SwipePass: Acoustic-based Second-factor User Authentication for Smartphones

YONGLIANG CHEN, TAO NI, and WEITAO XU*, City University of Hong Kong Shenzhen Research Institute, China and City University of Hong Kong, China
TAO GU, Macquarie University, Australia

Pattern lock-based authentication has been widely adopted in modern smartphones. However, this scheme relies essentially on passwords, making it vulnerable to various side-channel attacks such as the smudge attack and the shoulder-surfing attack. In this paper, we propose a second-factor authentication system named *SwipePass*, which authenticates a smartphone user by examining the distinct physiological and behavioral characteristics embedded in the user's pattern lock process. By emitting and receiving modulated audio using the built-in modules of the smartphone, *SwipePass* can sense the entire unlocking process and extract discriminative features to authenticate the user from the signal variations associated with hand dynamics. Moreover, to alleviate the burden of data collection in the user enrollment phase, we conduct an in-depth analysis of users' behaviors under different conditions and propose two augmentation techniques to significantly improve identification accuracy even when only a few training samples are available. Finally, we design a robust authentication model based on CNN-LSTM and One-Class SVM for user identification and spoofer detection. We implement *SwipePass* on three off-the-shelf smartphones and conduct extensive evaluations in different real-world scenarios. Experiments involving 36 participants show that *SwipePass* achieves an average identification accuracy of 96.8% while maintaining a false accept rate below 0.45% against various attacks.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**.

Additional Key Words and Phrases: User authentication, Acoustic sensing, Smartphone, Deep learning

ACM Reference Format:
Yongliang Chen, Tao Ni, Weitao Xu, and Tao Gu. 2022. SwipePass: Acoustic-based Second-factor User Authentication for Smartphones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3, Article 106 (September 2022), 25 pages. https://doi.org/10.1145/3550292

1 INTRODUCTION

Over the past decades, smartphones are becoming increasingly popular and have significantly changed the way people live. Since smartphones store extensive private and personal data such as social contacts and bank information, a secure and user-friendly user authentication system is highly desirable. Biometric technologies have been widely adopted to verify a user's identity in commercial smartphones. Specifically, biometric-based authentication systems authenticate the users based on their unique physiological biometric characteristics, such as the face, fingerprint, and iris. However, these mechanisms not only need costly specialized sensors

[*] Corresponding author.

Authors' addresses: Yongliang Chen, cs.ylchen@my.cityu.edu.hk; Tao Ni, taoni2-c@my.cityu.edu.hk; Weitao Xu, weitaoxu@cityu.edu.hk, City University of Hong Kong Shenzhen Research Institute, Shenzhen, China and City University of Hong Kong, Hong Kong, China; Tao Gu, Macquarie University, Sydney, Australia, tao.gu@mq.edu.au.

2474-9567/2022/9-ART106 $15.00
https://doi.org/10.1145/3550292

Table 1. Comparison of pattern lock based authentication methods (❍–Low, ◗–Medium, ●–High).

Scheme	Sensing Technique	Generalizability	Pattern Diversity	Data Efficiency	User Experience
Angulo *et al.* [4]	Touch Screen	❍	❍	◗	❍
Liu *et al.* [14]	Touch Screen	❍	❍	◗	❍
TouchID [45]	IMU, Touch Screen	❍	●	◗	◗
FCBBS [27]	IMU, Touch Screen	❍	●	❍	◗
TouchPrint [7]	Acoustic	●	❍	◗	❍
SwipePass	Acoustic	●	●	●	●

but also suffer from the hazard of spoofing attacks [1, 2, 11]. In addition, there are still nearly 20% of modern smartphones without these biometric sensors [32]. Therefore, traditional password-based authentication methods are still widely used, among which the pattern lock is one of the most popular schemes. A typical pattern lock scheme requires a user to unlock their smartphone by swiping on a 3 × 3 grid of contactable points. Because of its simplicity and convenience, it has been widely used in many commercial devices (e.g., Android-based smartphones which have 71.59% global market share [31]) and mobile applications (e.g., Alipay [3], Wechat [34], and AppLock [21]).

Despite its popularity, pattern lock has exposed to several vulnerabilities recently, which may lead to serious security hazard. For example, a recent study [30] shows that although a theoretically huge password space can be provided (e.g., 389, 112 for a 3 × 3 pattern lock), people tend to select simple unlock patterns ending up in a much smaller subspace. Moreover, the unlock pattern can be inferred by various side channels, such as smudges on touchscreen [5], radio signal [44], camera [42] and inaudible acoustic signal [47]. These side-channel attacks reveal that directly using the unlock pattern is far from being secure to adversarial attacks.

To strengthen the security of pattern lock, many efforts have been made by utilizing the rich on-board sensors on smartphone such as touch sensor [4, 14], IMU sensor [27, 45], and audio [7]. While these systems adopt different sensing modalities, they leverage essentially on the same idea that different people show distinct behaviors when they swipe unlock patterns due to the difference in hand geometry biometrics (e.g., palm size and finger length) and behavioral characteristics (e.g., speed and pressure) [24]. In some systems [4, 14], sensor data collected from touch screen are used to extract temporal features such as swiping speed and pressure to identify the user. Another line of research uses the built-in IMU sensor to capture the minor movement of smartphone when a user performs the unlocking pattern [27, 45]. Recent works focus on using the acoustic signals which can be sent and received by the embedded audio modules (i.e., microphone and speaker) to sense user's unlocking behaviors [7]. Despite these efforts, several limitations exist in prior studies which we summarize in Table 1.

- **Generalizability:** A user may unlock his/her smartphone with different postures (e.g., standing and sitting) in a variety of environments (e.g., home and office), therefore an authentication system should be able to achieve high performance in different scenarios. Unfortunately, existing systems either have low accuracy or only work in limited scenarios. For example, data recorded from touch screen can only capture limited biometric information about on-screen behaviors, resulting in a low authentication accuracy (10.4% EER in [4]). The system in [14] utilizes more features to improve performance, however both systems do not consider working in different scenarios, i.e., different environments and postures. IMU-based methods [27, 45] only work when user is holding a smartphone but fail to work when the phone is placed somewhere else, e.g., on a table. Moreover, IMU-based methods are sensitive to user's movement, hence the system performance may drop significantly when user is walking or on a vehicle.

- **Pattern diversity:** Since there is a large password space for a typical 3×3 pattern lock, authentication should be robust to different patterns, including both simple and complicated patterns. However, most of the existing systems only work well when complicated patterns are used. In this case, more sensor data are collected and hence more useful information can be extracted for accurate authentication. Correspondingly, users have to either choose complicated, multi-twist patterns [4] or long patterns [14] to achieve high security.
- **Data efficiency:** Existing systems typically require a large amount of the legitimate user's data to achieve satisfactory performance, resulting in low data efficiency. For example, FCBBS [27] requires extensive data collection in a variety of contexts to achieve high accuracy. While obtaining more data can be beneficial in improving authentication performance, extensive data collection is known to be labor-intensive and time-consuming.
- **User experience:** Pattern lock-based second-factor authentication systems should not pose any additional restrictions on the unlocking process, which may reduce user experience. However, a recent work TouchPrint [7] assumes the user pauses in several fixed positions (i.e., turning points), which is unrealistic and user-hostile. This assumption requires users to unlock a device in limited environments with carefully chosen patterns and more data, leading to poor user experience.

The aforementioned problems raise a critical question: *can we securely authenticate users when they perform pattern lock naturally without imposing much burden of data collection during the enrollment?* To answer this question, we develop a novel user authentication system, *SwipePass*, which utilizes the built-in microphone and speaker to emit and receive acoustic signals to capture the user's unique physiological and behavioral characteristics inherited from the pattern lock input. *SwipePass* can be seamlessly integrated into off-the-shelf smartphones as a second-factor authentication scheme to enhance the security level of traditional pattern lock without sacrificing user experience. While the idea is straightforward, we need to address several non-trivial challenges.

- **Challenge 1:** *how to extract robust and discriminative features from dynamic contexts?* Although well-defined acoustic signals are resilient to some environmental variations such as light conditions and audible ambient noise, they are sensitive to the surrounding objects (e.g., irrelevant body motions and furniture). Moreover, the attackers can imitate the way a legitimate user unlocks the smartphone by shoulder-surfing attacks. Therefore, it is challenging to extract features robust to the dynamic contexts but still user-specific. In *SwipePass*, we propose several novel methods to exclude the variations of surrounding contexts while preserving the informative components. Thereafter, we propose two types of features from the audio signal, namely *Magnitude Profile* and *Phase Contour Profile*, which depicts user-specific characteristics.
- **Challenge 2:** *how to achieve accurate user identification and spoofer detection while keeping computation and energy efficient for smartphones?* Smartphone users may frequently unlock their smartphones every day. Therefore, the computation and energy efficiency are crucial factors in designing our system. In *SwipePass*, we propose several efficient methods to segment the hand-induced signal variations and extract discriminative features, and design an efficient deep-learning authentication model to achieve system performance.
- **Challenge 3:** *how to design a reliable and secure authentication model in a limited-data regime?* Most of the existing systems require a large amount of training data to achieve high accuracy, essentially reducing their applicability. To address this problem, we conduct an in-depth analysis of smartphone users' behaviors and propose two novel data augmentation methods to imitate the possible variations based on a few samples, which significantly reduces the tedious enrollment process.

Table 1 illustrates the properties of *SwipePass* while comparing with the existing schemes. We further summarize our contributions as follows:

- We propose a novel second-factor authentication system named *SwipePass*, which leverages the built-in audio modules on smartphone to strengthen the security level of the traditional pattern lock scheme. *SwipePass* enables efficient, secure, and user-friendly authentication for smartphone users.
- We propose a novel approach to extract useful information from noisy environments. Specifically, our method first extracts the dynamic hand information by estimating the multi-path acoustic propagation channel, then it extracts discriminative features from acoustic signal strengths and delays to characterize user-specific biometric patterns.
- We propose a novel method to efficiently locate the informative audio segments caused by hand motion only. Together with the lightweight classification model, *SwipePass* is able to achieve high computation and energy efficiency.
- We conduct a comprehensive study on different factors that may affect the unlocking process. Based on our findings, we propose two data augmentation techniques to significantly improve the identification accuracy even with only a few pattern swipes in the enrollment phase, mitigating the burden of data collection.
- We implement *SwipePass* on three off-the-shelf smartphones and conduct extensive experiments in three different real-world environments and four different postures. Results show that the proposed system achieves an average identification accuracy of 96.8% while maintaining a false accept rate of below 0.45% against the credential-aware attack, the mimicry attack and the replay attack. Moreover, we conducted a user study by recruiting 40 participants to use *SwipePass* over one week. The survey results demonstrate the practicability of *SwipePass* in real-world applications.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 presents the design details of *SwipePass*. Then, Section 4 presents the evaluation results and Section 6 concludes the paper.

2 RELATED WORK

2.1 Adversarial Attacks on Pattern Lock

Pattern lock is popular among smartphone users nowadays, however, it is vulnerable to various attacks. Aviv *et al.* [5] recover the unlock pattern using smudges left on the smartphone with images under different camera settings. Researchers also find that the moving trajectories can be revealed based on variations in wireless signals [44] and acoustic signals [47]. Moreover, Snoopy [15] is able to recognize the input patterns using IMU sensors when users unlock their smartwatches. Thus, augmenting the security of pattern lock is necessary.

2.2 Pattern Lock-based Second-factor Authentication Systems

Although some researchers have investigated the security of pattern lock and provided advice to construct more secure patterns [30, 36], it is still vulnerable to shoulder-surfing attacks. To enhance the security of the pattern lock, a set of pioneering efforts have been made by exploring different sensing modalities such as the touching screen [4, 29], IMU sensor [27, 45] and acoustic signals [7]. Early approaches [4, 14] mainly rely on sensory data collected by the touch screen. Specifically, Angulo *et al.* [4] leverages the traversing speed across dots in the pattern as features to authenticate users. However, it assumes that both data from legitimate users and attackers are available, which is impractical. In [14], Liu *et al.* proposed to extract more comprehensive biometric patterns by including touch pressure features in the feature space. These work are examined to be resistant to some malicious attacks, but they do not take actual complicated usage contexts (e.g., variations in swiping speed, surrounding environments and unlocking postures) into consideration. The multi-touch authentication scheme proposed by Song *et al.* [29] assumes that the user swipes the screen with multiple fingers, which is difficult to be directly deployed on current smartphones. FCBBS [27] and TouchID [45] leverage the sensor data from both IMU sensor and touch screen collected during the unlocking period to authenticate a user. Both systems work well if users unlock in a hand-holding manner, since variations in IMU data can capture the hand-held smartphone motion, which in turn reflects the user-specific identity information. However, if the smartphone is placed stably on a table, the IMU data may not change significantly and hence cannot provide any useful information. The performance thus decreases. A close work to *SwipePass* is TouchPrint [7], which also uses acoustic features to authenticate the users during the unlocking process. However, it assumes the smartphone user has a temporal pause on turning points which is user-hostile and unrealistic. Compared with TouchPrint, *SwipePass* uses acoustic signals to sense the whole dynamic process of unlocking without any assumptions on the user's habitual behaviors. Another drawback of the previous systems is that they need much more training data to achieve the same level of accuracy as *SwipePass*. Therefore, compared with these earlier works, *SwipePass* is more user-friendly, data-efficient, and generalizable.

2.3 Acoustic-based Authentication Systems

To improve the security of smartphones, instead of developing second-factor authentication systems similar to *SwipePass*, another line of work uses the acoustic signal to sense other biometric patterns. In these works, similar to *SwipePass*, the user-specific biometric characteristics are captured from the propagated multi-path acoustic signals. Chen *et al.* proposed EchoFace [8], an acoustic-based liveness detection system that can resist media attacks in face recognition applications by analyzing the reflected multi-path echos. Similarly, EchoPrint [46] leverages the signals reflected from human face, together with the assistance of the camera, to identify the identity of an incoming user. Lip movement is another biometric pattern that can be resorted to distinguish people, based on which Lu *et al.* [16] proposed LipPass to authenticate a user using the lip-induced Doppler shift when speaking words. Another system named VocalLock [17] performs authentication based on the reflected acoustic signals from the vocal tract during speaking, which is examined to exhibit individual uniqueness. Besides smartphone-based applications, Schneegass *et al.* [25] developed a biometric system on wearable devices to authenticate a user by analyzing audios propagated through the human's skull. Fan *et al.* [9] proposed that echos captured by earphones can reveal the unique in-ear structure of a user, which can be leveraged for authentication. These works mainly focus on providing novel ways of user authentication, which are parallel to *SwipePass*. In contrast, *SwipePass* is based on a widely adopted unlocking scheme and hence shows higher usability.

2.4 Other Acoustic-based Applications

Besides authentication, acoustic signals have been widely used in many other sensing applications, such as motion tracking [13, 18, 33], gesture recognition [23, 38], and health monitoring [28, 43]. The relatively slow propagation speed of acoustic waves in common media (e.g., compared with Radio Frequency) makes it an ideal medium for accurate sensing while only occupying a narrow bandwidth. This facilitates their convenient deployment on commercial smart devices [6]. To track a moving target (e.g., the human hand), one major challenge is to resolve the target reflected echo from the mixed multi-path signals. Nandakumar *et al.* proposed FingerIO [18], which uses the cross-correlation of the OFDM modulated signals to locate the target finger. They discover the sample error is linearly correlated to the phase change and it can be compensated to achieve sub-centimeter level tracking accuracy. In [33], Sun *et al.* proposed VSkin, a system that supports fine-grained gesture-sensing on the back of mobile devices. They track the moving finger by estimating the phase change of the impulse response. Based on the microphone array, Li *et al.* proposed FM-Track [13], which leverages multi-dimensional acoustic information to track multiple targets of interest simultaneously. To recognize a gesture performed, Ruan *et al.* [23] investigated the Doppler frequency shifts of different gestures and successfully classify six basic gestures in a train-free manner. Wang *et al.* [38] introduced a multi-frequency modulation scheme to mitigate the frequency selective fading effect in acoustic sensing, which significantly boosts the performance in gesture

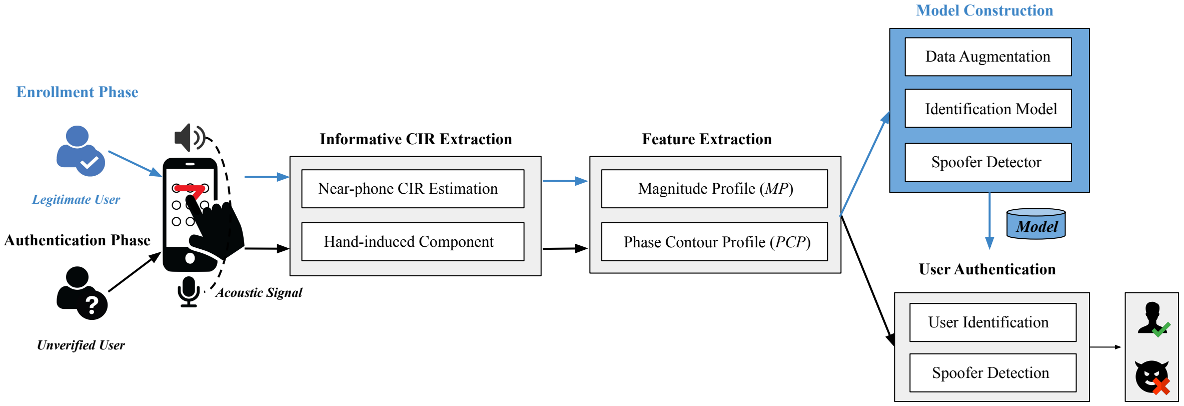


Fig. 1. System architecture of *SwipePass*.

recognition. In health monitoring area, Song *et al.* [28] proposed SpiroSonic, a system that measures the humans' chest wall motion via acoustic sensing. The result is then interpreted into lung function indices, based on the clinically validated correlation between them. By investigating the phase change cycle, Zhang *et al.* [43] further proposed to monitor heartbeat from the minor chest displacement using smart speakers. The above applications demonstrate the huge benefit that acoustic sensing can bring to our life.

3 SYSTEM DESIGN

3.1 Overview

As shown in Fig. 1, *SwipePass* works in two phases: enrollment phase and authentication phase.

In the enrollment phase, the legitimate users need to perform their pre-defined unlock patterns several times to train a user-specific model. When users swipe on their smartphones, their behaviors will be reflected by the inaudible acoustic signals that are sent and received by the built-in audio modules (e.g., speaker and microphone). After receiving the acoustic signal, we first locate the near-phone acoustic channels and estimate their Channel Impulse Response (CIR), which reflects the variations in response to external environmental changes. Then, we extract the informative CIR components related to the moving hand only. Based on the informative CIR, we propose to extract two types of features, which are named as *Magnitude Profile* and *Phase Contour Profile*. After that, we apply the proposed data augmentation methods on the extracted features to enhance the training dataset so that fewer enrollment efforts are required to achieve satisfactory performance. Finally, a user identification model and a spoofer detection model are trained based on the augmented dataset.

In the authentication phase, *SwipePass* performs the same informative CIR extraction and feature extraction as above. The extracted features are then fed into the pre-trained model. If the user's data is authenticated as legitimate by the model built in the enrollment phase, the user successfully passes the authentication. Otherwise, the login request will be rejected.

3.2 Transceiver Design

To sense the biometric patterns of the user, we are going to design a transceiver with the microphone and speaker on the smartphone for emitting and receiving the ultrasonic acoustic signal.

3.2.1 Acoustic Signal Design. We select Zadoff-Chu (ZC) sequence as the transmitted signal because it has an ideal auto-correlation property, which can effectively separate signals received from multiple propagation

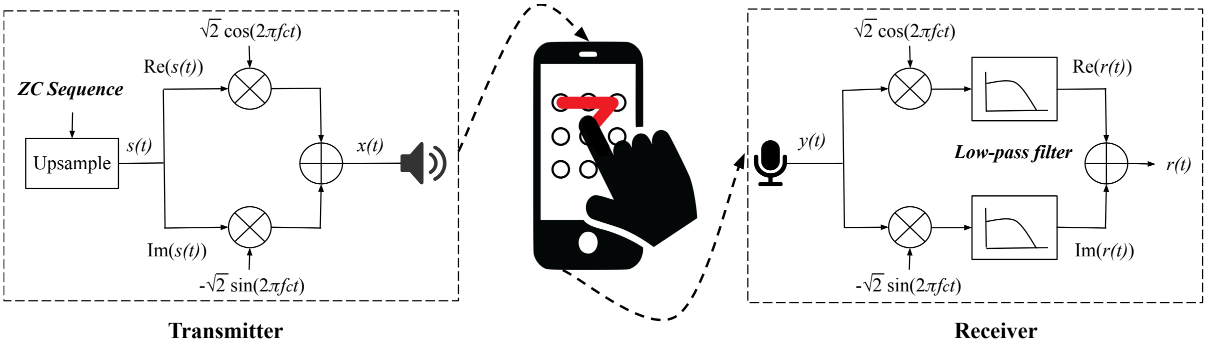


Fig. 2. The transceiver design.

paths [33, 37]. Specifically, the ZC sequence of length N_{ZC} is generated by:

$$ZC[n] = e^{-j\frac{\pi u n(n+1+2q)}{N_{ZC}}}, \tag{1}$$

where $0 \leq n < N_{ZC}$, q is a constant integer, and $u \in [0, N_{ZC}]$ is an integer coprime to N_{ZC}. Then, we need to up-sample the basic ZC sequence so that it can be fit into a targeting signal bandwidth B. To achieve this, we perform FFT on the N_{ZC}-length sequence and zero-pad the frequency spectrum by inserting zeros at the center. This step can separate the positive and negative frequency components, and interpolate the spectrum to be N_{ZCI}-length, where $N_{ZCI} = f_s N_{ZC}/B$. Next, IFFT is performed to convert the frequency signal back to obtain the ZC baseband signal in time domain $s(t)$. In *SwipePass*, we set $N_{ZC} = 127$, $u = 63$, $N_{ZCI} = 1024$, $B = 5.953$ kHz, and $f_s = 48$ kHz. We denote one 1024-length ZC sequence $s(t)$ as an acoustic frame; therefore, each acoustic frame lasts for only 21.3 ms, which is short enough to capture user's hand motions during unlocking.

3.2.2 Transmitter. The design of the transmitter is shown in the left block of Fig. 2. After obtaining the baseband ZC signal $s(t)$, we up-convert it into an inaudible signal in real format. Specifically, the real and imaginary components of $s(t)$ are multiplied by $\sqrt{2}\cos(2\pi f_c t)$ and $-\sqrt{2}\sin(2\pi f_c t)$, respectively. Then, the summation of the two parts is in real format and can be processed by the speaker. The carrier frequency of the modulated signal is $f_c = 20$ kHz, and the modulated signal occupies the frequency band between 17 kHz and 23 kHz, which is inaudible to most people [39]. The modulated acoustic frame is played repetitively by the speaker when a user is unlocking the smartphone.

3.2.3 Receiver. At the receiver, the microphone also works at 48 kHz sampling rate, and the recorded passband acoustic signal $y(t)$ needs to be converted back to the baseband signal. As shown in the lower right block of Fig. 2, the down-conversion is carried out by multiplying $y(t)$ with $\sqrt{2}\cos(2\pi f_c t)$ for the real part and $-\sqrt{2}\sin(2\pi f_c t)$ for the imaginary part. Then, a low-pass filter is used to eliminate the frequency components higher than half of the allocated bandwidth B. Finally, we combine the two parts to obtain the baseband signal $r(t)$.

3.3 CIR Estimation

After receiving the acoustic signal, we need to estimate the variations of different propagation channels. The signal $r(t)$ received by the microphone is a superposition of multiple copies of the transmitted signal $s(t)$ with different delays and extents of attenuation, which can be treated as a multi-path propagation model in a Linear Time-Invariant system [35]:

$$r(t) = \sum_{i=1}^{L} A_i e^{-j\theta_i} s(t - \tau_i) = h(t) * s(t), \tag{2}$$

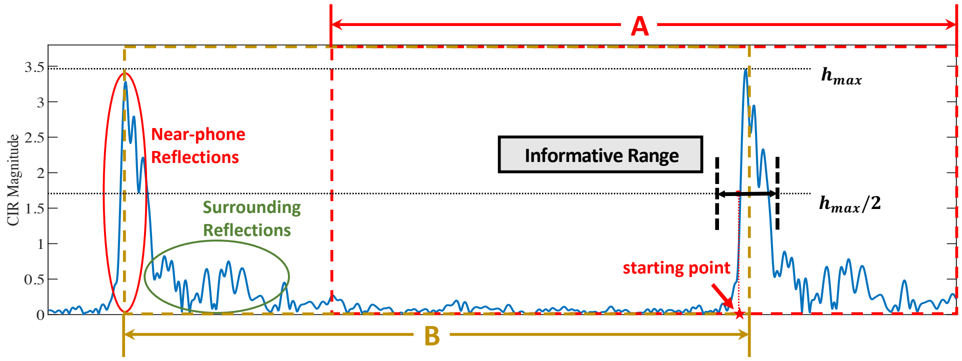


Fig. 3. Near-phone CIR estimation.

where L is the number of propagation paths, A_i demonstrates the attenuation in magnitude, and $\theta_i = 2\pi f_c \tau_i$ is the phase offset induced by the time delay τ_i in the i-th path. The formula can be further written in the convolution between the transmitted signal and the CIR $h(t)$. Here, $h(t) = \sum_{i=1}^{L} A_i e^{-j\theta_i} \delta(t - \tau_i)$ and $\delta(t)$ is Dirac's delta function. Since $\delta(t)$ is zero everywhere except at $t = 0$, we can derive the path variation with delay τ_i directly through $h(\tau_i) = A_i e^{-j\theta_i}$.

In practice, we can estimate the discrete CIR, $h[n]$, with an interval of $T_s = 1/f_s$, to measure the channel variations. Given the fixed-length transmitted signal, for each acoustic frame, the sampled $h[n]$ can reveal the variations of N_{ZCI} taps (i.e., paths), from 0 to $N_{ZCI} - 1$, covering the range up to $N_{ZCI}/f_s \times 340 = 7.253$ m away from the speaker, which is far enough for sensing the hand-induced near-phone signal variations. Thanks to the ideal property of the ZC sequence, its auto-correlation is non-zero only at the point with zero delay, which can well approximate the Dirac's delta function. Given a sequence of down-converted received signal sampled by an N_{ZCI}-length window starting at the n_t-th point, we can calculate the cross-correlation $R[n_t]$ by:

$$\begin{aligned} R[n_t] &= \sum_{l=0}^{N_{ZCI}-1} \Big(\sum_{i=1}^{L} A_i e^{-j\theta_i} s_l[n_i]\Big) \times s[l] \\ &= \sum_{i=1}^{L} A_i e^{-j\theta_i} \Big(\sum_{l=0}^{N_{ZCI}-1} s_l[n_i] \times s[l]\Big) \end{aligned} \tag{3}$$

where $s_l[n_i] = s[(l - n_i) \bmod N_{ZCI}]$ is the transmitted signal circularly shifted by n_i samples. Since the right part of Eq. 3 is the cross-correlation term between the original ZC sequence and the delayed ones, the result can be obtained as:

$$R[n_t] = \begin{cases} N_{ZCI} A_i e^{-j\theta_i}, & n_i = 0 \\ 0, & n_i \neq 0 \end{cases} \tag{4}$$

where only the auto-correlated component (i.e., signals arrive at the n_t-th sample with zero delay) is reserved. Therefore, we can estimate the CIR $h[n]$ within each acoustic frame by sliding the moving window on the received signal and calculating the cross-correlation. Note that the 48 kHz sampling rate leads to the distance interval of $1/48000$ second × 340 m s^{-1} = 0.7 cm between adjacent channel taps, which is sufficiently small to capture the differences in hand biometrics of different users.

3.4 Informative CIR Extraction

The CIR contains not only the information of the moving hand but also other irrelevant information like nearby objects and the moving body. However, only channel variations caused by the moving hand are required for

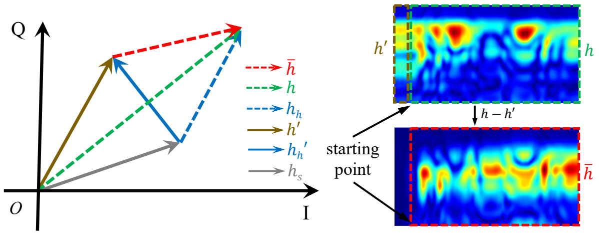


Fig. 4. The removal of static components.

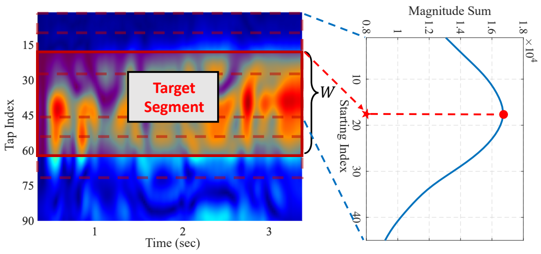


Fig. 5. Final hand-induced components.

authentication. To solve this problem, *SwipePass* first locates and estimates the near-phone CIR and then further extracts the hand-induced CIR (called informative CIR), which are detailed below.

3.4.1 Near-phone CIR Estimation. Fig. 3 shows the CIR magnitude of two consecutive acoustic frames in one unlock process. The higher peaks (in the red oval of Fig. 3) can be witnessed in the taps related to the reflections of the near-phone area (e.g., covering the area of hands, arms, and the smartphone itself), since these echoes travel less distance and the acoustic attenuation is not significant. The latter lower peaks (in the green oval of Fig. 3) denote signals reflected by the surroundings with a longer propagation distance and lower energy. The small rest fluctuations represent reflections from more distant objects. Since the unlocking behavior only affects the propagation channels close to the smartphone, our goal is to estimate the CIR of the near-phone taps.

To achieve this, a straightforward solution is to apply a threshold-based method for the CIR of each acoustic frame. Though effective, it is inefficient because 1024 points of multiplications and summations are required to estimate one CIR value (Eq. 3). Since the acoustic frame has a fixed length of 1024, we only need to locate the near-phone taps in one acoustic frame. Specifically, we randomly apply a window on the early received audio samples and estimate the CIR within this window. The window length is 1024, which is the same as the length of an acoustic frame. Under this setting, in most cases, the window will only contain near-phone reflections of one frame. We set a threshold that is half of the highest peak's value and search for the front-most sample reaching the threshold, which is set as the starting point. This case is denoted as case "A" in Fig. 3. However, there is another possibility: the window may contain near-phone reflections in two frames, which is denoted as case "B" in Fig. 3. In this case, two starting points will be detected, and we will set the rightmost one as the starting point. Then, we set the informative range, which contains the near-phone taps, to be the range between the 25 points before and 65 points after the starting point. Thereafter, the informative ranges in the later acoustic frames can be obtained by adding 1024 points delay to the first one because the acoustic frames have a fixed length of 1024. In practice, the computational consumption can be significantly reduced by only estimating the CIR of the near-phone taps within these ranges.

3.4.2 Hand-induced Component Extraction. The estimated near-phone CIR consists of two parts: the static components (e.g., the smartphone, the hand holding the smartphone, or the table the smartphone is on), and the dynamic components (e.g., the unlocking hand, the irrelevant displacement of arms and other body components). To obtain the informative hand-induced CIR components, we first calculate the difference of CIR to eliminate the static components and then further narrow the tap range in each acoustic frame so that only the taps corresponding to the unlocking hand are reserved.

As shown in Fig. 4, the near-phone CIR can be decomposed into vector h_s, which denotes the channel states affected by the static components, and vector h_h, which represents the time-varying dynamic components involving the unlocking hand. By analyzing people's common unlocking behaviors, we observe a transient time after the user first touches the screen but before swiping to unlock. We term the CIR estimated from this short period as the initial state (denoted as h'), during which the CIR values remain relatively stable. Given that h_s

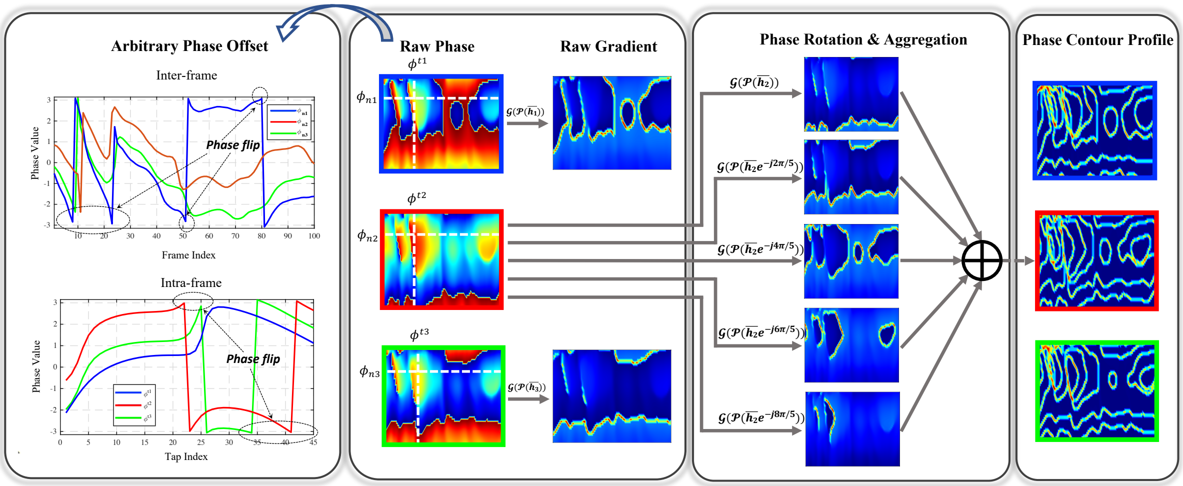

Arbitrary Phase Offset
Inter-frame
Phase Value
Frame Index
Phase flip
ϕ_{n1}
ϕ_{n2}
ϕ_{n3}
Intra-frame
Tap Index
ϕ^{t1}
ϕ^{t2}
ϕ^{t3}
Raw Phase
Raw Gradient
$\mathcal{G}(\mathcal{P}(\overline{h_1}))$
$\mathcal{G}(\mathcal{P}(\overline{h_3}))$
Phase Rotation & Aggregation
$\mathcal{G}(\mathcal{P}(\overline{h_2}))$
$\mathcal{G}(\mathcal{P}(\overline{h_2}e^{-j2\pi/5}))$
$\mathcal{G}(\mathcal{P}(\overline{h_2}e^{-j4\pi/5}))$
$\mathcal{G}(\mathcal{P}(\overline{h_2}e^{-j6\pi/5}))$
$\mathcal{G}(\mathcal{P}(\overline{h_2}e^{-j8\pi/5}))$
Phase Contour Profile

Fig. 6. Construction of phase contour profile. $\mathcal{P}(\cdot)$ and $\mathcal{G}(\cdot)$ denote phase extraction and gradient calculation.

can be regarded as unchanged during the whole unlocking process, we can further represent the initial state as $h^{'} = h_s + h^{'}_h$, where $h^{'}_h$ represents the CIR component of the hand touching the screen during this time. Thereafter, we can remove the static component and obtain the dynamic components (denoted as $\bar{h}$) by:

$$\bar{h} = h - h^{'} = (h_s + h_h) - (h_s + h^{'}_h) = h_h - h^{'}_h. \tag{5}$$

A previous study has shown that the biometrics of the holding hand (i.e., captured by $h^{'}_h$) with the same gesture is highly consistent for the same person [7]. Thus, measuring $\bar{h}$ is the same as measuring h_h. The actual starting point of swiping on the screen can be simply determined by a threshold-based method on the difference between the magnitude-sum of the adjacent CIR frames. If the normalized difference value exceeds the threshold (0.05 in this paper), the average of the CIR frames before that timestamp is adopted as h', and the latter frames constitute h. Thereafter, we can obtain the dynamic components from the residual between h and h', which is shown in the right part of Fig. 4.

To extract the hand-induced dynamic CIR components, we apply a moving summation window to tap sequence and select the segment with the highest value, as shown in Fig. 5. This is because the acoustic signals reflected by the close-phone unlocking hand travel shorter distances compared with those reflections from the other dynamic parts of the human body. Thus, the taps of interest should have higher CIR magnitudes. The window size is set to be 45 based on our preliminary experiments, which can effectively preserve the hand-induce portions while eliminating the other noisy taps. The selected segment is the final informative hand-induced component which will be used to extract discriminative features.

3.5 Feature Extraction

In this section, we present how to extract discriminative features from the hand-induced CIR component.

3.5.1 *Magnitude Profile.*

The magnitude of the CIR represents the strength of the acoustic signal received from different propagation paths. Intuitively, different hand sizes and moving behaviors can be reflected on distinguishable magnitude patterns. Thus, we use the magnitude of CIR as the first feature profile, which is named as *Magnitude Profile* (MP).

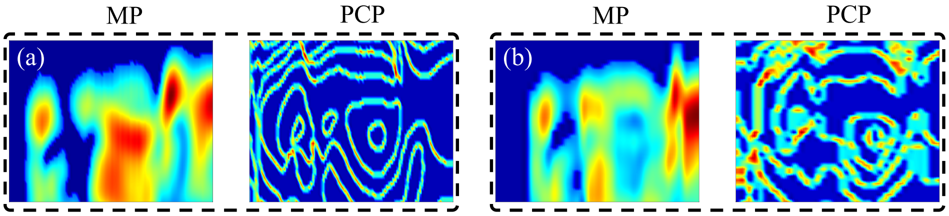


Fig. 7. The profiles of slow speed (a) and fast speed (b).

3.5.2 Phase Contour Profile. Besides the signal strength reflected by the magnitude, the phase of the CIR contains information related to the time delay of each acoustic channel. However, due to the asynchronous speaker and microphone, arbitrary phase offsets exist in the estimated CIR. Therefore, the absolute values of phase cannot be used directly. In *SwipePass*, we propose to use the relative changes of phase, which can be well represented by the gradient. The intuition of our method is that hand biometrics at a fixed time can be represented by the intra-frame difference, and the movement pattern over time can be characterized by the inter-frame variation. To enhance the significant changes and eliminate the noise, we define the phase gradient at the n-th tap and t-th frame as:

$$g_n^t = (\frac{\phi_n^{t+1} - \phi_n^{t-1}}{2})^2 + (\frac{\phi_{n+1}^t - \phi_{n-1}^t}{2})^2, \tag{6}$$

where the first and second terms account for the inter-frame and intra-frame difference, respectively.

However, phase flip—the phase value shifts circularly when it exceeds the limits $\pm\pi$—poses a challenge to feature extraction. Fig. 6 illustrates the impact of phase flip and our solution. The first block plots the raw inter-frame phase value (sliding horizontally at the same tap in the phase matrices, denoted as ϕ_{ni}) and the raw intra-frame phase value (sliding vertically at the same frame in the phase matrices, represented as ϕ^{ti}). The curves in three colors are based on the three phase samples in the second block of Fig. 6. It can be seen that the overall patterns of these three curves are similar. However, phase flip exists when the phase value approaches $\pm\pi$, making significant changes in adjacent phase values. Correspondingly, if we use the above phase gradient as a feature, the phase flip will cause clearly different patterns in the gradients, as shown in the second block.

To solve this problem, we shift the phase by N_r times with a fixed offset $2\pi/N_r$. In *SwipePass*, by setting $N_r = 5$, we can obtain 5 different phase gradient images, which are shown in the third block of Fig. 6. Then, to keep all the informative features, all the 5 phase images with different offsets are aggregated together to generate the final gradient image as shown in the fourth block of Fig. 6. We can see that although the three raw gradient images directly generated from the phase values are distinct, the three aggregated gradient images are highly consistent. The feature extraction process is similar to drawing contours of the phase value matrix, so we name it *Phase Contour Profile* (PCP).

After obtaining the MP and PCP, the two feature matrices are resized to be 45×100 so that they can be processed by the model with fixed-size input. Then the values are normalized into the range of $[0, 1]$, and small values below 0.2 are discarded.

3.6 Data Augmentation

To understand how the unlocking process is affected by different factors, we conduct a comprehensive study. Our preliminary study shows that *SwipePass* is immune to audible ambient noise and different volumes of the speaker since the system is working at a high-frequency band and uses normalized feature values. However, we find that the following two factors have a significant impact on the extracted features.

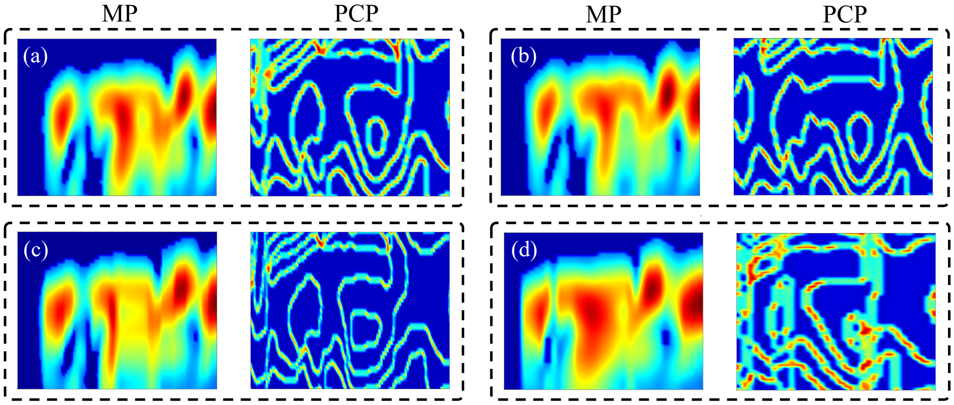


Fig. 8. (a) Raw profiles. After augmentation: (b) initial state variation; (c) slower speed; (d) faster speed.

3.6.1 Speed. To investigate the impact of the unlocking speed, we perform the unlock pattern at slow and fast speeds, respectively. From the extracted features (i.e., MP and PCP) in Fig. 7, we can see that the example at slow-speed has more smooth edge patterns in both profiles and clearer separation between contours in the PCP. This is because the fast unlocking can induce a larger inter-frame difference, the contours will be merged together and cause the aliasing effect.

With this in mind, we model the variation by modifying the number of acoustic frames in the extracted CIR since variations in the obtained frame number can effectively imitate the changes of speed when given a fixed frame rate. Specifically, given the CIR frames of one trial, we randomly pick a value r_s within $[-0.4, 0.4]$. If $r_s \geq 0$, each frame of the CIR will be dropped with probability r_s. Otherwise, each frame interval will be interpolated with a new frame in probability $-r_s$. The random setting allows us to imitate the real situations that users may unlock at incoherent speeds.

3.6.2 Initial State Variation. Since the hand-induced CIR $\bar{h}$ is obtained by subtracting the initial state h', the estimation of h' plays a crucial role. As defined in Sec. 3.4.2, the initial state means the state before the user starts swiping on the screen. There are two factors affecting the initial state CIR for different trials: (1) the holding hand phase of the same user may be slightly different; (2) the asynchronous microphone and speaker produce unpredictable phase offset. For the first factor, we find that the magnitude is not consistent with a small difference in the holding hand phase. To imitate these variations, we augment the raw sample by multiplying the h' with a factor F. To imitate the impact of the second factor, we randomly rotate the phase of $\bar{h}$. Technically, to imitate the possible variations in magnitude and phase, the augmented sample can be obtained by $\bar{h} = (h - F \cdot h')e^{-j2\pi r_o}$, where F and r_o are random values within $[0.9, 1.1]$ and $(0, 1)$.

Fig. 8 plots the feature profiles of different augmented samples. Compared with the raw signal (Fig. 8(a)), the speed-augmented samples (Fig. 8(c) and Fig. 8(d)) show different degrees of smoothness and aliasing, which are consistent with our observations. The augmented sample with initial state variation (Fig. 8(b)) contains different levels of variations in overall patterns. The evaluation results in Sec. 4.2 demonstrate that our augmentation methods can improve identification accuracy significantly with limited samples.

3.7 Authentication Model

After the feature profiles are extracted, they will be fed into an authentication model to identify whether the current user is a legitimate user or spoofer. Fig. 9 shows the architecture of the authentication model. We present the details below.

3.7.1 *CNN-LSTM Feature Extractor.*

First, the two feature profiles are fed into a CNN-LSTM neural network. Since the essence of feature profiles are images, we use CNN as the first part of the feature extractor because it performs well in extracting informative features from images. As shown in Fig. 9, three CNN layers are concatenated to capture features in different receptive fields with different semantic meanings. The first convolutional layer can only perceive simple low-level visual features, such as corners and edges of the profiles. These features are related to the partial geometries of the user's hand. The second convolutional layer extracts feature based on the output of the first layer, making it capable of capturing mid-layer features related to parts of the profiles, which are related to the hand gesture types. The third layer further aggregates the output of the former layers and can extract high-level features with semantic meaning of the whole hand movement patterns. The kernel sizes of the three layers are 5×5, 5×5, and 3×3, respectively. In addition to the spatial features, the profiles also contain important temporal features because they are extracted from time-series data. Therefore, an LSTM layer with five cells is used after the CNN to further extract the temporal features during the unlocking process.

To make sure the CNN-LSTM model can extract important informative features, we train it in an auto-encoder manner, which learns to map the network input to a compact representation in a latent space from which the data can be recovered with minimal information loss. Specifically, during training, three deconvolutional layers are deployed after the LSTM to reconstruct the two input feature profiles. Given the input profile X and the auto-encoder output X', our task is to minimize the reconstruction loss:

$$\mathcal{L}_{rec}(X, X') = \frac{1}{N} \sum_{n=1}^{N} (X_n - X'_n) + \lambda \times \Omega, \tag{7}$$

where N is the number of training samples and λ is the coefficient for the L_2 parameter regularization Ω. We denote the output of the CNN-LSTM extractor as O.

3.7.2 *User Identification and Spoofer Detection.*

Without loss of generality, we assume there are K legitimate users for a smart device. It should be noted that K can be larger than one. This is because there may be multiple legitimate users for the same smart device. For example, the members of the same family may share one tablet. In this situation, the authentication system not only needs to recognize whether the current user is a legitimate user or an attacker, but also has the capability to identify the user identity so that a number of personalized services can be enabled. To provide a generic solution to satisfy this requirement, our system firstly identifies the user's identity from multiple registered users, then further performs detection to resist adversarial login from unexpected spoofers.

In *SwipePass*, we adopt a two-layer design. The first layer is a multi-class classifier that recognizes the current user to one of the K enrolled users. The second layer is a One-Class Support Vector Machine (OC-SVM) spoofer detector, which checks whether the current user is indeed the legitimate user or a spoofer. To identify different users, we append a fully-connected layer with soft-max activation function to the end of the previous network. The output is the posterior probability of each class $P(U_k|O)$, where O is the output of the CNN-LSTM model, U_k is the class label, and $k = 1, 2, ..., K$ is the index of K legitimate users. The identification result will be the user with the maximum posterior probability. To train this model, the loss function is modified as:

$$\mathcal{L}(X, X', U) = (1 - \alpha)\mathcal{L}_{rec}(X, X') + \alpha \mathcal{L}_{CE}(P(X), U) \tag{8}$$

where $\mathcal{L}_{CE}(P(X), U) = \sum_{i=1}^{N} -P(X_i)_{U_i} \log U_i$ is the cross-entropy loss for the posterior probability $P(X)$ and user label U, and $\alpha \in (0, 1)$ is a parameter to balance the two loss terms. Note that for the situation there is only one legitimate user, the loss function degrades to Eq. 7.

Based on the classification result of the first layer classifier, the authentication result will be made based on the similarity between the incoming sample and the registered legitimate user's data. An OC-SVM with RBF kernel is trained to construct the authentication model for each of the legitimate user. Equipped with kernel functions,

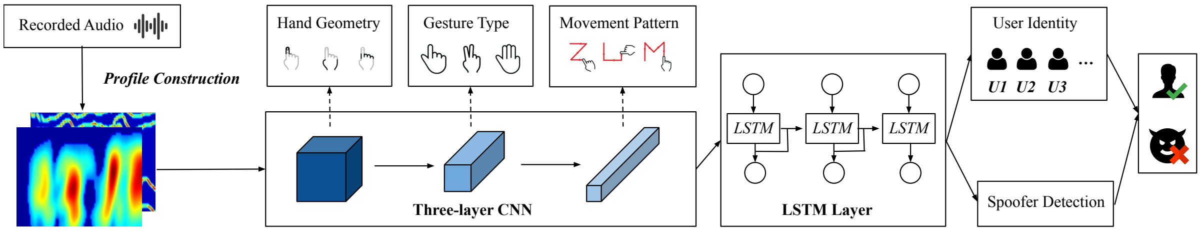


Fig. 9. The architecture of the authentication model.

the OC-SVM can learn a hyperplane which holds the legitimate samples on one side and maximizes the margin to the origin in a higher-dimensional space. This hyperplane, which is obtained based on the legitimate user's data only, can serve as a decision boundary to separate normal samples and anomalies (e.g., legitimate users and spoofers). In practice, the well-trained OC-SVM can output a similarity score between the testing data and the legitimate user's data, and a threshold can be empirically set for spoofer detection. The threshold is a user-specific parameter, so it is not fixed in *SwipePass*. Compared to multiple classification solutions [22], our method is more realistic because we cannot assume the availability of attackers' data in practical applications. To this end, in *SwipePass*, a person is allowed to log in only after they perform the correct unlock pattern, has been identified correctly as one of the legitimate users and passes the spoofer detection. Otherwise, the login will be denied.

4 EVALUATION

4.1 Experimental Setup

4.1.1 Implementation. We implement *SwipePass*[1] as a mobile application on three commercial Android smartphones, i.e., Samsung S10 (2.84 GHz CPU, 8 GB RAM, Android 11), Nexus 6P (1.95 GHz CPU, 3 GB RAM, Android 8) and Redmi Note 10 Pro (2.3 GHz CPU, 6 GB RAM, Android 11). The microphone and speaker on each smartphone vary in size and position, as shown in Fig. 10. It should be noted that some phones have two pairs of microphones and speakers while others have one pair only. To make our solution general, we only use one pair of microphone and speaker in our evaluation. The use of two pairs of microphone and speaker may increase authentication accuracy but will not be discussed in this paper due to space limitations. The deep-learning identification model and the OC-SVM spoofer detectors are implemented in PyTorch [19] and scikit-learn [20]. These models are first trained offline on a desktop PC with Intel i7-9700 CPU, 64 GB RAM, and RTX 2080 Ti GPU, then deployed on the smartphones.

4.1.2 Attack Scenarios. We consider the following three attack scenarios:

- *Credential-aware attack:* The attacker obtains the credential (i.e., the pattern) from some side channels such as the smudge on the screen or a quick glance and knows nothing about the unlocking hand gesture or moving patterns of the legitimate user. Therefore, the attacker can only try to unlock the smartphone in their preferred ways.
- *Mimicry attack:* The attacker observes the whole process when the user inputs the pattern through shoulder-surfing or secretly filmed videos, which means genuine unlocking behavior of the legitimate user is known. Then, the attacker can try to unlock the smartphone by mimicking the user's unlocking behavior.
- *Replay attack:* The attacker knows the credential and secretly places another microphone near the user to record the emitted audio signals when the user unlocks the phone. Then, the attacker unlocks the smartphone and plays back the recorded audio to the authentication system at the same time to spoof it.

[1] A video demo is available at: https://youtu.be/wom0x8u9J2c

Table 2. Summary of participants. Mean values are presented in parentheses.

Properties	Gender		Background		Hand Size (cm)		Age		
	Male	Female	Academia	Other	Width	Length	20-25	26-40	41-59
Details	25	11	30	6	8.9-11.6 (9.8)	14.4-18.5 (16.2)	17	11	8

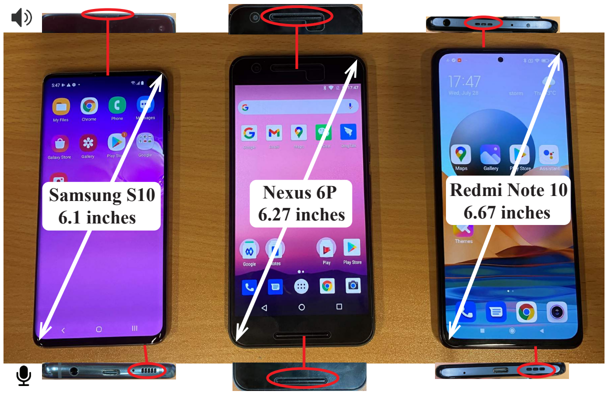


Fig. 10. Experimental devices.

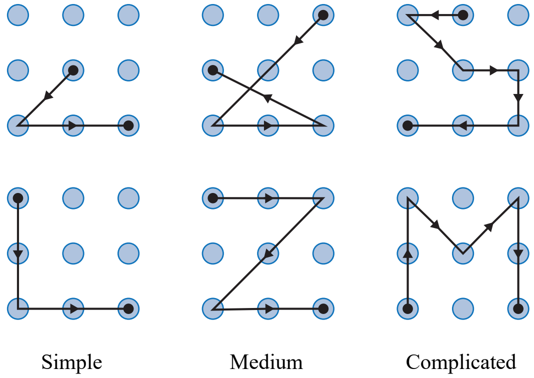


Fig. 11. Six patterns used in the experiment.

4.1.3 Data Collection. To evaluate the performance of *SwipePass*, we invite 36 volunteers [2] with a variety of age and hand size. The demography of the participants is shown in Table 2. All the participants are experienced smartphone users and know how to use *SwipePass*. Before conducting the experiment, the users signed the consent forms, which clearly state the purpose, procedure, and data usage of the study. For example, they are aware that the experiments would not cause any harm to their health, and the recorded data would only be used for research purposes and would not be leaked to any other third parties. Each participant was given a small gift (e.g., a notebook with a nice cover) for compensation. As shown in Fig. 11, we consider six different patterns for a 3 × 3 grid pattern lock scheme, which are grouped into three categories: simple, medium and complicated. We choose these patterns for two reasons. First, they are among the commonly used patterns according to a recent study [47]. Second, they represent different levels of complexity in terms of the number of lines and corners. As shown in Fig. 11, the simple category represents the low complexity with only two lines and three corners, while medium and complicated categories contains patterns with relatively higher complexities with more lines and corners. For simplicity, we use **S**, **M** and **C** to represent the simple, medium, and complicated categories.

The participants are asked to perform these three patterns in their preferable manners. Each participant performs the same pattern 27 times, of which the first 10 samples are used for training, and the remaining samples are used for testing. Compared to random splitting, this setting is more realistic because it is consistent with the authentication practice on smartphones where training data used in registration are collected before the test data. To understand the robustness of *SwipePass* in different real-world scenarios, we collect data in three real-life environments–an office, a public co-working space, and a busy street. These environments represent common scenarios in indoor and outdoor, quiet place, and noisy place. In addition, users may have different postures when using their smartphones. Therefore, we consider four common postures in data collection. The

[2] Ethical approval has been obtained (No. H002554).

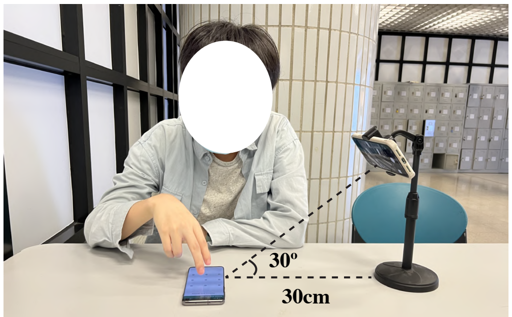


Fig. 12. The video filming setting for mimicry attack.

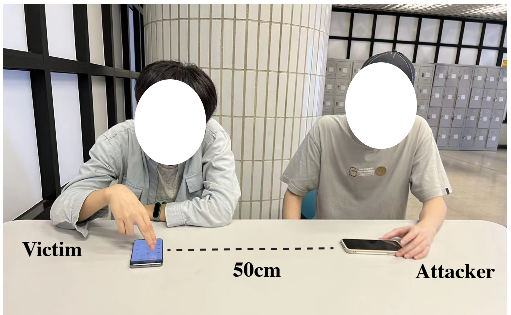


Fig. 13. The audio recording setting for replay attack.

first one is a static scenario that a user places his/her smartphone on a table and unlocks it with a finger. In the other three scenarios, a user holds his/her smartphone with one hand and swipes to unlock it with another hand while sitting, standing, and walking. We choose these settings because they are representative environments and postures in the literature [16, 27, 45]. The data is collected every two days with a time span of seven days to account for the changes in user behaviors.

To evaluate the performance of *SwipePass* against different attacks (Sec. 3.1), we conduct the following experiments. Among the participants, 26 of them are randomly selected as legitimate users, and the remaining participants are treated as attackers. Since the credential-aware attacker only knows the pattern itself, the data of the 10 attackers are used to spoof the authentication model directly. To evaluate the mimicry attack, during data collection, we use a device to film the unlocking processes of the 26 legitimate users. To evaluate the shoulder-surfing attack, as shown in Fig. 12, we place another smartphone 30 cm away beside the victim with an angle of depression at around 30° to ensure that the whole unlocking process is well-recorded by clearly showing the used gesture and graphic pattern. Then, the 10 attackers are asked to watch the videos of all the legitimate users and try to mimic their unlocking behaviors. In these experiments, each attacker is given 10 trials to attack each of the legitimate users. To evaluate the replay attack, the attacker handholds another smartphone to record the audio emitted from the victim's smartphone from different directions and distances when the user is unlocking his/her smartphone. An example is shown in Fig. 13, where the attacker records the audio 50 cm away at the left side of the victim. The recorded audios are then replayed to the victim's smartphone to spoof our system. The evaluation of these attacking experiments lasts for one month to consider the changes of user's unlocking behaviors and attacker's learning ability.

4.1.4 Metrics and Methodology. In the evaluation, the training data mentioned above are augmented to construct a much larger dataset by the proposed methods in Sec. 3.6. As an authentication system, we consider the following three metrics that are widely used in previous studies [7, 26]:

- *Identification Accuracy*: The probability that the identity of the user is correctly classified by our system.
- *False Accept Rate (FAR)*: The probability that a spoofer is authenticated as a legitimate user.
- *False Reject Rate (FRR)*: The probability that a legitimate user is authenticated as a spoofer.

4.2 Impact of Data Augmentation

In this experiment, we evaluate whether the proposed data augmentation methods can improve the accuracy with limited training samples. We compare the accuracy of four settings: (1) without augmentation, (2) augmentation with initial state variation only, (3) augmentation with speed variation only, (4) augmentation with both variations. The results are shown in Fig.14 (a). We observe that the accuracy of setting (1) is the worst, which indicates

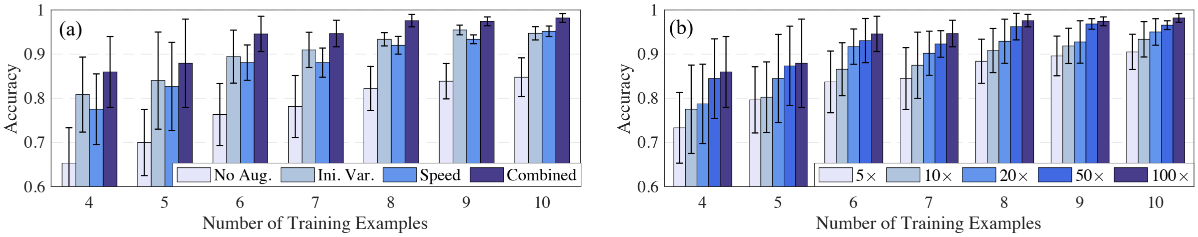


Fig. 14. Evaluation results: impact of different (a) augmentation methods and (b) augmentation rates.

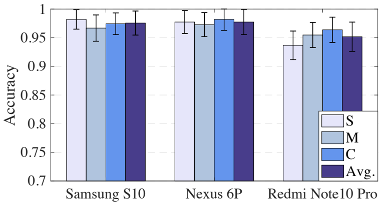


Fig. 15. Different devices.

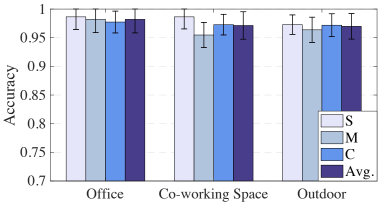


Fig. 16. Different environments.

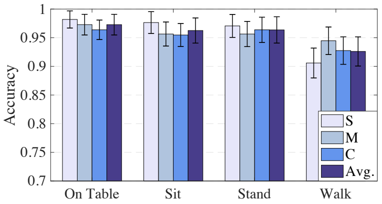


Fig. 17. Different postures.

the poor performance of the model trained directly on the limited samples. In terms of the two independent augmentation methods, the accuracy of setting (2) is slightly higher than that of setting (3). This is because the same user has relatively stable speed, but unstable initial states in different unlock trials. Not surprisingly, by using both augmentation methods, we can achieve the best accuracy. The result illustrates that the proposed two augmentation methods can improve the model performance by generating more training data that are close to real-world variations.

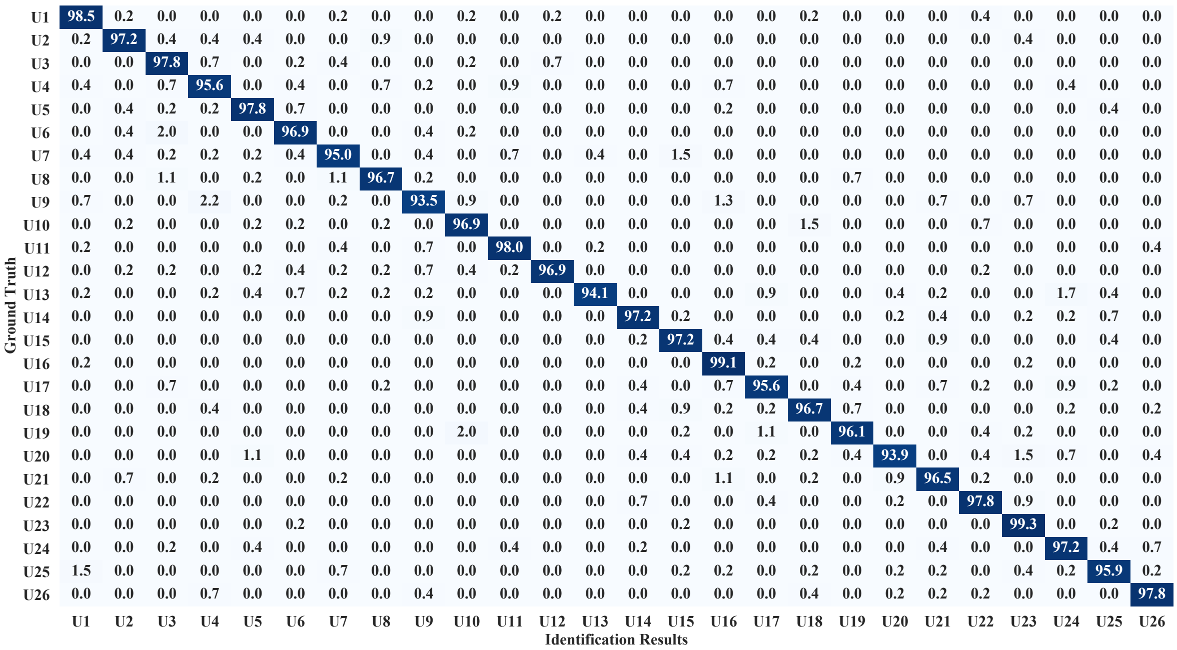
Next, we investigate the influence of different augmentation rates to the performance of our system. From the results in Fig.14 (b), we observe that when only four training examples are used, the accuracy with 5× augmentation rate achieves less than 75% only. However, the accuracy improves to 85% when the augmentation rate reaches 100×. If six samples per user are used, *SwipePass* achieves approximately 95% accuracy with 100× augmentation rate. The accuracy is further improved up to 96% when more samples are used, but we find that the improvement of accuracy diminishes when more than eight training samples are used. Based on the above results, we draw the conclusion that the proposed augmentation methods can significantly improve the accuracy of *SwipePass* in a data-efficient manner, greatly reducing the tedious and repetitive data collection process.

4.3 User Identification Performance

In this subsection, we evaluate the identification accuracy of *SwipePass* under different conditions (e.g., different smartphones, environments, postures, and unlock patterns).

Fig. 15 shows the accuracy using different smartphones. We observe that Samsung and Nexus achieve over 97% accuracy on average, while the accuracy of Redmi is slightly lower. To find out the reason, we compare the extracted near-phone CIRs of the three smartphones and find the variations of Redmi's are the least observable. This means, if we compare the signals bypassing the hand-interaction area with the LOS signal, it is the weakest in Redmi among the three smartphones. This can be explained by the different relative locations of the microphone and speaker (Fig. 10), that the opposite-side layout of Redmi results in weaker signals traveling above the screen. Nevertheless, *SwipePass* still achieves an accuracy of 95% approximately on Redmi.

Fig. 16 shows the accuracy in different environments. We observe that the accuracy in the noisy street is slightly lower than that in the indoor office and co-working place due to a relatively higher noise level. Nevertheless,



	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18	U19	U20	U21	U22	U23	U24	U25	U26
U1	98.5	0.2	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0
U2	0.2	97.2	0.4	0.4	0.4	0.0	0.0	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0
U3	0.0	0.0	97.8	0.7	0.0	0.2	0.4	0.0	0.0	0.2	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
U4	0.4	0.0	0.7	95.6	0.0	0.4	0.0	0.7	0.2	0.0	0.9	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0
U5	0.0	0.4	0.2	0.2	97.8	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0
U6	0.0	0.4	2.0	0.0	0.0	96.9	0.0	0.0	0.4	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
U7	0.4	0.4	0.2	0.2	0.2	0.4	95.0	0.0	0.4	0.0	0.7	0.0	0.4	0.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
U8	0.0	0.0	1.1	0.0	0.2	0.0	1.1	96.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
U9	0.7	0.0	0.0	2.2	0.0	0.0	0.2	0.0	93.5	0.9	0.0	0.0	0.0	0.0	0.0	1.3	0.0	0.0	0.0	0.0	0.7	0.0	0.7	0.0	0.0	0.0
U10	0.0	0.2	0.0	0.0	0.2	0.2	0.0	0.2	0.0	96.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0
U11	0.2	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.7	0.0	98.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4
U12	0.0	0.2	0.2	0.0	0.2	0.4	0.2	0.2	0.7	0.4	0.2	96.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0
U13	0.2	0.0	0.0	0.2	0.4	0.7	0.2	0.2	0.2	0.0	0.0	0.0	94.1	0.0	0.0	0.0	0.9	0.0	0.0	0.4	0.2	0.0	0.0	1.7	0.4	0.0
U14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0	0.0	0.0	0.0	97.2	0.2	0.0	0.0	0.0	0.0	0.2	0.4	0.0	0.2	0.2	0.7	0.0
U15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	97.2	0.4	0.4	0.4	0.0	0.0	0.9	0.0	0.0	0.0	0.4	0.0
U16	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.1	0.2	0.0	0.2	0.0	0.0	0.0	0.2	0.0	0.0	0.0
U17	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.7	95.6	0.0	0.4	0.0	0.7	0.2	0.0	0.9	0.2	0.0
U18	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.9	0.2	0.2	96.7	0.7	0.0	0.0	0.0	0.0	0.2	0.0	0.2
U19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.2	0.0	1.1	0.0	96.1	0.0	0.0	0.4	0.2	0.0	0.0	0.0
U20	0.0	0.0	0.0	0.0	1.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.4	0.2	0.2	0.2	0.4	93.9	0.0	0.4	1.5	0.7	0.0	0.4
U21	0.0	0.7	0.0	0.2	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.1	0.0	0.2	0.0	0.9	96.5	0.2	0.0	0.0	0.0	0.0
U22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.4	0.0	0.0	0.2	0.0	97.8	0.9	0.0	0.0	0.0
U23	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.3	0.0	0.2	0.0
U24	0.0	0.0	0.2	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	97.2	0.4	0.7
U25	1.5	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.2	0.0	0.2	0.0	0.2	0.2	0.0	0.4	0.2	95.9	0.2
U26	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.2	0.2	0.2	0.0	0.0	0.0	97.8

Fig. 18. Confusion matrix (%).

SwipePass achieves over 95% identification accuracy in all the environments. The results demonstrate that *SwipePass* can effectively extract the CIR caused by the moving hand in different real-world environments and is immune to most of the audible environment noise.

Fig. 17 shows the accuracy of different postures, from which we can see that the average accuracy of the three static cases (i.e., on table, sitting, and standing) are all over 95%. As for walking, although the irregular waggling of hands causes accuracy decline, *SwipePass* still achieves 92.6% on average, which demonstrates that our proposed hand-induced CIR extraction can effectively eliminate interference from other dynamic components such as the torso. In addition, there is no obvious difference among different unlock patterns, which means *SwipePass* can strengthen the security of pattern lock scheme regardless of the complexity of unlocking patterns. The overall performance of *SwipePass* in identifying legitimate users is shown in Fig. 18 with 96.8% accuracy on average.

4.4 User Authentication Performance

The detailed replay attack scenarios are shown in Fig. 19, where the attacker places another smartphone to record the audio emitted by the victim's smartphone from different directions (i.e., front, left, right and back) and distances (i.e., 0.5 m, 1 m and 2 m). 10 samples are recorded at each position and are replayed back to the authentication system to spoof it. As shown in Fig. 20, the average FAR of the replay attack is below 0.1%, showing that the system is secure enough to be resistant to this type of attack. This is because the audio recorded by the attacker experience completely different multi-path propagation compared with the legitimate one. Fig. 21 and Fig. 22 show the FAR of *SwipePass* against the credential-aware attack and mimicry attack, respectively. The average FARs of these two attacks are pretty close to each other: it is 0.44% for the credential-aware attack and 0.38% for the mimicry attack. If we convert the FAR to success rate, it requires approximately 230–1000 attempts

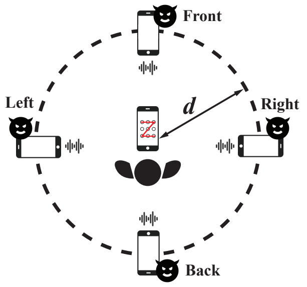


Fig. 19. Replay attack setting.

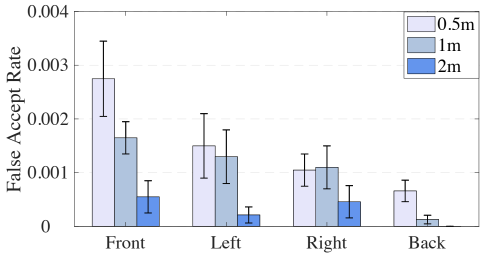


Fig. 20. Replay attack.

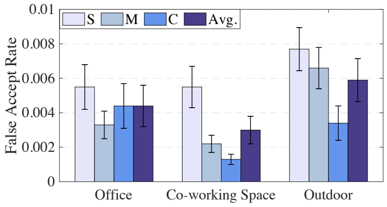


Fig. 21. Credential-aware attack.

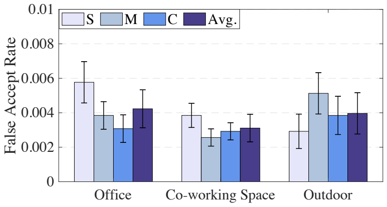


Fig. 22. Mimicry attack.

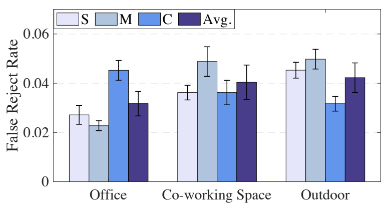


Fig. 23. False reject rate.

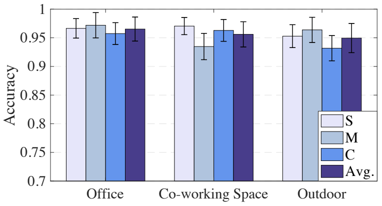


Fig. 24. Unseen environment.

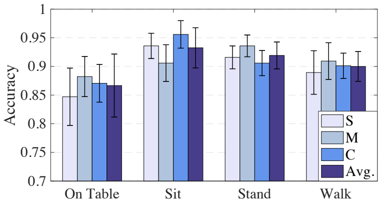


Fig. 25. Unseen posture.

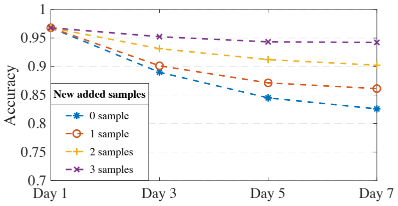


Fig. 26. Unseen time.

to successfully spoof our authentication system. Since modern smartphones usually trigger screen lock after several wrong attempts, our system is secure enough to resist these common attacks.

For a legitimate user, FRR is an important metric because it is related to user experience. From the results in Fig. 23, the FRR of *SwipePass* is 3.8% on average, which means out of 100 attempts, the legitimate user will be incorrectly rejected by approximately 3.8 times. In fact, there is a trade-off between FAR and FRR by setting the threshold in the OC-SVM spoofer detector. A higher level of security can be achieved by sacrificing the user experience. We consider the trade-off as a user-defined parameter to meet the requirement of different applications. From the above experiments, we observe that *SwipePass* can provide comprehensive protection to the smartphone regardless of the patterns used, which alleviates the dependency of the security level on the pattern complexity. In other words, by loosing the restrictions on the way how users choose patterns, both the generalization ability and user experience of the system is enhanced.

4.5 Generalization Ability Analysis

4.5.1 Unseen Environment. We conduct leave-one-environment-out validation to evaluate the accuracy of *SwipePass* in unseen environments. Specifically, the data collected in one environment is used to test the model

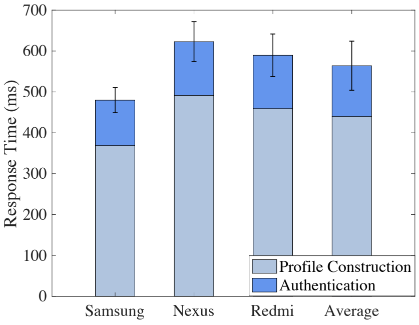


Fig. 27. Response time.

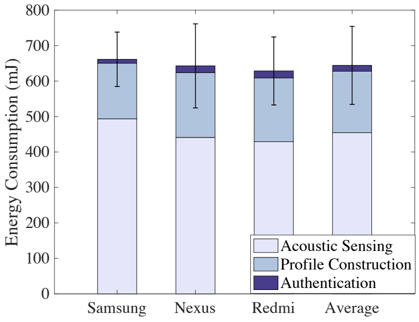


Fig. 28. Energy consumption.

trained in other environments. As shown in Fig. 24, the classification accuracy maintains at the similar level compared with the result in Fig. 16, where the data of the same environment is used in both training and testing. This is because *SwipePass* is inherently resilient to ambient noise.

4.5.2 Unseen Posture. We conduct leave-one-posture-out validation to evaluate the accuracy of *SwipePass* in unseen postures. The results are plotted in Fig. 25. Compared with the result in Fig. 17, the accuracy only drops by 3%–4% in sitting, standing, and walking scenarios but drops by around 10% when the smartphone is placed on the table. The result can be explained by the fact that biometric patterns of the two-hand involved unlocking scenarios are different from those of using only one hand, which is caused by the differences in the relative leaning angles and subconscious movements between the phone and the hand. To solve this issue, we may apply deep learning such as few-shot learning and domain adaptation, which we leave for our future work.

4.5.3 Unseen Time Period. The unlocking behavior of the same user may change gradually over time. Therefore, we investigate the accuracy of *SwipePass* by training and testing the system using data from different days. Specifically, we use the data collected on Day 1 to train the model, and then we use the data collected on Day 3/5/7 to test the model. As shown in Fig. 26 (blue dash line), the accuracy decreases gradually over time. To solve this problem, inspired by incremental learning, we fine-tune the model by using a few new samples collected on the testing day. Specifically, combining with the newly added samples, we fix the CNN layers and fine-tune the LSTM and FC parameters with 10 epochs before testing. The results of adding a different number of samples are plotted in Fig. 26, from which we observe that the accuracy improves significantly even when only three new samples are added. In practice, *SwipePass* can add new samples to update the model after the user is successfully authenticated, which guarantees the robustness of our system even when the user's unlocking behavior changes.

4.6 Response Time and Energy Consumption

We evaluate the response time and energy consumption of *SwipePass* on the three smartphones. The response time is obtained from the Android Studio console and the energy consumption is calculated based on Android's built-in API [10]. The averaged results after 100 runs are shown in Fig. 27 and 28. From Fig. 27, we find that the time consumption mainly falls in the profile construction which includes the segmentation and feature extraction. This is because the CIR estimation in segmentation is the most computationally intensive process. As for the devices, Samsung takes the least time to finish one authentication in 479 ms, while the Nexus takes the most time in about 622 ms. The difference can be explained by their different processing ability. The average response time is 564 ms, meaning that *SwipePass* will not cause significant delay.

Fig. 28 shows the energy consumption of each component in *SwipePass*. The acoustic sensing consumes the most energy since it takes more time compared with the processing stages. It is not surprising that the

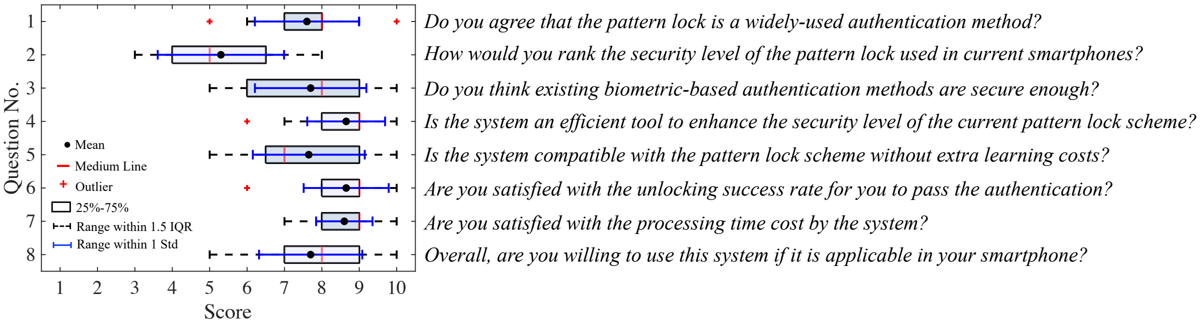


Fig. 29. Results of user study questionnaire.

authentication with a lightweight model costs the least energy, which is only 16.6 mJ on average. Generally, the three smartphones consume similar energy at around 644.3 mJ, which implies that one-time unlocking with *SwipePass* only requires less than 0.013‰ of the smartphone batteries. The experiments show that *SwipePass* is both computation and energy efficient.

4.7 User Study

In addition to the above experiments, we also conduct a user study by inviting 40 participants to fill out a questionnaire on their experience after using *SwipePass* for one week. The participants come from different industries (e.g., students, office staff, and drivers), and their ages span from 22 to 52. We would provide them with our testing smartphones if they do not have an Android phone or do not want to install extra software on their smartphones. Their data is collected on the first day, and the trained *SwipePass* is then deployed on the smartphone. In the following days, they are required to use *SwipePass* for authentication at least ten times per day. After one week's use, they are invited to complete a questionnaire. Since the goal of the user study is not only to understand the user experience but also to evaluate the performance of *SwipePass* in the wild, we design eight system-specific questions instead of using general standard questionnaires such as the User Experience Questionnaire [12]. The questions and the results are shown in Fig. 29. For each question, the answer is a score number taken from 1 to 10, based on the level of agreement, satisfaction, or preference. For the opinions on the current pattern lock on smartphones (Q1&Q2), most participants agree that the pattern lock is still being widely-used although it suffers from some potential security issues. One participant gives a *neutral* answer to Q1, with the reason that most smartphones nowadays provide various more convenient and safer solutions to smartphone authentication, such as fingerprinting. However, when it comes to the question (Q3) with respect to the biometric-based authentication methods, participants give a wide range of responses. Although the main portion of answers fall in the *agree* section (median at 8 and mean at 7.7), there are over 25% of the participants choose *neutral* or *weakly agree* options (5-7). As for the reason, some of the participants mentioned there is news reporting data leakage on smartphones due to incorrect authentication, and others have expertise in smartphone security-related knowledge and are aware of the possible vulnerabilities in the commonly-used technologies.

When it comes to *SwipePass*, the participants response positively to the capability of the system. In Q4 and Q6, a large majority of participants agree that the system can enhance the security of pattern lock efficiently and the success rate of the authentication is satisfactory. On average, the failure rate in the user study is about 10%. Based on the feedback from the participants, a large majority of failing cases occurred when the user is in a mobile environment, such as unlocking in a moving car/bus or when the user is walking. They think that compared with other mature commercial biometric-based methods, the system's robustness is still immature, and it should be further improved. In Q7, all participants are satisfied with the time consumed by *SwipePass*. As for Q5, over half of the participants agree that the system is easy to use without extra learning costs. However, for the rest of participants with neutral opinions, they mention that the requirement of unlocking the phone following the same hand gesture could sometimes be troublesome. This problem, that how to alleviate the requirement of the consistent unlocking gesture while not affecting system performance, is common in the community and is worthwhile to be further explored in the future. In the last question, most of the participants agree that it is a good choice to deploy *SwipePass* in their smartphones. participants with different opinions mainly consider pattern lock as a supplementary authentication method when using less security-sensitive applications. Therefore, they think that it is unnecessary to install *SwipePass*.

Overall, a large portion of the participants have acknowledged the potential security issues in the current authentication methods. *SwipePass* provides them with a new perspective of combining secure biometric-based method with pattern lock and they have shown great interest. Although there is still much room for improvement, we believe that *SwipePass* pushes the limits of existing authentication methods and moves a big step forward in mobile device security.

5 LIMITATIONS AND FUTURE WORK

In this paper, we propose a novel acoustic-based second-factor authentication system based on pattern lock. Although the above experiments demonstrate the superior performance of *SwipePass* in different real-world scenarios, there still exist several limitations, which we will try to tackle in future work:

- **Robustness in the long term:** The principle of *SwipePass* is that the locking behaviors of different users are distinct, such that the system can distinguish among people from the collected sensory data. In *SwipePass*, the unlocking gesture, hand movement pattern, and hand geometry together construct a unique and hard-to-copy profile for each user, making it secure from adversarial attackers. However, recent studies [40, 41] reveal that people's behaviors may change slightly over time, posing a significant challenge for our system. Therefore, how to enable tolerance for slightly different unlocking styles while maintaining secure authentication is the key to improving the practicality of the system. In the future, we will try to adopt the knowledge of deep learning, such as domain adaption and lifelong learning, to make the unlocking style in the enrollment phase transferable to other possible situations.
- **Multiple swipes in enrollment.** Although *SwipePass* is data-efficient compared with the previous work by using the proposed data augmentation techniques, it still needs the user to swipe eight to ten times to achieve the best performance. In contrast, only one swipe is required for the traditional pattern lock to set up the password. In future work, we will study how to reduce the number of registrations by using recently developed deep learning techniques, such as pre-training and few-shot learning.
- **Low generalizability across smartphones.** In this paper, we implement *SwipePass* on multiple smartphones and show that our system can provide reliable authentication on these devices. However, the current system is device-specific, which means a well-trained model on one smartphone cannot be generalized to other smartphones. This is because different devices have different sizes and microphone/speaker locations, as well as varying hardware standards. Therefore, the acoustic features of the same user may vary when using different devices. In large-scale deployment, it is impractical to collect a large amount of data for every mobile device. One possible solution is to normalize the sensory data to a unified scale and extract more robust features such that the gaps between different devices can be minimized. We can develop a more general model that can be easily deployed on different devices once the profiles of a person are equal across different devices.

6 CONCLUSION

In this paper, we propose a second-factor authentication system to augment the security of pattern lock, which is named *SwipePass*. The system uses inaudible acoustic signals to sense users' unlock behaviors and adopts several novel methods to extract unique features for different users. Additionally, we propose two augmentation methods to reduce the extensive data collection efforts to train a deep-learning model. The extensive evaluation demonstrates that *SwipePass* is capable of providing reliable authentication while remaining computation and energy-efficient.

ACKNOWLEDGMENTS

The work described in this paper was substantially sponsored by the project 62101471 supported by NSFC and was partially supported by the Shenzhen Research Institute, City University of Hong Kong. The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 21201420). The work described in this paper was partially supported by Shenzhen Science and Technology Funding Fundamental Research Program (Project No. 2021Szvup126), NSF of Shandong Province (Project No. ZR2021LZH010), Changsha International and Regional Science and Technology Cooperation Program (Project No. kh2201023), and a grant from Chow Sang Sang Group Research Fund sponsored by Chow Sang Sang Holdings International Limited (Project No. 9229062). The work was also partially supported by CityU MFPRC grant 9680333, CityU SIRG grant 7020057, CityU APRC grant 9610485, CityU ARG grant 9667225 and CityU SRG-Fd grant 7005666.

REFERENCES

[1] 2014. How to Fool a Fingerprint Security System As Easy As ABC. https://www.instructables.com/How-To-Fool-a-Fingerprint-Security-System-As-Easy-/.

[2] 2017. Galaxy S8 face recognition already defeated with a simple picture. https://arstechnica.com/gadgets/2017/03/video-shows-galaxy-s8-face-recognition-can-be-defeated-with-a-picture/.

[3] Alibaba. 2022. Alipay. https://intl.alipay.com.

[4] Julio Angulo and Erik Wästlund. 2011. Exploring touch-screen biometrics for user identification on smart phones. In *Privacy and Identity Management for Life*. Springer, 130–143.

[5] Adam J Aviv, Katherine L Gibson, Evan Mossop, Matt Blaze, and Jonathan M Smith. 2010. Smudge attacks on smartphone touch screens. *Woot* 10 (2010), 1–7.

[6] Chao Cai, Rong Zheng, and Jun Luo. 2022. Ubiquitous Acoustic Sensing on Commodity IoT Devices: A Survey. *IEEE Communications Surveys & Tutorials* (2022).

[7] Huijie Chen, Fan Li, Wan Du, Song Yang, Matthew Conn, and Yu Wang. 2020. Listen to Your Fingers: User Authentication Based on Geometry Biometrics of Touch Gesture. *ACM IMWUT* 4, 3 (2020), 1–23.

[8] Huangxun Chen, Wei Wang, Jin Zhang, and Qian Zhang. 2019. Echoface: Acoustic sensor-based media attack detection for face authentication. *IEEE IoTJ* 7, 3 (2019), 2152–2159.

[9] Xiaoran Fan, Longfei Shangguan, Siddharth Rupavatharam, Yanyong Zhang, Jie Xiong, Yunfei Ma, and Richard Howard. 2021. HeadFi: bringing intelligence to all headphones. In *ACM MobiCom*. 147–159.

[10] Google. 2022. Android Developers. https://developer.android.com/.

[11] Rosa González Hautamäki, Tomi Kinnunen, Ville Hautamäki, Timo Leino, and Anne-Maria Laukkanen. 2013. I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry.. In *Interspeech*. 930–934.

[12] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and usability engineering group*. Springer, 63–76.

[13] Dong Li, Jialin Liu, Sunghoon Ivan Lee, and Jie Xiong. 2020. FM-track: pushing the limits of contactless multi-target tracking using acoustic signals. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 150–163.

[14] Chao-Liang Liu, Cheng-Jung Tsai, Ting-Yi Chang, Wang-Jui Tsai, and Po-Kai Zhong. 2015. Implementing multiple biometric features for a recall-based graphical keystroke dynamics authentication system on a smart phone. *Journal of Network and Computer Applications* 53 (2015), 128–139.

[15] Chris Xiaoxuan Lu, Bowen Du, Hongkai Wen, Sen Wang, Andrew Markham, Ivan Martinovic, Yiran Shen, and Niki Trigoni. 2018. Snoopy: Sniffing your smartwatch passwords via deep sequence learning. *ACM IMWUT* 1, 4 (2018), 1–29.

[16] Li Lu, Jiadi Yu, Yingying Chen, Hongbo Liu, Yanmin Zhu, Yunfei Liu, and Minglu Li. 2018. Lippass: Lip reading-based user authentication on smartphones leveraging acoustic signals. In *IEEE INFOCOM*. IEEE, 1466–1474.

[17] Li Lu, Jiadi Yu, Yingying Chen, and Yan Wang. 2020. Vocallock: Sensing vocal tract for passphrase-independent user authentication leveraging acoustic signals on smartphones. *ACM IMWUT* 4, 2 (2020), 1–24.

[18] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1515–1525.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[21] Google Play. 2022. AppLock. https://play.google.com/store/apps/details?id=com.sp.protector.free&hl=en&gl=US.

[22] Aditya Singh Rathore, Weijin Zhu, Afee Daiyan, Chenhan Xu, Kun Wang, Feng Lin, Kui Ren, and Wenyao Xu. 2020. Sonicprint: a generally adoptable and secure fingerprint biometrics in smart devices. In *ACM MobiSys*. 121–134.

[23] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 2016. AudioGest: enabling fine-grained hand gesture detection by decoding echo signal. In *ACM UbiComp*. 474–485.

[24] Raul Sanchez-Reillo, Carmen Sanchez-Avila, and Ana Gonzalez-Marcos. 2000. Biometric identification through hand geometry measurements. *IEEE TPAMI* 22, 10 (2000), 1168–1171.

[25] Stefan Schneegass, Youssef Oualil, and Andreas Bulling. 2016. SkullConduct: Biometric user identification on eyewear computers using bone conduction through the skull. In *ACM CHI*. 1379–1384.

[26] Cong Shi, Jian Liu, Hongbo Liu, and Yingying Chen. 2017. Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT. In *ACM MobiHoc*. 1–10.

[27] Dai Shi, Dan Tao, Jiangtao Wang, Muyan Yao, Zhibo Wang, Houjin Chen, and Sumi Helal. 2021. Fine-Grained and Context-Aware Behavioral Biometrics for Pattern Lock on Smartphones. *ACM IMWUT* 5, 1 (2021), 1–30.

[28] Xingzhe Song, Boyuan Yang, Ge Yang, Ruirong Chen, Erick Forno, Wei Chen, and Wei Gao. 2020. SpiroSonic: monitoring human lung function via acoustic sensing on commodity smartphones. In *ACM MobiCom*. 1–14.

[29] Yunpeng Song, Zhongmin Cai, and Zhi-Li Zhang. 2017. Multi-touch authentication using hand geometry and behavioral information. In *IEEE S&P*. IEEE, 357–372.

[30] Youngbae Song, Geumhwan Cho, Seongyeol Oh, Hyoungshick Kim, and Jun Ho Huh. 2015. On the effectiveness of pattern lock strength meters: Measuring the strength of real world pattern locks. In *ACM CHI*. 2343–2352.

[31] Statcounter. 2022. Mobile Operating System Market Share Worldwide. https://gs.statcounter.com/os-market-share/mobile/worldwide.

[32] Statista. 2022. Share of active phones with enabled biometrics in North America, Western Europe & Asia Pacific from 2016 to 2020. https://www.statista.com/statistics/1226088/north-america-western-europe-biometric-enabled-phones/.

[33] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals. In *ACM MobiCom*. 591–605.

[34] Tencent. 2022. WeChat. https://www.wechat.com.

[35] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication*. Cambridge university press.

[36] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the security of graphical passwords: The case of android unlock patterns. In *ACM CCS*. 161–172.

[37] Haoran Wan, Shuyu Shi, Wenyu Cao, Wei Wang, and Guihai Chen. 2021. RespTracker: Multi-user Room-scale Respiration Tracking with Commercial Acoustic Devices. In *IEEE INFOCOM*. IEEE.

[38] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. 2020. Push the Limit of Acoustic Gesture Recognition. *IEEE TMC* (2020).

[39] Weitao Xu, Zhenjiang Li, Wanli Xue, Xiaotong Yu, Bo Wei, Jia Wang, Chengwen Luo, Wei Li, and Albert Y Zomaya. 2021. InaudibleKey: Generic Inaudible Acoustic Signal based Key Agreement Protocol for Mobile Devices. In *ACM IPSN*. 106–118.

[40] Weitao Xu, Yiran Shen, Chengwen Luo, Jianqiang Li, Wei Li, and Albert Y Zomaya. 2020. Gait-Watch: A Gait-based context-aware authentication system for smart watch via sparse coding. *Ad Hoc Networks* 107 (2020), 102218.

[41] Weitao Xu, Yiran Shen, Yongtuo Zhang, Neil Bergmann, and Wen Hu. 2017. Gait-watch: A context-aware authentication system for smart watch based on gait recognition. In *IoTDI*. 59–70.

[42] Guixin Ye, Zhanyong Tang, Dingyi Fang, Xiaojiang Chen, Kwang In Kim, Ben Taylor, and Zheng Wang. 2017. Cracking Android pattern lock in five attempts. In *NDSS*. Internet Society.

[43] Fusang Zhang, Zhi Wang, Beihong Jin, Jie Xiong, and Daqing Zhang. 2020. Your Smart Speaker Can" Hear" Your Heartbeat! *ACM IMWUT* 4, 4 (2020), 1–24.

[44] Jie Zhang, Xiaolong Zheng, Zhanyong Tang, Tianzhang Xing, Xiaojiang Chen, Dingyi Fang, Rong Li, Xiaoqing Gong, and Feng Chen. 2016. Privacy leakage in mobile sensing: Your unlock passwords can be leaked through wireless hotspot functionality. *Mobile Information Systems* 2 (2016).

[45] Xinchen Zhang, Yafeng Yin, Lei Xie, Hao Zhang, Zefan Ge, and Sanglu Lu. 2020. TouchID: User Authentication on Mobile Devices via Inertial-Touch Gesture Analysis. *ACM IMWUT* 4, 4 (2020), 1–29.

[46] Bing Zhou, Jay Lohokare, Ruipeng Gao, and Fan Ye. 2018. EchoPrint: Two-factor authentication using acoustics and vision on smartphones. In *ACM MobiCom*. 321–336.

[47] Man Zhou, Qian Wang, Jingxiao Yang, Qi Li, Feng Xiao, Zhibo Wang, and Xiaofeng Chen. 2018. Patternlistener: Cracking android pattern lock using acoustic signals. In *ACM CCS*. 1775–1787.