Squeez'In: Private Authentication on Smartphones based on Squeezing Gestures

Xin Yi
yixin@tsinghua.edu.cn
Tsinghua University
Beijing, China
Zhongguancun Laboratory
Beijing, China

Shuning Zhang
zhang-sn19@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Ziqi Pan
pzq19@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Louisa Shi
louisa.shi@gmail.com
Tsinghua University
Beijing, China

Fengyan Han
hfy19@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Yan Kong
ky21@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Hewu Li
lihewu@cernet.edu.cn
Tsinghua University
Beijing, China
Zhongguancun Laboratory
Beijing, China

Yuanchun Shi
shiyc@tsinghua.edu.cn
Tsinghua University
Beijing, China

ABSTRACT

In this paper, we proposed *Squeez'In*, a technique on smartphones that enabled private authentication by holding and squeezing the phone with a unique pattern. We first explored the design space of practical squeezing gestures for authentication by analyzing the participants' self-designed gestures and squeezing behavior. Results showed that varying-length gestures with two levels of touch pressure and duration were the most natural and unambiguous. We then implemented *Squeez'In* on an off-the-shelf capacitive sensing smartphone, and employed an SVM-GBDT model for recognizing gestures and user-specific behavioral patterns, achieving 99.3% accuracy and 0.93 F1-score when tested on 21 users. A following 14-day study validated the memorability and long-term stability of *Squeez'In*. During usability evaluation, compared with gesture and pin code, *Squeez'In* achieved significantly faster authentication speed and higher user preference in terms of privacy and security.

CCS CONCEPTS

• **Security and privacy** → **Authentication**; • **Human-centered computing** → **Touch screens**; **Interaction techniques**.

KEYWORDS

touch pressure, capacitive sensing, motion-based authentication

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3581419

ACM Reference Format:
Xin Yi, Shuning Zhang, Ziqi Pan, Louisa Shi, Fengyan Han, Yan Kong, Hewu Li, and Yuanchun Shi. 2023. Squeez'In: Private Authentication on Smartphones based on Squeezing Gestures. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3544548.3581419

1 INTRODUCTION

Nowadays, smartphones have become the most pervasive personal device in our daily lives. To protect security in various scenarios (e.g., online payment, APP log-in, and personal information access), authentication has become one of the most frequent interactions on smartphones. Currently, smartphones are also widely used for cross-authentication on websites and IoT devices (e.g., Amazon Web Services IoT[1] and Xiaomi IoT[2]). As a result, efficient, secure, and natural smartphone authentication has become a key demand in the era of mobile computing.

Authentication methods on today's mainstream smartphones can be categorized into two kinds: code-based (e.g., pin code and graphical gesture) and biometric features (e.g., face ID and fingerprint). Being widely commercialized, these methods are simple to use and can achieve a high authentication speed. However, pin code and graphical gestures are prone to shoulder-surfing when used in public, and their code space is limited (usually 6-digit codes or 9-dot gestures). Moreover, users usually feel it difficult to remember abstract passwords [3]. Biometric authentication relies on the unique physical characteristics of the users, which guarantees high security. However, the unchangeable biometric information also

[1]https://aws.amazon.com/cn/iot/solutions/connected-home/
[2]https://iot.mi.com/new/doc/guidelines-for-access/cloud-access/overview.html
[3]https://www.forbes.com/sites/daveywinder/2019/12/14/ranked-the-worlds-100-worst-passwords/

poses significant risks when copied, and users may hesitate to use such techniques due to privacy concerns [4].

Aiming at these challenges, researchers have proposed motion-based authentication techniques [2, 42], which improved the memorability of the codes using gestures, and also leveraged dynamic behavioral features in addition to static biometric features. However, the designed motions were usually obvious to be spotted, limiting their usability in real life. To overcome this problem, researchers have also proposed implicit authentication techniques [25, 49], which tracked the biometric features continuously to avoid explicit authentication behavior. However, the demand for continuous tracking also limits the applicable scenarios of these techniques and may lead to privacy issues.

In this paper, we proposed *Squeez'In*, a private motion-based authentication technique for smartphones: users hold and squeeze the phone with a unique pattern, and both the squeezing gesture and the user-specific behavioral features are considered during authentication. The advantages include: 1) squeezing gestures with multiple dimensions are more expressive compared with codes, allowing for a bigger code space; 2) the behavioral features during squeezing are dynamic and specific to users, making it not easy to copy; 3) squeezing gestures are subtle and hard to be snooped.

To explore the feasibility of *Squeez'In*, we first conducted a user study to explore the design space of squeezing gestures. We asked the participants to design squeezing gestures for authentication in three scenarios and analyzed the features they leveraged. Results showed that code length, touch pressure, and touch duration were the most preferred features. A following user study further examined the users' ability to perform gestures with different feature levels and found that they were capable of controlling two distinguishable levels of touch pressure and touch duration, respectively.

We implemented *Squeez'In* on an off-the-shelf smartphone. Touch pressure and touch duration were inferred based on the capacitive sensing data from the touchscreen. Furthermore, we incorporated SVM and GBDT with data augmentation to achieve robust authentication based on only 3 samples from the user. In a simulation with the data from 21 users, *Squeez'In* achieved 99.6% accuracy of accepting the correct user, with 93.6% probability of rejecting the attack from other users. A 14-day user study was conducted to verify the long-term stability of *Squeez'In*. Results showed that the users could accurately memorize the squeezing codes, and the recognition accuracy did not tend to drop over time.

We conducted a final user study to test the usability of *Squeez'In* and mainstream authentication techniques. The participants were asked to pick up and unlock the phones placed at different locations. Results showed that *Squeez'In* achieved competitive authentication accuracy compared with pin code and gesture authentication, but were significantly faster. Moreover, *Squeez'In* was highly preferred in terms of privacy, security, generalizability, and social acceptance.

The contributions of this paper are three-folded:

- For the first time, we systematically explored the features of user-defined squeezing gestures for authentication. We proposed applicable features and corresponding values for designing natural and robust squeezing gestures.
- We proposed and implemented *Squeez'In* on off-the-shelf capacitive sensing smartphones. *Squeez'In* incorporated SVM and GBDT with data augmentation to achieve robust authentication based on only 3 samples from the user.
- Through simulation and user studies, we validated the authentication performance, long-term stability, and usability of *Squeez'In* in comparison with mainstream techniques.

2 RELATED WORK

Currently, commercial products adopt two major kinds of authentication techniques: code-based (e.g., pin code and graphical track) and biometric features (e.g., fingerprint and face ID). To facilitate the usability of authentication techniques, researchers have also proposed to leverage a wider range of auxiliary features (e.g., behavioral feature [2, 42] and contextual information [25]). In this paper, we mainly reviewed authentication techniques leveraging capacitive sensing and touch pressure.

2.1 Authentication Techniques on Capacitive Touchscreens

The capacitive touchscreen is the most common interaction interface on today's mobile devices. Due to its sensitivity to the human body, many researchers have tried to leverage it for authentication. Bodyprint [13] applied the capacitive screen on smartphones as an image scanner and authenticated the users based on the contact feature of different body parts (e.g., ear, palm, and finger). Similarly, Rilvan et al. [41] captured the capacitance image when touching the phone with the ear or fingers, and used SVM and random forest for authentication. CapAuth [12] also used SVM to authenticate the users based on the contact area when they put their fingers together on the screen. These techniques all relied on static biometric features for authentication, ensuring high security and potentially high authentication speed. However, similar to fingerprint authentication, they also posed privacy issues and may be insecure if these features were copied.

Aiming at this problem, some researchers paid attention to behavioral features, as these were dynamic and more expressive. Feng et al. [10] extracted features from gesture strokes on a touchscreen and incorporated it with the data sequence from IMU for authentication. Rilvan et al. [40] monitored the frames of capacitance data during swiping and performed authentication using SVM. Meng et al. [28] surveyed different biometric authentication techniques (e.g., fingerprint, face ID, palm, and gait) and constructed a multi-phase authentication system, which significantly reduced the false rate of single biometric systems. These techniques demonstrated the possibility of authentication during touch interaction. However, none of them have explored using squeezing gestures, which were subtle, private, and potentially efficient as they could be performed during grasping.

2.2 Authentication Techniques Leveraging Touch Pressure

Beyond capacitive sensing, touch pressure was also widely used for authentication [22, 23, 30], as it was inherent during touching and was easy to control. Force-pin [22] explicitly combined touch pressure with a 4-digit pin code, significantly expanding the code

[4] https://www.bbc.com/news/technology-49343774

space and achieving higher security. Also, researchers have leveraged the implicit touch pressure during pin code [45] and gesture authentication [9]. These techniques improved the accuracy and expressiveness of traditional authentication techniques by leveraging touch pressure, but at the cost of limited generalizability.

There were also works that only used touch pressure for authentication. Murao et al. [30] placed pressure sensors on the lateral and back of a phone to sense the pressure during gripping for authentication. However, they only focused on grip gestures with different orders of fingers, and did not fully explore the features of user-defined gestures. As a result, the authentication error rate was not low (4% on 5 participants). Meng et al. [29] proposed to use a touch sequence with two levels of touch pressure on a fixed keyboard for authentication. Iso et al. [17] extracted the personal pressure features during key strokes and used a statistic probabilistic model for authentication. However, the accuracy was still unsatisfying (~ 90%). Kim et al. [20] evaluated a number of authentication schemes during multi-touch interaction on the tabletop and found that pressure-based authentication significantly enhanced shoulder-surfing resistance.

These works have demonstrated the potential of pressure-based authentication. However, none have systematically explored the design space of squeezing gestures for authentication. As we will show in this paper, with carefully designed gestures and algorithms, satisfying authentication accuracy and usability can be achieved at the same time.

2.3 Interaction Techniques Leveraging Touch Pressure

Due to its rich expressiveness, touch pressure has been leveraged in different kinds of interaction tasks. ForceBoard [56] allowed the users to adjust the touch pressure to control a cursor on a keyboard for text entry, demonstrating the users' ability to continuously control the touch pressure. PalmBoard [55] proposed a Markov-Bayesian model that leveraged implicit touch pressure during one-handed touch typing for input decoding, which achieved an input speed of 32.8 WPM. Presstures [39] extended multi-finger gestures by using the initially applied pressure level for implicit mode switching. Evaluation results showed that two levels of touch pressure would yield satisfying interaction performance.

Some researchers have tried to use squeezing for interaction. Wilson et al. [53] proposed to intentionally control the touch pressure of different fingers while squeezing the phone for various interaction tasks (e.g., rotation and zooming), which proved the feasibility of controlling touch pressure during squeezing. Quinn et al. [37] evaluated the usability of using squeezing gestures on a Google Pixel 2 smartphone for different activation tasks. However, the design was limited to one time of squeezing and has not focused on typical authentication tasks on which users may have different preferences of features.

The above works inspired us to use squeezing gestures for authentication. However, whether the complexity of squeezing gestures could satisfy the need for authentication and how users would perform and perceive squeezing-based authentication has never been explored. In this regard, this work provided the first empirical results on the design and evaluation of squeezing-based authentication.

3 STUDY 1: EXPLORING THE DESIGN SPACE OF SQUEEZING GESTURES

In this section, we asked the participants to design squeezing gestures for authentication purposes in different scenarios, with the aim to explore the design space of practical squeezing gestures for *Squeez'In*.

3.1 Participants and Apparatus

We recruited 15 participants (7 male, 8 female, age = 21.5, SD=1.1) from the campus. A Huawei Mate 30 Pro smartphone was used as the apparatus. The phone was $158 \times 73mm$ in size, with a screen resolution of 2400×1176. The left and right edges of the touchscreen were curved and touchable (see Figure 1). The operating system was EMUI 11.0.0.165 (Android version 10.0). Each participant was compensated $10.

(a) (b)

Figure 1: Experiment apparatus. (a) Front view. (b) Bottom view, showed the curved screen.

3.2 Experiment Design

As with pin code and graphical gesture, the squeezing gestures of *Squeez'In* would be designed by individual users for themselves during usage. Therefore, similar to existing works [50], we asked the participants to design their preferred squeezing gestures for authentication freely and analyzed the features they leveraged. Specifically, we tested three scenarios: *Payment*, *APP Login* and *Phone Unlock*. These were all typical smartphone authentication scenarios but were usually perceived with different levels of security risk.

3.3 Procedure

We first introduced the idea of *Squeez'In* to the participants. They were then asked to design one squeezing gesture for each of the three scenarios respectively and describe the designed gesture to the experimenter. The instruction was, "Please design a squeezing gesture for authentication in this scenario that you would like to use in your real life."

To inspire the participants during designing, at the beginning of the study, we introduced seven features to them that were frequently used in other gesture and pressure-based interaction techniques [9, 30, 37]: code length (number of presses), touch pressure, touch duration, gripping hand (e.g., both hands vs right hand), number of fingers (that performed the squeeze), finger identity and touch location. During designing, we also encouraged the participants

to include any other features they liked, and they were allowed to freely try the gestures on the apparatus until determined.

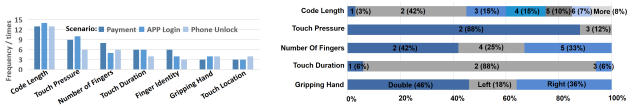
3.4 Results

3.4.1 Usage of Features. In total, we collected 15 participants × 3 scenarios = 45 squeezing gestures. Table 1 showed some examples of the proposed gestures. Although the participants were allowed to involve any features they liked, most of the gestures (40/45) only leveraged the seven features we introduced. In the remaining 5 gestures, three mentioned "finger angle" (e.g., "use the thumb to press on the bezel with an angle of 45°") and two mentioned "slide/squeeze" (e.g., "use the thumb and the index finger to slide on both bezels upwards, from lightly to heavily"). Concerning the low frequency of these two features, we did not include them in the following analysis.

Table 1: Examples of the gestures proposed by the participants.

Scenario	Squeezing Gesture
Payment	• Hold the phone with four fingers, squeeze in 'heavy, light, light, heavy' pattern, with the same duration. • Squeeze the phone with increasing pressure for three times. • Hold the phone in one hand, use the thumb and middle finger to squeeze lightly for three times, followed by three heavy squeezes.
APP Login	• Single hand, double tap with medium pressure. • Hold the phone with two hands, squeeze shortly in 'heavy, light' pattern.
Phone Unlock	• Use two fingers to squeeze both sides of the phone heavily for two times. • Use four fingers of one hand to squeeze shortly in 'light, heavy' pattern.

The average number of features within one gesture was 2.98 (SD = 0.54). This number implied that participants tended not to involve too many features simultaneously for authentication purposes, which may cause an unnecessary mental and physical burden. Figure 2a showed the number of gestures that leveraged each feature in each scenario. In total, *code length* was the most frequently used feature (40/45). Meanwhile, about half of the gestures leveraged *touch pressure* (25/45), *touch duration* (16/45), and *number of fingers* (19/45). In comparison, *finger identity* (13/45), *gripping hand* (11/45) and *touch location* (10/45) were only used in a few gestures. Repeated Measures Analysis of Variance (RM-ANOVA) found that the frequency of different features varied significantly ($F_{6,84} = 15.7, p < .05$). Meanwhile, no significant difference was found between different scenarios ($F_{2,28} = 1.08, p = .58$), indicating that the participants did not tend to leverage more features to design complex squeezing gestures for scenarios with a higher security risk. This result suggested the potentially high generalizability of squeezing gestures.



(a) Number of gestures with different features in each scenario. **(b) Distribution of the feature values.**

Figure 2: Analysis of the features used in the designed gestures.

3.4.2 Feature Value Distribution. We further analyzed the distribution of the feature values in the gestures. As *finger identity* and *touch location* were always dependent on other factors (e.g., *number of fingers*) and were complex to aggregate, we did not analyze these two features.

As shown in Figure 2b, the distribution of *code length* was rather scattered, ranging from 1 to over 6. 42% of the gestures' length was only two, implying that the participants favored short gestures but would also accept longer ones. Meanwhile, most gestures (88%) were designed to contain two levels of touch pressure, which were usually described in a temporal sequence (e.g. "light, heavy, heavy, medium, heavy"). Similarly, 88% of the gestures only contained two levels of touch duration (long vs. short). Regarding *number of fingers*, 42% gestures only leveraged two fingers (usually the thumb and index finger), while 33% leveraged all five fingers to hold and squeeze the phone stably. 54% of gestures were performed with only one hand, which suggested the potential advantage of one-handed interaction.

It could be noticed that *code length*, *touch pressure* and *touch duration* described the procedure of the squeezing behavior, while *number of fingers*, *finger identity* and *gripping hand* described the gripping posture. Considering that the former three factors were generally more preferred by the participants (see Figure 2a), and that detecting the gripping posture (e.g. finger identity) was not feasible on off-the-shelf smartphones, we concluded that *code length*, *touch pressure* and *touch duration* was the most recommended features when designing squeezing gestures for authentication.

3.4.3 Theoretical Code Space. The above results shaped the theoretical code space of *Squeez'In*. According to Figure 2b, in most cases (excluding the values < 10%), code length $L \in \{2, 3, 4, 5\}$, touch pressure $N_p \in \{2, 3\}$, number of fingers $N_f \in \{2, 4, 5\}$, and touch duration $N_d \in \{2\}$. We did not take the gripping hand into calculation as it was highly mixed with the number of fingers. Therefore, the theoretical code space of *Squeez'In* could be calculated as $\sum_{L,N_p,N_f,N_d}(N_p \times N_f \times N_d)^L \approx 3.82 \times 10^7$, which is 38 times bigger than that of 6-digit pin code (1×10^6), and 100 times bigger than that of 9-dot gesture (no bigger than 3.62×10^5 [51]). When considering only features of the squeezing behavior, the code space would be $\sum_{L,N_p,N_d}(N_p \times N_d)^L \approx 1.07 \times 10^4$, which is achievable on off-the-shelf smartphones. As the procedure of the squeezing behavior was generally more subtle and difficult to be observed than the posture, therefore the corresponding features are expected to be more robust to shoulder surfing. In real use, the combination of features could be selected to suit the security demand of different target scenarios.

4 STUDY 2: EXAMINING THE DISTINGUISHABILITY OF THE USERS' SQUEEZING BEHAVIOR

To explore whether the users could accurately perform squeezing gestures with different levels of features, we conducted another user study to examine the distinguishability of the users' squeezing behavior.

4.1 Participants and Apparatus

We recruited 14 right-handed participants (6 male, 8 female, age = 20.2, SD = 0.9) from the campus, none of whom participated in Study 1. We used the same apparatus as with Study 1. We developed an Android application to sense and visualize the users' touch pressure based on the capacitance data of the touchscreen. The algorithm details will be described in Section 5.1. Each participant was compensated $10.

4.2 Experiment Design

This study aimed to explore whether the users could perform squeezing gestures with different feature values that could be distinguished. As *code length* was easy to control (by squeezing and releasing) and detect, we focused on *touch pressure* and *touch duration* in this study, resulting in two kinds of tasks:

Touch Pressure: we tested gestures with 2 levels (heavy-light-heavy-light) and 3 levels (light-heavy-medium) of pressure, which covered all the levels in Figure 2b. According to the pilot study, providing visual feedback could help the users to control touch pressure more accurately. Therefore, we tested both in eye-engaged and eyes-free conditions. The visual feedback was designed as a bar displayed on the phone, with its length indicating the current touch pressure.

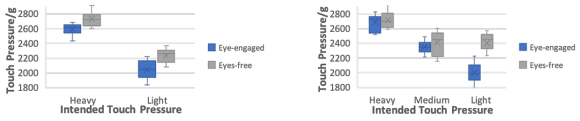
Touch Duration: we tested gestures with 2 levels (long-short-long-short) and 3 levels (long-short-medium) of touch duration, covering 94% of the cases in Figure 2b. As the pilot study found that providing visual feedback was not so helpful in this task, we did not provide visual feedback. Instead, we asked the participants to perform the gestures both with heavy and light pressure in order to examine the interaction between touch pressure and duration.

4.3 Procedure

The participants were first allowed 5 minutes for warm-up. Then they completed two sessions of touch pressure tasks in random order, with and without visual feedback, respectively. In each session, they performed the 2-level and 3-level gestures twice, in random order, resulting in a total of 4 gestures. We did not specify the touch duration in this task. After that, the participants completed two sessions of touch duration tasks in random order, with light and heavy pressure, respectively. In each session, they were asked to perform the 2-level and 3-level gestures twice, in random order, resulting in a total of 4 gestures. During the entire experiment, the participants held the phone and performed the squeezing code only with their dominant hand. A short break was enforced between the two kinds of tasks.

4.4 Results

4.4.1 Touch Pressure. Figure 3 showed the average touch pressure in different conditions. As expected, the exhibited touch pressure increased monotonically with the increase of intended touch pressure. For two-level gestures, RM-ANOVA found that both intended touch pressure ($F_{1,13} = 672, p < .001$) and visual feedback ($F_{1,13} = 109, p < .001$) yielded significant effect on the exhibited touch pressure, but with no significant interaction between these two factors ($F_{1,13} = 1.66, p = .22$). This implied that the users could stably perform distinguishable heavy vs. light touches, regardless of visual feedback. However, providing visual feedback would cause the participants to press slightly lighter (heavy: 2592g(SD = 85g)vs. 2733g(SD = 103g), light: 2042g(SD = 134g) vs. 2236g(SD = 90g)).



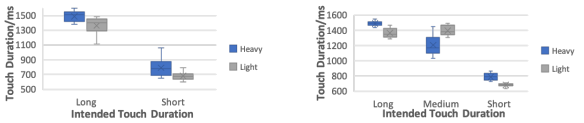
(a) Two levels. **(b) Three levels.**

Figure 3: Average touch pressure in different conditions.

For three-level gestures, RM-ANOVA found a main effect of intended touch pressure on the exhibited value both in eye-engaged ($F_{2,26} = 139, p < .001$) and eyes-free condition ($F_{2,26} = 23.3, p < .001$). However, *post hoc* analysis found that medium (2413g) and light (2404g) touches in the eyes-free condition were not significantly different ($p = .84$). This implied that distinguishing three pressure levels was only applicable with the help of visual feedback. Again, providing visual feedback was found to lower the touch pressure significantly ($F_{2,26} = 21.9, p < .001$, heavy: 2690g(SD = 111g) vs. 2717g(SD = 107g), medium: 2348g(SD = 79g) vs. 2413g(SD = 156g), light: 1999g(SD = 126g) vs. 2404g(SD = 157g)). Therefore, we recommended 2 levels of touch pressure, which supported eyes-free interaction that facilitated interaction speed and generalizability.

4.4.2 Touch Duration. Figure 4 showed the average touch duration in different conditions. Generally, the exhibited touch duration increased with the increase of the intended touch duration. For two-level gestures, RM-ANOVA found that both touch pressure ($F_{1,13} = 21.5, p < .001$) and intended touch duration ($F_{1,13} = 1685, p < .001$) yielded a significant effect on the exhibited touch duration, with no significant interaction between them ($F_{1,13} = 0.205, p = .66$). This implied that the users could stably perform long vs. short touches that were distinguishable, regardless of touch pressure. When pressing more heavily, users also tended to press longer (long: 1490ms(SD = 71ms) vs. 1364ms(SD = 112ms), short: 792ms(SD = 111ms) vs. 681ms(SD = 56ms)).

For three-level gestures, RM-ANOVA found a main effect of intended touch duration on the measured value both with heavy ($F_{2,26} = 300, p < .001$) and light pressure ($F_{2,26} = 721, p < .001$) (heavy vs. light for long: 1492ms(SD = 30ms) vs. 1368ms(SD = 64ms), medium: 1207ms(SD = 126ms) vs. 1398ms(SD = 63ms), short: 787ms(SD = 46ms) vs. 680ms(SD = 19ms)). However, *post hoc* analysis found that the medium (1398ms) and long (1368ms) touches with light pressure were not significantly different ($p = .28$). This implied



(a) Two levels. (b) Three levels.

Figure 4: Average touch duration in different conditions.

that distinguishing three duration levels was difficult when touched lightly. RM-ANOVA found that touch pressure significantly affects touch duration ($F_{2,26} = 56.1, p < .001$). However, no consistent trend was found. Considering that in real use, different levels of touch pressure and touch duration were often mixed together, we concluded that the 2-level of touch duration was more robust.

5 IMPLEMENTATION OF *SQUEEZ'IN*

As with any other authentication methods, the interaction of *Squeez'In* contained two parts: *Registration* and *Authentication*. During registration, the user designed his/her unique squeezing gesture and performed it on the phone several times as samples. During authentication, the user could perform the squeezing gesture on the phone. A recognizer would analyze the similarity between the registered samples and the input and judge whether the authentication could be accepted. In this section, we described the implementation details of *Squeez'In* on an off-the-shelf capacitive sensing smartphone.

5.1 Touch Pressure Sensing based on Capacitance Data

As shown in previous studies, touch pressure was one of the main features when designing squeezing gestures. However, commercial smartphones were mostly equipped with capacitive-sensing touchscreens. Although existing works [7, 16, 31] have explored the possibility of inferring touch pressure based on capacitance data, none of them can be directly implemented in our case, due to different contact shape and different hardware module. Therefore, we derived our own algorithm for touch pressure sensing based on the collected user data on our apparatus.

5.1.1 Data Collection. We recruited 14 participants (8 male, 6 female) with a mean age of 20.2 (SD = 0.9) from the campus. Each participant was compensated $10. The same apparatus as in previous studies was used. We requested the operating system with developer-level permission for the raw capacitance data from the touchscreen, which formed an 18×36 matrix with values in $[-1000, 3000]$ at 30Hz (see Figure 5a). To capture the ground truth of the touch pressure, we also attached an RP-C18.3-ST piezo-resistive pressure sensor on the user's left thumb, which could sense pressure between 20–6000g at $10Hz$ with drift error < 5%. The pressure sensor sent analog signals to the computer through Arduino-Uno (see Figure 5b). As the pressure sensor would significantly affect the capacitance value around the contact area, we asked the participants to squeeze the phone simultaneously with both thumbs in opposite directions. According to Newton's third law, the touch pressure on the left and right edges would be identical as long as the phone is kept stable. Therefore, we could sense the touch pressure on the left edge and the aligned capacitance data on the right edge.

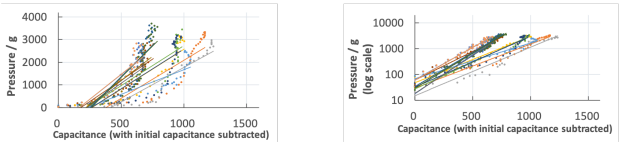
(a) (b)

Figure 5: The apparatus and the squeezing posture during data collection. The capacitance data was not displayed during the study.

Participants were asked to hold the smartphone with both hands and remain still. They then performed two rounds of squeezing gestures, each containing five discrete fast squeezes for a warm-up and one long crescendo squeeze from the lightest to the heaviest pressure they were able to perform. We used the data during the crescendo squeeze for further analysis.

5.1.2 Inferring Touch Pressure from Capacitance Data. Inferring touch pressure from capacitance data was a fundamental problem for capacitive pressure sensors, which some researchers have also explored [7, 19]. According to physics equations, touch pressure and capacitance should be linearly correlated [34, 47], when assuming the contact area to be constant. However, due to the change of contact area during pressing, and the difference in hardware modules, existing works have also found other correlations (e.g., exponential correlation [44]).

In total, we collected 14 participants × 2 repetitions = 28 crescendo squeezes. A significant variance was found in the range of touch pressure across different participants, with the minimum and maximum pressure being 10g and 3684g, respectively. We performed linear fitting between capacitance value and touch pressure for individual participants. However, the correlation was not strong, with an average R^2 of only 0.72 (SD = 0.06) (see Figure 6a).



(a) Linear fitting. (b) Exponential fitting.

Figure 6: Fitting between capacitance value and touch pressure across all 14 participants. Capacitance data was aligned by subtracting the initial value in the first frame (c_0).

Although not linearly correlated, the data exhibited an exponential correlation trend. A linear fitting between capacitance value and logged pressure for individual participants yielded an average R^2 of 0.91 (SD = 0.04) (see Figure 6b). We speculated that this correlation was due to different sensor structures. Furthermore, it could

be observed that the fitted lines in Figure 6b formed two clusters with different slopes, depending on different initial capacitance (ranging from 1109 to 1914, which may be due to the design of the sensor). Therefore, we proposed a general model for inferring touch pressure based on the exponential of capacitance data:

$$p = \begin{cases} (7.6 \times 10^{-4} c_0 - 6.5 \times 10^{-1}) e^{0.00375c}, & if\ c_0 < 1500 \\ (4.8 \times 10^{-7} c_0 - 9.5 \times 10^{-3}) e^{0.0006c}, & if\ c_0 \geq 1500 \end{cases} \quad (1)$$

where p and c denoted pressure and capacitance, respectively. And c_0 denoted the first frame of capacitance data when the finger touched the screen. The general model achieved an R^2 of 0.91 on the collective data from all participants.

5.2 Authentication Algorithm

5.2.1 Algorithm Design. As shown in Figure 7, the task of the authentication algorithm contains two parts: 1) during the registration phase, derives the personalized model automatically based on the samples provided by the user; 2) during the authentication phase, tests the input gesture on the model to determine whether to accept or reject the authentication. To achieve that, we first performed hyperparameter pre-training based on collected user data to optimize the performance (details in Section 6.2).

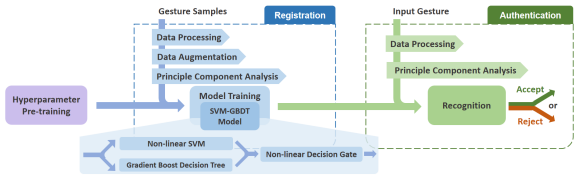


Figure 7: Algorithm design of *Squeez'In*.

5.2.2 Personal Model Derivation. To derive a robust personalized model based on a very limited number of gesture samples (< 5 in real use), we designed the personal model derivation process that contained three steps:

Step 1: Data Processing. We continuously formed the raw capacitance data (18 × 36, 30fps) into a temporal tensor consisting of different frames. We sequentially adopted spatial and temporal clustering algorithms for different frames to aggregate connected touch areas into gesture events. Sliding window smoothing (size = 5, rolling mean filter adopted) was then applied to reduce the input noise and eliminate accidental touches.

Step 2: Data Augmentation. We used data augmentation to synthesize additional training samples based on the limited samples provided by the user, which has been proven effective in improving the recognition performance [24, 33, 54]. Specifically, we added Gaussian noise sampled from a distribution with mean 0 to the raw data to create augmented data [52]. The baseline standard deviation and augmentation ratio were determined according to the collected user data (see Section 6.2). We treated synthesized samples with noise ≤ and > 3 standard deviations away from the mean value as positive and negative samples, respectively. In addition, we also added pure noise samples as negative samples.

Step 3: Feature Extracting and Model Training. We used Principle Component Analysis (PCA) to extract features from the processed data. Following existing works [5, 15, 45], we trialed different models and finally ensembled non-linear Support Vector Machine (SVM) and Gradient Boost Decision Tree (GBDT) for recognition. A non-linear decision gate with 3 branches ("based on SVM to accept", "based on GBDT to accept" and "combining SVM and GBDT to accept") would judge whether the authentication could be accepted. Based on the augmented data, we used Scikit-learn for training. We chose RBF kernel for SVM [35], regularization (C in scikit-learn) was set to 3, and tol was fixed as 5×10^{-3} to balance the training speed (against lengthy iterating) and accuracy (against overfitting). For GBDT, we set $n\_estimators = 100$, $subsample = 1.0$ and $alpha = 0.9$ as default.

5.2.3 Gesture Recognition. During authentication, *Squeez'In* first performed the same data processing and PCA procedure as described above on the user's input gesture. Then, the trained SVM-GBDT model would perform the recognition and judge whether the authentication could be accepted.

6 STUDY 3: AUTHENTICATION PERFORMANCE EVALUATION

In this section, we evaluated the authentication performance of *Squeez'In* on the collected data from real users. We also conducted a 14-day study to validate the memorability and long-term stability of *Squeez'In*.

6.1 Data Collection

We recruited 21 right-hand participants (6 male, 15 female, age = 20.0, SD=1.4) from the campus. None of them participated in the previous studies. The same smartphone was used to collect the data as in previous studies. Each participant was compensated $15.

In order to control the variety due to participants freely designing squeezing gestures, we allocated one gesture for each participant from a pre-defined gesture set with 21 gestures (see Table 2) as his/her designed code. According to the results from Study 1 and 2, we leveraged three features when designing the gestures: code length (5-7), touch pressure (heavy vs. light), and touch duration (short vs. long). Moreover, we defined *Gesture Distance* as the sum of the edit distance between two gestures on all three dimensions. Generally, gestures with closer distances were harder to be distinguished. We carefully designed the gestures to ensure that different levels of gesture distance were covered (from 1 to 12). Therefore, the results indicated the authentication performance of *Squeez'In* in real life.

During the experiment, the 21 gestures were randomly assigned to the 21 participants. The participants first familiarized themselves with the apparatus and the gesture. Then they performed the gesture on the apparatus repetitively for 8 times using their dominant hand, during which the experiment platform recorded the capacitance data. After each repetition, the participants were asked to drop and pick up the phone again after a short break (30-60 seconds), which ensures the variety in gripping postures and data independence. The study lasted about 10-15 minutes for each participant. In total, we collected 21 participants × 8 repetitions = 168 gestures.

Table 2: Gesture set used in Study 3. For touch pressure, H and L denoted heavy and light, respectively. For touch duration, S and L denoted short and long, respectively.

Length = 5		Length = 6		Length = 7	
Pressure	Duration	Pressure	Duration	Pressure	Duration
HLLLL	LSSLL	HLLHLH	LSSLSL	HLLHLHL	SSLSSSL
HLLHH	LSSLL	LHHLLL	LSSLSL	HLHLLLH	SSLSSSL
LHHLL	LSSLL	LLLHLH	LSSLSL	LHHLLLH	SSLSSSL
HLLLH	LSSLL	HLHHLL	LSSLSL	HLLLLHL	SSLSSSL
LHHLL	SLSLL	LHHLLL	LSLLSL	HLLHLHL	SLLSSSL
LHHLL	SLLSS	LHHLLL	SLSLSL	HLLHLHL	LSLSSLS
LHHLL	LSSLS	LHHLLL	SLSSLL	HLLHLHL	SLSLLLS

6.2 Algorithm Parameter Optimization

Based on the collected user data in Study 3, we optimized the remaining algorithm parameters of *Squeez'In* for data augmentation and model derivation (see Section 5.2.2) using a grid search and line search [11, 21]. Thresholds for non-linear gates were first optimized using line search and other parameters were then optimized using grid search respectively to minimize searching cost while obtaining optimized results. For each parameter's setting, we took 3 repeated tests (with different random seeds). For each test, we randomly chose 7 samples from each participant as registration samples and used the remaining 1 sample as test data. Then we applied the data processing, data augmentation and model derivation procedure to the data and got the final authentication accuracy. Finally, the parameter value with the highest average accuracy over 3 repeated tests would be chosen and used in *Squeez'In*.

According to the simulation, we set the standard deviation of Gaussian noise during data augmentation to 1/3 of the standard deviation in the users' data and augmentation ratio to 20. The dimension of the PCA layer was compressed from 23 (the dimension of the input vector after padding was 23) to 20. And the thresholds of the non-linear gate were set to 0.45, 0.65, 0.75 and 0.85, respectively.

6.3 Simulation Design

This study aimed to evaluate *Squeez'In*'s ability to accept the input from the correct user, while rejecting the input from other users. We used the algorithm design and parameters described above (also see Figure 7), and used 8-fold cross-validation. In each round of the simulation, we used 7 samples from each participant as registration samples to derive his/her personalized model and tested the authentication accuracy of all the models using the remaining samples from all the participants (21 samples × 21 models = 441 tests). We calculated the metrics by averaging the performance of all the 21 personalized models in 8 rounds.

6.4 Results

6.4.1 Authentication Accuracy. We calculated authentication accuracy as the total number of accepted positive samples and rejected negative samples, divided by the number of all test samples [8]. Across all participants, the average authentication accuracy was 99.3% (SD = 1.1%), proving the robustness of *Squeez'In*. On average, the recall, precision and F1-score were 93.6%, 91.6% and 0.93, respectively. The *False Acceptance Rate (FAR)* and *False Rejection Rate (FRR)* [6] were 0.4% (SD = 0.1%) and 6.4% (SD = 0.8%) respectively. This suggested that *Squeez'In* could accurately accept the correct input, while rejecting the input from other users.

6.4.2 Model Combination Strategy. To test the contribution of PCA and the model components (SVM and GBDT), we performed an ablation study [36] with the same simulation design and parameter values. Compare

d with the accuracy of 99.3% on the original algorithm, omitting PCA slightly dropped the accuracy to 95.0% (SD = 4.1%). More significantly, the algorithm only achieved 73.3% (SD = 18.8%) and 63.3% (SD = 20.1%) accuracy without SVM and GBDT, respectively. We observed that SVM performed better for longer gestures and explicit features (e.g., touch pressure and duration), and GBDT performed better for shorter gestures and implicit features (e.g. user-specific behavioral features). Therefore, we concluded that the components were all necessary for the effectiveness of our algorithm.

6.4.3 Effect of Sample Size. The effort for registration was important for any authentication techniques. Therefore, we also tested the influence of *Squeez'In* 's effectiveness when reducing the number of registration samples. Following the 8-fold cross-validation experiment design, we varied the number of chosen samples (N) from each participant, with $1 \leq N \leq 7$. Figure 8 showed the average accuracy with different sample sizes. As expected, a significant effect of sample size was found on accuracy using RM-ANOVA ($F_{6,120} = 806, p < .001$). With more samples, the accuracy increased monotonically. However, reducing the sample size was still possible. *Post hoc* analysis found no significant difference between adjacent pairs when $N \geq 3$ (all p value $> .14$). And for ≥ 3 samples, the accuracy was all above 91.0%.

Meanwhile, we found that data augmentation played an important role in avoiding over-fitting. Reducing the augmentation ratio by half would cause the accuracy to drop to below 50%. Therefore, we concluded that a minimum number of three samples were required during registration and proper data augmentation should be added to ensure accuracy in real use. This was close to the repetitions of graphical gesture and pin code (one for entering and one for confirmation).

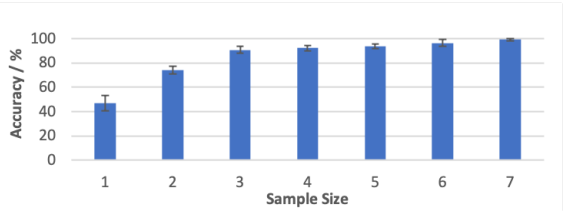
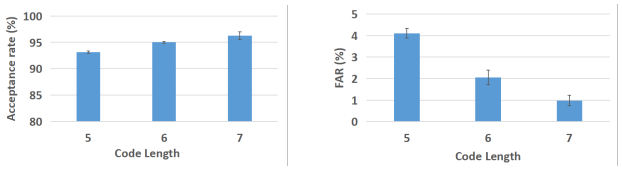


Figure 8: Authentication accuracy with different sample sizes. Error bar showed one standard deviation.

6.4.4 Robustness to Gesture Collision. Code collision was a severe challenge for any code-based authentication technique, which happened when two users accidentally designed the same code. Considering the user-specific behavioral features during squeezing, *Squeez'In* was expected to distinguish different users even when

they performed the same gestures. To verify this, we collected more data to run another simulation.

We recruited another 8 participants (6 male, 2 female, age = 19.2, SD=0.5, no overlap with the 21 users), and asked them to perform the same gestures. We chose three gestures with different lengths from Table 2 (shown in red), and each participant was asked to perform all three gestures eight times, similar to in Section 6.1. We tested the three gestures separately to simulate gesture collision. During simulation, we used the same algorithm parameters and 8-fold cross-validation design (see Section 6.3).



(a) (b)

Figure 9: (a) Acceptance rate and (b) FAR for different gestures. Error bar showed one standard deviation.

Figure 9 showed the acceptance rate and FAR for different gestures. Across the three gestures, the average acceptance rate for positive samples was 94.8%, and the FAR was 2.4%. Interestingly, with the increase in code length, the acceptance rate also increased monotonically (93.1%(SD=0.2%), 95.0%(SD=0.2%), 96.3%(SD=0.7%)), while FAR dropped monotonically (4.1%(SD=0.2%), 2.1%(SD=0.3%) and 1.0%(SD=0.2%)). We speculated this was due to more information on the user-specific behavioral features when performing longer gestures. For gestures with length ≥ 7, the FAR was only 1%. Considering the very low frequency of gesture collision in actual use, we concluded that *Squeez'In* could achieve high robustness and accuracy during authentication, and has significant advantages over conventional code-based authentication techniques (e.g., pin code and graphical gestures) when handling code collision.

6.5 Long-term Stability

Similar to existing works [22, 29], we conducted another 14-day user study with 24 participants to validate the long-term stability of *Squeez'In*, which was important for evaluating its memorability and robustness in real world.

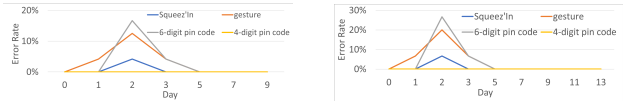
6.5.1 Participants and Apparatus. We recruited 24 participants (15 male, 9 female, age = 20.9, SD=1.4) from the campus. None of them have participated in previous studies and three of them were left-handed. The same smartphone as in previous studies was used as the apparatus. Each participant was compensated $15~$20.

6.5.2 Design and Procedure. In this study, we adopted a within-subjects design with one factor: technique (*Squeez'In*, 4-digit pin code, 6-digit pin code and gesture). We designed the complete user study to last for 14 days. On the first day (day-0), we asked all the participants to design a unique code or pattern for each of the techniques "that would be used for a long period of time in the future, and had not been used before". They then performed the registration and authentication process once for all the techniques in random order. Similar to existing works [22, 29], we asked them to come back and authenticate again with increasing time intervals (i.e., on day 1, 2, 3, 5, 7, 9, 11 and 13).

Each day, the participants completed the authentication task for all four techniques in random order. For each technique, to reflect memorability, we first asked them to write down the code. If they failed to correctly recall the code, we would present the correct code as feedback. After that, the participants were asked to perform the authentication on the apparatus once. As the accuracy of pin code and gesture were very high given the correct code, we only tested *Squeez'In*, which may be affected by different body conditions (e.g., humidity and muscle strength).

6.5.3 Results. Because of COVID-19, 9 participants dropped out after day-9, the rest 15 participants finished the 14-day study. We analyzed the error rate of memorization and recognition separately.

Designed Code for *Squeez'In* We analyzed the length of the participants' designed code for *Squeez'In*. The average code length was 5.6 (SD = 1.3). 4/24 participants designed code with length ≥ 7, while 5/24 participants designed code with length of 4. This implied that most participants preferred codes with length between 5 and 6. We noticed that to facilitate the memorability of the code, over half of the participants (16/24) correlated their code with phrases of a song (e.g., by mapping the touch pressure and duration to the rhythm of a song, similar with the finding in existing works [14]), poetry or rhythms, resulting in longer codes compared with Study 1.



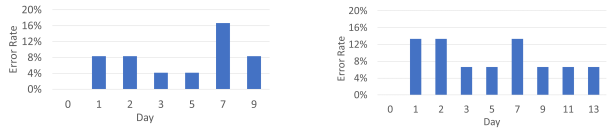
(a) Error rate for all 24 participants across 9 days. **(b) Error rate for 15 participants across 14 days.**

Figure 10: Memorization error rate of different techniques.

Memorability Figure 10a and 10b showed the memorization error rate of different techniques across different days. We performed two-way ANOVA on the 9-day data from 24 participants, no significant difference was found among different techniques ($F_{3,24} = 1.95$, $p = .15$). All the memorization errors occurred within day-5, with the most errors occurring on day-2. Generally, *Squeez'In* achieved a higher memorization accuracy than 6-digit pin code and gesture. Meanwhile, *Time* yielded a significant effect on the error rate ($F_{8,24} = 4.41$, $p < .01$). However, *post hoc* analysis found no significant difference among the error rate on different days.

Recognition Error Rate Figure 11a and 11b showed the recognition error rate of *Squeez'In* across different days. Noticeably, no error was observed on day-0, confirming that the participants could successfully learn to use *Squeez'In* at the first trial. By analyzing the data from all 24 participants, we found that the error rate increased to 8.3% on day-1, and kept between 4.1% and 16.6% in the following week, with an average value of 7.1%. This was close to the recall rate (93.6%) in the previous study. One-way ANOVA did not find a significant effect of *time* on error rate ($F_{6,161} = 0.981$, $p = .44$) ,

implying that *Squeez'In* could stably recognize the squeezing gestures from users across time. One-way ANOVA on the 14-day data from 15 participants did not find different results ($F_{8,126} = 0.378$, $p = .93$).



(a) Recognition error rate for all 24 participants across 9 days. **(b) Recognition error rate for 15 participants across 14 days.**

Figure 11: Recognition error rate of *Squeez'In* across different days.

7 STUDY 4: USABILITY EVALUATION IN REAL SCENARIOS

In Study 3, we have demonstrated the authentication performance of *Squeez'In* through simulation. In this section, we further evaluated the usability of *Squeez'In* in real scenarios.

7.1 Participants and Apparatus

We recruited 18 participants (9 male, 9 female, age = 20.0, SD=1.2) from the campus. Four of them have participated in previous studies. Each participant was compensated $15. The same smartphone as in previous studies was used as the apparatus with the same operating system. We developed the experiment platform using Android Studio. The capacitance data on the touchscreen was collected by the platform, and was sent to a MacBook Pro laptop for online recognition (developed using Python 3.8) via wireless network. The computing and network delay was less than 500ms.

7.2 Experiment Design

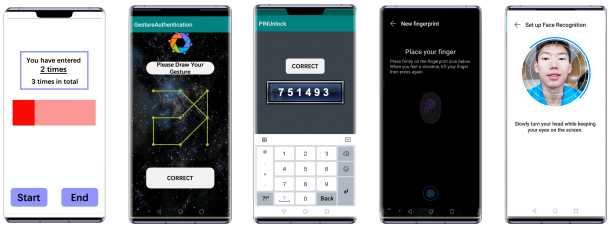
In this study, we used a two-factor, within-subjects design with *technique* and *scenario* as the factors. We tested *Squeez'In* as well as four common authentication techniques as baselines: gesture, pin code, fingerprint, and face ID. For each technique, the participants first registered the gesture, code or bio-information. We then asked them to unlock the phone using it. The detailed interaction for each technique was:

Squeez'In: According to previous results (see Figure 8), we designed the number of required samples during registration to be three. During entering, the platform showed the current touch pressure as the visual feedback (see Figure 12a). Participants were free to design their own gestures with lengths between 4 and 6 (consistent with Study 3), and with 2 levels of pressure and duration respectively. During authentication, the screen was black with no visual feedback until successfully unlocked. The algorithm for authentication was the same as described in Study 3.

Gesture: Participants designed their own gesture that covered at least 6 out of the 9 dots and repeated it twice during registration (to ensure an acceptable security [3, 48]). During authentication, the dots and the drawn gesture would be displayed as visual feedback (see Figure 12b).

Pin Code: Participants designed their own pin code with 6 digits and repeated it twice during registration (consistent with the built-in pin code of iPhone 13 and Huawei Mate30 smartphones). During authentication, the keyboard and the input code would be displayed as visual feedback (see Figure 12c).

Fingerprint and Face ID: We used the built-in fingerprint and face ID authentication of the operating system. Participants were asked to register their right thumb's fingerprint or their face ID following the system's guidance (see Figure 12d and Figure 12e). During authentication, no visual feedback was provided.



(a) *Squeez'In* **(b) Gesture** **(c) Pin code** **(d) Fingerprint** **(e) face ID**

Figure 12: User interface of the five authentication techniques.

To simulate the authentication experience in real life, we designed three scenarios with different phone locations: *At Hand* (hold the phone on the dominant hand with the screen off), *On Table* (sit at a table with the phone placed on it) and *In Pocket* (walk with the phone in the trouser pocket) [18]. In all the scenarios, we distracted the participants by chatting, and a ring sound would be played at a random time. Upon hearing it, the participants were asked to unlock the phone as quickly as possible, simulating the scenario of answering phone calls and notifications.

7.3 Procedure

Participants were first allowed 5 minutes for warm up. They then completed five sessions of tasks, each corresponding to one authentication technique, in random order (according to the output of a shuffle algorithm). For each technique, the participants first completed the registration procedure, and then completed three blocks of authentication tasks, one for each scenario, in random order. In each block, we first started chatting with them to distract their attention. After a random time between 30–60 seconds, a ring sound would be played on the phone, and the participants were required to pick up and unlock the phone using the specified technique as quickly as possible. They were asked to keep trying until completion, and the platform recorded all the attempts for analysis. After the experiment, questionnaires and interviews were conducted to gather their subjective ratings and feedback. The whole study took about 45 minutes.

7.4 Results

In total, We collected data from 18 participants × 5 techniques = 90 registration tasks and 18 participants × 3 scenarios × 5 techniques

= 270 authentication tasks. We used RM-ANOVA for statistical tests, and Friedman test for non-parametric tests.

7.4.1 Registration Time. We measured the registration time as the time spent on entering the code samples, not including the time for designing codes. As showed in Figure 13, the average registration time for *Squeez'In*, gesture, pin code, fingerprint and face ID was 14.5s (SD = 4.7s), 4.1s (SD = 2.6s), 3.6s (SD = 1.9s), 75.6s (SD = 11.9s), and 20.5s (SD = 7.4s) respectively. Significant difference was found between different techniques ($F_{4,68} = 324, p < .001$). *Post hoc* analysis found that *Squeez'In* was significantly faster than fingerprint ($p < .001$) and face ID ($p < .05$), but significantly slower than gesture ($p < .001$) and pin code ($p < .001$).

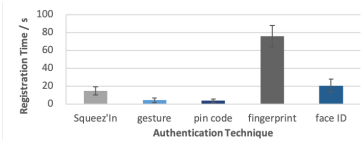
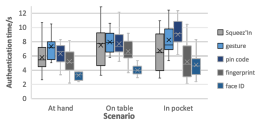


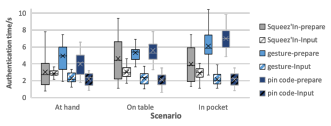
Figure 13: Average registration time for different techniques. Error bar showed one standard deviation.

Compared with gesture and pin code, the registration of *Squeez'In* required more repetitions (3 vs. 2), and performing gestures were generally slower than taps [4, 32], therefore this result was expectable. However, *Squeez'In* was 29% faster than face ID, and even 81% faster than fingerprint, which was inspiring as they all collected user-specific features (behavioral feature or bio-information). Note that in real usage, registration was only performed once for each code, therefore the difference was not crucial to the usability.

7.4.2 Authentication Time. We calculated the authentication time as the elapse between the moment the ring sounded and authentication was accepted (including reaction time, picking up the phone and authenticating). As shown in Figure 14a, the average authentication time for *Squeez'In*, gesture, pin code, fingerprint and face ID across all scenarios was 6.7s (SD = 0.5s), 7.8s (SD = 0.3s), 7.6s (SD = 0.3s), 5.7s (SD = 0.2s) and 4.0s (SD = 0.2s), respectively, which was significantly different ($F_{4,68} = 26.8, p < .001$). In all three scenarios, *Squeez'In* achieved significantly faster authentication speed than gesture and pin code (post-hoc test $p < .05$). While fingerprint and face ID achieved the fastest authentication speed, as they did not require code entering. Meanwhile, scenario was found to yield a significant effect on authentication time ($F_{2,34} = 42.6, p < .001$). As expected, *at hand* yielded the shortest authentication time, while *in pocket* was generally the slowest.



(a) Authentication time.



(b) Preparation time and input time.

Figure 14: Authentication time, preparation time and input time for different techniques.

We then split the authentication time of the three code-based techniques into *preparation time* and *input time*, which denoted the time before and after the first touch event, respectively. As showed in Figure 14b, both scenario ($F_{2,34} = 17.8, p < .001$) and technique ($F_{2,34} = 11.2, p < .001$) was found to significantly affect the preparation time. Post hoc analysis found that the preparation time of *Squeez'In* was significantly shorter than gesture ($p < .01$) and pin code ($p < .001$). We observed that when using *Squeez'In*, participants could learn to start performing the gesture during picking up the phone, which helped shorten the preparation time, especially when *in pocket*.

The input time of *Squeez'In* was significantly longer than gesture and pin code ($F_{2,34} = 22.0, p < .001$), but the difference was not big (< 0.7 s). We speculated this was due to a slower speed of performing squeezing gestures compared with taps, and the usage of long duration in the squeezing gestures. However, the difference in input time was much smaller than that in preparation time. Therefore, *Squeez'In* still showed advantages in terms of the total authentication time in all cases (see Figure 14a).

The average input time of *Squeez'In* across the three scenarios was 2.91s (SD = 0.71s), which was comparable with existing pressure-based pin code authentication techniques (e.g., 3.66s using 4-digit pin code [22] and 2.30s after 10 days of usage [29]). Considering the longer code length in our study (average length = 5.4), *Squeez'In* achieved a faster input speed per digit (0.54s vs 0.92s vs. 0.82s). Moreover, *Squeez'In* allowed the users to perform the input while picking up the phone, therefore further shortening the total authentication time compared with the techniques that required tapping on the touchscreen.

7.4.3 Authentication Accuracy. Consistent with existing works [9, 57], we calculated authentication accuracy as the ratio between the number of successful trials and the total number of detected attempts. We did not analyze the authentication accuracy for face ID, as the total number of attempts was undetermined. When *at hand* and *in pocket*, all the techniques achieved a 100% accuracy except for gesture when *at hand* (94.7%, failed once) and *Squeez'In* when *in pocket* (94.7%, one unintentional squeeze to avoid the phone from slipping away).

When *on table*, the accuracy of the techniques all dropped, with only fingerprint remaining a 100% accuracy. *Squeez'In* achieved 90.0% accuracy with two failed cases. Gesture achieved 94.7% accuracy with one failed case. Among all the failed cases, the participants were able to authenticate successfully in the second trial, confirming the reliability of the techniques. In comparison, the pin code only achieved 78.2% accuracy as one participant forgot the code and retried it five times. No significant different was found between *Squeez'In*, gesture and pin code ($F_{2,34} = 0.551, p = .58$).

7.4.4 Subjective Ratings. We collected the participants' subjective ratings towards the techniques using a 7-point Likert-scale questionnaire (7: most positive, 1: most negative). Dimensions include: *Security* (how hard to be guessed out), *Privacy*, *Social Acceptance* (willingness to use in public), *Generalizability* and *Memorability*. Figure 15 showed the average rating of different techniques.

A Significant difference was found between the five techniques in terms of *Privacy* ($\chi^2(4) = 17.8, p < .001$), *Social Acceptance* ($\chi^2(4) = 18.7, p < .001$), *Generalizability* ($\chi^2(4) = 19.8, p < .001$),

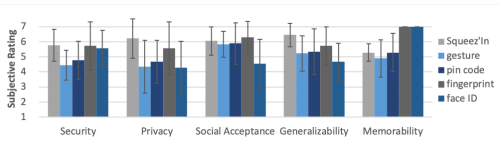


Figure 15: Subjective ratings of different techniques (7: most positive, 1: most negative). Error bar showed one standard deviation.

and *Security* ($\chi^2(4) = 17.0, p = .002$). Among the five techniques, *Squeez'In* received the highest rating on all the five dimensions except for *Memorability* and *Social Acceptance*. However, no significant difference was found between *Squeez'In* and fingerprint on *Social Acceptance* ($\chi^2(2) = 0.40, p = .53$), or between the three code-based techniques on *Memorability* ($\chi^2(2) = 1.75, p = .42$).

8 DISCUSSION

8.1 Feasibility of *Squeez'In*

8.1.1 Theoretical Code Space. In Study 1, we have analyzed the theoretical code space of *Squeez'In* based on users' preferred features, which is greater than pin code and gesture authentication. According to the results in Study 2, we recommended that $N_p = N_d = 2$ was optimal for practical use, resulting in $N = 4^L$. This may seem smaller than pin code (10^L), but *Squeez'In* also leveraged the behavioral biometrics of the user, and could leverage more auxiliary features (see Figure 2a) to significantly improve the distinguishability between gestures. In Study 3, we demonstrated that *Squeez'In* could achieve a FAR of only 2.4% even for the same gestures from different users. Therefore, we believed that the "effective" code space of *Squeez'In* was sufficient for real use.

8.1.2 Security. The design of *Squeez'In* combined code-based and behavioral biometric authentication techniques, achieving a high-security level by combining respective advantages: 1) When designing squeezing codes, a number of features could be leveraged by the users, which formed a vast code space. Compared with gesture and pin code, the user-specific behavioral features make attacking difficult even if the code were known by others (see Section 6.4.4); 2) Compared with fingerprint and face ID, squeezing codes are hidden when not in use. Therefore, they cannot be stolen during sleeping, and users are able to change the codes once the data was attacked. 3) Squeezing gestures are subtle (especially the squeezing procedure features), and could be performed during picking up the phone eyes-freely. Therefore, *Squeez'In* was robust to shoulder-surfing, and was more socially acceptable (see Figure 15).

Regarding the potential drawback of robustness, our SVM-GBDT model was proved effective to recognize the user-specific behavioral features, and tolerate input noise (see the results of the long-term stability study).

8.1.3 Usability. In Study 4, *Squeez'In* achieved faster registration speed than fingerprint and face ID and faster authentication speed than gesture and pin code, which were all mainstream authentication techniques. Besides, *Squeez'In* was more preferred in terms of security, privacy, generalizability and social acceptance, confirming its usability in real scenarios. We deferred this to five reasons: 1) the design of squeezing gestures was intuitive, which were based on the preferred features proposed by the participants (see Study 1); 2) the performing of squeezing gestures fitted humans' motion control ability, thus minimizing the mental and physical effort (see Study 2); 3) the required number of samples during registration was optimized to balance the authentication accuracy and effort (see Figure 8); 4) the authentication accuracy of *Squeez'In* was satisfying; therefore users could successfully authenticate within one trial in most of the time, even during grasping the phone (see Study 4); 5) *Squeez'In* could be used with a single hand and eyes-freely, allowing a wider range of application scenarios (e.g. when the other hand or the eyes were occupied).

In the long-term stability study, some participants mentioned that they became more familiar with the technique over time, and could perform squeezing gestures faster and more comfortably, which suggested a learning trend. Also, providing more samples during registration could help further increase the authentication accuracy (see Figure 8). Therefore, we believed that with more practice and more samples (e.g., using online learning), the usability of *Squeez'In* could be further improved in scenarios with higher demand.

As we used intentional squeezing behavior for authentication, natural squeezing behavior during grasping may lead to unintentional triggering. During the interview in Study 4, participants mentioned that they would design the squeezing gestures such that the combination of touch duration and pressure was different from that in natural squeezing behaviors. And in result, no unintentional authentication was triggered during picking up and holding the phone, only one happened when the phone slipped away.

8.1.4 Memorability. In Figure 15, *Squeez'In* was rated higher than gesture authentication in terms of memorability. According to the interview, this was due to the richer expressiveness of squeezing gestures compared with a simple gesture on only 9 dots. This made the squeezing gestures less abstract, and thus easier to memorize. When designing pin codes, most participants used meaningful digits (e.g. birthday) to help memorize the code. Similarly, this strategy was also observed when designing squeezing gestures in the long-term stability study (see Section 6.5) and Study 4, which they felt was very helpful and interesting. Of course, unlike fingerprint and face ID, squeezing gestures still need to be memorized, which we see as a trade-off between flexibility and usability. And we concluded that memorizing squeezing gestures was achievable and applicable with proper design.

8.2 Design Implications

In Study 1 and Study 2, we found that touch pressure and duration were the most preferred features when designing squeezing gestures, and the users could stably distinguish two levels for each feature. This result not only supported the design of *Squeez'In*, but could also direct other squeezing-based interaction techniques. Moreover, recognizing gripping posture and finger identity would be valuable for further improving the expressiveness of squeezing-based interaction (see Figure 2a). Finally, we found that providing visual feedback enabled the users to distinguish more levels of touch pressure (see Figure 3), which may help techniques that required more precise control (e.g., pointing).

The user studies in this paper demonstrated the usability of the squeezing gesture in authentication. The results and interview in Study 4 also pointed out other application possibilities. For example, the eyes-free interaction ability of *Squeez'In* made it helpful for visually impaired users [1]. And squeezing could be incorporated with other subtle interaction techniques (e.g., [26, 38, 46]) to complete more complex tasks on smartphones, and for more privacy-sensitive scenarios [11, 43].

9 LIMITATION AND FUTURE WORKS

First, in this paper, we emphasized code length, touch pressure and duration as the features when designing squeezing gestures, as these were the most preferred and expressive features. However, as proposed by the participants, other features (e.g., finger identity) could also be leveraged to enrich *Squeez'In*. We planned to improve the algorithm for sensing more features during squeezing, and explore the feasibility of leveraging these features in real use.

Second, we implemented *Squeez'In* on a curved-screen smartphone, as it could provide more information on the bezel during squeezing. Although curved-screen smartphone was popular, this may limit the generalizability of the algorithm to conventional smartphones. During the studies, we observed that squeezing could also affect the capacitive sensors on the edge of the front screen (see Figure 5a). Therefore, sensing squeezing gestures on the front screen (e.g. using CNN to estimate finger orientation [27]) was worth exploring. We deferred this to future work.

Third, we evaluated the authentication performance and usability of *Squeez'In* through simulation and controlled studies, and demonstrated its feasibility in unlocking tasks. Moreover, our participants were highly homogeneous in terms of age. Although this was helpful for ensuring the internal validity of the results, validating the performance in the wild with more participants, more diverse user demographics and more scenarios (including payment and APP login) was also worthwhile. We planned this in future work.

ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China (NSFC) under Grant No. 62132010, and the grant from the Institute for Guo Qiang, Tsinghua University No. 2019GOG0003.

10 CONCLUSION

In this paper, we proposed *Squeez'In*, a technique that leveraged squeezing gestures to support subtle authentication on smartphones. In the first user study, we analyzed the user-defined squeezing gestures to elicit the common features, and selected code length, touch pressure and duration as the most recommended features. We further tested the users' motion control ability during squeezing in terms of touch pressure and duration, and found that two-level was the most applicable density. We implemented *Squeez'In* on a capacitive sensing smartphone, and tested its authentication performance on the data from 21 real users. Results showed that an SVM-GBDT model with 7 samples could reach an accuracy of 99.3% and an F1-score of 0.93. A following 14-day user study verified the memorability and long-term stability of *Squeez'In*. We finally evaluated the usability of *Squeez'In* with four mainstream authentication techniques in three scenarios. Results showed that *Squeez'In* achieved competitive accuracy with gesture and pin code, while receiving significantly faster authentication speed and higher ratings on privacy and generalizability. This paper demonstrated the feasibility of squeezing gestures for interaction, and proposed *Squeez'In* as an efficient, accurate and private authentication technique on smartphones.

REFERENCES

[1] Tousif Ahmed, Roberto Hoyle, Kay Connelly, David Crandall, and Apu Kapadia. 2015. Privacy Concerns and Behaviors of People with Visual Impairments. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3523–3532. https://doi.org/10.1145/2702123.2702334

[2] Adnan Bin Amanat Ali, Vasaki Ponnusamy, and Anbuselvan Sangodiah. 2019. User behaviour-based mobile authentication system. In *Advances in Computer Communication and Computational Sciences*. Springer, 461–472.

[3] Panagiotis Andriotis, Theo Tryfonas, and George Oikonomou. 2014. Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method. In *International conference on human aspects of information security, privacy, and trust*. Springer, 115–126.

[4] Ahmed Arif, Michel Pahud, Ken Hinckley, and WIlliam Buxton. 2013. A Tap and Gesture Hybrid Method for Authenticating Smartphone Users. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Munich, Germany) *(MobileHCI '13)*. Association for Computing Machinery, New York, NY, USA, 486–491. https://doi.org/10.1145/2493190.2494435

[5] Mozhgan Azimpourkivi, Umut Topkara, and Bogdan Carbunar. 2017. Camera Based Two Factor Authentication Through Mobile and Wearable Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 35 (sep 2017), 37 pages. https://doi.org/10.1145/3131904

[6] Uladzislau Barayeu, Nastassya Horlava, Arno Libert, and Marc Van Hulle. 2020. Robust Single-Trial EEG-Based Authentication Achieved with a 2-Stage Classifier. *Biosensors* 10, 9 (2020), 124.

[7] Tobias Boceck, Sascha Sprott, Huy Viet Le, and Sven Mayer. 2019. Force Touch Detection on Capacitive Sensors Using Deep Neural Networks. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 42, 6 pages. https://doi.org/10.1145/3338286.3344389

[8] Mario Parreño Centeno, Yu Guan, and Aad van Moorsel. 2018. Mobile Based Continuous Authentication Using Deep Features. In *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning* (Munich, Germany) *(EMDL'18)*. Association for Computing Machinery, New York, NY, USA, 19–24. https://doi.org/10.1145/3212725.3212732

[9] Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. 2012. *Touch Me Once and i Know It's You! Implicit Authentication Based on Touch Screen Patterns*. Association for Computing Machinery, New York, NY, USA, 987–996. https://doi.org/10.1145/2207676.2208544

[10] Tao Feng, Ziyi Liu, Kyeong-An Kwon, Weidong Shi, Bogdan Carbunar, Yifei Jiang, and Nhung Nguyen. 2012. Continuous mobile authentication using touchscreen gestures. In *2012 IEEE Conference on Technologies for Homeland Security (HST)*. 451–456. https://doi.org/10.1109/THS.2012.6459891

[11] Neil Zhenqiang Gong, Mathias Payer, Reza Moazzezi, and Mario Frank. 2016. Forgery-Resistant Touch-Based Authentication on Mobile Devices. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security* (Xi'an, China) *(ASIA CCS '16)*. Association for Computing Machinery, New York, NY, USA, 499–510. https://doi.org/10.1145/2897845.2897908

[12] Anhong Guo, Robert Xiao, and Chris Harrison. 2015. CapAuth: Identifying and Differentiating User Handprints on Commodity Capacitive Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (Madeira, Portugal) *(ITS '15)*. Association for Computing Machinery, New York, NY, USA, 59–62. https://doi.org/10.1145/2817721.2817722

[13] Christian Holz, Senaka Buthpitiya, and Marius Knaust. 2015. Bodyprint: Biometric User Identification on Mobile Devices Using the Capacitive Touchscreen to Scan Body Parts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3011–3014. https://doi.org/10.1145/2702123.2702518

[14] Feng Hong, Meiyu Wei, Shujuan You, Yuan Feng, and Zhongwen Guo. 2015. Waving Authentication: Your Smartphone Authenticate You on Motion Gesture. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI EA '15)*. Association

for Computing Machinery, New York, NY, USA, 263–266. https://doi.org/10.1145/2702613.2725444

[15] Safaa Hriez, Nadim Obeid, and Arafat Awajan. 2019. User Authentication on Smartphones Using Keystroke Dynamics. In *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems* (Dubai, United Arab Emirates) *(DATA '19)*. Association for Computing Machinery, New York, NY, USA, Article 34, 4 pages. https://doi.org/10.1145/3368691.3368725

[16] Kaori Ikematsu and Shota Yamanaka. 2020. ScraTouch: Extending Touch Interaction Technique Using Fingernail on Capacitive Touch Surfaces. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–10.

[17] Toshiki Iso, Tsutomu Horikoshi, Masakatsu Tsukamoto, and Takeshi Higuchi. 2012. Personal Feature Extraction via Grip Force Sensors Mounted on a Mobile Phone: Authenticating the User during Key-Operation. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia* (Ulm, Germany) *(MUM '12)*. Association for Computing Machinery, New York, NY, USA, Article 30, 4 pages. https://doi.org/10.1145/2406367.2406404

[18] Ryo Izuta, Kazuya Murao, Tsutomu Terada, Toshiki Iso, Hiroshi Inamura, and Masahiko Tsukamoto. 2016. Screen Unlocking Method Using Behavioral Characteristics When Taking Mobile Phone from Pocket. In *Proceedings of the 14th International Conference on Advances in Mobile Computing and Multi Media* (Singapore, Singapore) *(MoMM '16)*. Association for Computing Machinery, New York, NY, USA, 110–114. https://doi.org/10.1145/3007120.3007162

[19] Minpyo Kang, Jejung Kim, Bongkyun Jang, Youngcheol Chae, Jae-Hyun Kim, and Jong-Hyun Ahn. 2017. Graphene-Based Three-Dimensional Capacitive Touch Sensor for Wearable Electronics. *ACS Nano* 11, 8 (2017), 7950–7957. https://doi.org/10.1021/acsnano.7b02474 arXiv:https://doi.org/10.1021/acsnano.7b02474 PMID: 28727414.

[20] David Kim, Paul Dunphy, Pam Briggs, Jonathan Hook, John W. Nicholson, James Nicholson, and Patrick Olivier. 2010. Multi-Touch Authentication on Tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 1093–1102. https://doi.org/10.1145/1753326.1753489

[21] Sowndarya Krishnamoorthy, Luis Rueda, Sherif Saad, and Haytham Elmiligi. 2018. Identification of User Behavioral Biometrics for Authentication Using Keystroke Dynamics and Machine Learning. In *Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications* (Amsterdam, Netherlands) *(ICBEA '18)*. Association for Computing Machinery, New York, NY, USA, 50–57. https://doi.org/10.1145/3230820.3230829

[22] Katharina Krombholz, Thomas Hupperich, and Thorsten Holz. 2016. Use the Force: Evaluating Force-Sensitive Authentication for Mobile Devices. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. USENIX Association, Denver, CO, 207–219. https://www.usenix.org/conference/soups2016/technical-sessions/presentation/krombholz

[23] Masashi Kudo and Hayato Yamana. 2018. Active Authentication on Smartphone Using Touch Pressure. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings* (Berlin, Germany) *(UIST '18 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 96–98. https://doi.org/10.1145/3266037.3266113

[24] Yantao Li, Hailong Hu, and Gang Zhou. 2018. Using data augmentation in continuous authentication on smartphones. *IEEE Internet of Things Journal* 6, 1 (2018), 628–640.

[25] Chen Liang, Chun Yu, Xiaoying Wei, Xuhai Xu, Yongquan Hu, Yuntao Wang, and Yuanchun Shi. 2021. *Auth+Track: Enabling Authentication Free Interaction on Smartphone by Continuous User Tracking*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411764.3445624

[26] Markus Löchtefeld, Christoph Hirtz, and Sven Gehring. 2013. Evaluation of Hybrid Front- and Back-of-Device Interaction on Mobile Devices. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia* (Luleå, Sweden) *(MUM '13)*. Association for Computing Machinery, New York, NY, USA, Article 17, 4 pages. https://doi.org/10.1145/2541831.2541865

[27] Sven Mayer, Huy Viet Le, and Niels Henze. 2017. Estimating the Finger Orientation on Capacitive Touchscreens Using Convolutional Neural Networks. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. Association for Computing Machinery, New York, NY, USA, 220–229. https://doi.org/10.1145/3132272.3134130

[28] Weizhi Meng, Duncan S Wong, Steven Furnell, and Jianying Zhou. 2014. Surveying the development of biometric user authentication on mobile phones. *IEEE Communications Surveys & Tutorials* 17, 3 (2014), 1268–1293.

[29] Zhangyu Meng, Jun Kong, and Juan Li. 2021. Utilizing binary code to improve usability of pressure-based authentication. *Computers & Security* 103 (2021), 102187. https://doi.org/10.1016/j.cose.2021.102187

[30] Kazuya Murao, Hayami Tobise, Tsutomu Terada, Toshiki Iso, Masahiko Tsukamoto, and Tsutomu Horikoshi. 2014. Mobile Phone User Authentication with Grip Gestures Using Pressure Sensors. In *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia* (Kaohsiung, Taiwan) *(MoMM '14)*. Association for Computing Machinery, New York, NY, USA, 143–146. https://doi.org/10.1145/2684103.2684116

[31] Hyoungsik Nam, Ki-Hyuk Seol, Junhee Lee, Hyeonseong Cho, and Sang Won Jung. 2021. Review of capacitive touchscreen technologies: Overview, research trends, and machine learning approaches. *Sensors* 21, 14 (2021), 4776.

[32] Matei Negulescu, Jaime Ruiz, Yang Li, and Edward Lank. 2012. Tap, Swipe, or Move: Attentional Demands for Distracted Smartphone Input. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (Capri Island, Italy) *(AVI '12)*. Association for Computing Machinery, New York, NY, USA, 173–180. https://doi.org/10.1145/2254556.2254589

[33] KENNEDY OKOKPUJIE, ABERE REUBEN, JOYCE C OFOCHE, BASUO J BIOBELEMOYE, and IMHADE PRINCESS OKOKPUJIE. 2021. A COMPARATIVE ANALYSIS PERFORMANCE OF DATA AUGMENTATION ON AGE-INVARIANT FACE RECOGNI-TION USING PRETRAINED RESIDUAL NEURAL NETWORK. *Journal of Theoretical and Applied Information Technology* 99, 6 (2021).

[34] Changhyun Pang, Ja Hoon Koo, Amanda Nguyen, Jeffrey M Caves, Myung-Gil Kim, Alex Chortos, Kwanpyo Kim, Paul J Wang, Jeffrey B-H Tok, and Zhenan Bao. 2015. Highly skin-conformal microhairy sensor for pulse signal amplification. *Advanced materials* 27, 4 (2015), 634–640.

[35] Kang Ryoung Park. 2011. Finger vein recognition by combining global and local features based on SVM. *Computing and Informatics* 30, 2 (2011), 295–309.

[36] Pramuditha Perera and Vishal M Patel. 2018. Dual-minimax probability machines for one-class mobile active authentication. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 1–8.

[37] Philip Quinn, Seungyon Claire Lee, Melissa Barnhart, and Shumin Zhai. 2019. Active Edge: Designing Squeeze Gestures for the Google Pixel 2. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300504

[38] Hanae Rateau, Yosra Rekik, Edward Lank, and Keiko Katsuragawa. 2021. *Edge-Mark Menu: A 1D Menu for Curved Edge Display Smartphones*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3447526.3472017

[39] Christian Rendl, Patrick Greindl, Kathrin Probst, Martin Behrens, and Michael Haller. 2014. Presstures: Exploring Pressure-Sensitive Multi-Touch Gestures on Trackpads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 431–434. https://doi.org/10.1145/2556288.2557146

[40] Mohamed Azard Rilvan, Jedrik Chao, and Md Shafaeat Hossain. 2020. Capacitive Swipe Gesture Based Smartphone User Authentication and Identification. In *2020 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*. 1–8. https://doi.org/10.1109/CogSIMA49017.2020.9215998

[41] Mohamed Azard Rilvan, Kolby Isiah Lacy, Md Shafaeat Hossain, and Bing Wang. 2016. User authentication and identification on smartphones by incorporating capacitive touchscreen. In *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 1–8.

[42] Baljit Singh Saini, Parminder Singh, Anand Nayyar, Navdeep Kaur, Kamaljit Singh Bhatia, Shaker El-Sappagh, and Jong-Wan Hu. 2020. A Three-Step Authentication Model for Mobile Phone User Using Keystroke Dynamics. *IEEE Access* 8 (2020), 125909–125922.

[43] Roland Schlöglhofer and Johannes Sametinger. 2012. Secure and Usable Authentication on Mobile Devices. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia* (Bali, Indonesia) *(MoMM '12)*. Association for Computing Machinery, New York, NY, USA, 257–262. https://doi.org/10.1145/2428955.2429004

[44] Martin Schmitz, Jürgen Steimle, Jochen Huber, Niloofar Dezfuli, and Max Mühlhäuser. 2017. Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1001–1014. https://doi.org/10.1145/3025453.3025663

[45] Sougata Sen and Kartik Muralidharan. 2014. Putting 'pressure' on mobile authentication. In *2014 Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. 56–61. https://doi.org/10.1109/ICMU.2014.6799058

[46] Shaikh Shawon Arefin Shimon, Sarah Morrison-Smith, Noah John, Ghazal Fahimi, and Jaime Ruiz. 2015. Exploring User-Defined Back-Of-Device Gestures for Mobile Devices. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Copenhagen, Denmark) *(MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 227–232. https://doi.org/10.1145/2785830.2785890

[47] Sung-Ho Shin, Sangyoon Ji, Seiho Choi, Kyoung-Hee Pyo, Byeong Wan An, Jihun Park, Joohee Kim, Ju-Young Kim, Ki-Suk Lee, Soon-Yong Kwon, et al. 2017. Integrated arrays of air-dielectric graphene transistors as transparent active-matrix pressure sensors for wide pressure ranges. *Nature communications* 8, 1 (2017), 1–8.

[48] Chen Sun, Yang Wang, and Jun Zheng. 2014. Dissecting pattern unlock: The effect of pattern strength meter on pattern selection. *Journal of Information Security and Applications* 19, 4-5 (2014), 308–320.

[49] José Torres, Sergio de los Santos, Efthimios Alepis, and Constantinos Patsakis. 2019. Behavioral Biometric Authentication in Android Unlock Patterns through

Machine Learning.. In *ICISSP*. 146–154.

[50] Radu-Daniel Vatavu and Jacob O. Wobbrock. 2022. Clarifying Agreement Calculations and Analysis for End-User Elicitation Studies. *ACM Trans. Comput.-Hum. Interact.* 29, 1, Article 5 (jan 2022), 70 pages. https://doi.org/10.1145/3476101

[51] Emanuel von Zezschwitz, Malin Eiband, Daniel Buschek, Sascha Oberhuber, Alexander De Luca, Florian Alt, and Heinrich Hussmann. 2016. On Quantifying the Effective Password Space of Grid-Based Unlock Gestures. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia* (Rovaniemi, Finland) *(MUM '16)*. Association for Computing Machinery, New York, NY, USA, 201–212. https://doi.org/10.1145/3012709.3012729

[52] Kensuke Wagata and Andrew Beng Jin Teoh. 2022. Few-Shot Continuous Authentication for Mobile-Based Biometrics. *Applied Sciences* 12, 20 (2022), 10365.

[53] Graham Wilson, Stephen Brewster, and Martin Halvey. 2013. Towards utilising one-handed multi-digit pressure input. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 1317–1322.

[54] Cong Wu, Kun He, Jing Chen, Ruiying Du, and Yang Xiang. 2020. CaIAuth: Context-Aware Implicit Authentication When the Screen Is Awake. *IEEE Internet of Things Journal* 7, 12 (2020), 11420–11430.

[55] Xin Yi, Chen Wang, Xiaojun Bi, and Yuanchun Shi. 2020. *PalmBoard: Leveraging Implicit Touch Pressure in Statistical Decoding for Indirect Text Entry*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376441

[56] Mingyuan Zhong, Chun Yu, Q. Wang, Xuhai Xu, and Y. Shi. 2018. ForceBoard: Subtle Text Entry Leveraging Pressure. *Proceedings of the 2018 CHI Conferen10.1145/3266037.3266113ce on Human Factors in Computing Systems* (2018).

[57] Wen Zhu, Nancy Zeng, Ning Wang, et al. 2010. Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *NESUG proceedings: health care and life sciences, Baltimore, Maryland* 19 (2010), 67.