Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones

Hui Xu[*] Yangfan Zhou[*†] Michael R. Lyu[*‡]
[*]Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications,
Shenzhen Research Institute, The Chinese University of Hong Kong
[†]MoE Key Laboratory of High Confidence Software Technologies (CUHK Sub-Lab)
[‡]Dept. of Computer Science & Engineering, The Chinese University of Hong Kong

ABSTRACT

Current smartphones generally cannot continuously authenticate users during runtime. This poses severe security and privacy threats: A malicious user can manipulate the phone if bypassing the screen lock. To solve this problem, our work adopts a continuous and passive authentication mechanism based on a user's touch operations on the touchscreen. Such a mechanism is suitable for smartphones, as it requires no extra hardware or intrusive user interface. We study how to model multiple types of touch data and perform continuous authentication accordingly. As a first attempt, we also investigate the fundamentals of touch operations as biometrics by justifying their distinctiveness and permanence. A one-month experiment is conducted involving over 30 users. Our experiment results verify that touch biometrics can serve as a promising method for continuous and passive authentication.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; D.4.6 [**Software**]: Security and Protection

General Terms

Human Factors, Security, Experimentation

Keywords

Smartphone, Continuous Authentication, Touch Biometrics

1. INTRODUCTION

Smartphones are becoming more and more popular in people's daily life. According to a recent report [31], the number of smartphone users has reached 56% of the American adult population, and smartphone sales continue to grow radically [11]. As a result of the extensive usage of smartphones, much of our sensitive and private information is kept by our phones. This inevitably poses great security risks to smartphone users [8, 13, 35].

To mitigate the risk of malicious user access, most smartphone systems adopt a traditional access control mechanism: Before using a phone, a user needs to unlock its screen with a password or a lock pattern (*i.e.*, several dots in the screen that should be visited in sequence in one finger move). Since a user may use her phone quite often in her daily life, password or lock pattern should be designed simple enough to facilitate the frequent unlock operations. This severely degrades the strength of the access control mechanism. Malicious users can break into the phone simply via peeping [9], or the smudge attack [5].

An enhanced mechanism, namely *continuous authentication* [14, 27], can be more effective in combatting malicious user access. It keeps authenticating the current user during system runtime, thus greatly increasing the complexity of potential intrusions. Examples for such mechanism include requiring fingerprint[1] or face authentication frequently, asking for the answers of a set of pre-defined security problems or passwords, or connecting to an accessory device owned by the valid user[2]. However, these approaches are either too intrusive (*e.g.*, keep asking for password or fingerprint) or costly (*e.g.*, require an extra device like fingerprint sensor or the "Skip"), not to mention the extra energy required to drive the sensors.

We observe that the user operations on touchscreen can be utilized for continuous authentication, with no requirement for extra hardware or user attention. As the dominant human-to-smartphone interface [34], touchscreen is equipped on most smartphones. Moreover, modern touchscreens can produce rich data to describe how users touch, including the curve, the timing, the size and the pressure of a touch operation. Such data can be collected in the background and analyzed to discriminate different users. In other words, while the user performs her normal operations, the authentication proceeds continuously without her notice, *i.e.,* in a passive way.

Using touch operations for continuous authentication has been suggested recently in [14], where a single type of operations (strokes or slides) is considered. Some promising results have been reported. For example, a 13% equal error rate (EER) for one single stroke, and 2% to 3% for 11 consequent strokes can be achieved [14]. However, stroke is not the only type of touch operations. They can also include other types, such as pinch and handwriting. Hence, consid-

Symposium on Usable Privacy and Security (SOUPS) 2014, July 9–11, 2014, Menlo Park, CA.

[1] Note that recently Apple and Samsung have embedded fingerprint sensor into their smartphones.

[2] For example, the "Skip" device introduced by Motorola for MotoX phone.

ering only strokes is not enough to continuously authenticate the user as she can perform other types of operations. A seamless continuous authentication mechanism should take multiple types of operations into account. Moreover, previous investigations (*e.g.*, [9],[14]) have based their designs on a rather straightforward idea that touch operations can be employed to identify users. Yet, the biometric properties of touch operations have not been comprehensively evaluated.

Our work, in contrast, takes advantage of multiple types of touch data to model a user. As a first attempt, we further investigate the underlying fundamentals of touch operations as biometrics by justifying their two critical properties: *distinctiveness* and *permanence*. In other words, we evaluate whether the data features are distinctive among various users, and whether the data features collected from the same user are temporally stable. Both properties are prerequisites for biometrics [17].

To this end, we have conducted a real-world experiment involving over 30 users for one month. Our results confirm that it is promising to implement a continuous authentication mechanism based only on the touch data collected during normal user operations.

The contributions of this paper are as follows:

- This work serves as the first attempt to comprehensively evaluate the biometric properties of touch data, and we study how such data can be used for continuous authentication.
- We propose a set of methods to model the multiple types of touch data via a separation-of-concern solution, which is quite effective.
- The findings and data from our real-world experiment involving over 30 users are publicly available, which can facilitate further follow-up work.

The rest of the paper is organized as follows. Section 2 provides the adversary model and some preliminaries of touch biometrics. Section 3 overviews the framework of touch-based authentication and goes through details about the feature extraction and classification method. In Section 4, we evaluate the performance of touch biometrics in distinctiveness, permanence and authentication error rate based on the framework. The related work is discussed in Section 5. Section 6 concludes our research and suggests potential future work.

2. BACKGROUND

In this section, we briefly introduce the adversary model and some technical preliminaries including smartphone touch operations, biometrics, and performance metrics.

2.1 Adversary Model and Assumptions

In this paper, we assume the following adversary. A malicious attacker has gained access to a person's smartphone equipped with a touchscreen. The smartphone is either unprotected (*e.g.,* no PIN) or the attacker has got into possession of the authentication secret, for instance by shoulder surfing the owner. The attacker can then perform undesirable actions with the device violating the owner's privacy (*e.g.,* browsing photos, reading SMS or e-mails). Afterwards, the phone's screen can be turned off and put back to its original place, appearing as if it was never touched. The owner will have no chance to figure out that it has been used by someone else. In this way, the owner's privacy could be severely violated. Our work targets such situations and tries to make this kind of manipulation impossible by analyzing touch behavior.

Table 1: Example of raw event data collected when tapping "1" and "2" on soft keyboard

Tap	Time	Position		Size	Pressure
		X	Y		
1	122382	62.869	550.312	0.169	0.233
1	122444	67.892	553.328	0.169	0.2
1	122461	70.057	550.008	0.067	0.067
1	122503	70.057	550.008	0.067	0.067
2	122731	202.578	553.308	0.141	0.167
2	122794	204.591	556.305	0.141	0.2
2	122811	204.574	554.170	0.141	0.2

2.2 Touch Operations

The smartphone systems accept user commands through interpreting touch. According to our knowledge, the most frequently used operations include *keystroke*, *slide*, *pinch*, and *handwriting*.

- *Keystroke:* A keystroke is a finger tap on the screen. Typical scenarios include using soft keyboard and unlocking screen with PIN.
- *Slide:* A slide is a finger move on the screen. A lot of applications use slide for navigating documents, *e.g.*, web pages, photo albums, messages, and contact list.
- *Pinch:* A pinch is a two-finger gesture typically used for zooming functionality.
- *Handwriting:* Handwriting is an important alternative input method on smartphone to enter characters.

When a touch operation is performed, the smartphone hardware automatically generates a set of data and reports them to the operating system as *raw events*. Taking Android as an example, a raw event reports the data of the position, pressure, and size of a touch, as well as a timestamp. The operating system generally extracts touch operations intended by the user by interpreting such raw events. Each row in Table 1 shows the data of a raw event. We observe in our practice that the *time* and *position* data are fine-grained, while the *size* and *pressure* are coarse-grained. To avoid noise, we choose to use statistical information (*e.g.*, average or standard deviation) of the size and pressure data instead of subtle data changes in the feature extraction process.

In practice, one single touch operation generates a series of raw events. Their positions form a trajectory sequence. We call the sequence of the corresponding raw event data a *touch data sequence* of the touch operation. Touchscreen can produce raw events every few milliseconds when being touched. As a result, even the simplest touch operation can generate quite a few raw events. Table 1 shows an example of raw events collected when tapping "1" and "2" on the soft keyboard. In this example, the tap on "1" and "2" produce four and three raw events. We will discuss how we model a touch operation based on the touch data sequence it generates in Section 3.1.

2.3 Biometrics

Biometrics refers to the automatic recognition of individuals based on their physiological and/or behavioral characteristics [17]. Common types of biometrics include face, fingerprint, hand geometry, iris, keystroke, signature, and voice [16]. When a biological characteristic qualifies to be a form of biometrics, it should generally bear the following four properties [17].

- *Universality*: Every person has the characteristic.
- *Distinctiveness*: Any two persons are distinguishable in terms of the characteristic.
- *Permanence*: The characteristic is stable over a period of time.
- *Collectability*: The characteristic can be measured in numbers.

Touch operation can be considered as of behavioral biometrics. Its universality and collectability are obvious, while its distinctiveness and permanence need to be assessed, which is a major focus of our work.

Note that there are also other issues that need to be considered for a practical biometric system, for example, recognition speed, overhead, and user-friendliness [17]. These implementation considerations are not the focus of this work.

2.4 Performance Metrics

Accuracy and error rate are two straightforward metrics for authentication performance. However, their information is rather limited and must be interpreted with much caution. It is therefore necessary to introduce the concepts of false acceptance rate (FAR), false rejection rate (FRR), equal error rate (EER) and receiver operating characteristic (ROC), which are more meaningful [24]. These terms are defined as follows:

- *FAR*: The rate that an attacker is wrongly accepted as the valid user.
- *FRR*: The rate that the valid user is wrongly rejected as an attacker.
- *EER*: The rate at which FAR and FRR are equal. In practice, FAR and FRR are sensitive to system settings and correlated with each other. FAR will usually increase as FRR decreases, and *vice versa*. EER is a metric of the trade-off between of FAR and FRR, which is widely used for indicating the performance of real authentication systems.
- *ROC*: A graphical plot that visualizes the performance of a binary classifier as its discrimination threshold varies. ROC is created by plotting the fraction of the true positive rate (*i.e.*, rejection rate when the user is invalid) vs the false positive rate (*i.e.*, rejection rate when the user is valid), at various threshold settings [1]. ROC is a more complicated indicator, which reflects the performance of a system under different settings.

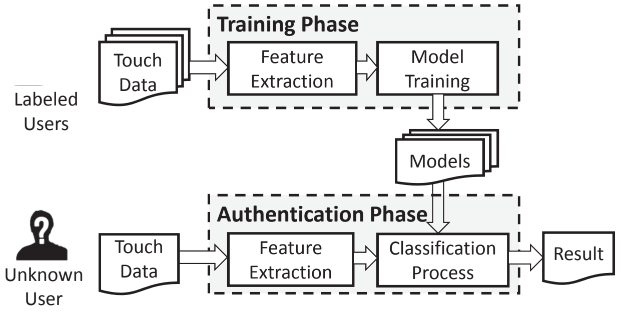


Figure 1: Overview of touch-based authentication approach

3. TOUCH DATA-BASED USER AUTHENTICATION

Our idea of using touch data for continuous authentication includes two phases: the training phase and the authentication phase. In the training phase, a number of *labeled* touch data (*i.e.*, the data together with whether it comes from a valid user) are processed so as to model the valid user. In the authentication phase, the touch data, which may come from the valid user or an attacker, are labeled according to the models generated in the training phase. In this way, we can authenticate the corresponding user of the touch data. Fig. 1 overviews the touch-based user authentication approach.

Centric to this approach is a statistical pattern recognition procedure that can discriminate different users according to the touch data. To design an effective touch data-based user authentication approach, two key steps need to be addressed: 1) how to model the user characteristics from the touch data, *i.e.*, what kind of features should be extracted from the data. 2) how to recognize users according to these features. We discuss these two issues in what follows.

3.1 Feature Extraction

Touchscreen can catch every subtle user touch and generate corresponding touch data sequence. We may directly consider touch data sequence as the basic granularity and model the user accordingly. However, since different sequences may belong to different types of touch operations, they may contain quite different characteristics. For example, a slide operation with one finger move is quite different from a pinch operation with two fingers. In order to address this problem, we propose a separation-of-concerns approach which considers each type of touch operations separately. In this way, each type of touch operations can be modeled separately with its corresponding sequence of raw events.

Let X denote the data of a raw event, where X = [Time, Position$_x$, Position$_y$, Pressure, Size]. Let $\{X_1, X_2, ..., X_n\}$ denote a sequence of raw events that jointly form a touch operation. Let F = [feature$_1$, feature$_2$, ..., feature$_m$] denote the feature vector of a touch operation. We should find how to map $\{X_1, X_2, ..., X_n\}$ to F, so that F can well describe the characteristics of the touch operation. In what follows, we will discuss the design of such a mapping according to the specifics of each type of touch operations.

3.1.1 Features of Keystroke

Keystroke operation typically involves a series of taps on the soft, on-screen keyboard. Keystroke dynamics on hard-

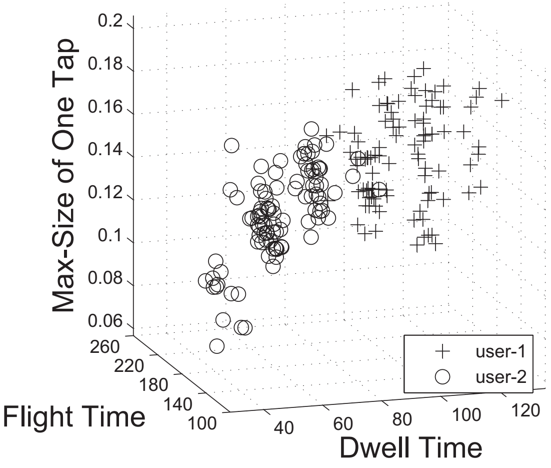


Figure 2: Keystroke feature vectors of 2 users in 3-dimensional space when tapping "1" within a number sequence "123456"

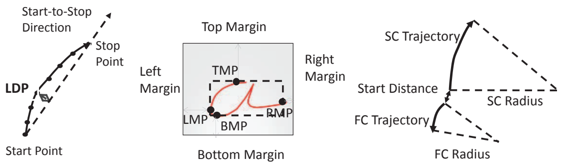
ware keyboard is a type of biometrics well studied in the literature [4, 22], which sheds light to our study on software keyboard. We adopt two features proven effective in the hardware keystroke dynamic field: the *dwell time* and *flight time* features. The former considers the duration of a keystroke and the latter considers the time interval between successive keystrokes. Even though some new features specially tailored for touchscreen based keystrokes have been proposed (*e.g.*, the detailed touch locations of each key [10]), there is no enough evidence to show that the recognition accuracy can be improved considerably [10]. Hence, we don't include these new features in our model.

The upper-left corner of Table 2 shows the four typical features for keystroke operation we propose. Besides dwell time and flight time, the other two features are self-explained by their names. As a demonstrating example, Fig. 2 shows the feature vectors extracted from 2 different users when they perform keystroke operations. We can easily see that different people have quite different characteristics in terms of the features we propose.

3.1.2 Features of Slide

A slide operation is a finger move from a start point to a stop point on the screen, *i.e.*, a curve. Besides these two points, we also consider the largest deviation point (LDP) in the slide curve. An LDP is the point that is farthest to the straight line between the start point and the stop point of the slide curve. Fig. 3(a) shows an example of such an LDP. The LDP can, to some extent, describe the curvature of the slide. Hence, we choose to extract features based on these three points. Our extraction process is designed as follows.

First, we consider the positions of these three points, and thus introduce the *trajectory features*. Trajectory features are the features that reflect the directional information of finger moving and those that measure the length of the moving trajectory. The latter is measured by the sum of the line segments between every two consecutive raw events occurring during the finger move. Secondly, we consider the dynamics of the slide move along these three points. Specially, we consider the pressure, size and velocity along them.



(a) Slide (b) Handwriting (c) Pinch

Figure 3: Demonstration of key metrics during feature extraction

Thirdly, there are several statistical features that have been taken into account. For example, the standard deviation of touch pressure occurring during a slide can reflect the distribution of touch strength. Table 2 provides the 37 suggested features for slide.

3.1.3 Features of Handwriting

Input via writing on the screen is an important input method for smartphones. Naturally, how to model such operations is the area of handwriting forensic. Handwriting forensic identifies handwriting through the analysis of various aspects of writing, including the arrangement, slant, baseline alignment, design of alphabets [32]. In this work, we also extract handwriting features with the handwriting forensic approach. We omit those features that are not computationally available [32] and customize 42 features for handwriting authentication, as provided in Table 2. Specifically, we consider the leftmost, rightmost, topmost, and bottommost points of a handwritten letter (denoted by LMP, RMP, TMP, and BMP, respectively). Fig. 3(b) demonstrates these four points of a handwritten "*a*".

Similar to slide operation, we propose the trajectory features of these four points, as well as dynamics of the finger move along these points. We also consider the statistical features of raw events which occur during the handwriting.

3.1.4 Features of Pinch

The trajectory of a pinch operation includes two curves, since it involves two fingers. The features of a pinch naturally include the features of both curves. The features of each curve can be extracted similarly as a slide. We also consider the features that can describe the correlation between the curves, as they are generated by two fingers of the same user. For example, we consider *start distance* and *stop distance*, which are the distances between two fingers when the pinch starts and stops respectively.

We notice some people would pinch with thumb and index finger, while others with index finger and middle finger, which will cause quite different characteristics of the resulting curves. Instead of distinguishing the two curves with finger name, we distinguish the two curves by their positional information: The curve with the start position on the left-hand side to the start position of the other curve is named the first curve (FC), and the other curve is named the second curve (SC). There are in total 49 features we propose for modeling the pinch as listed in Table 2.

In the discussions above, we have provided a set of features for each type of touch operations based on their specifics. It is worth noting that these features may not all be effective for user authentication. In our experimental study, we will evaluate these features and select a subset for modeling each type of touch operations.

Table 2: The features we proposed for touch operations(Pos. and Traj. stand for position and trajectory, respectively). For each feature, we present the feature evaluation result in accuracy according to Section 4.2.

Keystroke Features			Handwriting Features			Pinch Features		
Feature Name	Accurary (%)	Ranking	Feature Name	Accurary (%)	Ranking	Feature Name	Accurary (%)	Ranking
Max-Size of One Tap	18.3761	1	Left Margin	12.3	27	FC Start Point Pos. X	19.2	3
Max-Pressure of One Tap	9.9343	4	Right Margin	11.1	31	FC Start Point Pos. Y	16	14
Dwell Time	13.1823	3	Top Margin	16.6	15	FC Start Point Size	14.3	20
Flight Time	13.4075	2	Bottom Margin	20.2	4	FC Start Point Pressure	13.8	23
Slide Features			LMP Size	19.8	5	SC Start Point Pos. X	13.9	22
Start Point Pos. X	20.6	1	RMP Size	9.9	35	SC Start Point Pos. Y	15.5	18
Start Point Pos. Y	16.4	7	TMP Size	23.3	1	SC Start Point Size	18	8
Start Point Size	18.7	2	BMP Size	17	12	SC Start Point Pressure	11	29
Start Point Pressure	10.1	18	LMP Pressure	8.7	36	FC Stop Point Pos. X	18.4	4
Start Point Velocity	10.6	16	RMP Pressure	4.7	38	FC Stop Point Pos. Y	14.3	21
LDP Pos. X	12.4	14	TMP Pressure	11.9	28	FC Stop Point Size	9	38
LDP Pos. Y	11.5	15	BMP Pressure	7.9	37	FC Stop Point Pressure	6.5	43
LDP Size	18.5	3	Vertical Direction	2.4	41	SC Stop Point Pos. X	16	15
LDP Pressure	10.4	17	Horizontal Direction	2.4	40	SC Stop Point Pos. Y	24.5	1
LDP Velocity	14.2	11	Avg. Size	18.2	9	SC Stop Point Size	12.6	25
Stop Point Pos. X	16.2	8	Avg. Pressure	20.9	2	SC Stop Point Pressure	9.4	37
Stop Point Pos. Y	14.5	10	Start Point Pos. X	11.5	29	FC Start Point Velocity	10.2	32
Stop Point Size	7.7	28	Start Point Pos. Y	16.6	14	FC Stop Point Velocity	8.1	41
Stop Point Pressure	5.5	30	Start Direction	3.2	39	SC Stop Point Velocity	9	39
Stop Point Velocity	8.5	26	Stop Point Pos. X	12.3	26	SC Start Point Velocity	9.8	34
Avg. Velocity	16.8	5	Stop Point Pos. Y	19	6	FC Traj. Length	15.6	16
Start-to-LDP Latency	8.7	25	Stop Direction	2	42	SC Traj. Length	16.7	13
Straight Start-to-LDP Length	9.4	21	Start-to-LMP Latency	11.5	30	FC Interval	19.6	2
Start-to-LDP Direction	4.7	32	Start-to-RMP Latency	19	7	SC Interval	18.3	6
Start-to-Stop Latency	10	19	Start-to-TMP Latency	13.8	21	FC Traj. Velocity	9.8	35
Straight Start-to-Stop Length	9.1	22	Start-to-BMP Latency	17	13	SC Traj. Velocity	11.8	27
Start-to-Stop Direction	3.4	36	Start-to-LMP Traj. Length	13	24	Start Distance	13.5	24
LDP-to-Stop Latency	14.1	12	Start-to-RMP Traj. Length	18.2	8	Stop Distance	15.6	17
Straight LDP-to-Stop Length	16.5	6	Start-to-TMP Traj. Length	16.2	16	Start Interval	8.2	40
LDP-to-Stop Direction	4	35	Start-to-BMP Traj. Length	17.8	11	Stop Interval	11	30
Straight LDP Length Ratio	7.7	27	Start-to-LMP Velocity	10.7	33	Mutual Interval	18.4	5
Start Direction	2.7	37	Start-to-RMP Velocity	13.4	23	Traj. Length Ratio	16.8	10
Stop Direction	4.2	33	Start-to-TMP Velocity	16.2	17	FC Moving Direction	3.3	46
Rotation	4	34	Start-to-BMP Velocity	14.6	20	SC Moving Direction	2.4	48
Traj. Length	17.1	4	Total Traj. Length	20.6	3	FC Moving Rotation	5.3	44
Straight to Traj. Length Ratio	5.6	29	Avg. Velocity	18.2	10	SC Moving Rotation	3.3	47
Avg. Distance	8.7	24	Width	13.4	22	FC Straight Length	18.3	7
Avg. Size	15.5	9	Height	15	19	SC Straight Length	16.8	11
Avg. Pressure	14	13	Area Size	15.8	18	Straight Length Ratio	16.8	12
Distance STD Deviation	4.9	31	Width-to-Height Ratio	10.7	32	FC Traj. Radius	5.3	45
Size STD Deviation	9.1	23	Size STD Deviation	12.6	25	SC Traj. Radius	1.7	49
Pressure STD Deviation	9.7	20	Pressure STD Deviation	10.3	34	Avg. Size of FC	15.5	19
						Avg. Size of SC	17.1	9
FC Pressure STD Deviation	7.7	42	FC Size STD Deviation	11	31	Avg. Pressure of FC	12.2	26
SC Pressure STD Deviation	9.8	36	SC Size STD Deviation	10.2	33	Avg. Pressure of SC	11.4	28

3.2 Classification

The major purpose of the classification process in Fig. 1 is to authenticate users using a classifier. We discuss our *authentication model* and classifier in this section. Moreover, since there is no systematic study of touch biometric properties so far, we further introduce our *discrimination model* for studying its biometric properties. The key difference of a discrimination model from an authentication model is that, in a discrimination model, we can have the data of each class for training. Fig. 4 compares these two models and visualizes their difference.

3.2.1 Discrimination Model

We define this model as a typical multi-class classification model: Given N classes, each having some samples, how to identify which one of these classes a new observation belongs to. In the training phase, a number of labeled touch data from N users are processed via the feature extraction process discussed in Section 3.1. We can obtain corresponding N classes of feature vectors. The vectors are then fed into a classifier for training purpose. While in the discrimination phase, a new touch data observation is also processed via feature extraction process first. The classifier then decides which class the obtained feature vector belongs to and then identify the user accordingly.

Obviously, when N grows, the identification process naturally becomes more difficult, and the accuracy would decrease. A form of good biometrics should exhibit good performance even when N is large. Hence, the discrimination model can reflect the distinctiveness of biometric properties by involving different numbers of users.

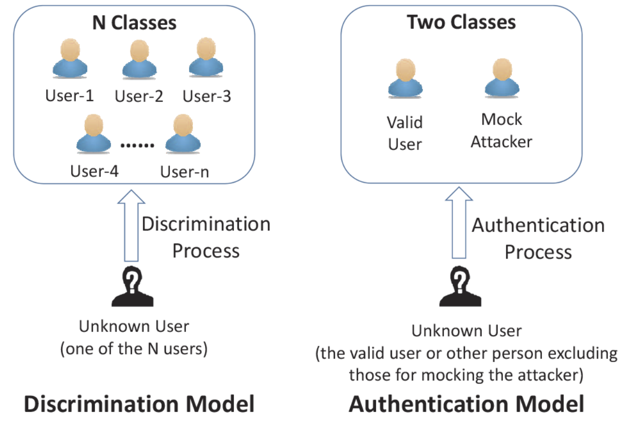


Figure 4: Comparison between discrimination model and authentication model

3.2.2 *Authentication Model*

In practice, we cannot know the models of the attackers beforehand. However, we can obtain the touch data of the valid user herself, and those of some other users[3]. We use these additional users to build a mock attacker model as an approximation to the real, unknown attacker.

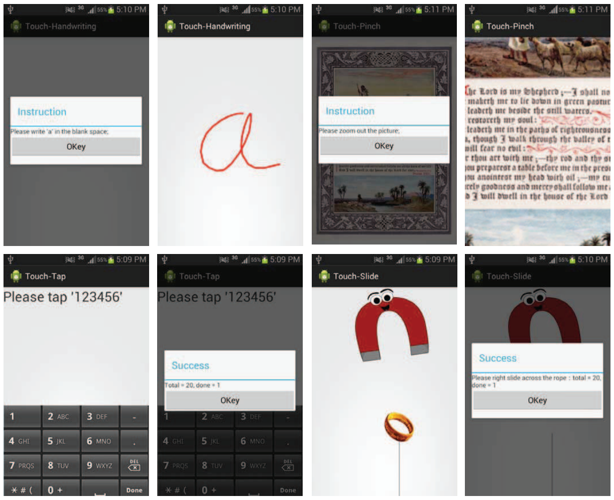
We define the authentication problem as a binary classification problem. Given two classes of samples, one including touch data of the valid user, and the other including those of the mock attackers, how to identify which class a *new* observation belongs to. In the training phase, given the touch data of both classes, we can obtain two corresponding classes of feature vectors via the feature extraction process discussed in Section 3.1. We can then turn to a classification algorithm: Input the two classes of feature vectors to train a classifier. After the classifier is trained, it can be used to determine whether a current user operation is from a valid user or not, by checking which class (*i.e.*, the valid user class or the attacker class) it belongs to.

3.2.3 *Classifier*

There are many classification algorithms we can choose. We adopt a state-of-the-art statistics-based classification method, *i.e.*, the Support Vector Machine (SVM) [6]. It can infer how two classes of vectors are different from each other by finding a hyperplane (*i.e.*, a boundary) that best separates the classes. With such a boundary, any unlabeled sample can then be classified according to which side of the boundary it locates.

We adopt SVM since it has long been proven successful in many classification applications. Moreover, it can seamlessly apply the kernel method, *e.g.*, via Radial Basis Function (RBF) kernel [6], and thus find a nonlinear boundary that best separates the two classes. This non-linear property is critical to our problem setting, since the discriminations between the touch data from the valid user and those from the attackers are nonlinear in nature.

Finally, note that SVM is not the only option of classifier for our user authentication approach. Other methods, for example, logistic regression and Naive Bayes classifier, can also be incorporated into our approach conveniently. A further comparison study is left to our future work.



Please write 'a' in the blank space;

Please zoom out the picture;

Please tap '123456'

Please tap '123456'

Total = 20, done = 1

Please right slide across the rope : total = 20, done = 1

Figure 5: User interface of our data acquisition tool. The first row demonstrates our handwriting and pinch experimental UIs, while the other demonstrates these of keystroke and slide.

4. EXPERIMENTAL STUDY

In the previous section, we have described our framework for continuous authentication based on touch operations. This section evaluates its performance via real-world experiments. First, we conduct a real-world experiment to collect touch data. Secondly, we evaluate the proposed features using these data. Thirdly, we study the distinctiveness and permanence properties of touch operation, and justify it qualifies to be a form of good biometrics. Finally, we evaluate the authentication performance of our proposed framework.

4.1 Data acquisition

We recruited 32 participants for our data acquisition experiment using an online advertisement. The only requirement was that the participants had to be users of smartphone with touchscreen. This was to guarantee that they were familiar to the touch operations required in the experiment. Each participant received a $6 gift for his/her participation.

In order to collect touch data, we programmed a data acquisition tool with Java, which runs on Android smartphone as a stand-alone application. This tool collects the four types of touch operations of interest, and saves their touch data sequences for further analysis. Fig. 5 shows the user interface of this tool. It was installed on a Samsung Galaxy SII smartphone with Android OS 4.1.2.

Before the experiment, the participants were informed that that their touch data would be collected for behavior analysis, and they were required to operate as they usually did. After they got familiar with the tool, we required them to start performing operations as the tool instructed. Each experiment took roughly 15 minutes. In this way, we collected 200 touch data sequences from each participant.

We further chose 3 volunteers among these 32 participants for a long-term study. We asked them to do the experiment

[3] These data are collectable in reality since it is not hard to collect the touch data of some other users who use the same smartphone model.

with the same settings repeatedly for 20 more times. The interval of each two consecutive experiments for each volunteer was one day by default except weekends. To be convenient, we only required them to perform tasks for about 5 minutes (*i.e.*, we thus collected 50 touch data sequences) in each experiment. The whole data acquisition experiment lasted for almost one month. We collected roughly 1200 touch data sequences from each volunteer in total[4].

4.2 Feature Evaluation

In Section 3.1, we have suggested a set of features for each type of touch operations. We now evaluate the effectiveness of each feature in classification accuracy. The idea is to discriminate users *solely* based on one feature at a time. We adopt the discrimination model in the feature evaluation process. To elaborate, in the training phase, we use only *one* feature to model the user at a time. The classifier then classifies a new sample based on this model. The classification accuracy can be obtained accordingly as an indication of the feature's effectiveness.

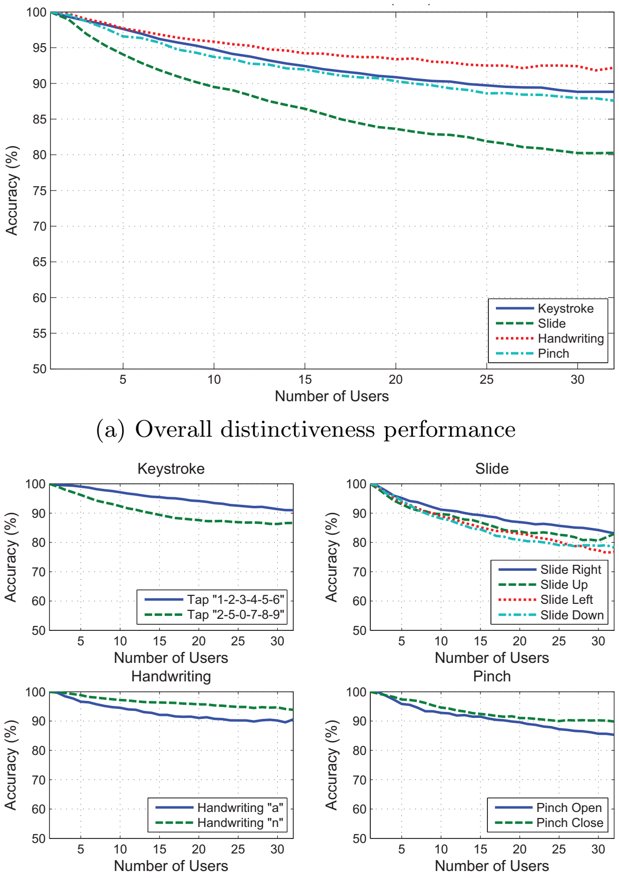
In our experimental settings, we use the data set of 32 participants. To evaluate each feature, the classifier performs a 10-fold cross validation based on the data of that particular feature. A 10-fold cross validation approach randomly partitions the data into 10 equal-size subsets. Each time nine subsets are used for training, and the remaining subset is retained for testing. The accuracy values are then averaged. Our evaluation results are provided in Table 2 along with the feature name, and the ranking according to the accuracy.

According to [15], a feature X is relevant in the process of discriminating class $Y{=}y$ from others if the conditional probability $P(Y{=}y|X{=}x)$ is different from the unconditional probability $P(Y{=}y)$ for some values $X{=}x$ for which $P(X{=}x)>0$. In our study, since the task is to discriminate one user among the 32 users, a naive guess can achieve a 1/32 accuracy (*i.e.*, the unconditional probability is 3.125%). Therefore, the features with accuracy lower than 3.125% are useless in discriminating users, and we thus remove these features.

In the rest of our study, we consider only the features with accuracy higher than 3.125% in Table 2. Noticing that some features on directional information are not discriminating. We believe such an evaluation study can enlighten future feature extraction method for touch-based continuous authentication.

4.3 Evaluation of Distinctiveness

In this section, we evaluate the distinctiveness of touch biometrics, *i.e.*, how well touch operations can be used to discriminate users. We adopt the discrimination model in this step. Our experiment is based on the data set of feature vectors from 32 users. We randomly pick N users and their vectors from the data set. Focusing on each type of touch operation at a time, we benchmark the classification accuracy with N users using a 10-fold cross validation approach. We change N from 2 to 32, and thus get the accuracy with different user sizes. Fig. 6(a) shows our experiment results. We can see that all types of touch operations are distinctive among users with a classification accuracy better than 80% even when we try to discriminate a user from 31 others.



(a) Overall distinctiveness performance

(b) Distinctiveness performance of touch operation subtypes

Figure 6: Distinctiveness performance of touch operations based on the data set of 32 users

We have noticed that there are still minor differences among the operations of each type. Specifically, a pinch may be pinch open or pinch close; A slide can have four possible directions; Handwriting can involve different letters; Keystroke operations can input different words. We study whether such *subtypes* have a considerable impact on the distinctiveness performance. Fig. 6(b) shows the experiment results, from which we can tell that the differences between subtypes are slight. Therefore, in the subsequent experiments, we will not consider these subtypes.

4.4 Evaluation of Permanence

We now study the permanence performance of touch biometrics, *i.e.*, if we model a user with her touch biometrics, whether the model is stable over a period of time for the same user. In this regard, our experiment is based on a 21-day data set from the 3 volunteers. As mentioned before, we collected their touch data from a 21-day long experiment. We use the discrimination model for evaluation. To elaborate, we model the user using their data collected in the first day. We then discriminate the data of each remaining day based on this model. If touch biometrics bears good permanence property, the model should be good enough in discriminating the data of the remaining days. Fig. 7 shows the results.

[4] The data set are available at the project homepage: http://www.cudroid.com/urmajesty.

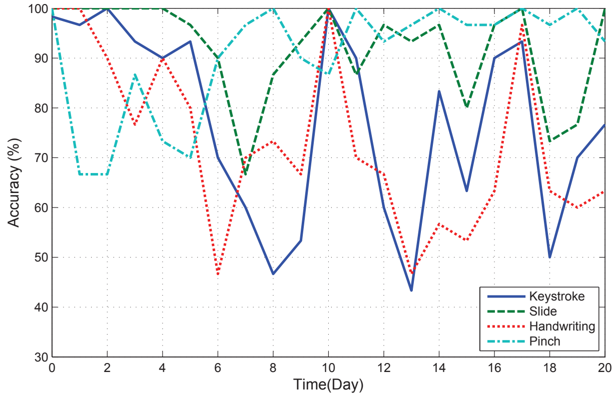


Figure 7: Permanence performance based on the data set of the 3 volunteers

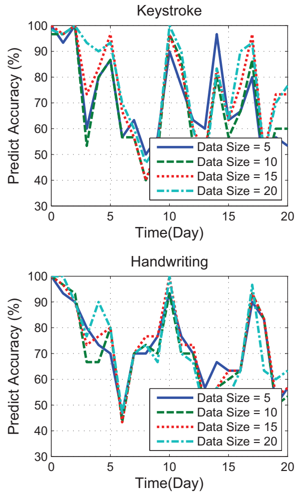


Figure 8: Permanence performance with different training data sizes

We can observe that the performance is not stable for all touch operations, even though pinch and slide are relatively better than keystroke and handwriting. It is probably because that our data used for training is too flaky to get a stable enough result. To further clarify this issue, we conduct another experiment using different sizes of training samples. The results in Fig. 8 show that the performance improves only a little as the data size grows. Therefore, we can infer that data size is not the key factor to the poor performance. As a result, we conclude that touch biometrics is not quite stable over time.

A common way to deal with the permanence issue in biometric systems is to consider an adaptive approach: The model will be adjusted according to new samples. We investigate whether such an adaptive approach is helpful for touch biometrics. For this reason, we improve the previous experiment in permanence evaluation using the same data set. When discriminating the data of the nth day, we model the users using all the touch data previous to the nth days, instead of the first day only. Fig. 9 shows the evaluation results. We can see that the results tend to be much more stable, especially after the 8th day. This shows that an adaptive approach can help tackle the permanence problem.

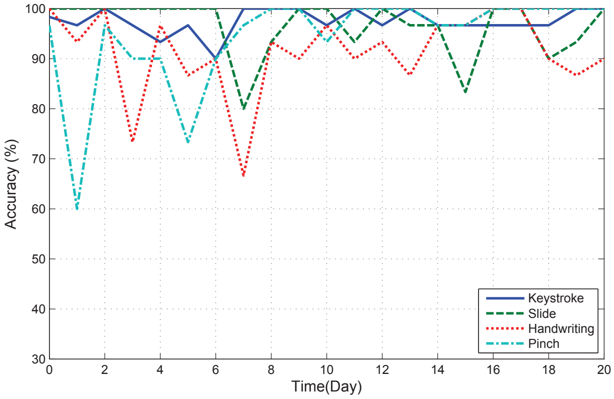


Figure 9: Permanence performance of adaptive approach based on the data set of the 3 volunteers

Table 3: Average error rate with different numbers of additional users to model the mock attacker

additional user #	Keystroke	Slide	Handwriting	Pinch
5	11.76%	11.24%	11.48%	7.38%
10	10.3%	10%	10.08%	4.96%
15	9.36%	4.85%	9.27%	3.87%
20	7.71%	1.53%	11.39%	3.75%
28	6.42%	0.75%	8.67%	3.33%
30	5.3%	1.3%	8.67%	3.33%

4.5 Evaluation of Touch-based Authentication

In this section, we study the performance of touch-based authentication. The major difference of this study is that we consider the practical case, where the attacker model is not known beforehand. In other words, the classifier cannot be trained with the touch data from the real attacker. We adopt the authentication model in this study. As discussed in Section 3.2.2, we assume that we can have the touch data of the valid user herself, and those of some other users to mock attackers.

Our experimental setting is discussed as follows. We consider each of the 3 volunteers at a time, and use her data of the previous 20 days to model the valid user. We then randomly select M additional users from the rest 31 users to model the mock attacker. The remaining data of the valid user and those of the rest users (those are not involved in the training process) are used for prediction. We study the performance in terms of average error rate (*i.e.*, (FAR+FRR)/2). Table 3 shows our experiment results when M varies. Each error rate within this table is an average of those of the three volunteers'.

From Table 3, we can observe that the performance improves as the additional users number increases. However, an overfitting for slide occurs when the number of additional users exceeds 28. But for the other 3 touch operations, the performance might further improve when involving more additional users.

In general, including more additional users can help reduce FAR, since it explores more diverse user characteristics. In other words, involving more additional users shrink the class boundary of the valid user and thus improve FAR.

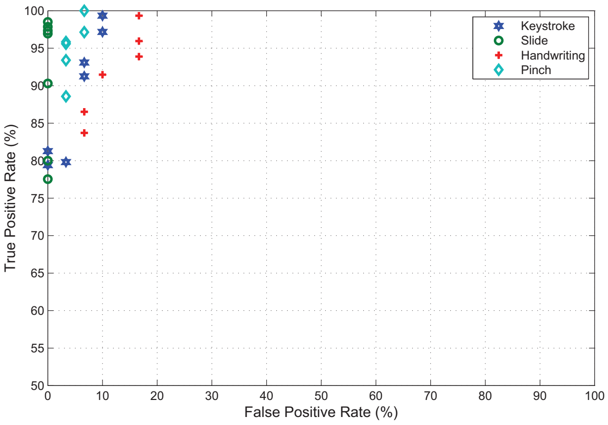


Figure 10: ROC plot when using different number of additional users to model the mock attacker.

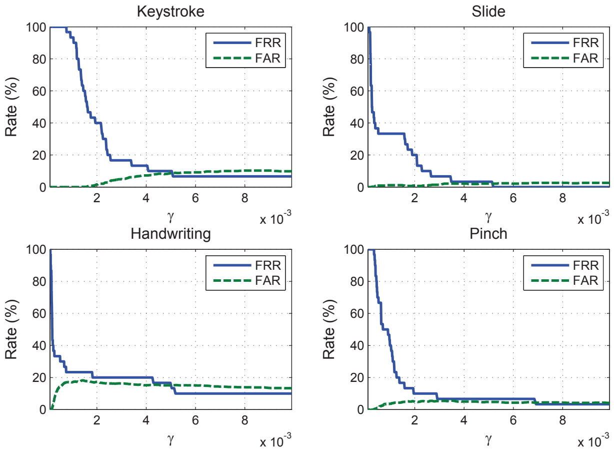


Figure 11: FAR/FRR plots where additional user number equals to 20 for keystroke, 25 for slide, 8 for handwriting and 20 for pinch

However, when the number of additional users are too high (*e.g.*, the 30 case), it may also deteriorate the authentication accuracy. This is not surprising: As the number increases, the attacker samples are getting more diverse, and the SVM will suffer overfitting to the attacker class. As a result, it tends to misclassify more operations of the valid users, causing a high FRR.

In practice, FAR and FRR are correlated with each other. To avoid bias, ROC is commonly used to evaluate biometric systems, which reflects the characterization of the trade-off between the true positive rate and the false positive rate. Fig. 10 visualizes such a trade-off for the average error rate achieved in Table 3.

Since our approach heavily relies on the SVM classifier, we tune the SVM parameters to get the EER. We adopt a commonly-used RBF kernel in the SVM classifier, defined as $K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$ [6]. We tune the value of γ and obtain corresponding FAR and FRR, which are plotted in Fig. 11. We observe that our biometric system can generally achieve EER values lower than 10% for all operation types. The slide operation performs the best by achieving an EER lower than 1%.

Table 4: Average error rate using consecutive sequences. To better visualize the improvement, some previous experiment results in Table 3 are also shown here for comparison purpose.

user # in training	Numbers of Operation 1	 3	 5
Keystroke			
10	10.3%	9.82%	9.71%
20	7.7083%	7.74%	3.32%
28	6.4167%	5.02%	0.88%
Slide			
10	10%	9.55%	9.33%
20	1.5278%	0.98%	0.64%
28	0.75%	0%	0%
Handwriting			
10	10.0758%	5.94%	5.62%
20	11.3889%	10.92%	15.8%
28	8.6667%	8.3%	13.89%
Pinch			
10	4.9621%	2.63%	2.1%
20	3.75%	1.47%	0.92%
28	3.333%	0%	0%

In practical scenarios, we can use a combination of consecutive operations jointly for making an authentication decision [14]. A convenient approach is to authenticate the user with each of the operations first. The system then decides whether a user is an attacker based on the majority of the results. To verify the applicability of this idea to our model, we conduct a comparison experiment with the same data set. This time, we try to authenticate users with 3 and 5 consecutive operations. Table 4 shows the experiment results, which confirm such an approach is helpful in improving authentication performance. According to Table 4, the performance improves a lot in most cases. For slide and pinch, the average error rate even approaches 0. However, the performance for handwriting does not improve much. We think the reason is that the average error rate for each handwriting operation is relatively high. From the permanence experiment, we could infer that consecutive handwriting operations are more likely to be similar. Therefore, errors would also tend to happen consecutively in a short interval, rather than distribute evenly over a period of time. When such case occurs, the performance will degrade due to the high error rate. Which will affect the performance when the error rate is too high. If the rate could be lower down (*e.g.*, by involving more additional users), the result would also improve. Details of such an evaluation are left to future work.

To conclude, when we model the mock attacker properly, the authentication performance can be very promising. Also, using consecutive sequences to authenticate a user is a helpful way to improving the error rate.

4.6 Lessons Learned

Our experiments have evaluated the distinctiveness and permanence properties of touch operations. The results show that touch operation can be a form of good biometrics. However, regarding the distinctiveness property, we find that there is still room for the accuracy to approach 100% when we discriminate the users. As a result, our touch-based continuous authentication approach cannot achieve an error rate very close to zero using one operation. This indicates a need for further research to make touch-based continuous authentication a practical solution. We believe that it is a promising solution to consider a set of touch operations jointly for making an authentication decision rather than using one at a time. We have shown that when considering 3 or 5 consecutive operations jointly, the biometric system achieves average error rates approaching 0% for slide or pinch, which can satisfy practical concerns. However, how to use these operation combinations effectively and efficiently should be studied in the future.

Regarding the permanence property, we find that touch biometrics are not strictly stable over time, especially for keystroke and handwriting. We have shown that a convenient adaptive approach can greatly improve the accuracy. Therefore, the permanence problem can be mitigated. However, a more sophisticated approach is still at large.

Finally, touch-based authentication inevitably requires a large number of touch operation samples for training purpose. We have shown that potential attackers can be modeled with data from a set of additional users. Such data can be preloaded into smartphone in practice. However, what is the adequate number of additional users should be further studied in the future. Moreover, we still need hundreds of training samples from the target valid user. How to design a user-friendly way to obtain so many data samples is still an open question for implementing touch-based authentication.

5. RELATED WORK

Continuous authentication on traditional PC has been extensively studied for years. Research on how to continuously authenticate PC users can be found in [2, 7, 18, 19, 20, 28, 30, 36]. Keystrokes, mouse dynamics, and face recognition are the main approaches. However, the usability of these technologies is still a question due to the low recognition accuracy and inconvenience.

Equipped with more sensors in smartphones (*e.g.*, gyroscopes), continuous authentication on smartphone started a new research area. Several projects have studied how to passively authenticate users based on a variety of sensory data. For example, SenSec [38] constantly collects sensory data from accelerometers, gyroscopes and magnetometers, and constructs the gesture model of how a user uses the device. The user studies has showed that SenSec achieved an accuracy of 75% in identifying the users and 71.3% in detecting the non-owners. Senguard [29] also investigates on a framework to continuously identify users based on a variety of sensory data. Touchscreen is one sensor of concern. However, the paper only visually shows that different users have different touch traces, without mentioning how to authenticate users based on these traces.

Using touch operations to authenticate users is a relatively new topic that has yet to capture extensive research attentions. Several recent work has studied how to improve the touch unlocking mechanism by considering touch biometrics. Such work includes [3, 9, 25, 26, 33]. De Luca *et al.* in [9] propose to track touch data of slide operations to unlock the screen. Touch data including time, position, size and pressure are used directly to authenticate users. Their work has achieved an overall accuracy of 77% using DTW (*i.e.*, Dynamic Time Warping) at best. Angulo *et al.* research on improving the lock patterns and introduce the notion of lock pattern dynamics [3]. Their work has achieved an EER of 10.39% using Random Forest machine learning classifier. Sae-Bae *et al.* focus on the specific five-finger touch gestures available on the Apple devices [25]. They model a user based on the movement characteristics of the five fingers and the palm center. An accuracy of 90% has been achieved over an Apple iPad. Shahzad *et al.* discuss a slide-based user authentication scheme, where a series of customized slides are used jointly to authenticate users [26]. It has been reported that a combination of three slides can achieve an average EER of 0.5%. Sun *et al.* propose TouchIn that allows user to draw on arbitrary regions with one or multiple fingers to unlock his mobile device. The user is authenticated based on the geometric properties of his drawn curves as well as his behavioral and physiological characteristics [33].

Other than improving screen locker security, several investigations focus on exploring the applicability of traditional keystroke-based authentication on smartphone with new features. KenSens [10] passively authenticates users via the specific location touched on each key, the drift from finger down to finger up, the force of touch, the area of press. The work in [23] also discusses the feasibility of employing keystroke dynamics to perform user verification on mobile phones and introduces a new statistical classifier. However, such work has not achieved great improvement in authentication accuracy. Zheng *et al.* propose to rely on more sensors (*e.g., accelerometers*) other than purely touchscreen [37]. They propose acceleration features which can reflect the magnitude of acceleration when the key is pressed and released. Their approach finally has achieved an average EER down to 3.65%.

Besides exploring touching biometrics on improving the screen lock or keystrokes, Frank *et al.* introduce the notion of continuous authentication via touch operations [14]. They focus on stroke operations. An EER of 13% for one single stroke, and 2% to 3 % for 11 consequent strokes have been achieved. Instead of only considering slide operation, Li *et al.* study both tap and slide, and achieved an accuracy of approximately 90% [21]. Feng *et al.* also study the continuous mobile authentication issues via touchscreen gestures [12]. They implement FAST (*i.e.*, Finger-gestures Authentication System using Touchscreen), where an extra glove equipped with sensors is used. FAST has achieved an FAR of 4.66% and an FRR of 0.13% using 7 touch sequences.

Our work also aims at exploring the applicability of continuous authentication relying only on touch operations. Unlike the existing work that using only one type of specific touch operation, our work comprehensively investigates a set of general, commonly-used types of touch operations on smartphone. Our authentication performance is better than that reported in [14] and [21] (the other existing work focuses on different problem settings, and is not comparable). More importantly, all existing work is based on the hypothesis that touch data qualifies good biometrics. Our work is the first to systematically evaluate the distinctiveness and permanence properties of touch biometrics. Such a study is the basis for touch-based authentication.

6. CONCLUSION

This work has suggested a touch-based authentication framework to continuously authenticate user. The authentication proceeds in a passive way while the user performs her normal touch operations. We proposed a set of methods targeting the problem of how to model multiple types of touch data produced by users. We further justified two critical properties of such data: distinctiveness and permanence. We presented our work together with a real-world experimental study. It is the first attempt to comprehensively evaluate the biometric properties of touch operations.

Although we have shown that touch operations bear good biometric properties, there is still a long way to implement a practical, touch-based continuous authentication system. First, the error rate when authenticating a user with one touch operation still cannot approach zero. We have hence suggested considering a set of touch operations jointly. Although we have shown some preliminary results with such a consideration, future research efforts (*e.g.*, consider the combination of different touch operations) are still required to examine it comprehensively. Secondly, our experiments have shown that the user features of touch operations are not stable over a period of time. Although we have suggested an adaptive approach that can mitigate such a problem, extensive future work is still needed to find an optimized adaptation method. Finally, there are quite a lot of other implementation issues of our touch-based continuous authentication framework. Examples include how to engineer a seamless touch operation tracing mechanism that runs silently as a smartphone background service and how to design a user-friendly mechanism to obtain data samples for training purpose.

Acknowledgements

The authors are grateful to our shepherd, Dr. Alex De Luca, who helps a lot in improving this manuscript. We also thank Xiaolei Zhang and the anonymous reviewers for their constructive comments. This work was supported by the Key Project of National Natural Science Foundation of China (Project No. 61332010), the National Basic Research Program of China (973 Project No. 2014CB347701), the National Natural Science Foundation of China (Project No. 61100077), the Shenzhen Basic Research Program (Project No. JCYJ20120619152636275), and the Research Grants Council of Hong Kong (Project No. CUHK 415113). Yangfan Zhou is the corresponding author.

7. REFERENCES

[1] Receiver operating characteristic. *http://en.wikipedia.org/wiki/Receiver_operating_characteristic*.

[2] A. Altinok and M. Turk. Temporal integration for continuous multimodal biometrics. In *Proc. of the Workshop on Multimodal User Authentication*, 2003.

[3] J. Angulo and E. Wästlund. Exploring touch-screen biometrics for user identification on smart phones. In *Privacy and Identity Management for Life*, pages 130–143. Springer, 2012.

[4] L. C. Araújo, L. H. Sucupira Jr, M. G. Lizarraga, L. L. Ling, and J. B. T. Yabu-Uti. User authentication through typing biometrics features. *IEEE Trans. on Signal Processing*, 53(2), Feb. 2005.

[5] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *Proc. of the 4th USENIX Conf. on Offensive Technologies*, pages 1–7, 2010.

[6] C. Bishop. *Pattern recognition and machine learning.* Springer, 2006.

[7] I. Brosso, A. La Neve, G. Bressan, and W. Ruggiero. A continuous authentication system based on user behavior analysis. In *Proc. of the 10th Int. Conf. on Availability, Reliability, and Security*, 2010.

[8] E. Chin, A. P. Felt, V. Sekar, and D. Wagner. Measuring user confidence in smartphone security and privacy. In *Proc. of the 8th Symposium on Usable Privacy and Security*, 2012.

[9] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you! implicit authentication based on touch screen patterns. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, 2012.

[10] B. Draffin, J. Zhu, and J. Zhang. Keysens: passive user authentication through micro-behavior modeling of soft keyboard interaction. In *Proc. of the 5th Int. Conf. on Mobile Computing, Applications and Services*, 2013.

[11] Egham. Gartner says smartphone sales grew 46.5 percent in second quarter of 2013 and exceeded feature phone sales for first time. *http://www.gartner.com/newsroom/id/2573415*, 2013.

[12] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In *Proc. of the IEEE 6th Int. Conf. on Biometrics: Theory, Applications and Systems*, 2013.

[13] I. Fischer, C. Kuo, L. Huang, and M. Frank. Smartphones: not smart enough? In *Proc. of the 2nd ACM workshop on Security and privacy in smartphones and mobile devices*, Oct. 2012.

[14] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Trans. on Information Forensics and Security*, 8(1), Jan. 2013.

[15] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature extraction: foundations and applications.* Springer-Verlag, 2006.

[16] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: a tool for information security. *IEEE Trans. on Information Forensics and Security*, 1(2), June 2006.

[17] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(1), Jan. 2004.

[18] R. Janakiraman, S. Kumar, S. Zhang, and T. Sim. Using continuous face verification to improve desktop security. In *Proc. of the 7th IEEE Workshops on Application of Computer Vision*, 2005.

[19] A. J. Klosterman and G. R. Ganger. Secure continuous biometric-enhanced authentication(cmu-cs-00-134). *CMU Technical Report*, 2000.

[20] G. Kwang, R. H. C. Yap, T. Sim, and R. Ramnath. An usability study of continuous biometrics authentication. *Advances in Biometrics*, (828-837), 2009.

[21] L. Li, X. Zhao, and G. Xue. Unobservable reauthentication for smart phones. In *Proc. of the 20th Network and Distributed System Security Symposium*, volume 13, 2013.

[22] D. T. Lin. Computer-access authentication with

neural network based keystroke identity verification. In *Proc. of the Int. Conf. on Neural Networks*, 1997.

[23] E. Maiorana, P. Campisi, N. González-Carballo, and A. Neri. Keystroke dynamics authentication for mobile phones. In *Proc. of the 2011 ACM Symposium on Applied Computing*, 2011.

[24] C. E. Metz. Basic principles of roc analysis. In *Seminars in Nuclear Medicine*, volume 8, pages 283–298. Elsevier, 1978.

[25] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, 2012.

[26] M. Shahzad, A. X. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it. In *Proc. of the 19th Annual Int. Conf. on Mobile Computing and Networking*, pages 39–50, 2013.

[27] C. Shen, Z. Cai, and X. Guan. Continuous authentication for mouse dynamics: a pattern-growth approach. In *Proc. of the 42nd Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks*, June 2012.

[28] S. J. Shepherd. Continuous authentication by analysis of keyboard typing characteristics. In *Proc. of the European Convention on Security and Detection*, 1995.

[29] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong. Senguard: passive user identification on smartphones using multiple sensors. In *Proc. of the 7th Int. Conf. on Wireless and Mobile Computing, Networking and Communications*, 2011.

[30] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar. Continuous verification using multimodal biometrics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(4), Apr. 2007.

[31] A. Smith. Smartphone ownership (2013 update). *http://pewinternet.org/Reports/2013/Smartphone-Ownership-2013.aspx*.

[32] S. N. Srihari, S. H. Cha, H. Arora, and S. Lee. Individuality of handwriting. *Journal of Forensic Sciences*, 47(4), July 2002.

[33] J. Sun, R. Zhang, J. Zhang, and Y. Zhang. Touchin: Sightless two-factor authentication on multi-touch mobile devices. *http://arxiv.org/abs/1402.1216*, 2014.

[34] S. Thomas. Touchscreen handsets dominanting uk mobile market! *http://www.3g.co.uk/PR/Nov2012/touchscreen-handsets-dominanting-uk-mobile-market.html*, 2012.

[35] D. Van Bruggen, S. Liu, M. Kajzer, A. Striegel, C. R. Crowell, and J. D'Arcy. Modifying smartphone user locking behavior. In *Proc. of the 9th Symposium on Usable Privacy and Security*, 2013.

[36] R. H. C. Yap, T. Sim, G. X. Y. Kwang, and R. Ramnath. Physical access protection using continuous authentication. In *Proc. of the IEEE Conf. on Technologies for Homeland Security*, 2008.

[37] N. Zheng, K. Bai, H. Huang, and H. Wang. You are how you touch: user verification on smartphones via tapping behaviors(wm-cs-2012-06). *Tech. Repo. of the College of William and Mary*, 2012.

[38] J. Zhu, P. Wu, X. Wang, and J. Zhang. Sensec: mobile security through passive sensing. In *Proc of the 13th Int. Conf. on Computing, Networking and Communications*, 2013.