recordings. We further plot the legitimate recordings and the replayed signals into a 3D cluster, where the three axes represent the average CNN score of Mic1 and Mic2, Mic1's SNR difference between the signal's front half and later half, and Mic2's SNR difference between the signal's front half and later half, respectively. As we can observe in Fig. 23, the cluster of the legitimate user is clearly separated from those of the side channel replay and the legitimate data replay.

B. Defending Against Replay Attacks

To quantitatively evaluate the spatial encoding mechanism's ability in helping the user authentication system defend against the replay attacks, we recruit 10 participants and use a Moto G8 (which is equipped with stereo speakers) for data collection. The data collection follows the strategy in Section IV. We then compare the performance of the proposed user authentication system with and without the spatial encoding mechanism in defending against the replay attacks. As we can observe in Table III, without spatial encoding, our system suffers from 6.3% and 8.7% FARs under the side channel replays and the legitimate data replays, respectively, when using 1 PCR digit. However, with the spatial encoding mechanism, the FAR for the side channel replays is decreased to 0% and the FAR for the legitimate data replay drops to 1.2%. This is because it is hard for the adversary to forge the spatial information, which includes the different signal delays and the special SNR relationships and patterns of the signals played by the stereo speakers and recorded by the two microphones. The results confirm the spatial encoding mechanism's efficiency in helping defend against the replay attacks. We can further leverage the delay between the two speakers to design a coding system, where different lengths of delay can be used to represent different coding channels. Moreover, the spatial encoding can be integrated with the temporal-frequential encoding to provide an enhanced security.

C. Combination of Different Encoding & Decoding Methods

Since we have two biometric encoding mechanisms (i.e., temporal-frequential and spatial) and two biometric decoding algorithms (i.e., CNN-based and transformer-based), we now compare our system's user verification performance when using different combinations of these encoding & decoding methods. As we can observe in Fig. 24(a), when verifying 1-digit PCR codes, if we use the temporal-frequential encoding mechanism, using the CNN-based decoding algorithm achieves 90.6% TAR and 10.1% FAR, while using the transformer-based decoding algorithm achieves 87.2% TAR and 10.4% FAR. If we use the spatial encoding mechanism, using the CNN-based decoding algorithm achieves 89.3% TAR and 8.8% FAR, while using the transformer-based decoding algorithm achieves 85.3% TAR and 10.5% FAR. The results show that all these encoding & decoding methods work well in verifying the user. Fig. 24(b) further presents the performance of verifying 4-digit PCR codes, when the temporal-frequential encoding mechanism is integrated with



(a) Verifying 1-digit PCR codes (b) Verifying 4-digit PCR codes

Fig. 24. Combination of encoding & decoding methods.

the spatial encoding mechanism. In that case, using the CNN-based decoding algorithm achieves 94.2% TAR and 4.6% FAR, while using the transformer-based decoding algorithm achieves 92.4% TAR and 6.9% FAR.

D. Spatial Encoding Under Noise

Similar to the temporal-frequential encoding, we also test the performance the spatial encoding under different ambient noises. In particular, we recruited 10 participants to collect the under-noise data with an S8 phone and test the same noise levels regarding both the daily noises and dedicated ultrasonic noises as specified in Section VII, we collected 40 instances for each noise level to test the CNN model, which is trained with only the office environment data (40 dB noise). We find that when using 1-digit PCR codes, our system achieves an average accuracy of 94.4%, 93.9%, 93.9%, 93.3%, 93.0%, and 92.8%, under the daily noise level of 40 dB, 55 dB, 60 dB, 65 dB, 70 dB, and 75 dB, respectively. The result shows that daily noises have no obvious or limited impact on single PCR digits. Longer PCR codes make such impacts hardly observed. In comparison, dedicated ultrasonic attacks have a greater impact. Specifically, under the ultrasonic noises with 30 dB, 40 dB, and 50 dB ultrasound SPL, our system achieves an average accuracy of 94.7%, 93.5%, and 85.9% by using 4-digit PCR codes. To further address the ultrasonic noise impact, we can leverage the diffusion-based augmentation method similar to that in Section VII or use longer PCR codes.

X. Related Work

Biometrics utilized for mobile devices can be classified into two categories. Physiological biometrics are extracted from static body traits, such as face, fingerprint and iris. Behavioral biometrics are a relatively new type of biometrics, which refer to the inherent dynamic behavioral patterns of human motions, such as gaits [43], voices [44], keystroke dynamics [45], and finger gestures [46]. However, due to the advanced mobile recording techniques (e.g., visual and acoustic), 3D printing and robotics, the physiological and behavioral biometrics are both under a high risk to be obtained by an adversary [3], [47], [48]. Furthermore, the biometrics' static nature makes them easy to be reused by an adversary for replay attacks.

To improve biometric security, some studies focus on multi-factor authentication, which combines multiple biometric and knowledge factors to achieve enhanced security. For example, the user's face, teeth and voice can be verified visually and acoustically for a fused decision [49], [50], [51], [52]. Safe et al. propose to display a secret icon on the screen during the face recognition and verify the eye gaze direction as a second factor [53]. Ometov et al. propose to combine the user's biometrics such as voice and face with a PIN entry for authentication [54]. But adding additional factors requires multiple entries from the user, which scarifies the usability. Some more advanced multi-factor authentication methods focus on integrating knowledge secrets and biometrics in one input, such as by extracting keystroke dynamics from a password entry [55], [56], capturing finger gesture behaviors from a signature [57], [58] or obtaining vibration signatures from the user's secret input on a solid surface [42]. But all these methods still reuse the same biometric data for every authentication session, which is vulnerable to replay attacks.

Challenge-response protocols are designed to prevent replay attacks [59]. The initial success of using the handshake protocol to verify humans is based on behavioral biometrics. When the user responds to a challenge (e.g, a task or a game), the inherent motion behaviors are verified. For example, Mohamed et al. [11] design a game challenge for users to select from a number of icons the preset secret ones. Both the selected icons and the drag-and-drop behaviors are verified as a response. Sluganovic et al. [12] propose to randomly show a dot on the screen as the challenge and capture the user's reflexive eye movements as the response. However, these methods require cognitive and behavioral activities from the user during authentication, which is intrusive and demands a long response time. Moreover, the great variability caused by behavioral inconsistency leads to high false rejection rates. The recent work Velody [17] utilizes a vibration motor and receiver to collect a large number of vibration responses from the user for authentication, and every used response is disposed of. But this method requires additional hardware and is thus hard to deploy on most handheld devices. Moreover, the system demands high efforts to train and refill a biometric pool periodically to support daily usage. Differently, PCR-Auth unobtrusively verifies the user's PCR with most handheld devices. It creates a huge biometric response universe at a minimum overhead and saves the trouble of biometric pool maintenance. The comparison with related work is presented in Table IV. To summarize, PCR-Auth outperforms the existing works in both verifying the users and defending against different types of attacks. In terms of user experience, PCR-Auth requires lower user participation as compared to the existing works and is thus more convenient to use. Additionally, PCR-Auth has a huge response pool, which supports daily authentication needs with enhanced security.

XI. Limitations

We now discuss the limitations of the proposed system: 1) While we demonstrate that biometric features can be encoded temporal-frequentially and spatially, the encoding algorithms are still basic. Further efforts are needed to integrate more advanced coding algorithms with biometric verification. 2) The proposed system is based on ultrasonic sound sensing, the dedicated ultrasonic interference or attack may degrade the system performance. In this paper, we show that using longer PCR

TABLE IV
COMPARISON WITH RELATED STUDIES

Work	Protocol	Modality	FNR	FPR			User	Dedicated	Response
				Impersonation	Replay	Synthesis	Participation	Hardware	Pool
LivDet [6]	Physiological	FingerPrint	11.96%	1.07%	N.A.	N.A.	Low	Yes	N.A.
Erdogmus et al. [3]		FaceID	5.5%	1.1%	N.A.	N.A.	Medium	Yes	N.A.
Menotti et al. [5]		Iris	0.16%	0.16%	N.A.	N.A.	Medium	Yes	N.A.
BiLock	Behavioral	Tooth click sound	5%	1.5%	5.6%	N.A.	High	Yes	N.A.
BreathPrint		Breathing gesture-induced sound	6%	2%	2%	N.A.	High	Yes	N.A.
Taprint		Tapping-induced vibration	1.74%	1.74%	N.A.	N.A.	High	Yes	N.A.
VibWrite [42]		Vibration response of dynamic gestures	10%	2%	N.A.	N.A.	High	Yes	N.A.
Eye Movement [12]	Challenge-response	Reflective eye movement	6.3%	6.3%	0.06%	N.A.	Medium	Yes	N.A.
Velody [17]		Vibration response	5.8%	5.8%	0%	0%	Medium	Yes	n
PCR-Auth		Vibration response of palm contact	0.4%	3.5%	0%	0%	Low	No	n^m

codes and diffusion models to denoise can effectively address the dedicated ultrasonic noises below 50 dB. Addressing the higher-power ultrasonic interference requires further research. 3) Our authentication method is device-dependent since the phone sizes, dimensions and the locations of the speakers and microphones dominate the signal patterns in addition to the user's hand. If the user changes a phone, the authentication model needs to be retrained. Our future work will focus on developing transfer learning and domain adaption algorithms to address this limitation, and allow few-shot learning to update the model for different phones.

XII. CONCLUSION

In this work, we propose a challenge-response user authentication system, PCR-Auth, based on the novel palm contact response. It is associated with the user's gripping hand biometric and can be extracted by narrow-band ultrasonic pulses unobtrusively, when the user holds a handheld device. The proposed system is designed to verify the user by examining both the biometric features and the coding sequence. In particular, we devise two biometric encoding techniques (i.e., temporal-frequency biometric encoding and spatial biometric encoding), to encode the biometric features into a biometric code, which responds to a given session challenge. The biometric encoding generates a large biometric response universe to support disposable biometric responses and prevent replays. We then develop two alternative decoding algorithms to decode the biometric code for user authentication. In addition, we investigate various attacks, including acoustic replays and 3D-printing attacks that can physically reproduce the user's gripping hand. We further exploit a latent diffusion model to address the noise discrepancy between training and testing data, which improves our system performance without requiring additional training data collection. Extensive experiments show a 6-digit PCR achieves 97% accuracy to distinguish users and rejects replay attacks with 100% accuracy.

ACKNOWLEDGMENT

Preliminary results of this paper have been presented in part in IEEE S&P 2022 [60].

REFERENCES

[1] EyeVerify, "EyeVerify survey reveals high consumer trust in biometrics for mobile banking and payments," 2017. [Online]. Available: https://www.globenewswire.com/news-release/2017/05/04/1078295/0/en/EyeVerify-Survey-Reveals-High-Consumer-Trust-in-Biometrics-for-Mobile-Banking-and-Payments.html

[2] M. Mohamed, B. Shrestha, and N. Saxena, "SMASheD: Sniffing and manipulating android sensor data for offensive purposes," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 4, pp. 901–913, Apr. 2017.

[3] N. Erdogmus and S. Marcel, "Spoofing face recognition with 3D masks," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 7, pp. 1084–1097, Jul. 2014.

[4] D. F. Smith, A. Wiliem, and B. C. Lovell, "Face recognition on consumer devices: Reflections on replay attacks," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 4, pp. 736–745, Apr. 2015.

[5] D. Menotti et al., "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 4, pp. 864–879, Apr. 2015.

[6] L. Ghiani et al., "LivDet 2013 fingerprint liveness detection competition 2013," in *Proc. 2013 Int. Conf. Biometrics*. IEEE, 2013, pp. 1–6.

[7] J. Shelton, K. Roy, B. O'Connor, and G. V. Dozier, "Mitigating iris-based replay attacks," *Int. J. Mach. Learn. Comput.*, vol. 4, no. 3, 2014, Art. no. 204.

[8] K. W. Bowyer and J. S. Doyle, "Cosmetic contact lenses and iris recognition spoofing," *Computer*, vol. 47, no. 5, pp. 96–98, 2014.

[9] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Proc. Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2014, pp. 1–5.

[10] R. Font, J. M. Espín, and M. J. Cano, "Experimental analysis of features for replay attack detection-results on the ASVspoof 2017 challenge," in *Proc. Interspeech*, 2017, pp. 7–11.

[11] M. Mohamed, P. Shrestha, and N. Saxena, "Challenge-response behavioral mobile authentication: A comparative study of graphical patterns and cognitive games," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, 2019, pp. 355–365.

[12] I. Sluganovic, M. Roeschlin, K. B. Rasmussen, and I. Martinovic, "Using reflexive eye movements for fast challenge-response authentication," in *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1056–1067.

[13] Z. Chen and M. Recce, "Handgrip recognition," *J. Eng. Comput. Archit.*, vol. 1, no. 2, 2007. [Online]. Available: https://www.academia.edu/1149541/Handgrip_recognition?sm=b

[14] C. J. Migos and D. H. Sloo, "Personalization using a hand-pressure signature," US Patent 8,172,675, 2012.

[15] A. S. Weksler, N. J. Peterson, and R. S. VanBlon, "Grip signature authentication of user of device," US Patent App. 14/098,180, 2015.

[16] J. Liu, Y. Chen, M. Gruteser, and Y. Wang, "VibSense: Sensing touches on ubiquitous surfaces through vibration," in *Proc. 14th Annu. IEEE Int. Conf. Sens., Commun., Netw.*, 2017, pp. 1–9.

[17] J. Li, K. Fawaz, and Y. Kim, "Velody: Nonlinear vibration challenge-response for resilient user authentication," in *Proc. 2019 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 1201–1213.

[18] C. Wang, S. A. Anand, J. Liu, P. Walker, Y. Chen, and N. Saxena, "Defeating hidden audio channel attacks on voice assistants via audio-induced surface vibrations," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, 2019, pp. 42–56.

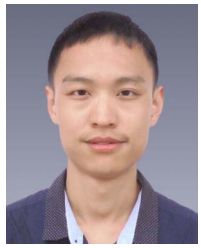
[19] FCC, "Code of federal regulations title 47: Part 18 industrial, scientific and medical equipment," 2020. [Online]. Available: https://www.ecfr.gov/cgi-bin/text-idx?SID=c58a65f109fafe947820c107581748fd$&$mc=true$&$node=pt47.1.18$&$rgn=div5

[20] K. Ashihara, "Hearing thresholds for pure tones above 16 KHz," *J. Acoust. Soc. Amer.*, vol. 122, no. 3, pp. EL52–EL57, 2007.

[21] B. Zhou, J. Lohokare, R. Gao, and F. Ye, "EchoPrint: Two-factor authentication using acoustics and vision on smartphones," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 321–336.

[22] D. Han, Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Proximity-proof: Secure and usable mobile two-factor authentication," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 401–415.

[23] K. Sun, T. Zhao, W. Wang, and L. Xie, "Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 591–605.
[24] Y.-C. Tung and K. G. Shin, "Expansion of human-phone interface by sensing structure-borne sound propagation," in *Proc. 14th Annu. Int. Conf. Mobile Syst., Appl., Serv.*, 2016, pp. 277–289.
[25] J. Jonsson and B. Kaliski, "Public-key cryptography standards (PKCS)# 1: RSA cryptography specifications version 2.1," Tech. Rep. RFC 3447, 2003.
[26] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2014, pp. 157–175.
[27] A. Shamir, "On the generation of cryptographically strong pseudorandom sequences," *ACM Trans. Comput. Syst.*, vol. 1, no. 1, pp. 38–44, 1983.
[28] Apple, "Use touch ID on iPhone and iPad," 2019. [Online]. Available: https://support.apple.com/en-us/HT201371
[29] M. Xu, J. Liu, Y. Liu, F. X. Lin, Y. Liu, and X. Liu, "A first look at deep learning apps on smartphones," in *Proc. World Wide Web Conf.*, 2019, pp. 2125–2136.
[30] "Flexible bendable mannequin hands for nails practice fake hand nail display tool," 2021. [Online]. Available: https://www.amazon.com/gp/product/B08VJ27BRN/ref=ox_sc_act_title_1?smid=A2CH949N64GN7G
[31] M. Gomez-Barrero, J. Galbally, C. Rathgeb, and C. Busch, "General framework to evaluate unlinkability in biometric template protection systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 6, pp. 1406–1420, Jun. 2018.
[32] M. Gomez-Barrero, C. Rathgeb, J. Galbally, C. Busch, and J. Fierrez, "Unlinkable and irreversible biometric template protection based on bloom filters," *Inf. Sci.*, vol. 370, pp. 18–32, 2016.
[33] S. Gudkov, "Ultrasound detector," 2018. [Online]. Available: https://play.google.com/store/apps/details?id=com.microcadsystems.serge.ultrasounddetector
[34] H. Liu et al., "Audioldm: Text-to-audio generation with latent diffusion models," 2023, *arXiv:2301.12503*.
[35] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.
[36] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
[37] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.
[38] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv: 2010.11929*.
[39] P. Delgado-Santos, R. Tolosana, R. Guest, F. Deravi, and R. Vera-Rodriguez, "Exploring transformers for behavioural biometrics: A case study in gait recognition," 2022, *arXiv:2206.01441*.
[40] G. Stragapede, P. Delgado-Santos, R. Tolosana, R. Vera-Rodriguez, R. Guest, and A. Morales, "Mobile keystroke biometrics using transformers," in *Proc. IEEE 17th Int. Conf. Autom. Face Gesture Recognit.*, 2023, pp. 1–6.
[41] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 2114–2124.
[42] J. Liu, C. Wang, Y. Chen, and N. Saxena, "Vibwrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration," in *Proc. 2017 ACM SIGSAC Conf. Comput. Commun. Secur.*, ACM, 2017, pp. 73–87.
[43] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1961–1974, Sep. 2015.
[44] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.
[45] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors," in in *Proc. IEEE 22nd Int. Conf. Netw. Protoc.*, 2014, pp. 221–232.
[46] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-rich gestures: A novel approach to authentication on multi-touch devices," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2012, pp. 977–986.
[47] A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni, "Fake finger detection by skin distortion analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 3, pp. 360–373, Sep. 2006.
[48] A. Serwadda and V. V. Phoha, "When kids' toys breach mobile phone security," in *Proc. 2013 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 599–610.
[49] D.-J. Kim, K.-W. Chung, and K.-S. Hong, "Person authentication using face, teeth and voice modalities for mobile device security," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2678–2685, Nov. 2010.
[50] C. McCool et al., "Bi-modal person recognition on a mobile phone: Using mobile phone data," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, IEEE, 2012, pp. 635–640.
[51] K. B. Raja, R. Raghavendra, M. Stokkenes, and C. Busch, "Multi-modal authentication system for smartphones using face, iris and periocular," in *Proc. 2015 Int. Conf. Biometrics*, IEEE, 2015, pp. 143–150.
[52] M. De Marsico, C. Galdi, M. Nappi, and D. Riccio, "Firme: Face and iris recognition for mobile engagement," *Image Vis. Comput.*, vol. 32, no. 12, pp. 1161–1172, 2014.
[53] A. Boehm et al., "Safe: Secure authentication with face and eyes," in *Proc. Int. Conf. Privacy Secur. Mobile Syst.*, 2013, pp. 1–8.
[54] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, no. 1, 2018, Art. no. 1.
[55] C. Giuffrida, K. Majdanik, M. Conti, and H. Bos, "I sensed it was you: Authenticating mobile users with sensor-enhanced keystroke dynamics," in *Proc. Int. Conf. Detection Intrusions Malware Vulnerability Assessment*, Springer, 2014, pp. 92–111.
[56] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and I know it's you! Implicit authentication based on touch screen patterns," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2012, pp. 987–996.
[57] N. Sae-Bae and N. Memon, "Online signature verification on mobile devices," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 6, pp. 933–947, Jun. 2014.
[58] Y. Ren, C. Wang, Y. Chen, M. C. Chuah, and J. Yang, "Critical segment based real-time e-signature for securing mobile transactions," in *Proc. 2015 IEEE Conf. Commun. Netw. Secur.*, IEEE, 2015, pp. 7–15.
[59] D. L. Davis and L. Smith, "Authentication system based on periodic challenge/response protocol," US Patent 6,088,450, 2000.
[60] L. Huang and C. Wang, "PCR-auth: Solving authentication puzzle challenge with encoded palm contact response," in *Proc. IEEE Symp. Secur. Privacy*, IEEE, 2022, pp. 1034–1048.



Long Huang (Student Member, IEEE) received the BEng degree in electrical and electronic engineering from the University of Electronic Science and Technology of China, in 2015 and the MSc degree in electrical and electronic engineering from the Stevens Institute of Technology, in 2019. He is currently working toward the PhD degree with Louisiana State University. His research interests include mobile computing, signal processing, and Internet of Things. He received the Best Paper Award from the 14th International Workshop on Wireless Sensor, Robot and UAV Networks, IEEE WISARN 2021.

Chen Wang (Member, IEEE) received the PhD degree from Rutgers University, in 2019. He is an associate professor of computer science with SMU Lyle and leads the Mobile and Internet SecuriTy (MIST) lab. His research interests include cyber security and privacy, sensing, mixed reality, robotics security and smart healthcare. He is the recipient of NSF CAREER award and has published a number of papers with high-impact conferences, including IEEE S&P, ACM CCS, ACM Mobicom and IEEE Infocom. He has received five Best Paper Awards from IEEE CNS 2018, IEEE CNS 2014, ASIACCS 2016, EAI HealthyIoT 2019 and IEEE INFOCOM WKSHPS 2021. From 2014 to 2023, his research studies have been reported by more than 170 media outlets, including IEEE Spectrum, NSF Science 360, CBS TV, BBC News, NBC, IEEE Engineering 360, Fortune, ABC News, MIT Technology Review, etc.