Stop Unauthorized Access to Your Smart Devices

Yutong Liu*, Linghe Kong*, Yifeng Cao*, Vahan Sarafian† and Long Cheng‡
*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
Email: {isabelleliu, linghe.kong, yyshxyh2013}@sjtu.edu.cn
†Institut national des sciences appliqu, Lyon, France
Email: vahan.sarafian@gmail.com
‡School of Computing, Clemson University, USA
Email: lcheng2@clemson.edu

***Abstract*—Smart devices (e.g., smartphones or tablets) have become an indispensable part of our daily lives for conducting mobile payment transactions, and storing both corporate and personal sensitive data. As a result, unauthorized access to smart devices can result in a catastrophic security breach. Lock screen provides the first line of defense against unauthorized access to smart devices, where users typically use PIN, pattern of drawing, or biometric to unlock their devices. Unfortunately, recent studies have revealed that individual unlocking methods are insufficient to prevent unauthorized access to smart devices.**

In this paper, we aim to increase the security barrier of smart device unlocking. We present the CP^3, a combined unlocking framework to achieve highly secure and usable authentication for commodity smart devices. We address several challenges of combing unlocking methods from different modalities, such as the high reliability and low latency. We implement a prototype of our approach based on the Android platform, which selects the fingerPrint authentication, bluetooth transmission Power authentication and facial Pattern verification as our typical Combination for the secure unlocking. We have made the source code of our implementation public[1]. Real-world experiments demonstrate the effectiveness of our solution. CP^3 achieves 88% accuracy and 2.88s operation latency, which guarantees both good user experience and high security level compared with existing methods.

***Index Terms*—Smartphone authentication; Combined unlocking; Fingerprint; Auxiliary authentication; Facial recognition**

I. Introduction

Smart devices (*e.g.*, smartphones or tablets) are becoming attractive targets for attacks as they are increasingly storing a tremendous amount of sensitive information [1]. Unauthorized access to smart devices may lead to catastrophic security breaches in the case these devices get lost or stolen. A Pew study from 2017 reported that 72% of people in 17 advanced economies like North America and much of Europe depend entirely on a smartphone to access online services and information [2]. An attacker could steal usernames and passwords used to access apps and online services extracted from a smartphone, resulting in personal information disclosure and legal consequences, etc. For example, numerous cases of celebrities losing their phones with private photos and secret information have been reported on the news. According to a Gartner group forecast [3], the global mobile payment applications will get over 450 million users and a transaction value of over US$721 billion by the end of the year 2017. Adversaries that gain unauthorized access to smart devices may jeopardize banking and payment information stored on these devices, leading to significant financial losses.

Locking screen provides the first line of defense against unauthorized access to the contents of a lost or stolen smart device. Typically, it requires a secret code (*e.g.*, PIN, drawing a pattern, or biometric) to gain access to their devices [4]. Unfortunately, recent studies have revealed that individual unlocking (*i.e.*, authentication) methods are insufficient to prevent unauthorized access to smart devices. The shoulder-surfing attack can acquire the secret code directly by observations, especially for traditional texture passwords, PINs or patterns [5]. Smudge attack succeeds in bypassing pattern-drawing authentication, where the oily residues left on the screen will be simply captured to guess the true secret code [6]. Even the fingerprint authentication, which is the most popular unlocking in shelved smartphones, can be broken by a simulated finger model created by hackers in Chaos Computer Club [7].

Biometrics-based unlocking (*e.g.*, fingerprints, face [8], voice or eye patterns [9, 10], gait and gesture behaviors [11, 12]) has attracted considerable attention recently due to its uniqueness and thus high immunity to the accurate replication by adversaries [10]. However, like the heart-beating [13] or breathing authentication [14], they are not practical for mobile users. And authentications like Eyeveri [10] or key-stroke based identification [15] have long operating latency to achieve high accuracy.

To increase the security of individual unlocking, Saevanee *et al.* proposed the combined authentication [16]. They designed a multi-modal behavioral biometric authentication system to overcome the weaknesses of individual protections. Despite different concepts of the combined authentication [17, 18] have been proposed in recent years, there still exist challenges that hinder the practical usage of such techniques. Firstly, selected methods for combination should be diverse. For security enhancement, combined authentication should have more than one input to defend against more attacks. Authors in [18] chose different methods only based on one input. The face, periocular and iris recognitions make use of one image to get three results. The adversaries can break the whole authentication simply by attacking this image. Secondly,

[1] https://goo.gl/oUwSfZ

protection on smart devices should also continue when they have been unlocked. To the best of our knowledge, there is no combination can monitor devices after unlocking process. Thirdly, average operation delay should be acceptable for users. Former studies have ignored this practical requirement. In [16], authors select behavior, keystroke and linguistic profiling to combine pursuing the high accuracy. Nevertheless, the operation on this authentication requires three behaviors (walk, tap and speak), which will waste too much time on input and make it unpractical for unlocking.

To address the above challenges, in this work, we introduce a new unlocking design to increase the security barrier of smart device authentication, named CP^3: a **C**ombined unlocking protection including finger**P**rint authentication, Bluetooth transmission **P**ower authentication and facial **P**attern verification. The contributions from this work are summarized as follows:

- We propose a new combined unlocking framework to achieve highly secure and usable authentication for commodity smart devices. Our design not only defends against various unlocking attacks (including forgery attack, hacking attack and snatch attack, etc), but also reduces the complexity and latency on combined authentications.
- We exploit and present a new Bluetooth authentication method making use of equipped devices for unlocking. Transmission powers are modulated as the covert channel to avoid wireless eavesdropping, and the background periodical detection ensures that unauthorized users cannot access to the device even after the legitimate user has unlocked it.
- We implement the CP^3 prototype on Android platform and make the source code of our implementation public. Real-world experiments achieve 88% authentication accuracy and 2.88s operation latency, which prove the feasibility of our solution.

The remaining part of the paper is organized as follows: Section II surveys the related work. Section III describes the attack model and design goals. Section IV and V introduce the details of the design and implementation of CP^3, respectively. Section VI reports the evaluation results. Section VII provides a security analysis of our approach and discusses in-depth practical issues. Finally, Section VIII concludes the paper.

II. Related Work

There are various unlocking methods to protect smart devices, but these methods also bring new attacks. In this section, we will discuss related work in two aspects: unlocking schemes and attacks.

A. Unlocking Schemes

Generally, there are three unlocking schemes: traditional authentication, biological authentication, and auxiliary authentication. Developments have been made in each scheme, but shortcomings still exist.

Traditional Authentication. It is considered as "what you know" authentication scheme. Traditional authentication includes texture passwords, PIN codes and the Android Pattern, which are most likely exposed to various adversaries [19]. XSide [19] exploits the front and the back of smartphones to enter stroke-based passwords. This design can enhance such authentications resistance to shoulder surfing. To avoid taping both on the front and back, PassMatrix [20] leverages a graphics-based login indicator. This indicator randomly generates pass-images, to achieve higher security and easier operation. While they enhance the security against the shoulder surfing, the hardware overhead cannot be neglected and they provide little resistance to the smudge attack.

Biological Authentication. It is considered as "what you are" scheme. Biological authentication recognizes and unlocks smart devices based on biological features, such as fingerprints, face [8], voice or eye patterns [9, 10], gait and gesture behaviors [11, 12]. Recent studies have shown that they are vulnerable to imitating attacks [21]. To avoid this, EyeVeri [10] captures eye-movements and extracts gaze pattern for access. However, the use of EyeVeri is limited by its unneglectable time delay, which can last more than 5 secs to achieve better accuracy. Key-stroke based authentication [15] exploits tapping strength on the screen which claims difficult to be copied. More than 30 profiles should be manually input which brings a complicated setting process. Additionally, CardiacScan [13] provides a novel approach to implement continuous heart-based user authentication. Also in [14], the users' breathing gestures are used as a biometric signature. But such authentications are not practical for current smartphone users.

Auxiliary Authentication. It is considered as "what you have" scheme. Authentication based on auxiliary devices is newly proposed in recent years. The paper [22] firstly studied using a smartphone to control the lock/unlock status of other smart devices, i.e, smart doors or smart watches, by Bluetooth transmission. But the instructions transmitted are just put in the plain text in Bluetooth packets, which will be hacked easily. Besides, IAuth [23] records behavioral characteristics through smart watches to verify the identity. But its training overhead cannot be ignored.

B. Unlocking Attacks

With the development of unlocking methods, more new attacks are also proposed. At first, *shoulder surfing* is considered as the most direct way to break unlocking system [5]. The paper [24] gives an improved *WiFi influence attack*, but this attack is not stable, as the WiFi signals can also be disrupted by other body movements except finger moving. A more accurate *video based attack* is studied. It constructs pattern password simply by analyzing the filmed finger footage [25, 26].

For auxiliary authentication, the typical attacks are *hacking attack* and *eavesdropping attack*, which aim to take control of any side of device or listen on the legitimate channel. Especially for Bluetooth communication, hackers will adapt malformed objects, which can control the victim's device to list the attackers' device as the trusted pair device [27]. They will also eavesdrop on a legitimate transmission to learn the password in a second, which is a completely random 64 or 128-bit key [28].

For biological authentication, *forgery attack* [7] and *imitating attack* [29] are widely used. The forgery attack is to acquire secret codes from known clues, like a simulated finger model generated from the fingerprints left somewhere else. It has been used by Chaos Computer Club to break fingerprints authentication [7]. And the imitating attack can mimic users' manners including gait patterns, eye movement patterns, or touch gestures [29].

From the above discussions, single authentication is not secure enough against various attacks. Combining multiple methods for higher performance is becoming another dimension for safely unlocking. In addition to PIN codes, the authors in [17] collect physical status, i.e, acceleration, pressure, time features when tapping. It assumes the consistent behaviors for users which is difficult to achieve in reality. The multi-model authentication proposed in [16] combines three biometrics: voice, facial features, and signature, but the problem lies in its unacceptable time delay. The framework in [18] makes an improvement by allocating weight factors to biometrics. The leverage on one image for three recognitions (face, periocular, iris) leaves this image as the simple target to be compromised. So, the low level of security and the bad user experience are important barriers to the currently combined authentication, which are also our motivations on this paper.

III. Attack Model and Design Goals

In this section, we first describe the threat model of this work. Then, we discuss the design objectives of CP^3.

A. Attack Model

We consider that an adversary wants to access the sensitive information and controls privacy-concerned applications on a target device which is protected by CP^3. We concern the following six different attacks that may bypass mobile device authentications:

- **Brute-force attack:** The adversary tries every possible combination of passwords to bypass the secret code based authentication [30].
- **Fingerprint forgery attack:** The adversary will get fingerprints left on somewhere else and make a simulated model to fool the fingerprint authentication [7].
- **Facial forgery attack:** The adversary uses a forged image (i.e., using a previously used photo to substitute the true facial patterns) to fool the face authentication [31].
- **Snatch attack:** This attack specially works when the device has been unlocked. The adversary may directly snatch the device when the owner is using it [32].
- **Eavesdropping attack:** The adversary will monitor the transmission on legitimate channel all the time to get packets that contain private information [28].
- **Device hacking:** The adversary tries to control the Bluetooth devices for executing designed commands [27].

We also assume that mobile applications on the target device are trustworthy. It is free of spyware, password modified capability and algorithm vulnerability. For example, no password attack application is pre-installed and the devices are not equipped with spied hardware. In addition, we assume that attackers do not know the exact combined methods before attacking and cannot try infinitely without being perceived by legitimate users.

B. Design Goals

Under the above attack model, we have three security goals:

- Defeat any individual unlocking attack that attempts to circumvent the smart devices authentication.
- Improve the robustness against potential combined attacks by increasing the difficulties that an attack could bypass the authentication.
- Achieve continuous authentication of smart device users to defend against snatch attacks.

To achieve our security goal without compromising the authentications usability, we identify the following design objectives for developing a secure and usable combined unlocking scheme.

- **Diversity:** There should be more than one input for the whole authentication. Unlike the paper [18] introduced in section II-A, one input image will be the target for attackers to simply bypass the authentication.
- **Low latency:** The simple combination of multiple authentications inevitably increases the operation time of authentication. The authentication in [16] requires all walking, tapping and speaking profiles, which causes an extremely long latency. So a practical combined unlocking scheme should be time-efficient and user-friendly.
- **Continuous authentication:** To the best of our knowledge, existing combined authentications [16]-[18] do not protect the smart devices after the unlocking process. In case of snatch attacks, the continuous monitoring is necessary to detect and react immediately to block attackers when using devices.

IV. Design of CP^3

In this section, we first introduce the rationale of combining different unlocking methods in CP^3. We then present the design overview, followed by the detailed description of key components, including the Bluetooth based auxiliary authentication and facial pattern verification.

A. Design Choices

In CP^3, we combine three different methods to prevent unauthorized access to smart devices: 1) fingerprint authentication; 2) Bluetooth based auxiliary authentication; and 3) facial pattern based authentication. We make these design choices due to their favorable features for practical deployment, including low overhead/cost, high accuracy, and short latency. Firstly, all these three methods are readily available on commodity smart devices. For example, mobile phones are typically equipped with standard fingerprint sensors, Bluetooth interfaces, and cameras. Thus, our approach will not incur extra equipment overhead/cost. Secondly, these methods can individually achieve high detection rates. For instances, deep learning based fingerprint and facial recognition algorithms

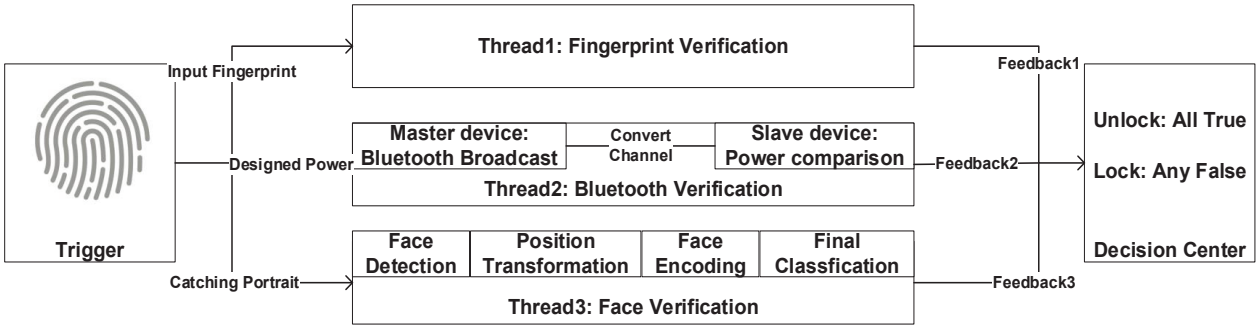


Fig. 1. Authentication process in CP^3

are recently shown to be quite reliable [8, 33]. In particular, we exploit pre-processing techniques in facial recognition to further reduce possible inaccuracies. For the Bluetooth based authentication, we utilize the covert channel in Bluetooth communication to improve its security. Thirdly, these methods individually incur reasonable latency, which guarantees the usability of our CP^3. We seamlessly integrate them to achieve a secure and usable unlocking scheme, where the the user-side operation is simply touch the fingerprint sensor while watching the screen. In Section VII, we report the accuracy and latency of experimental results and demonstrate the feasibility of our design.

B. System Overview

An unlocking scheme is normally composed of two stages: setting stage (*e.g.*, data collection and feature extraction) and authentication stage, which are introduced as follows.

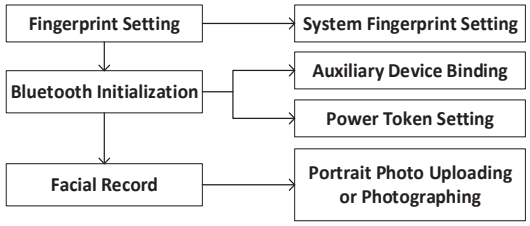


Fig. 2. Setting steps in CP^3

1) Setting: When a user initiates CP^3 at the first time, there are three setting steps as shown in Fig. 2:

- *Fingerprint settings:* It first collects the user's fingerprints, *e.g.*, sampling center and edge fingerprints for three times.
- *Bluetooth initialization:* It then asks the user to specify her secret power token (*e.g.*, k-bit key), which will be transmitted via the Bluetooth covert channel from the auxiliary device to the master device for authentication. We use the standard Bluetooth low energy (BLE) mode. We refer the authentication device as the master device, and the auxiliary device (*e.g.*, a smart watch) as a slave/peripheral device.
- *Facial record setting:* The user finally uploads at least one clear portrait photo or takes a camera photo using the mobile device (glass wearing is acceptable for authentication).

2) Authentication Process: Fig. 1 illustrates the authentication process in CP^3. There are three parallelled threads in the authentication process. Once the fingerprint sensor is triggered, the camera captures the user's facial images. Meanwhile, the Bluetooth module begins to send authentication messages and wait for feedback. For the thread 1, when capturing an input from the fingerprint sensor, we compare the input fingerprint with the training dataset. If their difference can be accepted, CP^3 returns `True` feedback on the first authentication, and then passes the result to the final decision center.

For the thread 2, the master device modulates the transmission power of Bluetooth broadcasts depending on the pre-specified secret power token. This modulation is considered as the convert channel to convey the token message. The auxiliary device continuously scans the wireless channels to receive any broadcasts from the master device. Received signal strength of the broadcast messages will be analyzed to decode the secret token. If the decoded token matches with the recorded one in the setting stage, the `True` feedback for Bluetooth authentication will be sent to the master device.

For the thread 3, the camera catches three frames of the portrait photo as its input. Through our facial recognition algorithms, two conditions are analyzed: (i) if the differences among three captured frames are below an empirical threshold; and (ii) if the captured frame matches the pre-stored pictures in the setting phrase. If they are all satisfied, the facial authentication returns the `True` feedback.

If three feedbacks are all `True`, the device will be unlocked. Otherwise, the authentication process fails.

C. Secure Bluetooth Based Auxiliary Authentication

CP^3 uses a covert channel to convey the secret token by modulating the transmission power of Bluetooth messages, which can hide the token deeply into normal traffics [34]. On the master device, the specified transmission power sequence (*i.e.*, secret token) is represented as a k-bit key (*e.g.*, a 5-bit token 01001 means "low, high, low, low, high"), which will be modulated on k broadcasts. Then, the feedback will be received from the auxiliary paired device.

We use the Received Signal Strength Indication (RSSI) to measure the received power at the auxiliary device. After receiving k broadcast messages, the auxiliary device gets k

RSSI values $(\delta_1, \delta_2, ..., \delta_k)$. CP^3 firstly calculates the arithmetic mean $\overline{\delta}$ of these k RSSIs. If $\delta_i > \overline{\delta}$, it is then considered as the high level (*i.e.*, bit 1) and smaller ones are the low level (*i.e.*, bit 0). By comparing the final decoded result with the pre-specified secret token, we obtain the authentication output.

Continuous Authentication Support: Our design inherently supports continuous authentication and thus can defend against the snatch attack. For example, the Bluetooth communication module runs as a background thread, and periodical detects the presence of the master device. Any failure can cause the immediate locking on the screen.

D. Efficient Facial Pattern Based Authentication

The facial pattern verification process is composed of four steps: 1) face detection, 2) position transformation, 3) face encoding, and 4) the final classification.

Face detection. This step is mainly to get the basic structure of a face from a photo. To avoid the influence of lightness, CP^3 firstly changes the color photo to a gray-scale map. Histogram of Oriented Gradients (HOG) is then used [35] to find the face area passed to the next step. It focuses on every single pixel and its surroundings. The gradients between them are denoted by the directions of arrows, directing from a lighter pixel to a darker one. In particular, to reduce the analysis delay, CP^3 partitions the image into 16×16 small squares and selects the "strongest" arrow which appears most to represent this square. The HOG figure of a face can be presented by these arrows, and then the face will be detected by comparing to the known HOG patterns in the setting stage.

Position transformation. It is inevitable that the faces in some photos are not towards the center. Making use of face landmark estimation [36], the problem can be simplified to locate 68 face landmarks on photos. No matter which direction people look, the relative positions of eyes and mouths will not change. Actually, the process of finding landmarks is a regression process from appearance to the basic shape. So the main purpose is to train a regressor using gradient tree boosting algorithm.

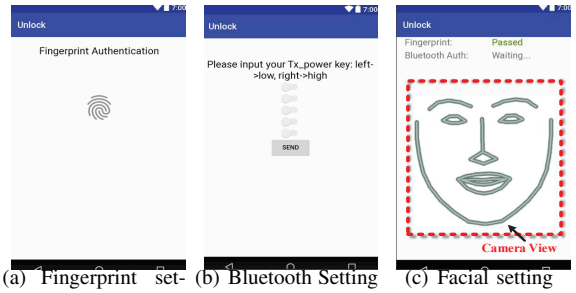
We next transform these positions to the center. Instead of fancy 3D wrap, which will introduce distortions to the changed pictures, CP^3 uses affine transformations including rotate, scale and shear to preserve parallel lines. Such transformation can improve the accuracy of the next measurement.

Face encoding. This step extracts features for deep learning based classification. We use a deep convolutional neural network for model training and get the face encoding. For instance, CP^3 uses the OpenFace [37], a widely used tool for deep learning based facial pattern classification, to get 128 measurements for our input images. It encodes a face image to a series of numbers.

Final classification. CP^3 uses linear SVM classifier [38] to find any match in the database for authentication. This step only takes milliseconds from input to classification output.

V. Implementation

To demonstrate the feasibility of our approach, we have implemented a prototype, which uses multiple off-the-shelf tools and libraries on Android platform. Our prototype is developed on Android Studio 2.3.3, where the compile sdk version is 21 and the build tool version is 25.0.0. In this section, we present key implementation aspects in our prototype. We also provide the source code of our implementation[2].



(a) Fingerprint setting (b) Bluetooth Setting (c) Facial setting

Fig. 3. CP^3 user interfaces in the setting stage

A. CP^3 Prototype

We use a mobile phone equipped with Android 6.0.1 system (API 23) and Bluetooth 4.2 version as the master device. For the auxiliary device, we use a smart watch equipped with Android 5.1 system (API 22) and Bluetooth 4.0 version. For the Bluetooth communication, the transmission range is around 10 meters, and the channel band is ISM 2.4GHz. Its transmission power has four levels: -18dBm, -6dBm, 0dBm and 3dBm. In our prototype, we only use the low level (*i.e.*, -18dBm) and the high level (*i.e.*, 3dBm) to modulate broadcast messages for the covert channel communication.

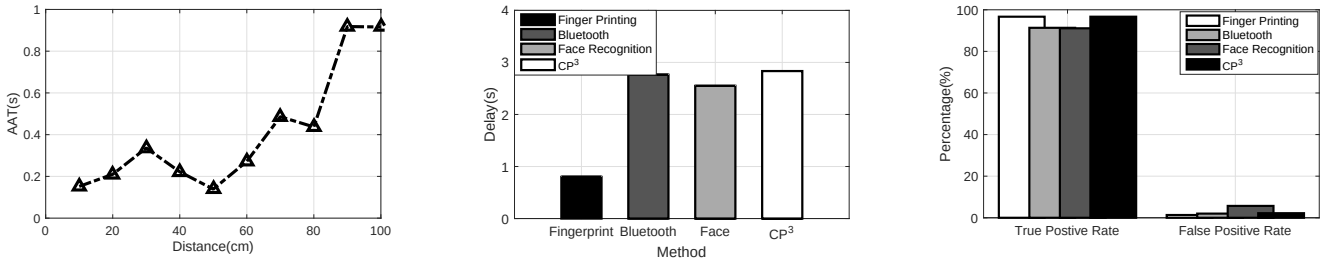
Fig. 3 shows the user interfaces for the fingerprint setting, the secret power token setting, and the facial pattern setting, respectively. In this prototype, the Bluetooth authentication frequency is 3 minutes once and the secret power token is a 5-bit key. In the authentication stage, once the fingerprint sensor captures any input, the fingerprint, Bluetooth and facial pattern recognition threads start in a parallel manner:

- **Fingerprint.** The implementation of fingerprint function is based on the built-in fingerprint package (*i.e.*, four classes in *FingerprintManager*) provided by Android platform.
- **Bluetooth.** *RepeatSendActivity* is in charge of sending modulated broadcasts and receiving the feedback from the auxiliary device. We use timer-based scanning to implement the continuous monitoring.
- **Facial pattern recognition.** We develop three activities for the face authentication: *FaceInitialization*, *FaceRecognition* and *FaceVerification*. The implementation of recognition is based on the offline Face++ package and SVM package provided by Android platform.

B. Permission Request

Considering the dynamic permission request from Android 6.0, we need the following permissions set by users (permissions only need to be given once at the initial phrase of CP^3):

[2]https://goo.gl/oUwSfZ



(a) The relationship between distance and AAT

(b) Authentication delay

(c) Authentication accuracy

Fig. 4. CP^3 vs. Individual Authentication

- *Positioning permission:* Bluetooth communication concerns about the location information. Only when such permission is given, the Bluetooth module can work.
- *Floating window permission:* To shield HOME button, we design the interfaces of our app on floating windows. But Android adds this permission to prevent rogue software occupying the screen maliciously. Thus, we need to ask users to give this permission.
- *Camera permission:* The camera access permission should be enabled so as to capture facial pictures.

VI. Performance Evaluation

In this section, we evaluate the performance of CP^3 compared with three individual authentications. We ask 15 volunteers, including 10 males and 5 females, to test each authentication method for 20 attempts, including 10 attempts for true positive test and another 10 attempts for false positive test. Our experiments aim to answer the following questions:

- Does our approach incur a large authentication delay, compared with other individual baseline authentications?
- How about the detection accuracy, including the true positive rate and false positive rate?
- What are the impacts if we allow multiple attempts in the authentication process?

A. CP^3 vs. Individual Authentication

In this experiment, it allows only one unlocking attempt per authentication. That is, if a user does not pass any of the individual authentications among fingerprint, Bluetooth and face, the authentication is failed. The true positive rate (TPR) in this subsection is defined as the value that the number of the true feedback for true inputs divided by the number of true verification. The false positive rate (FPR) is also defined as the value that the number of the true feedback for false inputs (*i.e.*, wrong fingerprints, wrong transmission power, and fake facial images) divided by the number of false verification.

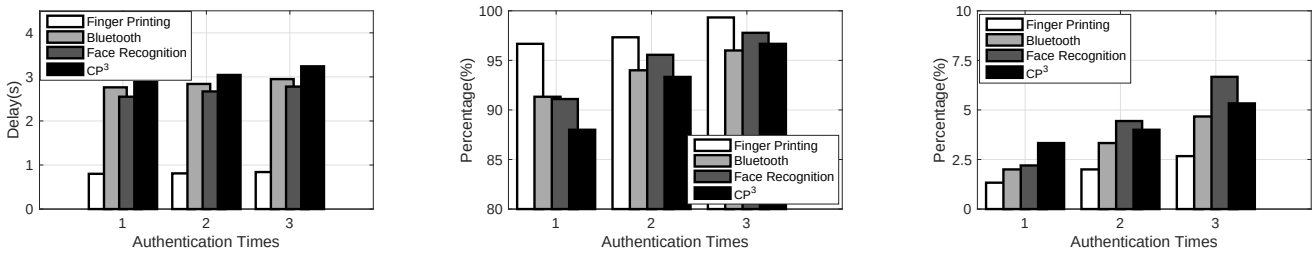
For fingerprint verification, each participant is required to record one finger in the system. To test the TPR, each participant presses the same finger on the sensing area for 10 attempts to see the result. Evaluations show that the recognition rate achieves up to 96%. For the FPR, each participant presses a different finger on the sensor. For instance, one records the index fingerprint but presses his ring finger on the screen. It shows that FPR is as long as 1.33%. We record the authentication delay for each correct recognition. The average authentication time (AAT) is around 0.8s.

For Bluetooth verification, participants are required to customize their own transmission power and start to test the unlocking results. The AAT of this method depends on the distance, as shown in Fig. 4(a). Longer distance will increase the transmission time on Bluetooth communication. According to this, we ask volunteers to wear the watch on the left hand and hold the phone in right hand, keeping an average distance 40 cm between two devices. The corresponding operation time is 2.763s. From the test results, we get TPR_b=91.33%. Participants then modify the power sequence to a false one, and 3 unlocking results come back, which makes FPR_b=2%. From our practical observation, participants cannot stay stable during the test period. Any move of the hands will cause the different RSSI value, which makes it possible to get the opposite result.

For facial verification, participants are required to stare at the camera for a while. The camera captures three following frames right after initialization. To simulate the light intensity in real life, several pictures in low light intensity are selected for evaluation in the experiment. Results show that the facial recognition achieves a TPR up to 91.1% and FPR about 2.2%. The false positive samples have two characters: 1) same skin color and 2) similar facial features. Especially, two volunteers are the father and son relationship, which contribute most to the FPR. The ATT is 2.55s, where 1.43s is for initialization time plus frame capturing time and 1.12s for computation.

For the CP^3 test, participants test on our implemented prototype. From 150 collections, the combined TPR_c=88% and the FPR_c=0%, where no pass for incorrect input in our samples. The time delay measured is 2.88s.

As shown in Fig. 4(b) and Fig. 4(c), CP^3 achieves comparable accuracy and latency as other individual baseline authentications. It only incurs 12.9% larger delay than the facial pattern based authentication. CP^3 yields 88% overall TPR, which demonstrates the feasibility of its practical deployment. In the next section, we show that the TPR can be further improved by allowing multiple unlocking attempts per authentication, without compromising much of the AAT.



(a) Authentication delay

(b) TPR vs. Authentication attempts

(c) FPR vs. Authentication attempts

Fig. 5. Authentication with Multiple Attempts

B. Authentication with Multiple Attempts

In this experiment, we evaluate the impact of unlocking attempts on the authentication performance by allowing multiple unlocking attempts per authentication. We change the unlocking attempts from one to three. The evaluation results are shown in Fig. 5.

It is obvious that the accuracy of all these methods will be improved by allowing multiple attempts, especially for CP^3. The increase rates for all these authentications from one to three attempts are: 3.33% for fingerprint, 4.67% for Bluetooth, 6.68% for face and 8.67% for CP^3. On the latency performance, with the increase of unlocking attempts, all methods do not incur a large increase on the operating delay, as most authentications can pass on one attempt. Take CP^3 for example, it takes 2.88s, 3.04s, and 3.24s, which only has 12.5% increase rate from one to three attempts.

C. Energy Consumption

The main energy cost in CP^3 comes from the continuous Bluetooth monitoring. However, it only runs when the screen is on. We test the energy consumption in two scenarios for 12 hours, where a smart device keeps playing music with and without the Bluetooth authentication running on the backend. Our experimental results reveal that Bluetooth based authentication consumes 2% higher of battery energy compared with the case without Bluetooth authentication. This result depends on the latest BLE mode, which reduces the broadcast channel time of RF to save energy.

VII. Discussion

In this section, we analyze the security guarantees of CP^3, and discuss practical considerations to improve our method.

A. Security Analysis

We discuss the security enhancement provided by CP^3 against different exploitation techniques. We summarize security guarantees of different unlocking schemes in Table I. In conclusion, our combined authentication has the high level of security, which can defend against individual attacks, limit combined attacks and the snatch attack.

Single attack defending. Generally, the single attack can only have one true feedback for the whole authentication. It cannot pass the final decision part in CP^3 where three feedbacks should be all true. For fingerprint forgery and facial forgery, it is possible to success but only for one protection. The other two attacks, including device hacking and eavesdropping attack, will not success depending on convert channel exploitation. Despite the adversary attempts to acquire the secret token or modify the MAC address, it cannot catch the transmission power deeply hidden into normal traffics and pass the comparison process on pair device.

TABLE I
Security guarantees of different unlocking schemes

Methods	Brute-force attack	Finger-print forgery	Blue-tooth hacking	Facial forgery	Snatch attack	Com-bined attack
Finger	N	N	Y	Y	N	N
Bluetooth	N	Y	N	Y	N	N
Face	N	Y	Y	N	N	N
CP^3	Y	Y	Y	Y	Y	L

[1] Y- enable to defend; N- unable to defend; L- defended by limited attempts

Combined attacks and brute-force attack limitation. Brute-force attack has low possibility of success proved by our FPR test results in section VI. As the assumption proposed in section III-A, it is extremely challenging to choose the correct attacks because of the undetectable of convert channel. Attackers also must collect the fingerprint and the facial photos for owners without being sensed, which is a difficult project.

Snatch attack prevention. The simple way to break the system is snatch attack. CP^3 has a periodical detection to prevent this attack. Only if attackers snatch both the main and the pair device, can he get into the system at the first time. But the second time he wakes up the screen, there are still three "doors" waiting for him.

B. Practical Considerations

Availability or DoS attacks on auxiliary devices. It is possible that the auxiliary device is out of battery or being compromised to shut down intentionally rather than accidentally. In this case, the Bluetooth authentication should be automatically disabled to ensure the normal operation. A possible solution is to let users flexibly enable or disable this authentication in the setting stage. Increasing the resilience of auxiliary authentication will be an interesting research topic. We leave it as an important part of our future work.

Authentication performance. As tested in Section VI, allowing multiple attempts in authentication improves the accuracy of CP^3. However, it also increases the FPR. The tradeoff between TPR and FPR should be considered when choosing the optimal unlocking times. As the CPU speed improves, authentication time is also expected to be further reduced where the analysis on three inputs can be performed more quickly.

Light impact on facial verification. Our facial verification changes the color images to gray ones to reduce the influence on light condition. But it will not work when the surroundings is totally dark, even the screen light is enabled. In such condition, we can improve CP^3 with context awareness, *i.e.*, automatically opening flashlight in front.

Applicability to diverse smart devices. In Section VI, we have evaluated CP^3 using the typical Android smartphone and smart watch. It is expected to apply CP^3 on different platforms, such as iOS or Windows platforms etc.

VIII. Conclusion

In this paper, we presented the CP^3, a new authentication solution that combines unlocking methods from different modalities. We introduced the combined unlocking framework and addressed several challenges to achieve highly secure and usable authentication for commodity smart devices, which seamlessly integrate fingerprint, Bluetooth and facial verifications. In particular, we proposed a new Bluetooth based auxiliary authentication method, which modulates transmission power as a convert channel to avoid wireless eavesdropping. A prototype of CP^3 have been implemented on Android platforms and the source code is on public. Real-world experiments with 15 participants demonstrate the effectiveness of our solution. CP^3 achieves 88% detection accuracy with very low false negatives and 2.88s operation latency, which guarantees both good user experience and high security level compared with existing methods. In the future, we will investigate the mutual authentication for smart devices, and address the DoS attacks on auxiliary authentication.

References

[1] "Cyber threats to mobile phones," https://www.us-cert.gov/sites/default/files/publications/cyber_threats-to_mobile_phones.pdf.

[2] "Social media use continues to rise in developing countries but plateaus across developed ones," http://www.pewglobal.org/2018/06/19.

[3] J. T. Isaac and Z. Sherali, "Secure mobile payment systems," *IT Professional*, vol. 16, no. 3, pp. 36–43, 2014.

[4] M. Harbach, A. D. Luca, and S. Egelman, "The anatomy of smartphone unlocking: A field study of android lock screens," in *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems (CHI'16)*, New York, NY, USA, 2016.

[5] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *AVI*, 2006.

[6] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens." *Woot*, vol. 10, pp. 1–7, 2010.

[7] A. C, "iphone 5s fingerprint sensor hacked by germany's chaos computer club," *Guardian News. Np*, vol. 23, 2013.

[8] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted facial recognition: an enhanced biometric authentication system for smartphones," in *MobiSys*, 2014.

[9] P. R. Kennedy, T. G. Hall, and W. C. Yip, "Radio telecommunication device and method of authenticating a user with a voice authentication token," 2000, uS Patent 6,084,967.

[10] C. Song, A. Wang, K. Ren, and W. Xu, "Eyeveri: A secure and usable approach for smartphone user authentication," in *INFOCOM*, 2016.

[11] M. O. Derawi, C. Nickel, P. Bours, and C. Busch, "Unobtrusive user-authentication on mobile phones using biometric gait recognition," in *IIH-MSP*, 2010.

[12] G. Bailador, C. Sanchez-Avila, J. Guerra-Casanova, and A. de Santos Sierra, "Analysis of pattern recognition techniques for in-air signature biometrics," *Pattern Recognition*, vol. 44, no. 10, pp. 2468–2478, 2011.

[13] F. Lin, C. Song, Y. Zhuang, W. Xu, C. Li, and K. Ren, "Cardiac scan: a non-contact and continuous heart-based user authentication system," *MobiCom*, 2017.

[14] J. Chauhan, Y. Hu, S. Seneviratne, A. Misra, A. Seneviratne, and Y. Lee, "Breathprint: Breathing acoustics-based user authentication," in *Mobisys*, 2017.

[15] S. Zahid, M. Shahzad, S. A. Khayam, and M. Farooq, "Keystroke-based user identification on smart phones," in *RAID*, 2009.

[16] H. Saevanee, N. Clarke, and S. Furnell, "Multi-modal behavioural biometric authentication for mobile devices," *Information Security and Privacy Research*, pp. 465–474, 2012.

[17] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors," in *ICNP*, 2014.

[18] K. B. Raja, R. Raghavendra, M. Stokkenes, and C. Busch, "Multi-modal authentication system for smartphones using face, iris and periocular," in *ICB*, 2015.

[19] A. De Luca, M. Harbach, E. von Zezschwitz, M.-E. Maurer, B. E. Slawik, H. Hussmann, and M. Smith, "Now you see me, now you don't: protecting smartphone authentication from shoulder surfers," in *SIGCHI*, 2014.

[20] S. H. M, C. S. T, Y. J. H, and et al, "A shoulder surfing resistant graphical authentication system," *TDSC*, 2016.

[21] L. Y, L. Y, Y. Q, and et al, "Seeing your face is not enough: An inertial sensor-based liveness detection for face authentication," *CCS*, 2015.

[22] J. Potts and S. Sukittanon, "Exploiting bluetooth on android mobile devices for home security application," in *Southeastcon*, 2012.

[23] L. W. H and L. R, "Implicit sensor-based authentication of smartphone users with smartwatch," *HASP*, 2016.

[24] J. Zhang, X. Zheng, Z. Tang, T. Xing, X. Chen, D. Fang, R. Li, X. Gong, and F. Chen, "Privacy leakage in mobile sensing: Your unlock passwords can be leaked through wireless hotspot functionality," *Mobile Information Systems*, vol. 2016, 2016.

[25] G. Ye, Z. Tang, D. Fang, X. Chen, K. I. Kim, B. Taylor, and Z. Wang, "Cracking android pattern lock in five attempts," in *NDSS*, 2017.

[26] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind recognition of touched keys on mobile devices," in *SIGSAC*, 2014.

[27] D. Browning and G. C. Kessler, "Bluetooth hacking: A case study," in *ADFSL*, 2009.

[28] A. Y. Lindell, "Attacks on the pairing protocol of bluetooth v2. 1," *Black Hat USA, Las Vegas, Nevada*, 2008.

[29] D. Gafurov, E. Snekkenes, and T. E. Buvarp, "Robustness of biometric gait authentication against impersonation attack," in *OTM*, 2006.

[30] P. Mihailescu, "The fuzzy vault for fingerprints is vulnerable to brute force attack," *arXiv preprint arXiv:0708.2974*, 2007.

[31] W. Karam, H. Bredin, H. Greige, G. Chollet, and C. Mokbel, "Talking-face identity verification, audiovisual forgery, and robustness issues," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. 4, 2009.

[32] A. Giri and S. Oswal, "A smart system for women security: A new innovation in the domain of women security," *International Journal of Computer Applications*, vol. 119, no. 23, 2015.

[33] S. Yang and I. Verbauwhede, "Automatic secure fingerprint verification system based on fuzzy vault scheme," in *ICASSP*, 2005.

[34] N. Tuptuk and S. Hailes, "Covert channel attacks in pervasive computing," in *PerCom*, 2015.

[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[36] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *CVPR*, 2014.

[37] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," 2016.

[38] T. Joachims, "Making large-scale svm learning practical," Tech. Rep., 1998.