PCR-Auth: Solving Authentication Puzzle Challenge with Encoded Palm Contact Response

Long Huang, Chen Wang
Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA
Email: lhuan45@lsu.edu, chenwang1@lsu.edu

2022 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-1316-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SP46214.2022.9833564

Abstract—**Biometrics have been widely applied as personally identifiable data for user authentication. However, existing biometric authentications are vulnerable to biometric spoofing. One reason is that they are easily observable and vulnerable to physical forgeries. Examples are the apparent surface patterns of human bodies, such as fingerprints and faces. A more significant issue is that existing authentication methods are entirely built upon biometric features, which almost never change and could be obtained or learned by an adversary such as human voices. To address this inherent security issue of biometric authentications, we propose a novel acoustically extracted hand-grip biometric, which is associated with every user's hand geometry, body-fat ratio, and gripping strength; It is implicit and available whenever they grip a handheld device. Furthermore, we integrate a coding technique in the biometric acquisition process, which encodes static biometrics into dynamic biometric features to prevent data reuse. Additionally, this low-cost method can be deployed on any handheld device that has a speaker and a microphone.**

In particular, we develop a challenge-response biometric authentication system, which consists of a pair of biometric encoder and decoder. We encode the ultrasonic signal according to a challenge sequence and extract a distinct biometric code as the response for each session. We then decode the biometric code to verify the user by a convolutional neural network-based algorithm, which not only examines the coding correctness but also verifies the biometric features presented by each biometric digit. Furthermore, we investigate diverse acoustic attacks to our system, by respectively assuming an adversary could present the correct code, generate similar biometric features or successfully forge both. Extensive experiments on mobile devices show that our system achieves 97% accuracy to distinguish users and rejects 100% replay and synthesis attacks with 6-digit codes.

I. Introduction

Biometrics such as faces, fingerprints and irises are increasingly exploited to verify users because they are convenient to use [1]. A recent report estimates that over 1.5 billion people might use biometrics for authentication by 2023 [2]. However, biometric security is attracting increasing public concerns. Due to the increasingly advanced recording technologies, 3D printing, wireless eavesdropping and malware [3], 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 [4], [5], [6], fingerprint [6], [7], iris [8], [9] and voice [10], [11]. Addressing the replay issues has become a critical task for ensuring biometric security.

An active research direction for preventing relay attacks is liveness detection. These approaches require motions to prove live faces [12] and leverage heatmaps to detect live fingers. But these methods require the user's participation to prove "liveness" or are subject to additional sensor overheads. They still have not fundamentally solved the two replay threats. Behavioral characteristics (e.g., gaits) are a rapidly growing category of biometrics, which can not 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 [13], [14]. 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, the 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.

This work aims to develop a biometric-based CR authentication system for handheld devices, which not only solves the above replay threats but also requires low user efforts. The hand-grip biometric inherently comes with handheld devices, and acquiring it requires no more efforts than obtaining a fingerprint. This biometric was traditionally extracted by an array of pressure sensors that enclose the handheld device [15], [16], [17]. 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.

We devise a novel biometric encoding technique to integrate the hand-grip biometric with the CR protocol. Based on that we develop the PCR-Auth system, whose handshake process is shown in Figure 1. When a user requests an authentication,

DOI 10.1109/SP46214.2022.00053

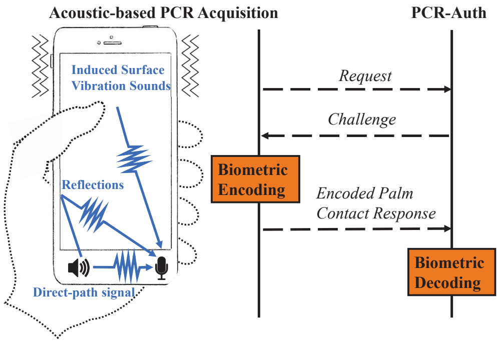

Acoustic-based PCR Acquisition
PCR-Auth
Induced Surface Vibration Sounds
Request
Challenge
Biometric Encoding
Reflections
Encoded Palm Contact Response
Biometric Decoding
Direct-path signal

Fig. 1: The handshake process of PCR-Auth.

PCR-Auth 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. The encoded PCR is then decoded, and 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 spectrogram to examine the user's hand-grip biometric features carried on each PCR digit. We develop a Convolutional Neural Network (CNN)-based algorithm to verify all PCR digits and leverage its multi-class classification capability to address human behavioral inconsistency. The CNN scores of each PCR digit are returned. We then apply a cluster-based method to integrate the CNN 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 authentications by integrating a coding technique with biometric acquisition. The authentication process requires neither active user participation nor additional hardware.
- **An 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.
- **Biometric Encoding:** We encode a user's biometric into biometric codes, which creates a huge response pool to support everyday CR authentications. Moreover, we develop a CNN-based method to decode the unique biometric code for each session, which not only verifies the biometric but also checks the code correctness.
- **New Attacks and Extensive Experiments:** While the CR authentication is designed to defeat replay attacks, we take one step further to investigate new attacks, assuming an adversary can repeat the code, replicate the biometric or forge both. The system is then evaluated on multiple devices under these attacks. Results show that our system verifies users with 97% accuracy and rejects up to 100% replay and synthesis attacks with 6-digit PCR codes.

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 [15], [16], [17]. 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 [18], [19], 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 Figure 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 [20]. 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 Equation 1,

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

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

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

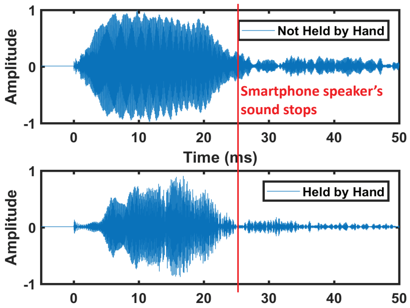


Fig. 2: The impact of hand-grip to the smartphone's sound.

where t and τ are the additional travel time of the reflected signal and the surface vibration sound, compared to the direct-path signal. Equation 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. This work is based on the assumption that people's gripping hands are distinctive. As a result, the combined signal at the microphone should present individually distinctive patterns. Moreover, the hand-grip biometric is implicit and hard to imitate. Even if an adversary perfectly replicates the hand geometry and the holding position, he/she could hardly repeat the body-fat ratio and gripping strength.

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

$$pcr = \langle H_d, H_r, H_v, F \rangle, \tag{3}$$

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

B. Motivational Study

We conduct a feasibility study to show how ultrasonic signals are impacted by the gripping hand. Specifically, we play a short 18k-22kHz chirp signal in 25 ms using a smartphone's speaker. Figure 2 shows the received signals at the smartphone's microphone, when it is held in the user's hand and placed on a table, respectively. From the comparison, we observe that the user's gripping hand suppresses the speaker's sound by an average of 3 dB. Moreover, when the chirp signal sweeps from 18k to 22kHz, the signal amplitudes are degraded by the gripping hand with different scales, which illustrates the PCR's frequency-selective nature. Furthermore, we find that while the direct-path signals dominate the microphone data before $25ms$, the speaker sound reflections and the induced surface vibration sounds become significant after the speaker's sound stops. They degrade over 20 dB after 75 ms. We thus propose to leverage the $0-75$ms sounds for the gripping-hand sensing. It is also worth noting that the sounds in the $25-75$ms range are the residue of the speaker sound, which are harder to forge via a speaker than direct-path signals.

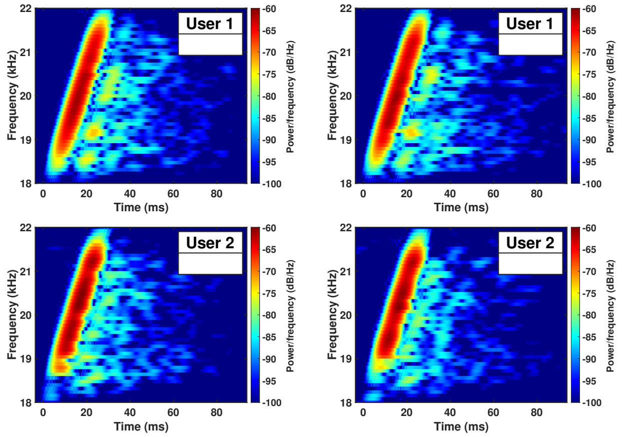


Fig. 3: Distinguishing users by palm contact responses.

We then study the feasibility of using PCRs to distinguish users. Figure 3 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 Figure 4, 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 device plays the OTC-encoded stimulus signal using its own speaker, and in the meanwhile, its microphones record the signals to obtain the encoded

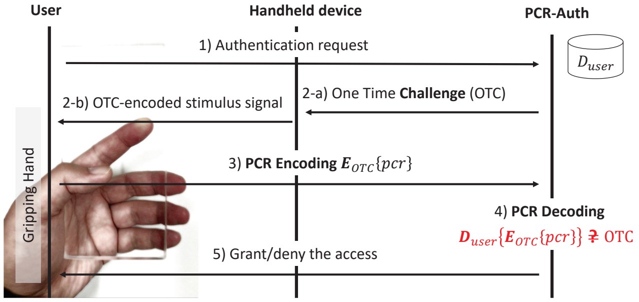

User
Handheld device
PCR-Auth
D_{user}
1) Authentication request
2-b) OTC-encoded stimulus signal
2-a) One Time Challenge (OTC)
3) PCR Encoding $E_{OTC}\{pcr\}$
4) PCR Decoding
$D_{user}\{E_{OTC}\{pcr\}\} \overset{?}{=}$ OTC
5) Grant/deny the access
Gripping Hand

Fig. 4: Our challenge-response authentication model.

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 Equation 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 biometric encoding, which serves as the basis of PCR-Auth and creates a huge response universe to support everyday CR authentications. 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

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

where pcr_i, $i = 1, 2, ...m$ is the i-th 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 [19]. As a result, the user does not need to refill the response pool periodically with new biometric features.

E. Framework Overview

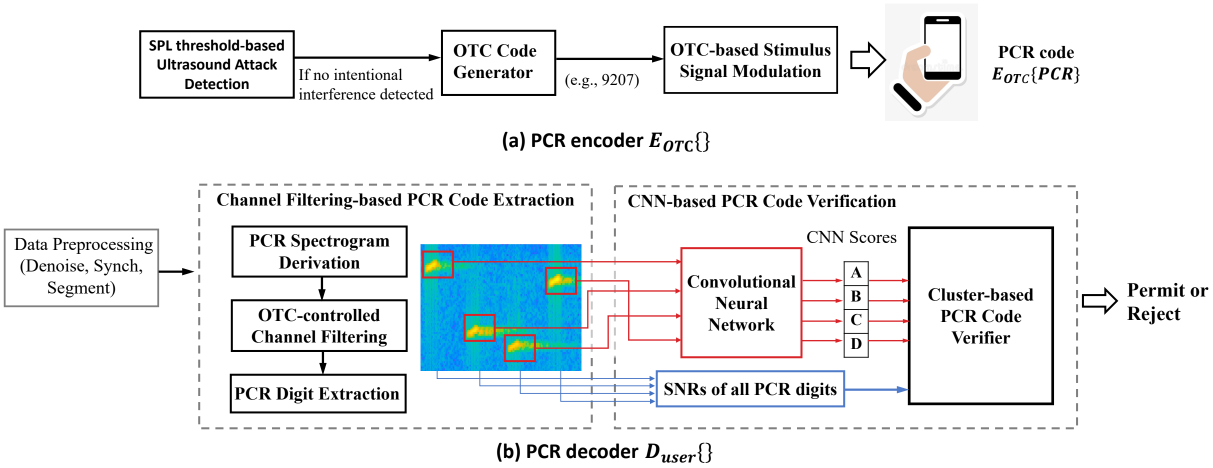
Based on the above CR model, we design the PCR-Auth framework as shown in Figure 5. Upon each authentication request, the microphone access permission is acquired, which is revoked after authentication through auto-reset. The PCR encoder first detects whether the device is under intentional ultrasonic interference by examining the ultrasonic band against a Sound Pressure Level (SPL) threshold, which is introduced in Section V-F2. If no dedicated ultrasound is detected, the PCR encoder generates an OTC, and the OTC-based Stimulus Signal Modulator selects ultrasonic pulses accordingly to encode the user hand-grip biometric into a PCR code.

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 spectrogram 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 CNN-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.

The CNN-based PCR Decoder exploits one CNN model with five convolutional layers to decode each PCR digit from its spectrogram. The CNN 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 CNN 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:


SPL threshold-based Ultrasound Attack Detection
If no intentional interference detected
OTC Code Generator
(e.g., 9207)
OTC-based Stimulus Signal Modulation
PCR code
$E_{OTC}\{PCR\}$
Data Preprocessing (Denoise, Synch, Segment)
Channel Filtering-based PCR Code Extraction
PCR Spectrogram Derivation
OTC-controlled Channel Filtering
PCR Digit Extraction
CNN-based PCR Code Verification
CNN Scores
Convolutional Neural Network
A
B
C
D
Cluster-based PCR Code Verifier
SNRs of all PCR digits
Permit or Reject

(a) PCR encoder $E_{OTC}\{\}$

(b) PCR decoder $D_{user}\{\}$

Fig. 5: The architecture of PCR-Auth.

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; We further consider a ***knowledgeable fake-hand attacker***, who uses a silicone fake hand to imitate the victim's hand with more freedom.

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) Listen-and-synthesis Attack: We design new attacks by assuming that the adversary could capture the OTC by listening to the acoustic channel and immediately stitching prior PCR digits in a correct sequence. This attack aims to forge both the code and the biometric. We also assume the adversary could obtain the victim's all PCR digits beforehand to attack with full freedom, which may be achieved through eavesdropping on the registration process or the disclosure of the stored biometric data. However, the implementation of this attack is still challenging. If playing the synthesized sound after waiting to eavesdrop on all OTC digits, it would be rejected because the microphone is off just after the speaker sound stops (e.g., with 80ms delay). So the only way is to listen and forge each PCR digit separately. In this scenario, each synthesis digit is still unavoidably delayed by at least $t_a$ms because the attacker needs this period for observation (buffer), A/D and D/A conversions, Fast Fourier Transform (FFT), cache accesses and the CPU scheduling time. The last item alone can be over 100ms and unpredictable. But the success of such attacks requires the short delay (less than one digit) and the strict synchronization. Otherwise, each synthesized digit may be corrupted by the speaker's next digit or partially segmented due to poor synchronization, resulting in a rejection.

To our knowledge, no prior work has implemented such attacks due to the strict real-time synchronization requirement. But we find that if using a Field Programmable Gate Array (FPGA) for the attack system implementation, the CPU scheduling time can be removed, and a determined delay might be achieved as low as 10ms for each synthesized digit. While implementing the FPGA-based attack system is beyond this paper's scope, we assume its feasibility and further investigate two new attacks based on simulation: ***synchronization disturbance attack*** overwhelms or replaces each original digit with louder delayed adversarial digits, which aims to fool the synchronization method into picking up the delayed adversarial digits for authentication; ***real-time perturbation attack*** attempts to generate perturbation noises based on adversarial learning and modify the original digit sounds to present correct biometric features.

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. Approach Design

A. Palm Contact Response Encoder

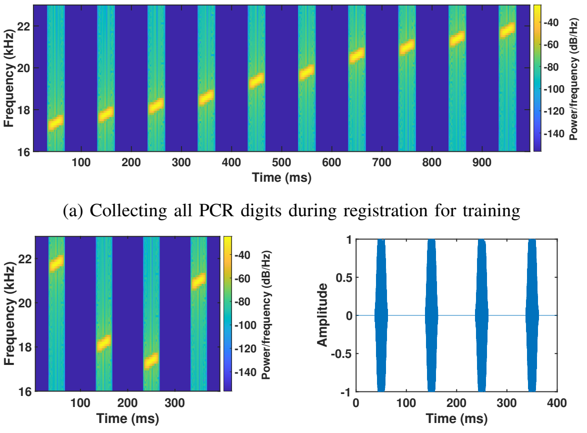
1) Stimulus Signal Design: The stimulus signal is used to interact with the user's palm and extract the PCR for authentication. In order to acoustically obtain sufficient biometric information, we exploit the upward frequency sweeping signals to capture the user's biometric in a frequency range rather 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 [21]. In particular, we apply the signals within the range 17k-22kHz, which has been demonstrated to be hardly audible [22] and widely applied in prior ultrasonic sensing work [23], [24], [25], [26]. 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 25ms long and 350Hz wide chirp pulses within the range $17k - 22k$Hz. The signal frequency bands are 10 times narrower than the prior acoustic sensing work (i.e., $4 - 6$kHz wide [23], [24], [25], [26]), 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 Figure 6.

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 [27], [28], [29]. For simplicity of description, we select 10 exclusive narrow-band channels from the range 17k-22kHz to represent decimal digits. These coding channels are all 350Hz wide and separated by a gap (e.g., 50Hz). It is important to note that not all channels are suitable for encoding, and we conduct extensive experiments to identify the good channels, which is introduced in Section V-A2.

Chirp pulse is used as the basic unit to encode the user's PCR onto the corresponding channel. When training the PCR decoder, the user's PCRs at all coding channels are collected as shown in Figure 6 (a). During the authentication, the PCR encoder scopes down to each coding channel indexed by the OTC to extract the corresponding PCR digit. Figure 6 (b) illustrates the stimulus signals for encoding a 4-digit PCR code, when the OTC code is "9207" and the selected chirp pulses are 21.6-21.95kHz, 18.8-19.15kHz, 21.6-21.95kHz and 20.8-21.15kHz 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. A longer code creates an exponentially greater response universe but at the cost of a longer waiting time.



(a) Collecting all PCR digits during registration for training

(b) Spectrogram of PCR code

(c) Stimulus signal for PCR code

Fig. 6: Stimulus signals for training and authentication.

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 bandpass 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. 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 25ms coding chirp and a 75ms 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 21.6-21.95kHz, 18.8-19.15kHz, 21.6-21.95kHz and 20.8-21.15kHz when the OTC code is "9207" as shown in Figure 6 (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 Spectrogram Derivation: We derive the spectrogram of each PCR digit as biometric features to describe the PCR defined by Equation 3 in the frequency domain. As shown in Figure 3, the spectrogram 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. The spectrogram is computed based on the Discrete-Time Short-Time Fourier Transform (DT-STFT) as

$$DTSTFT(t,f) = \sum_{\tau=t}^{t+T-1} s(\tau)w(\tau - t)e^{-j2\pi f\tau}, \qquad (7)$$

where t and f are the time and frequency indexes, and $w(t)$ is a window function with length T. Each pixel of the 2D image, spectrogram, at the coordinate (t, f) is then computed as

$$spectrogram(t,f) = |DTSTFT(t,f)|^2. \qquad (8)$$

4) PCR Spectrogram Time Series: In order to balance the spectrogram resolution and the decoding algorithm's complexity, we divide each PCR spectrogram into three pieces, which separately describe three different stages of the PCR. Specifically, the first spectrogram (0-25ms) mainly captures the palm's impact on the dominant direct-path signal. The second (25-50ms) and third (50-75ms) focus on the reflected signals and the induced surface vibration sounds. All of the three spectrogram pieces show user-distinctive patterns and are input in 2D-image time series into the PCR decoder for verification. The 75-100ms subsegment is not utilized, because the sound degrades over 20dB in this period. Furthermore, using spectrogram 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.

5) CNN-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 Convolutional Neural Network (CNN) model best meets the above requirements. 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 for different times, the grabbing actions may result in more than one patterns. We thus leverage the CNN model's strong multi-class classification capability to label multiple hand-grip patterns for each user. 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 [30]. 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 [31].

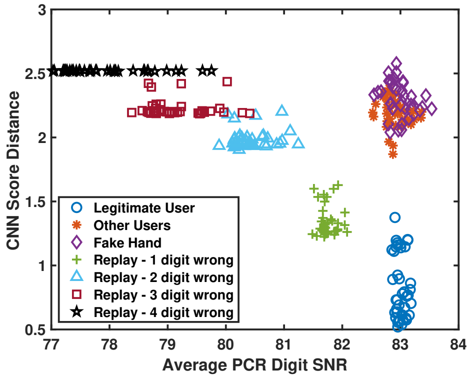


Fig. 7: Illustration of the cluster-based PCR Code Verifier.

The architecture of our CNN model is shown in Appendix Table VI. In particular, 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 overfitting. The cross-entropy is used as the loss function, and the softmax layer outputs the final CNN scores of each input.

CNN Scores of the Input. We resize each spectrogram into a 98×40 time-frequency image as the input of our CNN-based algorithm. Two CNN scores (i.e., probabilities) are computed, which are associated with two classes, *User* and *Non-User*. A higher CNN 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 $12 = 2 \times 3 \times 2$ CNN scores. For each $m-$digit PCR code, we thus obtain $12m$ CNN scores as the biometric features for verification.

6) Cluster-based PCR Code Verifier: We develop a cluster-based algorithm to verify a PCR code by integrating the biometric features ($12m$ CNN scores) and the coding features ($2m$ 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, ..., $m-1$ correct PCR digits. We then cluster these PCR codes based on their CNN 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.

Figure 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, 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. We find that though the user cluster has a wide dispersion along the CNN score distance due to behavioral inconsistency, it is small enough to be separated from the other clusters. Figure 7 indicates that only breaking the coding sequence or replicating the biometric features alone is hard to attack PCR-Auth. We thus further design two attacks that can forge both simultaneously in Section V-E3.

IV. Methodology and Evaluation Criteria

Experimental Setup. We experiment with six different mobile device models ranging from \$140 to \$350, including Samsung Galaxy Note5 and S8, 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 40 participants (26 males and 14 females) aged from 24 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 Institutional Review Board with Application No. 4305.

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 Figure 6(a) is repeatedly played 20 times, and the participants are asked to re-grab the device from a table for each time to present behavioral inconsistency. *re-grab-1* is collected, 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 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 Figure 6(b). 40 PCR codes are collected from each participant, when a re-grab is required each time to imitate an authentication session.

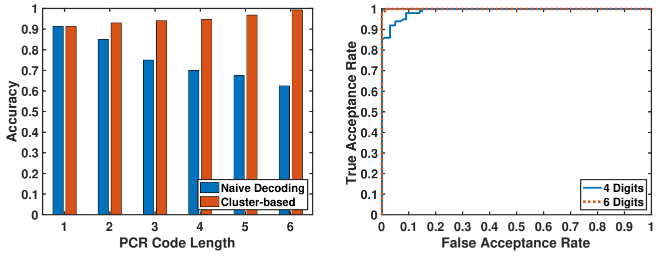
Evaluation Metrics. We first conduct the basic PCR analysis to examine the *accuracy* performance of using the biometric PCR to distinguish users, which is defined as the ratio of accurately classified test instances over all test instances. We then evaluate the authentication performance of PCR-Auth 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

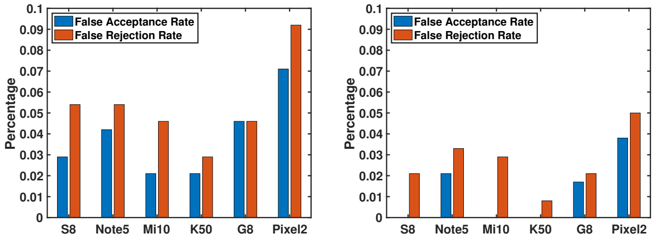
1) Stimulus Signal Duration and Bandwidth: When using a chirp signal for biometric encoding, a question is how to select its duration and bandwidth. Using coding chirps with narrower bands allows extracting more different PCR digits for biometric encoding, and using shorter chirps enables encoding more digits within the limited time. However, the shorter duration and narrower bandwidth also mean the lower resolution to describe the user's PCR. To address the above trade-off, a critical task is to explore the extent of the stimulus signal's duration and bandwidth. We first fix the chirp bandwidth to be 100Hz and examine the chirp lengths from 5ms to 500ms. We find that the PCRs are distinguished accurately by all the tested short chirps equaling or greater than $10ms$. In particular, the accuracy performance increases fast from 80% to 92.5% when the signal duration changes from 5ms to 10ms. After 10ms, the performance has a slower increasing trend. For example, when using 20ms and 500ms signals, we achieve 93% and 95% accuracy respectively. The detailed signal duration study is shown in Appendix Figure 16(a). The results confirm the feasibility to extract PCR using short-duration signals.

We next fix the chirp duration to be 10ms and examine the bandwidth by changing it from 100Hz to 4kHz. We observe that the user's PCR is verified accurately when the bandwidth is no less than 100Hz, starting from which the performance has a slow increasing trend. For example, the accuracy performances are 92.5%, 95.2%, 98%, 98.7% at 100Hz, 200Hz, 350Hz and 500Hz. When the bandwidth is lower than 100Hz, the performance has a drastic drop. The detailed signal bandwidth study is shown in Appendix Figure 16(b). The results confirm the feasibility of using narrow-band signals to distinguish PCRs.



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

Fig. 8: Performance of PCR-Auth and coding gains.



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

Fig. 9: Performance of different device models.

2) PCR Coding Channels and Mic1&2: Based on the above study, we choose 25ms chirps with a 350Hz bandwidth as coding signals. In the ultrasonic frequency range 17-22kHz, we find 12 exclusive channels as candidates. We now evaluate the verification performance of these channels. Table I presents the accuracy performances of these channels, when Mic1, Mic2 or both mics of S8 phone are used, respectively. We find that the 40 participants' PCRs are distinguished accurately on 10 channels, which achieve an average of 91.3% accuracy 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 89.6%, 89.1%, 96.4% and 93.1%, 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. The reason is that Mic 2 sounds have much higher SNR and are less sensitive to the user's behavioral inconsistency. The integration of the two mics makes a more robust authentication system.

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. Figure 8 (a) shows the user verification accuracy when 1-digit to 6-digit PCR codes are used respectively. We observe that both methods achieve 91.3% 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% accuracy with 4-digit, 5-digit and 6-digit PCR codes. 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.

TABLE I: Performance at all PCR coding channels (the 25ms and 350Hz chirp is used).

Ch.	Freq. Range	Mic 1	Mic 2	**Mic 1+2**
0	17.2-17.55kHz	0.742	0.847	**0.864**
1	17.6-17.95kHz	0.716	0.835	**0.852**
2	18.0-18.35kHz	0.696	0.868	**0.896**
3	18.4-18.75kHz	0.718	0.908	**0.939**
-	18.8-19.15kHz	0.653	0.737	**0.763**
4	19.2-19.55kHz	0.774	0.842	**0.891**
5	19.6-19.95kHz	0.781	0.887	**0.929**
-	20.0-20.35kHz	0.599	0.625	**0.653**
6	20.4-20.75kHz	0.827	0.954	**0.964**
7	20.8-21.15kHz	0.836	0.870	**0.927**
8	21.2-21.55kHz	0.902	0.682	**0.933**
9	21.6-21.95kHz	0.895	0.805	**0.931**
Average		0.789	0.850	**0.913**

The ROC curves of the 4-digit and 6-digit PCR codes in Figure 8 (b) further confirm the high performance of PCR-Auth, and both codes achieve a high TAR and a low FAR. Moreover, the 6-digit PCR code's ROC curve is above that of the 4-digit code. 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 Device Models: We next evaluate the performance of our system on six different smartphone models, when fifteen participants are involved. Figure 9 shows the FAR and FRR performance of PCR-Auth when 4- and 6-digit codes are used. We observe that all the six devices achieve a low FAR and a low FRR. When using 6-digit PCR codes, S8, Mi10 and K50 all achieve 0% FARs, and their FRRs are 2.1%, 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 K50 are 2.1% and 2.9% 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

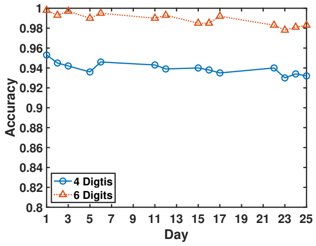


Fig. 10: Long-term performance of PCR-Auth.

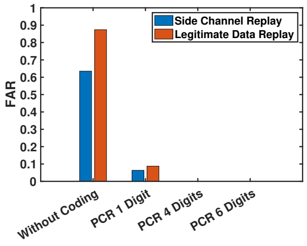


Fig. 11: Performance under replay attacks.

study. 40 PCR codes are collected for each participant for each day, which are only used for testing. Figure 10 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. Impersonation Attack

1) Setup: We perform three 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 four 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. For *knowledgeable fake-hand impersonation*, we use a silicone fake hand [32] to replicate each user's gripping hand. The attackers attempt 40 times for each target user's OTC.

2) Result: Table II presents the performance of the 4- and 6-digit PCR codes under the three impersonation attacks. We observe that both PCR codes achieve a low FAR and a low FRR in the three 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 and the knowledgeable fake-hand impersonations slightly degrade the performance of the 6-digit PCR code. But its EERs are still low under the two advanced impersonation attacks, which are 3.1% and 3.0%, respectively. The 4-digit PCR code has a lower performance compared to the 6-digit PCR code, whose EERs are 5.7%, 6.3%, and 6.2% in the random, knowledgeable, and knowledgeable fake-hand impersonations, respectively. The results indicate the difficulty of replicating the user's PCR via impersonation attacks.

TABLE II: Performance of PCR-Auth under impersonation and replay attacks.

Code	**FRR**	**FAR**								
		Impersonation Attack			Replay (#Err Digit)					
		Rand.	Knowl.	Fake	1	2	3	4	5	6
4 Digits	0.063	0.046	0.061	0.058	0	0	0	0	-	-
6 Digits	0.032	0.010	0.029	0.026	0	0	0	0	0	0

D. 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.

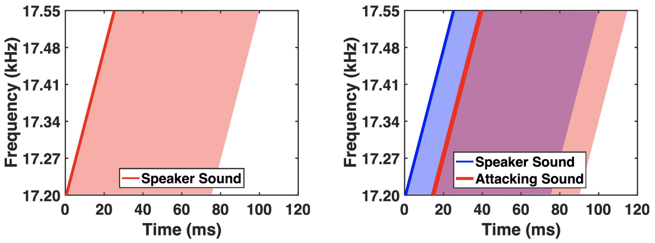
2) Result: Table II shows the performance of PCR-Auth under replay attacks. We find that both the 4- and 6-digit PCR codes prevent all replay attacks with 0% FAR, including the case when there is only 1-digit error. The reason is that each PCR code is only used once, and coding errors can be accurately detected based on the physical layer SNR of each digit. Figure 11 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.

E. Defending New Synthesis Attacks

1) When to Stop Recording?: PCR-Auth is designed to work in a way in which the speaker and the microphone should be turned on and off simultaneously during authentication. However, in practical implementations, such a concurrency could not be achieved due to the audio latency [33]. Then, a critical question is when to stop the mic to improve the data integrity, block additional inputs and prevent the speech privacy leakage. While the audio latency is not avoidable for both users and attackers, it is more important to know whether this latency is stable, so that we can apply a threshold to cut off the recordings. We conduct experiments with different devices to measure the ending time of the recorded PCR codes. Table III presents the variation range of the ending time measurements, when two different Android player classes are used. We find the measured ending time of PCR code is not fixed, which has up to 78.5ms variations. This is caused by unpredictable hardware latency and the CPU scheduling time. Moreover, the variation range is related to the device and the player class. Because this range is not determined, our threshold needs to cover its maximum to ensure each entire PCR code to be recorded. An adversary can also leverage this additional time to attack PCR-Auth too. As our experiments

TABLE III: The variation ranges of PCR code ending time (ms) on different devices.

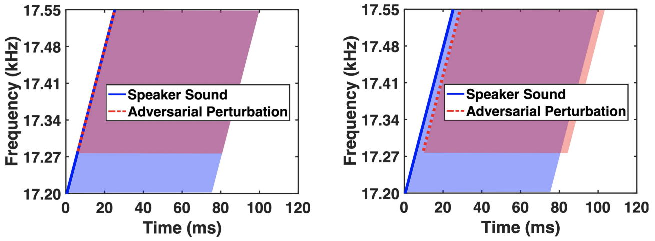
Andorid.Media	S8	Note5	K50	G8	Pixel 2
AudioTrack	43.604	39.730	20.538	78.518	28.338
MediaPlayer	60.618	40.067	33.184	45.485	45.705



(a) User's one digit sound

(b) synchronization disturbance

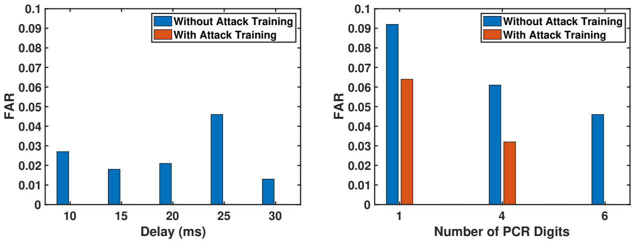
Fig. 12: Illustration of synchronization disturbance attack.



(a) Synchronized (ideal)

(b) Unsynchronized (non-ideal)

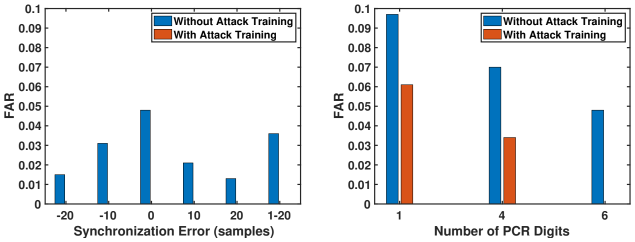
Fig. 13: Illustration of adversarial perturbation attack.



(a) Impact of delays (6 digits)

(b) PCR code length (25ms delay)

Fig. 14: Under synchronization disturbance attack.



(a) Impact of synch error (6 digits)

(b) PCR code length (no synch error)

Fig. 15: Under adversarial perturbation attack.

show that it is less than one digit's time (100ms), the adversary has to launch the attack simultaneously with the phone's speaker sound. Additionally, as the audio of a PCR code can be less than 1 second, which is hard to record the user's complete speeches to cause privacy concerns.

2) Listen-and-synthesis Attack Setup: As discussed above, to present the correct coding sequence to fool PCR-Auth, the adversary has to listen to each coding chirp of the target device during authentication and simultaneously sends the corresponding malicious signal. This process is repeated for every coding chirp, and in this process, the phone may be on a table or in the adversary's hand. We assume the adversary uses FPGA to remove the CPU scheduling time. By considering the time required for observation (3ms for 64-sample buffer), A/D and D/A conversions (2ms), FFT (2ms), bus and signal propagation time (3ms) [34], we find an adversary needs at least 10ms to attack each digit. We thus use 10ms as the delay of adversarial digits for attack simulation. We simulate the *synchronization disturbance attack* by replaying the user's PCR digit with double loudness, which postpones the entire synchronization to pick up adversarial digits. As illustrated in Figure 12, a stronger malicious sound is mixed with the original speaker sound with a delayed phase. We also simulate the *adversarial perturbation attack* by using the perturbation noises to modify the original speaker sound into the user's PCR digit. Due to the delay, the attacking sound can only modify partial frequencies. This attack has two cases depending on whether the perturbations are synchronized well with the speaker sound as illustrated in Figure 13.

3) Results: We evaluate PCR-Auth under the two types of attacks with two training models, the original one and the one updated with the attack data. Figure 14(a) shows the 6-digit PCR code performance under synchronization disturbance attack, when the entire adversarial PCR code is sent with different delays. We observe that without attack training, the FARs are already low for all delays. In particular, when the adversarial digits are delayed by 25ms, which is exactly behind the speaker's coding chirp, the attack achieves the highest FAR, which is 4.6%. When the attack data of all delays are included in training, the FARs fall to 0% for all cases. We further examine the impact of code lengths when the delay is 25ms (the worst case). Figure 14(b) shows that a longer code has a stronger capability to defend this synchronization attack.

We next evaluate the adversarial perturbation attack, and different synchronization errors are considered. The system performance of the 6-digit PCR code is shown in Figure 15(a). We observe that the FARs are all below 4.8% without training the model with attack data. This highest FAR is obtained when the synchronization error is 0 sample. When the attack data is included in training, the 6-digit PCR code achieves 0% FAR for all the cases of adversarial perturbation attack. We further present the FARs of different code lengths when there is no synchronization error (worst case). As shown in Figure 15(b), the FARs are 9.7%, 7%, and 4.8% when the 1-, 4-, and 6-digit codes are used without attack training. When the attack data is trained, the FARs are reduced to 6.1%, 3.4%, and 0%.

F. Under Daily Noises and Dedicated Ultrasonic Interference

While the above results are obtained in the regular office scenarios with a 40dB noise level, we next evaluate PCR-Auth by asking 10 participants to further conduct experiments with eight more different types of noise including the natural ambient noises and the dedicated ultrasonic interference.

1) Impact of Daily Noises: The daily ambient noises we test include an open area scenario at a large parking lot (55dB), a working Air Conditioner (AC) (60dB), regular conversations (65dB), in-car scenarios (70dB) and a train station (75dB). We use the Ultrasound Detector App [35] to measure the SPL at the ultrasound band 17k-22kHz. Table IV 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 not 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 slightly decrease under higher SPL noises. In particular, the average accuracy of all channels is 94.7%, 94.4%, 94.3%, 94%, 93.8% and 93.5% under the noise levels 40dB, 55dB, 60dB, 65dB, 70dB and 75dB. 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 Ultrasound or DoS Attacks: We are aware of that an adversary can 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 17kHz to 22kHz. Table IV presents the performance of PCR-Auth under three ultrasonic SPLs ($17k$-$22k$Hz). When the SPL of the ultrasonic noise increases to 30dB, 40dB and 50dB, the 4-digit PCR code's accuracy drops to 96.2%, 95.4%, and 89.1%, and the 6-digit PCR code's accuracy decreases from 100% to 98.9% and 93.2%. The results indicate the potential of PCR-Auth to work with 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 30cm distance of ultrasound transmission if using normal mobile devices. If the ultrasound SPL is greater than this level, PCR-Auth will notify the user of potential DoS attacks and recommend the user to change a location to restart PCR-Auth. It is important to note that the above results are obtained when PCR-Auth uses the 50% volume. We can further push this DoS attack detection threshold higher when using the phone with higher volumes.

VI. Discussion & Future Work

While this is the first biometric encoding work to implement biometric CR authentication, the following are some issues that could be considered for further improvement.

This work evenly divides the inaudible frequency range $17k$-$22k$Hz into 12 channels and selects the 10 best channels for biometric encoding. After extensive experiments with 40 participants and 6 devices, we find that some channels perform better than the others. Thus, new ways of selecting the coding channels to better leverage the "good frequencies" need to be explored. Moreover, to achieve the optimal system performance, more efforts are needed to balance the base value of the coding system, the channel bandwidth and the coding signal duration. Whether other signal patterns are better than frequency sweeping signals for biometric encoding is worth further exploration. We may use the more advanced time-frequency images and deep learning methods to improve the performance. Additionally, the impacts of user postures (e.g., sitting and laying down), age group and PCR transferability (e.g., from one device to another) need to be further studied.

To cheat the biometric CR authentication, an adversary must capture the authentication challenge in addition to obtaining the user's all biometric data. This work investigates two new synthesis attacks that eavesdrop on every challenge digit via a side-channel and launch attacks immediately to inject each malicious digit. We then discuss the possibility of implementing these attacks by using an FPGA, which achieves both the short processing time and the determined delay to meet the strict synchronization requirement. The detailed implementation of the two attacks requires future work. Moreover, in our attack simulations, we assume the adversary needs at least 10ms to listen, process and attack, by referring to the current parameters of hardware. The shorter delay may be achieved by the future FPGA, and we need to further counteract it, such as by exploring the potential of using shorter coding signals.

VII. 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 [37], voices [38], keystroke dynamics [39], and finger gestures [40]. However, due to the advanced mobile recording techniques (e.g., visual and acoustic), 3D printing and robotics, the physiological and behavioral biometrics are

TABLE IV: 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	0.920	0.920	0.920	0.917	0.914	0.916	0.908	0.838
Ch.1	0.962	0.959	0.959	0.952	0.950	0.948	0.951	0.944	0.872
Ch.2	0.944	0.940	0.938	0.935	0.934	0.932	0.934	0.927	0.851
Ch.3	0.931	0.928	0.927	0.926	0.923	0.921	0.925	0.912	0.847
Ch.4	0.958	0.953	0.951	0.948	0.944	0.940	0.940	0.932	0.869
Ch.5	0.929	0.929	0.927	0.925	0.924	0.922	0.921	0.915	0.842
Ch.6	0.973	0.973	0.971	0.970	0.968	0.965	0.962	0.955	0.891
Ch.7	0.932	0.929	0.928	0.924	0.922	0.920	0.925	0.918	0.858
Ch.8	0.961	0.957	0.955	0.952	0.950	0.945	0.947	0.942	0.881
Ch.9	0.957	0.953	0.950	0.945	0.943	0.941	0.939	0.930	0.853
Average	0.947	0.944	0.943	0.940	0.938	0.935	0.936	0.928	0.860
4-digit PCR Code	0.979	0.975	0.973	0.971	0.965	0.962	0.962	0.954	0.891
6-digit PCR Code	1	1	1	1	1	1	1	0.989	0.932

both under a high risk to be obtained by an adversary [4], [41], [42], [43]. 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 [44], [45], [46], [47]. 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 [48]. Ometov *et al.* propose to combine the user's biometrics such as voice and face with a PIN entry for authentication [49]. 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 [50], [51], capturing finger gesture behaviors from a signature [52], [53] or obtaining vibration signatures from the user's secret input on a solid surface [36]. But all these methods still reuse the same biometric data for every authentication session, which is vulnerable to replay attacks.

There is active research on liveness detection to defend against replay attacks during face and voice authentications. Fathy *et al.* propose a method, which asks the user to show some motions during the face recognition and leverages the video frames to verify a dynamic face [54]. Chen *et al.* [12] ask the user to move the camera around the head to construct a 3D face for authentication, and the liveness detection is based on the consistency between the camera and the motion sensor data. Chen *et al.* [55] detect the magnetic fields emitted by machine speakers to prevent non-live human sounds from attacking voice authentications. VoiceLive [56] and VoiceGesture [57] derive the vocal tract movements and articulatory gestures from human speech sounds to make sure the voice is live. But these methods either require the user's participation to prove the liveness or are subject to additional overheads. They are still unable to prevent the biometric data replay.

Challenge-response protocols are designed to prevent replay attacks [58]. 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.* 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 [13]. Sluganovic *et al.* propose to randomly show a dot on the screen as the challenge and capture the user's reflexive eye movements as the response [14]. 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 [19] 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 performance comparison with related work is in Table V.

VIII. 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 and the coding sequence. In particular, we devise a biometric encoding technique, which uses acoustic signals to encode the biometric into biometric codes to respond to the current authentication challenge. The biometric encoding generates a large biometric response universe to support massive CR authentication requests and prevent replay attacks. Furthermore, we develop a deep learning-based algorithm to decode the biometric code and investigate new attacks by assuming that the adversary is able to break the coding sequence, replay the biometric data, or replicate both, respectively. Extensive experiments show that a 6-digit PCR achieves a 97% accuracy to distinguish users and reject both replay and synthesis attacks with 100% accuracy.

Acknowledgments. This work was partially supported by LEQSF(2020-23)-RD-A-11. We would like to also thank our anonymous reviewers for helping us improve the paper.

TABLE V: Comparison with related studies.

Work	Protocol	Modality	FNR	FPR			User	Dedicated	Response
				Impersonation	Replay	Synthesis	Participation	Hardware	Pool
LivDet [7]	Physiological	FingerPrint	11.96%	1.07%	N.A.	N.A.	Low	Yes	N.A.
Erdogmus et al. [4]		FaceID	5.5%	1.1%	N.A.	N.A.	Medium	Yes	N.A.
Menotti et al. [6]		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 [36]		Vibration response of dynamic gestures	10%	2%	N.A.	N.A.	High	Yes	N.A.
Eye Movement [14]	Challenge-response	Reflective eye movement	6.3%	6.3%	0.06%	N.A.	Medium	Yes	N.A.
Velody [19]		Vibration response	5.8%	5.8%	0%	0%	Medium	Yes	n
PCR-Auth		Vibration response of palm contact	3.2%	2.9%	0%	0%	Low	No	n^m

REFERENCES

[1] EyeVerify, "EyeVerify Survey Reveals High Consumer Trust in Biometrics for Mobile Banking and Payments," 2017, 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] PYMNTS, "1.5B Mobile Users To Rely On Biometrics Security By 2023," 2018, https://www.pymnts.com/authentication/2018/biometrics-smartphone-mobile-security-fingerprint-sensors/.

[3] M. Mohamed, B. Shrestha, and N. Saxena, "Smashed: Sniffing and manipulating android sensor data for offensive purposes," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 901–913, 2016.

[4] N. Erdogmus and S. Marcel, "Spoofing face recognition with 3d masks," *IEEE transactions on information forensics and security*, vol. 9, no. 7, pp. 1084–1097, 2014.

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

[6] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 864–879, 2015.

[7] L. Ghiani, D. Yambay, V. Mura, S. Tocco, G. L. Marcialis, F. Roli, and S. Schuckcrs, "Livdet 2013 fingerprint liveness detection competition 2013," in *2013 International Conference on Biometrics (ICB)*. IEEE, 2013, pp. 1–6.

[8] J. Shelton, K. Roy, B. O'Connor, and G. V. Dozier, "Mitigating iris-based replay attacks," *International Journal of Machine Learning and Computing*, vol. 4, no. 3, p. 204, 2014.

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

[10] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE, 2014, pp. 1–5.

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

[12] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted facial recognition: an enhanced biometric authentication system for smartphones," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, 2014, pp. 109–122.

[13] M. Mohamed, P. Shrestha, and N. Saxena, "Challenge-response behavioral mobile authentication: a comparative study of graphical patterns and cognitive games," in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 355–365.

[14] I. Sluganovic, M. Roeschlin, K. B. Rasmussen, and I. Martinovic, "Using reflexive eye movements for fast challenge-response authentication," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1056–1067.

[15] Z. Chen and M. Recce, "Handgrip recognition," *Journal of Engineering, Computing and Architecture*, vol. 1, no. 2, 2007.

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

[17] A. S. Weksler, N. J. Peterson, and R. S. VanBlon, "Grip signature authentication of user of device," June 11 2015, uS Patent App. 14/098,180.

[18] J. Liu, Y. Chen, M. Gruteser, and Y. Wang, "Vibsense: Sensing touches on ubiquitous surfaces through vibration," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2017, pp. 1–9.

[19] J. Li, K. Fawaz, and Y. Kim, "Velody: Nonlinear vibration challenge-response for resilient user authentication," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1201–1213.

[20] 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 *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 42–56.

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

[22] K. Ashihara, "Hearing thresholds for pure tones above 16 khz," *The Journal of the Acoustical Society of America*, vol. 122, no. 3, pp. EL52–EL57, 2007.

[23] B. Zhou, J. Lohokare, R. Gao, and F. Ye, "Echoprint: Two-factor authentication using acoustics and vision on smartphones," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 321–336.

[24] D. Han, Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Proximity-proof: Secure and usable mobile two-factor authentication," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 401–415.

[25] K. Sun, T. Zhao, W. Wang, and L. Xie, "Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 591–605.

[26] Y.-C. Tung and K. G. Shin, "Expansion of human-phone interface by sensing structure-borne sound propagation," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, pp. 277–289.

[27] J. Jonsson and B. Kaliski, "Public-key cryptography standards (pkcs)# 1: Rsa cryptography specifications version 2.1," RFC 3447, February, Tech. Rep., 2003.

[28] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 157–175.

[29] A. Shamir, "On the generation of cryptographically strong pseudo-random sequences," *ACM Transactions on Computer Systems (TOCS)*, vol. 1, no. 1, pp. 38–44, 1983.

[30] Apple, "Use touch id on iphone and ipad," April 2019, https://support.apple.com/en-us/HT201371.

[31] M. Xu, J. Liu, Y. Liu, F. X. Lin, Y. Liu, and X. Liu, "A first look at deep learning apps on smartphones," in *The World Wide Web Conference*, 2019, pp. 2125–2136.

[32] "Flexible bendable mannequin hands for nails practice fake hand nail display tool," 2021, https://www.amazon.com/gp/product/B08VJ27BRN/ref=ox_sc_act_title_1?smid=A2CH949N64GN7G.

[33] Android, *Audio Latency Measurements*. Google, 2020, https://source.android.com/devices/audio/latency/measurements.

[34] M. Ottewill, *Latency in audio production systems*, http://www.planetoftunes.com/sound-recording/digital-audio-latency.htm.

[35] S. Gudkov, "Ultrasound detector," 2018, https://play.google.com/store/apps/details?id=com.microcadsystems.serge.ultrasounddetector.

[36] J. Liu, C. Wang, Y. Chen, and N. Saxena, "Vibwrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 73–87.

[37] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," *IEEE Transactions on Mobile Computing*, 2014.

[38] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE transactions on speech and audio processing*, vol. 3, no. 1, pp. 72–83, 1995.

[39] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors." in *ICNP*, vol. 14, 2014, pp. 221–232.

[40] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-rich gestures: a novel approach to authentication on multi-touch devices," in *proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 977–986.

[41] A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni, "Fake finger detection by skin distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 3, pp. 360–373, 2006.

[42] A. Serwadda and V. V. Phoha, "When kids' toys breach mobile phone security," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 599–610.

[43] S. Kim, C. Kim, and J. H. Park, "Human-like arm motion generation for humanoid robots using motion capture database," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3486–3491.

[44] D.-J. Kim, K.-W. Chung, and K.-S. Hong, "Person authentication using face, teeth and voice modalities for mobile device security," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2678–2685, 2010.

[45] C. McCool, S. Marcel, A. Hadid, M. Pietikäinen, P. Matejka, J. Cernockỳ, N. Poh, J. Kittler, A. Larcher, C. Levy *et al.*, “Bi-modal person recognition on a mobile phone: using mobile phone data,” in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*. IEEE, 2012, pp. 635–640.

[46] K. B. Raja, R. Raghavendra, M. Stokkenes, and C. Busch, “Multi-modal authentication system for smartphones using face, iris and periocular,” in *Biometrics (ICB), 2015 International Conference on*. IEEE, 2015, pp. 143–150.

[47] M. De Marsico, C. Galdi, M. Nappi, and D. Riccio, “Firme: Face and iris recognition for mobile engagement,” *Image and Vision Computing*, vol. 32, no. 12, pp. 1161–1172, 2014.

[48] A. Boehm, D. Chen, M. Frank, L. Huang, C. Kuo, T. Lolic, I. Martinovic, and D. Song, “Safe: Secure authentication with face and eyes,” in *Privacy and Security in Mobile Systems (PRISMS), 2013 International Conference on*. Citeseer, 2013, pp. 1–8.

[49] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, “Multi-factor authentication: A survey,” *Cryptography*, vol. 2, no. 1, p. 1, 2018.

[50] C. Giuffrida, K. Majdanik, M. Conti, and H. Bos, “I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics,” in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2014, pp. 92–111.

[51] 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 *proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 987–996.

[52] N. Sae-Bae and N. Memon, “Online signature verification on mobile devices,” *IEEE transactions on information forensics and security*, vol. 9, no. 6, pp. 933–947, 2014.

[53] Y. Ren, C. Wang, Y. Chen, M. C. Chuah, and J. Yang, “Critical segment based real-time e-signature for securing mobile transactions,” in *2015 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2015, pp. 7–15.

[54] M. E. Fathy, V. M. Patel, and R. Chellappa, “Face-based active authentication on mobile devices,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2015.

[55] S. Chen, K. Ren, S. Piao, C. Wang, Q. Wang, J. Weng, L. Su, and A. Mohaisen, “You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones,” in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 183–195.

[56] L. Zhang, S. Tan, J. Yang, and Y. Chen, “Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1080–1091.

[57] L. Zhang, S. Tan, and J. Yang, “Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 57–71.

[58] D. L. Davis and L. Smith, “Authentication system based on periodic challenge/response protocol,” Jul. 11 2000, uS Patent 6,088,450.

Appendix

A. *The Configuration of The CNN-based PCR Decoder*

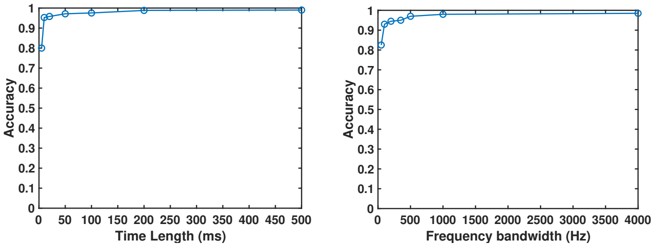
Table VI shows the detailed architecture of our CNN-based PCR Decoder to decode the spectrograms of each PCR digit.

TABLE VI: The architecture of our CNN model to verify PCR spectrograms.

Layer	Output Shape	Parameter #	Activation #
Input: PCR Spectrogram	(40,98,1)	0	3920
Conv2D + RecLineU	(40,98,12)	120	47070
Max Pooling	(20,49,12)	0	11760
Batch Normalization	(20,49,12)	24	11760
Conv2D + RecLineU	(20,49,24)	2616	23520
Max Pooling	(10,25,24)	0	6000
Batch Normalization	(10,25,24)	48	6000
Conv2D + RecLineU	(10,25,48)	10416	12000
Max Pooling	(5,13,48)	0	3120
Batch Normalization	(5,13,48)	96	3120
Conv2D + RecLineU	(5,13,48)	20784	3120
Conv2D + RecLineU	(5,13,48)	20784	3120
Max Pooling	(5,1,48)	0	240
Dropout	(5,1,48)	0	240
Fully Connected + Softmax	(2)	482	2
Output: Probability Distribution	(1)	0	0

B. *Stimulus Signal Duration and Bandwidth Study*

Figure 16 illustrates the impacts of the stimulus signal's duration and bandwidth on the PCR verification accuracy.



(a) Duration

(b) Bandwidth

Fig. 16: Impact of coding chirp duration and bandwidth.