ShakeIn: Secure User Authentication of Smartphones with Single-Handed Shakes

Hongzi Zhu, *Member, IEEE*, Jingmei Hu, Shan Chang, *Member, IEEE*, and Li Lu, *Member, IEEE*

Abstract—Smartphones have been widely used with a vast array of sensitive and private information stored on these devices. To secure such information from being leaked, user authentication schemes are necessary. Current password/pattern-based user authentication schemes are vulnerable to shoulder surfing attacks and smudge attacks. In contrast, stroke/gait-based schemes are secure but inconvenient for users to input. In this paper, we propose ShakeIn, a handy user authentication scheme for secure unlocking of a smartphone by simply shaking the phone. With embedded motion sensors, ShakeIn can effectively capture the unique and reliable biometrical features of users about *how* they shake. In this way, even if an attacker sees a user shaking his/her phone, the attacker can hardly reproduce the same behavior. Furthermore, by allowing users to customize the way they shake the phone, ShakeIn endows users with the maximum operation flexibility. We implement ShakeIn and conduct both intensive trace-driven simulations and real experiments on 20 volunteers with about 530,555 shaking samples collected over multiple months. The results show that ShakeIn achieves an average equal error rate of 1.2 percent with a small number of shakes using only 35 training samples even in the presence of shoulder-surfing attacks.

Index Terms—User authentication, smartphone application, single-handed shakes, inertial sensors, biometrics

1 Introduction

Last decade has witnessed the booming development of smartphones. With the powerful computing and sensing capabilities and a large storage of a modern smartphone, instead of just making phone calls, a rich set of complex applications, such as taking photos, investing in stocks, sending emails and banking, are made possible to run on such devices. According to the report of the European Union Agency for Network and Information Security [1], data leakage resulting from device loss or theft and unintentional disclosure of data has been the top two information security risks for smartphone users. The security problem of private information (e.g., personal photos, contact list, emails and bank accounts) stored on smartphones therefore is of great importance to smartphone users.

In both the industry and the literature, there is a rich set of user authentication schemes. The most widely adopted scheme is to let a smartphone lock itself after a short period of inactivity and prompt a user to input a password or some graphic pattern to unlock the phone. For example, iPhones use a four-digit password and Android systems use a geometric pattern on a grid of nine points. On one hand, when short passwords or simple patterns are adopted, these schemes are vulnerable to shoulder-surfing attacks where the passwords or the graphical patterns are easy to spy [2], [3]. Moreover, studies have also shown that finger smudges left on the touch screen of a smartphone can be used to infer short passwords and simple graphic patterns [4]. On the other hand, long passwords or complex patterns are inconvenient for users to input frequently, leading to unpleasant user experience. Recently, another new category of user authentication schemes based on user biometrics has received much attention. Either physiological characteristics (e.g., fingerprints and face recognition) or behavioral characteristics (e.g., voices, typing and stokes on touch screens) can be utilized to label or describe individuals. In general, these schemes focus on how users input as the authentication secret. Physiological-characteristics-based schemes can achieve satisfactory performance. For instance, newly distributed iPhones have a fingerprint sensor integrated with the Home button, which can actively read the fingerprint of users and unlock the phone [5]. Nevertheless, such schemes heavily rely on special sensors embedded on smartphones and often suffer from biometrics hacking attacks.

Current behavioral-characteristics-based schemes such as gait recognition, keystroke dynamics and phone usage statistics need an enormous amount of time to determine the legitimacy of a user and have low accuracy. Most recent schemes based on strokes on touch screens such as [6] can achieve very high accuracy but need two-handed operations which limits its applicable scenarios.

In this paper, we propose a smartphone user authentication scheme, called *ShakeIn*, based on customized single-handed shakes. As shown in Fig. 1a, a *shake* refers to a

- *H. Zhu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, P. R. China. E-mail: hongzi@sjtu.edu.cn.*
- *J. Hu is with the Department of Computer Science, School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138. E-mail: jingmei_hu@g.harvard.edu.*
- *S. Chang is with the School of Computer Science and Technology, Donghua University, Shanghai 201620, P. R. China. E-mail: changshan@dhu.edu.cn.*
- *L. Lu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 201620, P. R. China. E-mail: luli2009@uestc.edu.cn.*

Manuscript received 20 Apr. 2016; revised 15 Sept. 2016; accepted 7 Jan. 2017. Date of publication 11 Jan. 2017; date of current version 29 Aug. 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2017.2651820

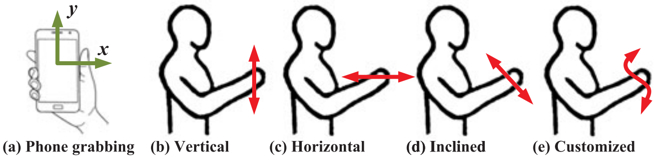


Fig. 1. Illustration of four shaking styles.

to-and-fro movement with one hand holding a smartphone and swinging the x- and y-axis coordinate plane of the phone around the elbow in the air. In essence, ShakeIn adopts a machine learning methodology, consisting of a training phase and an authentication phase. More specifically, in the training phase, ShakeIn first asks a legitimate user to choose his/her preferred shaking styles and collects a small number of shakes. For each of such shakes, unique and reliable biometrical features are derived from the raw readings of the embedded 3D accelerometer and the gyroscope sensors, and then utilized to establish a Supporting Vector Machines (SVM) classifier. In the authentication phase, ShakeIn use the pre-trained classifier to verify the legitimacy of shaking attempts from a user and unlock the phone if the user passes the verification. The key insight behind ShakeIn is that people have consistent and distinguishing physiological characteristics (e.g., the physical structure of the arm) and behavioral characteristics (e.g., shaking behavior patterns) while doing shakes.

To implement ShakeIn, there are three main challenges. First, as most smartphones are equipped with low-end motion sensors, both acceleration and rotation readings are error-prone. How to eliminate such errors while the attitude of the phone keeps changing is very difficult. In this work, by leveraging an inherent characteristic of shaking movements, we can find periodical *transition points* from which the shaking direction starts to reverse, which can be well exploited to remove accumulative errors. Second, how to choose features that can characterize the user is not straightforward. In ShakeIn, with transition points, we divide continuous shakes into segments and extract two behavioral patterns of *motion velocity* and *angular speed* and one physiological pattern of *shaking radius* based on shaking segments. Last, smartphones are often used in various conditions. For example, normal transport mobility (e.g., on a bus or subway train) can greatly affect the motion sensor readings and therefore the performance of ShakeIn. To tackle this challenge, we notice another tricky characteristic of shaking movements that the time duration between two consecutive transition points are quite short (e.g., about 100 ms), which means that the transport mobility change in such a short period of time can be negligible. As a result, shaking segments can be calibrated even without knowing the underlying transport mobility before features are extracted.

Compared with the state-of-art smartphone user authentication schemes, the novelty of ShakeIn is four-fold. First, ShakeIn is more difficult to compromise as it is very hard for an imposter to generate the same shakes as legitimate users do, especially for user-customized shakes, through shoulder surfing or biometrics hacking attacks. Second, ShakeIn allows a user to unlock his/her phone with single-handed operation, making it an easier choice for people with missing digits or in various scenarios where only one hand is available. Moreover, ShakeIn has no mandatory instructions on how users should shake, which endows users with the maximum flexibility. Third, ShakeIn is quite reliable and works well with various modes of transport such as cars, buses and subway trains, and in different user postures such as sitting and standing. Last but not least, as accelerometers and gyroscopes are widely available sensors in most off-the-shelf smartphones, it is easy to deploy ShakeIn. Furthermore, ShakeIn is lightweight, needing only a small number of shakes for training models and authenticating users. We implement ShakeIn on two Google Nexus 4 phones running Android, and evaluate the performance of ShakeIn via real-world experiments and trace-driven simulations with 530,555 shaking samples collected from 20 volunteers over multiple months. The results show that ShakeIn is very resilient to shoulder-surfing attacks and can achieve an average equal error rate (EER) of 1.2 percent with a few shakes of three different shaking styles.

The remainder of this paper is organized as follows. Section 2 compares ShakeIn with related work. We present the data collection and pre-processing in Section 4. Section 5 introduces the architecture of ShakeIn. Deriving effective signals representing shaking movements from raw sensory data is elaborated in Section 6. Section 7 describes how to extract and select effective features from motion signals. The procedures of training classifiers for single and multiple shaking styles and verifying the legitimacy of a user with those classifiers are introduced in Section 8. We discuss the reliability of ShakeIn under various conditions that may be encountered in real-world deployment are discussed in Section 9. Section 10 presents the performance evaluation and experiment results. Finally, we present concluding remarks of our work and summarize the directions for future work in Section 11.

2 Related Work

2.1 Physiological Characteristic Based

Several schemes [7], [8] have been proposed that utilize the accelerometer in smartphones to recognize human biometric gait. In general, these schemes have low true positive rates as it is sensitive to many uncontrollable factors such as the phone placement and the types of the ground surface and shoes. Other physiological characteristics such as fingerprints [9], face and sound could be utilized for authentication. However, requirements like large memory usage, high processing latency, and external devices make these kinds of authentication schemes unrealistic to be widely deployed on smartphones.

2.2 Behavioral Characteristic Based

Typing behavior with physical keyboards can be utilized to authenticate users [10], [11] but the performance of these schemes when applied to smartphones is uncertain as typing behavior on touch screens is more difficult to model. Some schemes [6], [12], [13] have been proposed to draw special gestures on the touch screen of a smartphone for authentication. For instance, GEAT [6] authenticates users based on distinguishing features such as finger velocity, device acceleration, and stroke time extracted when doing gestures. GEAT can achieve very low equal error rate and defend shoulder-surfing attacks. Sae-Bae et al. [13] propose to use the timing of performing five-fingered gestures on multi-touch capable devices for authentication. The limitation of these schemes is that they certainly require two-handed operations and their reliability under different modes of transport is unknown.

OpenSesame [14] and uWave [15] are the two schemes mostly related to our work. OpenSesame allows users to shake or roll their phones with no special requirements and derives four types of geometric features with three-axis raw acceleration readings. Probability density functions (PDFs) of those feature samples are further used to train classifiers and verify a user. UWave can verify the legitimacy of a user by comparing the time series of three-axis acceleration readings of a testing gesture drawn in the air to a pre-defined template library by employing dynamic time warping (DTW). These schemes have relatively high false positive errors especially under shoulder-surfing attacks. ShakeIn differs from both schemes essentially in how features are extracted. In ShakeIn, both physiological and behavioral characteristics are considered, which makes ShakeIn easy to use and at the same time resilient to shoulder-surfing attacks.

2.3 Machine-to-Machine Authentication Based on Shakes

ShaVe and ShaCK [16], [17] are two methods that use shaking for mutual authentication between a pair of mobile devices. Similarly, Bichler et al. [18] present an approach to establish a secure connection between two devices by shaking them together. Shot [19] is a scheme where accelerometer readings are leveraged to assist in the secure exchange of information between smartphones while maintaining the limited user involvement. Although these schemes are not for user authentication, they are very interesting and valuable for study.

3 System Model

In the system of user authentication of smartphones, we consider the following three key entities:

- *Smartphones:* are the devices to be protected. We require such a target smartphone to have an onboard accelerometer, a gyroscope, and a digital compass, which can constantly measure the motion and attitude of the smartphone, respectively. We have very limited requirements on the computation and storage capabilities of the smartphone and rely on no other special hardware.
- *Legitimate users:* have the right to access a smartphone. We require a user to be relatively stable during the training phase and the authentication phase, which means that, for one particular shaking style, the user should keep the way how he/she shakes as consistent as possible. Note that as we do not mandate any shaking styles, users can choose their preferred or habitual shaking styles which tend to be stable during a short period of time. We will demonstrate this point through intensive trace analysis and performance evaluation in Sections 9 and 10, respectively.
- *Imposters:* are deliberate or unintentional attackers who try to unlock an unauthorized smartphone. We assume imposters cannot have physical access to a smartphone during the training phase of ShakeIn. After the training phase, imposters have the following three capabilities. First, they can have physical access to the phone in cases such as thieves stealing a smartphone, finders finding a lost smartphone, and friends holding a smartphone when the owner temporarily leaves. Second, imposters can launch shoulder surfing attacks by spying or even recording the owner when he/she performs shakes. Third, imposters have necessary equipment and technologies to launch biometrics hacking attacks.

4 Data Collection

In this section, we describe the process for collecting and pre-processing raw shaking data from smartphones.

4.1 Collecting Shake Data

We collect shake data with Google Nexus 4, a standard Android smartphone, with which raw readings on each axis of the 3D accelerometer and the gyroscope embedded on the phone can be recorded. The sampling frequency is 200 Hz and the measure range of the 3D accelerometer is $[-4G, 4G]$, where G is the gravitational constant.

We recruit 20 volunteers, five females and fifteen males, aged from 18 to 43, including five undergraduate students, nine graduate students, three faculty members, and three office staff. In general, each volunteer helps collect their shaking data for three times a day, i.e., in the morning, after lunch, and in the evening. For each time, each volunteer is asked to shake a phone in two postures (i.e., sitting or standing) and for each posture, the following four shaking styles as illustrated in Figs. 1b, 1c, 1d, and 1e are performed: *vertical* where the phone is shaken in a vertical plane, *horizontal* where the phone is shaken in a horizontal plane, *inclined* where the phone is shaken in a plane inclined from upper right to lower left before the body, and *customized* where the phone is shaken in an arbitrary plane chosen by the volunteer. For each style, each volunteer was asked to shake the phone for twenty times. It should be noted that we require volunteers to shake in the first three given styles only for comparison and have no such requirement when applying ShakeIn in practice.

We collect two data sets of shakes over two periods in the year of 2014, i.e., two weeks from Jul. 1 to Jul. 14 (denoted as *trace* $\mathcal{A}$) and over one month from Sep. 15 to Oct. 20 (denoted as *trace* $\mathcal{B}$).

4.2 Removing Noise

The time series of acceleration and angular speed along each axis can be treated as signals, which contains high-frequency noise. This can be seen in Fig. 2a which shows the y-axis acceleration readings of several vertical shakes. We consider frequencies above 15 Hz as noise because it can be seen that most energy is contained in frequencies below 15 Hz as we can see from the fast Fourier transform (FFT) result shown in Fig. 2b. We need to remove such high-frequency noise as it would affect the results of shaking velocity and angular speed calculation.

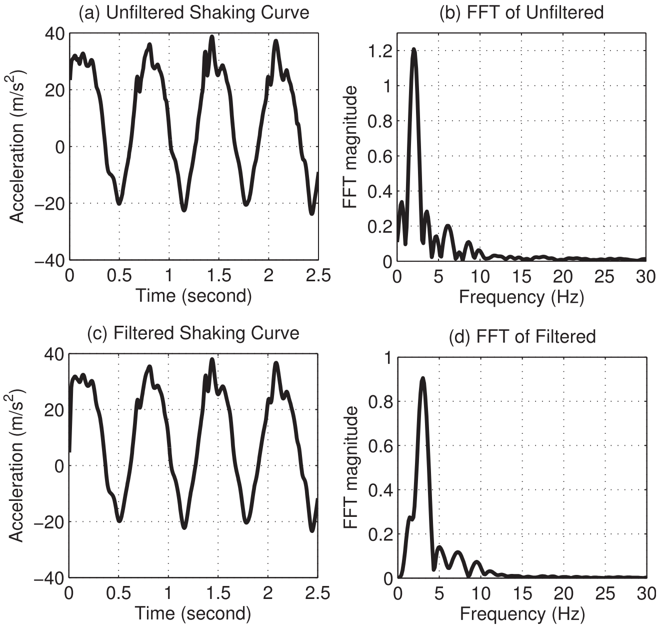


Fig. 2. Unfiltered and filtered shaking signals of acceleration.

5 ShakeIn Architecture

In general, ShakeIn consists of two phases: *training phase* and *authentication phase* as shown in Fig. 3. In the training phase, shaking motion of a legitimate user is first captured by the *motion estimator* (ME) which outputs three motion signals from the raw readings of the embedded accelerometer and gyroscope. With those derived motion signals, the *feature extractor* (FE) further profiles the user's shaking behavior based on a group of pattern-based features. Combined with the shaking style information which is estimated by the *phone attitude estimator*, features are used to train a corresponding *SVM classifier* corresponding to this shaking style. In the authentication phase, a user needs to shake the phone in order to unlock the phone. These testing shakes are also used to resolve the shaking styles and features, which are sent to the *verifier* to determine the legitimacy of the user based on pre-trained classifiers.

Motion Estimator. ME performs two key functions. First, it determines whether or not a user is shaking the phone. Second, if shakes are detected, it produces three motion signals, i.e., the angular speed, the tangential velocity and the shaking radius, to represent the shaking motion from the raw sensor readings. The key to ME is that it should always get accurate motion estimation despite how the user shakes the phone and what posture the user lies in or what transport vehicle the user is taking (described in Section 6).

Feature Extractor. The key function of the FE is to extract effective features to characterize the user. With the real-world traces we have collected, different types of features are studied. FE extracts three pattern-based features from corresponding motion signals, which are preferable for authentication, i.e., being both consistent for the same user over time and distinguishing between different users, as described in Section 7.

Phone Attitude Detector (PAD). In ShakeIn, users are allowed to customize the way they shake a phone for authentication. The main function of PAD is to track the shaking attitude of the phone by calculating the Euler angle of the z-axis of the phone in the terrestrial coordinate system, which is presented in Section 8.

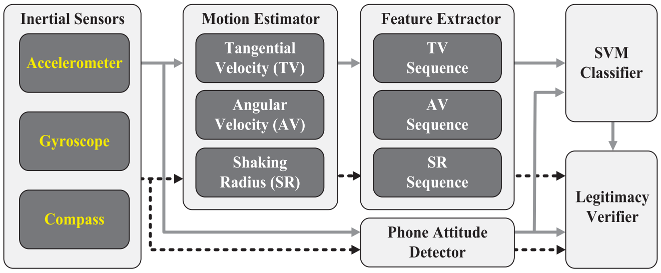


Fig. 3. Architecture of ShakeIn. Solid arrow lines show the data flow in the training phase and dashed arrow lines presents the data flow in the authentication phase.

Training Classifiers. As ShakeIn are designed to authenticate users, therefore, feature vectors derived from legitimate users are used to train classifiers. In ShakeIn, we train a one-class SVM classifier for each shaking style of a legitimate user, as described in Section 8.

Legitimacy Verifier (LV). The function of LV is to verify the legitimacy of a user trying to unlock the phone. The decision can be made based on the classification result of testing shakes in one shaking style or the result of multiple voting on individual classification results of multiple shaking styles. We describe the procedure in Section 8.

6 Motion Estimation

Before any meaningful features can be extracted, we need to characterize the shaking behavior with certain motion signals instead of using raw sensory data. In ME, as shown in Fig. 4, we estimate three related signals, i.e., the tangential velocity, the angular speed, and the shaking radius, denoted as V, Ω and R, respectively. As the angular speed can be directly obtained from the z-axis gyroscope readings, we elaborate the process of deriving the tangential velocity and the shaking radius.

6.1 Deriving Tangential Velocity

Suppose the y-axis of the accelerometer is along the shaking direction of the phone as shown in Fig. 4. To obtain the tangential velocity at time instant t_i, one basic solution is to calculate the integral of the acceleration readings along y-axis a_y over time

$$V(t_i) = \int_{t_0}^{t_i} a_y(t)\mathrm{d}t + V(t_0), \qquad (1)$$

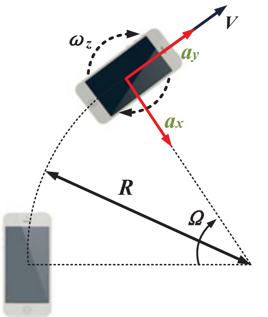


Fig. 4. Signals of interest in a shake.

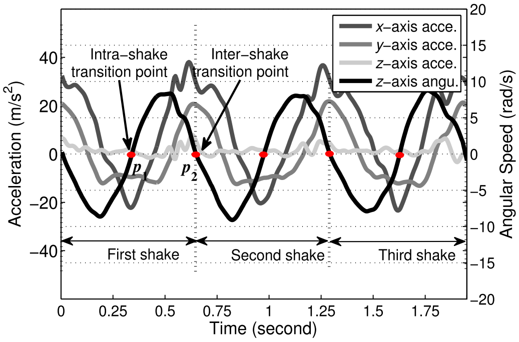


Fig. 5. Identifying transition points.

where $a_y(\cdot)$ is the acceleration function of time and $V(t_0)$ is the initial velocity at time instant t_0. As the accelerometer takes the acceleration samples at a certain sampling rate instead of producing a continuous function $a_y(\cdot)$, therefore, $V(t)$ can be calculated as

$$V(t) = \sum_{i=0}^{t \cdot k} \frac{1}{k} \cdot a_y(i) + V(t_0), \tag{2}$$

where k is the sample rate of the accelerometer and $a_y(i)$ is the ith received reading from the accelerometer's y-axis.

Though the basic solution is simple, it is very challenging to achieve high-accuracy velocity estimates of the phone due to the inherent noise from sensor readings. As a result, the estimation errors are accumulated when integrating the accelerometer's readings over time. For example, the solid curve in Fig. 6 shows the tangential velocity estimates obtained with the basic solution over three shakes. It can be seen that the integral estimates grow rapidly over time. Therefore, in order to get accurate shaking velocity, the accumulative error must be eliminated.

To this end, we thoroughly investigate the motion of shaking a phone and have two key observations. First, we refer to the time point when the direction of a shake is going to change as an *intra-shake transition point*, and the time point just before the next shake starts as an *inter-shake transition point*. In general, transition points are the moments when the direction of shaking movement starts to change and therefore the tangential and the angular velocities of the phone should both be zero. Second, agreed with observations found in prior work [20], [21], we find such accumulative errors of integral is an approximate linear function of time. For example, the integral velocities at transition points in Fig. 6 can be well fitted by a linear function of time. Given the fact that the true velocities at transition points are zero, the linear model of errors can be derived and utilized to infer true tangential velocities. In specific, transition points can be identified when the angular speed along z-axis reaches zero as illustrated by the dots in Fig. 5. Let p_1 and p_2 denote two transition points and $V(p_1)$ and $V(p_2)$ denote the integral velocity values at p_1 and p_2 calculated using (2), respectively. With linear accumulative errors, the slope of the linear model, i.e., the constant reading errors of the accelerometer, can be estimated with

$$err_a = \frac{V(p_2) - V(p_1)}{p_2 - p_1}. \tag{3}$$

As a result, the true tangential velocity between p_1 and p_2 can be estimated as

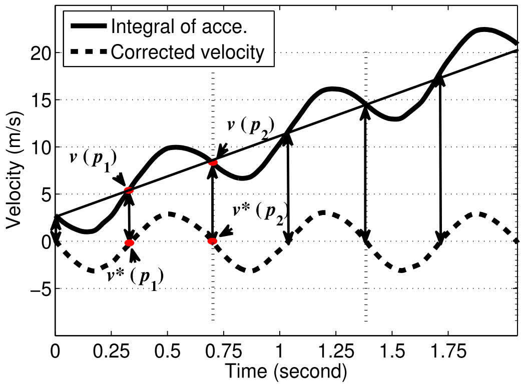


Fig. 6. Corrected velocity estimation.

$$V^*(t) = V(t) - V(p_1) - err_a \times (t - p_1). \tag{4}$$

Thanks to the inherent nature of shaking motion, we can constantly get transition points and therefore obtain correct tangential velocities (as illustrated by the dashed curve shown in Fig. 6).

6.2 Deriving Shaking Radius

Shaking can be regarded as back-and-forth movements of complex arcs, involving several parts of an arm such as the forearm, the elbow, the upper arm and even the shoulder. Therefore, the shaking radius information contains rich physiological characteristics of the user and could be utilized to label the user. Although the whole shaking trajectory in practice cannot be a perfect circular arc, if we divide time into short time slots, then the movement in such a short time slot can be treated as circular. With this approximation, we can calculate the shaking radius at time instant t as follows:

$$R(t) = \frac{V^*(t)}{\Omega(t)}, \tag{5}$$

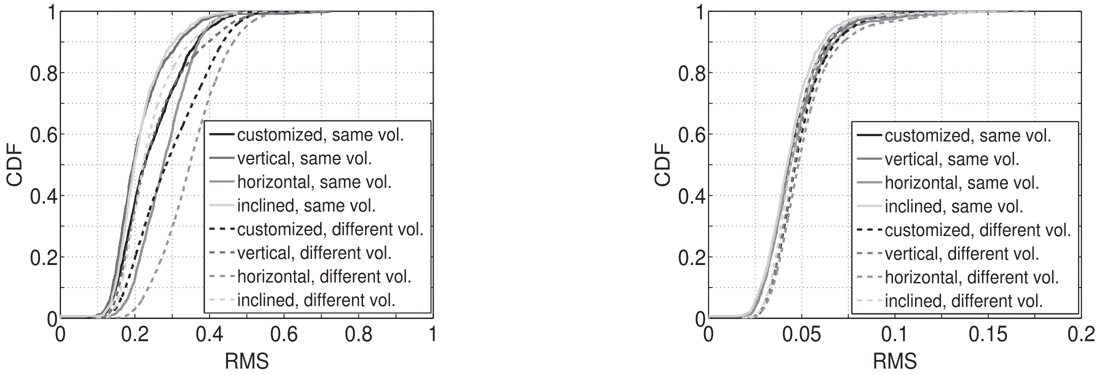
where $V^*(t)$ and $\Omega(t)$ are the corrected tangential and angular velocities of the phone at time instant t. The angular speed of the phone can be retrieved from z-axis gyroscope readings. Since the calculation of $R(t)$ involves division operation, in order to avoid divide-by-zero errors at transition points, we use a sliding window to average $V^*(t)$ and $\Omega(t)$ before doing division.

6.3 Pre-Processing Motion Signals

For the convenience of further analysis, we first divide long time series of all three motion signals of multiple shakes into short ones of individual shakes, according to inter-shake transition points. As the durations of shakes vary, in order to compare two shakes, we re-sample the corresponding time series of motion signals to a fix and sufficiently-large size of N elements. Hereafter in this paper, without specification, all time series of motion signals are segmented and re-sampled with the same length.

7 Feature Extraction

With derived motion signals about shaking, we examine what features can effectively capture the unique biometrics of users in this section.



(a) Shaking duration difference

(b) Average angular speed difference

Fig. 7. Efficacy of magnitude-based features.

7.1 Extracting Biometric Features

We examine three potential types of features as follows:

Duration-Based. Intuitively, people tend to shake a phone at a constant frequency in a normal mood and therefore the durations of shakes might be utilized to authenticate a person. For one shake, we extract two time durations from the angular speed signal, i.e., one from the starting of the shake to the intra-shake transition point and the other from the intra-shake transition point to the end of the shake. Therefore, a vector of $2k$ duration values can be obtained for a consecutive k shakes.

Magnitude-Based. Different participants may produce shakes with distinct magnitudes of all motion signals. To extract magnitude-based features, we divide each shake into two sub-shakes according to the intra-shake transition point and calculate the average, the variance, and the extrema of all motion signals based on sub-shakes. As a result, for each of such magnitude-based features, we can obtain a vector of $2k$ values for a consecutive k shakes.

Pattern-Based. To represent the differences between two shakes performed by distinct users, pattern-based features such as histograms and cumulative distribution functions (CDFs) of motion signals over multiple shakes can be used. In order to capture both temporal and structural patterns, we use the *normalized sequences* of motion signals of one shake. In specific, in addition to the re-sampling operation (as introduced in Section 6.3) which makes the length of all shakes equal to N values, all time series of motion signals of one shake are normalized. As a result, all values within the resolved time series lie in the range of $[0, 1]$, which removes the bias caused by uneven magnitude of different shakes. After that, the normalized time series are further divided into M segments with each segment having N/M values. Thus, we obtain the normalized sequences of M values of all motion signals of one shake, each value of which is calculated as the average of the N/M values in the corresponding segment.

7.2 Selecting Effective Features

Being consistent for the same user over time and space and being distinguishing among different users are essential to good features for authentication. To examine the efficacy of above features, given the same shaking style, we calculate the root mean squared (RMS) values between feature vectors extracted from shakes performed at different time by the same participant and between feature vectors extracted from shakes performed by different participants, over all participants and time.

In specific, let X_1 and X_2 denote two feature vectors. We calculate the root mean squared value of those vectors by subtracting the normalised values of X_1 from the normalised values of X_2 as follows [6]

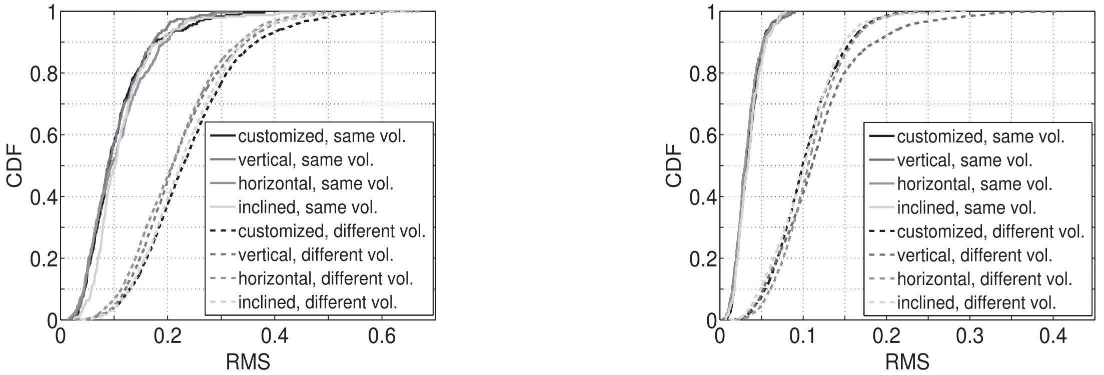
$$RMS = \sqrt{\frac{1}{n_2}\sum\nolimits_{i=1}^{n_2}\left(\hat{X}_2[i]-\hat{X}_1[i]\right)^2}, \qquad (6)$$

where $\hat{X}[i]$ presents the ith value in the feature vector of X. We study the RMS distribution of different types of features using all traces of all participants and have the following observations.

Shaking Durations are not Stable. For example, Fig. 7a plots the CDFs of RMS values of duration-based features. We have two following observations. First, shakes performed by the same participant are more similar than those performed by different participants as the RMS values between feature vectors extracted from the same participant are smaller than those extracted between different participants. For instance, over 60 percent RMS values are less than 0.25 when a participant shakes a phone at different time in the customized style whereas the value is 0.32 when comparing customized shakes between different participants. Second, the effectiveness of the shaking duration feature largely depends on how users shake. For instance, the differences between the CDFs obtained when the phone is shaken in the customized and horizontal styles are relatively larger than those achieved when the phone is shaken in the vertical and inclined styles.

Magnitude-Based Features are not Distinguishing. For example, Fig. 7b plot the CDFs of RMS values of the average angular speed. It can be seen from both figures that this feature has very good consistency performance over time. For instance, over 90 percent RMS values obtained from shakes of the same participant are less than 0.07 despite different shaking styles. However, this feature has very limited capability to distinguish different users as the RMS values obtained from shakes of different participants are very close to those obtained from shakes of the same participant. We have similar observations about other motion signals and other magnitude-based features and omit their CDFs of RMS values due to the page limitation.

Pattern-Based Features are Reliable and Unique. For example, Figs. 8a and 8b plot the CDFs of RMS values obtained



(a) Radius pattern difference

(b) Angular speed pattern difference

Fig. 8. Efficacy of pattern-based features.

with the normalized sequences of the shaking radius and the angular speed, respectively. We have three main observations as follows. First, pattern-based features have excellent consistency as the overall RMS values are quite small. Second, these features also have strong capability to distinguish different users, which is confirmed by the obvious gaps between the CDFs obtained from the same participant and those obtained from different participants. Third, these features also have supreme stability over various shaking styles. As a result, pattern-based features can perfectly capture how users behave when they shake their phones. We have similar observations for the tangential velocity.

As a result, we extract pattern-based features from all three motion signals to profile users.

8 Classifier Training and Authentication

In this section, we describe the details of ShakeIn on training its classifiers and how ShakeIn conducts authentication.

8.1 Detecting Phone Attitude

It should be noted that the recognition of shaking styles is automatic, which means in both the classifier training phase and the authentication phase, the user does not need to be interrupted. In specific, when the user shakes the phone in one arbitrary plane, ShakeIn constantly tracks the Euler angle of the z-axis of the phone in the terrestrial coordinate system. One simple scheme is to utilize the Android system call to get the rotation matrix transforming the geomagnetic vector into the same coordinate space as gravity. Suppose the ration matrix is

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, \tag{7}$$

the z-axis Euler angle of the phone can be calculated as $arc\tan(r_{21}/r_{11})$. Due to the presence of magnetic materials often affects the compass of the phone, we adopt the scheme proposed in [21] to mitigate the error.

8.2 Training Single Shaking Style Classifier

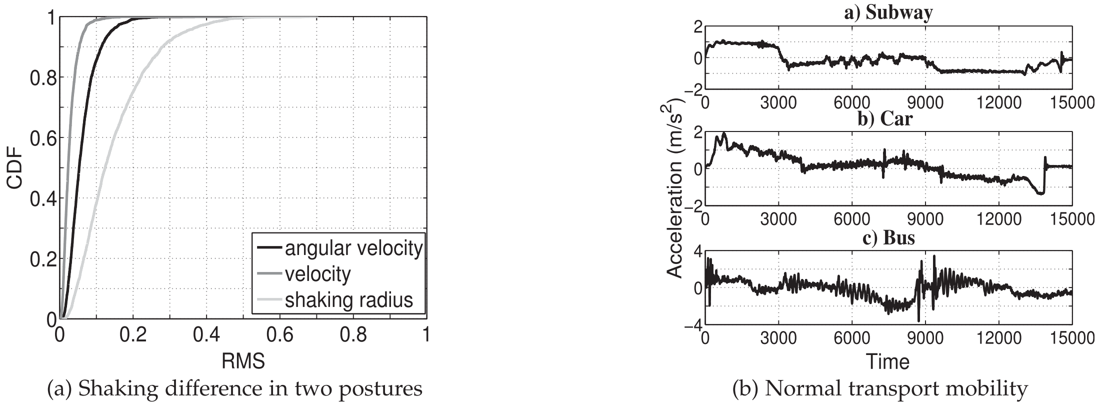
Since training samples are all from the legitimate user of a smartphone, one-class Support Vector Machine (SVM) classifier [22], [23] with the Radial Basis Function (RBF) kernel function [24] is effective and efficient [6], [25]. In ShakeIn, for each training shake, one feature vector which combines the normalized sequences of the shaking radius, the angular speed and the tangential velocity is obtained. We train a one-class SVM classifier using a group of training shakes for each shaking style of the user. We use the open source implementation of one-class SVM in libSVM [26].

In the RBF kernel function of SVM, there are two parameters, i.e., the penalty parameter c and the gamma parameter g, which impact the effect of training model. To obtain the appropriate parameters of c and g for one-class SVM, we adopt the scheme proposed in [6] and conduct a grid search over the same range of $[2^{-8}, 2^{8}]$ with cross validation on the training group. As all shaking samples are all from the same user, cross validation during the grid search only measures the true positive rate (TPR). As learned from our empirical experiments with the trace, a SVM classifier even with a rough configuration of both parameters trained using pattern-based features of shakes can easily reject a testing shake performed by an imposter. In ShakeIn, we choose the parameter values of c and g when the grid search finds the highest value of TPR as the best configuration to train SVM classifiers.

8.3 Verifying Legitimacy of Users

Authentication with Single Shaking Style (SSS). Given a set of testing shakes of a specified shaking style, we extract the normalized sequences of the desired motion signals from the testing shakes and form a set of feature vectors. Then we feed each feature vector to the classifier trained for the legitimate user. If the ratio of the number of accepted vectors to the total number of test vectors is higher than an *acceptance threshold*, ShakeIn accepts this testing shake as legitimate and unlocks the phone; otherwise, the user performing this group of test shakes is considered as illegitimate.

Authentication with Multiple Shaking Styles (MSS). In ShakeIn, we allow a legitimate user to define multiple shaking styles to increase the security level for user authentication. In the training phase, for each shaking style, ShakeIn trains a separate classifier. In addition, ShakeIn associates the phone attitude information with the trained classifier. In the authentication phase, the legitimate user sets a value of n, the number of shaking styles that the user needs to do in each authentication attempt. When a user tries to unlock a phone, ShakeIn prompts the user to shake the phone in n shaking styles. Then, testing shakes are associated to the corresponding classifiers according to the



(a) Shaking difference in two postures

(b) Normal transport mobility

Fig. 9. Reliability of ShakeIn in various conditions.

phone attitude of z-axis Euler angle. Last, ShakeIn classifies each shaking style as described in Section 8.2 and uses majority voting to determine the final legitimacy of the user.

9 Reliability

The reliability of ShakeIn under various working conditions is critical for real world deployment. In this section, we discuss the scenarios that ShakeIn might encounter in practice.

9.1 Different Postures

In ShakeIn, as a user mostly uses his/her smartphone in one of the two postures, i.e., sitting or standing. We compare the similarity of time series of motion signals achieved while a participant is sitting and while he/she is standing. Fig. 9a plots the CDFs of RMS values of all motion signals obtained in the postures of sitting and standing over all participants in our traces. It can be seen that there do exist slight differences between the motion signals achieved in sitting posture and achieved in standing posture, especially for the shaking radius. The reason might be that the motion of the arm is constrained when seated due to the limited room available.

To deal with different postures, one possible solution is to train a separate classifier for each posture. When testing, the shake input of a user is classified with each of those classifier. If one of the classifiers labels the input as legitimate, we accept the user and skip the rest classifiers. If no classifier accepts the input, then this user is treated as illegitimate. Although doing this would increase the reliability of the authentication scheme, it also significantly increases the burden of users for training classifiers. In contrast, another solution is to train a unified classifier for both postures, which simplifies the training procedure but comes at a cost of performance loss. We further examine the performance of both solutions in the performance evaluation.

At current stage, one main limitation of ShakeIn is that we do not consider other postures such as walking and running. The reason is two-fold: first, shaking behavior in those postures can vary significantly affected by too many factors such as different types of surfaces and shoes; second, it is easy and safe for a user to temporarily stop before the user trying to use the phone.

9.2 Transport Mobility

In ShakeIn, as shaking features are derived from motion sensors, the mobility of transport vehicles could also be perceived, polluting the desired tangential velocity of shakes.

We examine that ShakeIn is immune to normal transport mobility. According to (3) and (4), the tangential velocity of shakes can be estimated even without knowing the true velocities at transition points, as long as the condition $V(p_1) = V(p_2)$ holds. With fast repeated shaking movements, the time duration between two consecutive transition points (one is intra-shake and the other is intershake) is quite short at a scale of one or two hundred milliseconds. As shown in Fig. 9b, the acceleration or deceleration process of a transport vehicle, however, is mild and normally happens at a much larger time scale of seconds. Thus, even when on a vehicle, the velocity change between two consecutive transition points is negligible, i.e., $V(p_1) \approx V(p_2)$. We further examine the impact of transport mobility to the performance of ShakeIn in the performance evaluation.

10 Evaluation

We evaluate the performance of ShakeIn through both trace-driven simulations and real-world experiments, considering three metrics, i.e., *false positive rate* (FPR), referring to the probability of treating an imposter as the legitimate user when testing, *false negative rate* (FNR), referring to the probability of rejecting the legitimate user when testing, and *equal error rate*, referring to the error rate when FNR equals FPR. For each experiment, we repeat that experiment for 10 times and present the average error rates.

10.1 Effect of Kernel Functions and Acceptance Threshold

In this experiment, we examine the effect of different kernel functions adapted in training classifiers and the acceptance threshold through trace-driven simulations. In specific, we use *trace* $\mathcal{B}$ and divide the trace into two parts, i.e., shaking samples for training collected in the first two weeks (denoted as set $T = \{T_1, T_2, \ldots, T_{14}\}$, where T_i is the set of shakes collected on the ith day since Sep. 15), and shaking samples for testing collected in the following two weeks (denoted as set $S = \{S_1, S_2, \ldots, S_{14}\}$, where S_i is the set of shakes collected on the ith day since Sep. 29). For each shaking style of each volunteer, we randomly select 50 shakes from T to train a one-class SVM classifier. For testing, we treat each volunteer as a legitimate user once and treat the rest as imposters for the current legitimate user, conducting *10-shake SSS authentication*, where 10 consecutive shakes

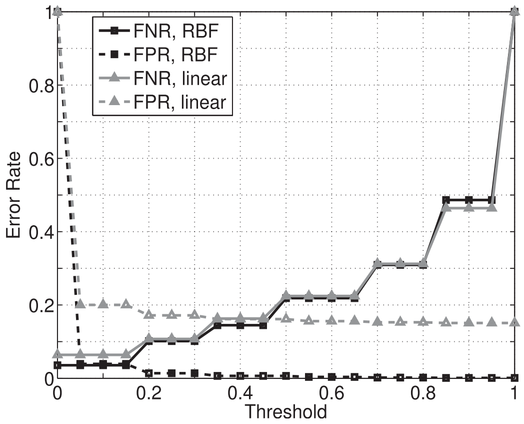


Fig. 10. FPR and FNR versus system parameters.

randomly selected from S are used in SSS authentication. To be fair, we do authentication 95 times for a legitimate user and 5 times for each imposter, which makes the number of tests from the legitimate user and that from all imposters equal. We adopt both the linear kernel function and the RBF kernel function. For the RBF kernel function, we conduct the grid searching as described in Section 8.2 to find the most appropriate parameter. We vary the acceptance threshold from zero to one with an interval of 0.05 and calculate the average error rates. Fig. 10 plots the average FPR and FNR as functions of the acceptance threshold. It can be seen that using RBF kernel function can achieve the best EER of about 4.6 percent with an acceptance threshold value of 0.1.

10.2 Effect of Training Data Age and Size

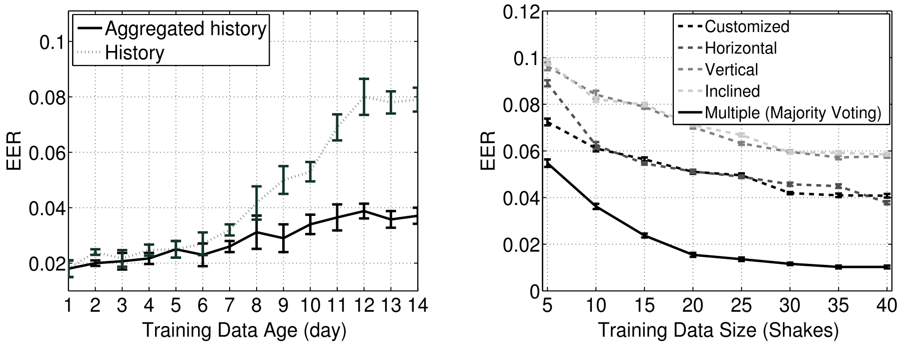
In this experiment, we first study how shakes evolve along time. The experiment setting is similar to the above experiment except that we vary the training data set. Specifically, for each shaking style of each volunteer, we build a separate SVM classifier using 20 shakes randomly selected from each T_{14-a+1} for $a = 1, \ldots, 14$, where a is denoted as the age of the training data. In addition, given an age a for $a = 1, \ldots, 14$, we also aggregate those chosen shakes selected from T_{14} to T_{14-i+1} to train a SVM classifier. In testing, in addition to 10-shake SSS authentication, we also conduct 4×3 *MSS authentication*, where four shakes of each of the three shaking styles (i.e., customised, horizontal and vertical) randomly selected from S are used in MSS authentication. Fig. 11a plots the average EER as a function of the data age. It can be seen that, in general, the average EER increases as the training data ages, especially when the training data are older than one week. In addition, using aggregated history data to train classifiers can achieve better performance.

We then study how many shakes are sufficient to profile a user. Suggested by the above observation, we use training data set of one week $T' = \{T_8, \ldots, T_{14}\}$ and vary the number of shakes used in training classifiers from 5 to 40 with an interval of five shakes. Fig. 11b shows the average EER as a function of the number of training shakes. We have several observations. First, the average EER drops as the training size increases and gradually stabilizes. Second, different shaking styles have distinct authentication effectiveness. The reason might be that a user performs more stably when shaking his/her phone in an easier or more natural way. Last, MSS authentication can significantly decreases the EER, even when individual classifiers have poor accuracy performance. For example, the average EER is 1.3 percent for the 4×3 MSS authentication using 35 training shakes for each shaking style.

The results suggest that ShakeIn can retain classifiers using most recent shakes which have successfully unlocked the phone during the last few days. According to the recent report that people might need to deal with on average 63.5 notifications per day with their smartphones [27], ShakeIn can easily obtain required shakes for retraining.

10.3 Effect of Authentication Complexity

We further examine the authentication complexity for a user to unlock a ShakeIn-enabled phone. For each shaking style, we use 35 shakes randomly selected from T' to train a classifier and vary the number of shakes used for a SSS authentication from one to six. Fig. 12 plots the average EER as a function of the number of shakes performed in authentication. It can be seen that, for SSS authentication, a small number of six shakes of single style can achieve satisfactory accuracy with an average EER value of 4.3 percent. With the 4×3 MSS authentication as described in the previous experiment, ShakeIn achieves an average EER value of 1.3 percent. In particular, when we set the true positive rate to 90 percent with other configuration being the same, the average FPR decreases to 0.9 and 0.4 percent for SSS and MSS authentication, respectively. As a result, ShakeIn can achieve excellent accuracy with low authentication complexity.



(a) Average EER vs. training data age (b) Average EER vs. training data size

Fig. 11. Impact of history data on training classifiers.

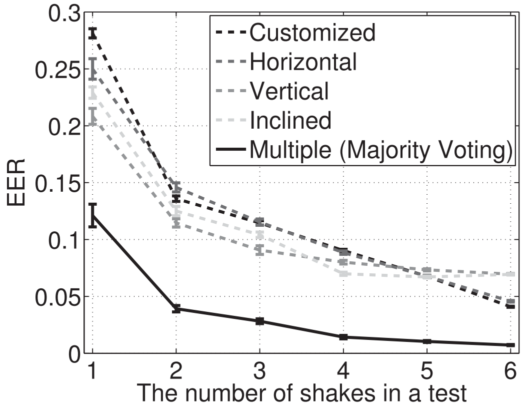


Fig. 12. Average EER versus authentication complexity.

10.4 Impact of Operation Postures

In this experiment, we examine the effect of operation postures to the performance of ShakeIn. For each shaking style, we train different models, i.e., *separate* and *unified*. For the separate model, we train a separate SVM classifier for sitting and standing, respectively. In the unified model, we do not distinguish postures and train a single classifier for all shakes. We set up the experiment similar with the above experiment. Fig. 13 shows the average FPR and FNR of the 4×3 MSS authentication as functions of the acceptance threshold achieved with both models. It can be seen that the separate model can achieve better average EER value of 1.3 percent. Using the unified model can achieve similar accuracy with the average EER value of 2.6 percent. Therefore, it is beneficial to distinguish different postures and build separate classifiers.

10.5 Impact of Transport Mobility

We investigate the impact of transport mobility to the performance of ShakeIn. We examine three different transport vehicles, i.e., subway trains, private cars and buses. Real-world shakes conducted on different transport vehicles were collected from 10 of our volunteers for one week (denoted as *trace $\mathcal{C}$*). For each transport mode, a volunteer was asked to conduct shakes in the four required styles during the acceleration, the deceleration and the complete stop periods. For comparison, we train new SVM models for each shaking style of each volunteer, using shakes collected during complete stops in the first three days. We conduct six-shake SSS authentication using the rest of the trace. We repeat the experiment for 10 times and calculate the average true positive rate over all shaking styles. Fig. 14 shows the bar plots of the average TPR when applying the original model trained with trace $\mathcal{B}$ and the new model trained with trace $\mathcal{C}$, respectively. It can be seen that using new models has better performance, which indicates that the shaking behavior of a user when standing still is slightly different from that when he/she takes a transport vehicle. However, it can also be seen that using original models can still achieve good accuracy, especially when taking a private car. This verifies that ShakeIn is very reliable to use for common transport modes.

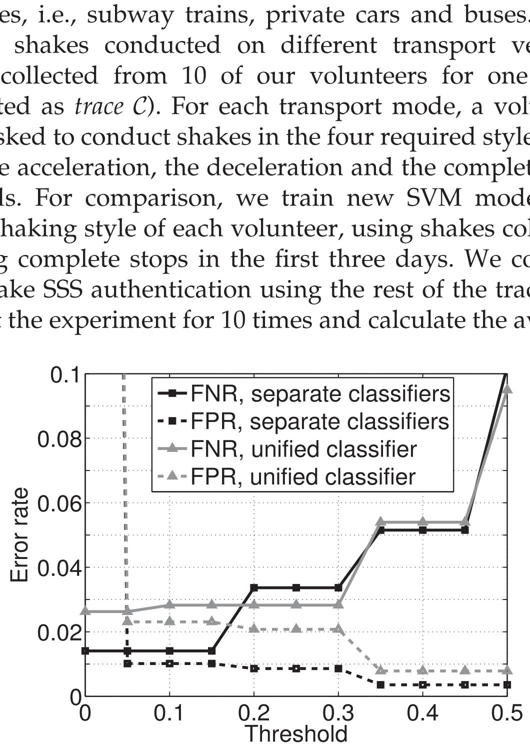


Fig. 13. FPR and FNR in different postures.

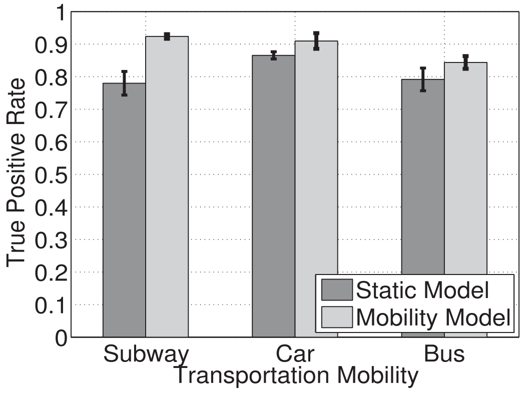


Fig. 14. TPR under different modes of transport.

10.6 Real-World Attack Experiment

We implement ShakeIn on five Google Nexus 4 Android smartphones equipped with a quad-core 1.5 GHz CPU and 2 GB memory, adopting the RBF kernel function. The average time for training one single SVM classifier with 35 training samples and verifying the legitimacy of a user with six shakes is 5.3 and 0.7 s, respectively.

We examine whether ShakeIn can defend shoulder-surfing attacks via real-world experiments, following the suggestions proposed in [28]. In specific, we randomly select five volunteers, two females and three males, as legitimate users, and 10 volunteers, three females and seven males, as imposters. For each legitimate user, we first let him/her to choose three most comfortable shaking styles and postures, and train corresponding SVM models for each shaking style on one of the five smartphones. Then, we ask each legitimate user to perform both 6-shake SSS authentication and 4×3 MSS authentication for twenty times each and record the whole process on tape. For imposters, they are allowed to perform live observations on how a legitimate user unlock his/her phone. In addition, they are allowed to watch the taped video as many times as they want as well. We then let imposters rehearse before requiring them to perform one hundred authentication attempts. The results turn out that the average TPR over all five users is 98 percent. Fig. 15 shows bar plots of the average FPR of each imposter over all five users. The average EER for SSS and MSS authentication schemes over all 10 imposters turned out to be 4.1 and 1.2 percent, respectively. The results show that ShakeIn is very resilient to shoulder-surfing attacks.

10.7 Comparison with Existing Schemes

We compare the performance of ShakeIn with two most related schemes, i.e., uWave [15] and OpenSesame [14]. We

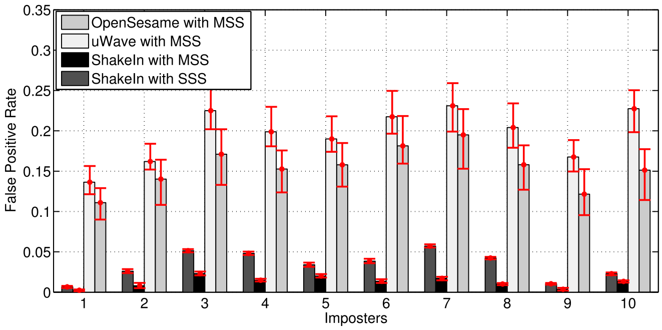


Fig. 15. Shoulder-surfing attacks against ShakeIn, uWave and OpenSesame.

recruit the same volunteers as the above experiment. For uWave, each legitimate volunteer defines three password gestures as suggested in [15] and performs each gesture for 35 times. To unlock the phone, each volunteer draws each pre-defined gesture for four times. To be fair, we also adopt the majority voting for all gestures in determining the legitimacy of users. For OpenSesame, each legitimate volunteer is also asked to perform shakes in three personalized styles. We use 35 shakes to train SVM models and conduct 4×3 MSS authentication. We conduct shoulder-surfing attacks on each scheme with a similar setting with the above experiment. The average TPR of uWave and OpenSesame is 93 and 88 percent, respectively. As to the security, Fig. 15 also shows bar plots of the average FPR of each imposter over all five users, achieved by using MSS-enabled uWave and OpenSesame, respectively. The average FPR of uWave and OpenSesame is 19.6 and 15.4 percent, respectively. It can be seen that ShakeIn outperforms both schemes.

11 Conclusion

In this paper, we have proposed a smartphone user authentication scheme, called ShakeIn, based on customised one-hand shakes. ShakeIn is resilient to shoulder-surfing and biometrics hacking attacks as it adopts both physiological and behavioural characteristics to profile users. Furthermore, ShakeIn is handy as it allows customised shakes and single-hand operations. ShakeIn is quite reliable and can work well with different modes of transport. As ShakeIn needs only off-the-shelf devices, it is easy to gain a wide deployment. Nevertheless, ShakeIn also has several limitations. For example, if a user forgets how he/she shakes during the training phase, it is very likely for ShakeIn to refuse this user for unlocking. We suggest that a user chooses the most comfortable shaking styles as his/her “passwords”. Another limitation of ShakeIn is that currently it can work with two common people postures, i.e., sitting and standing. It would be more practical if more postures are supported. In addition, extending ShakeIn to other mobile devices bigger than smartphones in size such as tablets is also challenging. Moreover, we would also investigate to use more advanced classifiers such as Structural Minimax Probability Machine [29] in the future.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (Grants No. 61472255, 61420106010, 61300199, 61672151, and 61472068) and the Project funded by the China Postdoctoral Science Foundation (Grant No. 2014M550466).

References

[1] European union agency for network and information security, “Top Ten smartphone risks,” (2011). [Online]. Available: https://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-applications/smartphone-security-1/top-ten-risks

[2] F. Tari, A. Ozok, and S. H. Holden, “A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords,” in *Proc. 2nd ACM Symp. Usable Privacy Secur.*, 2006, pp. 56–66.

[3] F. Schaub, R. Deyhle, and M. Weber, “Password entry usability and shoulder surfing susceptibility on different smartphone platforms,” in *Proc. 11th ACM Int. Conf. Mobile Ubiquitous Multimedia*, 2012, Art. no. 13.

[4] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, “Smudge attacks on smartphone touch screens,” *Proc. 4th USENIX Conf. Offensive Technol.*, 2010, vol. 10, pp. 1–7.

[5] The Apple Inc., “About touch ID security on iPhone and iPad,” (2015). [Online]. Available: https://support.apple.com/en-us/HT204587

[6] 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. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 39–50.

[7] J. R. Kwapisz, G. M. Weiss, and S. Moore, “Cell Phone-based Biometric Identification,” in *Proc. 4th IEEE Int. Conf. Biometrics: Theory Appl. Syst.*, 2010, pp. 1–7.

[8] D. Gafurov, K. Helkala, and T. Søndrol, “Biometric gait authentication using accelerometer sensor,” *J. Comput.*, vol. 1, no. 7, pp. 51–59, 2006.

[9] C. Yuan, X. Sun, and R. Lv, “Fingerprint liveness detection based on multi-scale LPQ and PCA,” *China Commun.*, vol. 13, no. 7, pp. 60–65, 2016.

[10] F. Monrose, M. K. Reiter, and S. Wetzel, “Password hardening based on keystroke dynamics,” *Int. J. Inf. Secur.*, vol. 1, no. 2, pp. 69–83, 2002.

[11] S. Zahid, M. Shahzad, S. A. Khayam, and M. Farooq, “Keystroke-based user identification on smart phones,” in *Proc. 12th Int. Symp. Recent Advances Intrusion Detection*, 2009, pp. 224–243.

[12] 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. SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 987–996.

[13] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, “Biometric-rich gestures: A novel approach to authentication on multi-touch devices,” in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 977–986.

[14] Y. Guo, L. Yang, X. Ding, J. Han, and Y. Liu, “OpenSesame: Unlocking smart phone through handshaking biometrics,” in *Proc. IEEE INFOCOM*, 2013, pp. 365–369.

[15] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, “User evaluation of lightweight user authentication with a single tri-axis accelerometer,” in *Proc. 11th ACM Int. Conf. Human-Comput. Interaction Mobile Devices Services*, 2009, Art. no. 15.

[16] R. Mayrhofer and H. Gellersen, “Shake well before use: Intuitive and secure pairing of mobile devices,” *IEEE Trans. Mobile Comput.*, vol. 8, no. 6, pp. 792–806, Jun. 2009.

[17] R. Mayrhofer and H. Glelersen, “Shake well before use: Authentication based on accelerometer data,” in *Proc. 5th Int. Conf. Pervasive Comput.*, 2007, pp. 144–161.

[18] D. Bichler, G. Stromberg, M. Huemer, and M. Löw, “Key generation based on acceleration data of shaking processes,” in *Proc. 9th Int. Conf. Ubiquitous Comput.*, 2007, pp. 304–317.

[19] A. Studer, T. Passaro, and L. Bauer, “Don’t bump, shake on it: The exploitation of a popular accelerometer-based smart phone exchange and its secure replacement,” in *Proc. ACM 27th Annu. Comput. Secur. Appl. Conf.*, 2011, pp. 333–342.

[20] H. Han, et al., “SenSpeed: Sensing driving conditions to estimate vehicle speed in urban environments,” in *Proc. IEEE INFOCOM*, 2014, pp. 727–735.

[21] P. Zhou, M. Li, and G. Shen, “Use it free: Instantly knowing your phone attitude,” in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 605–616.

[22] B. Gu and V. S. Sheng, “A robust regularization path algorithm for v-support vector classification,” *IEEE Trans. Neural Netw. Learn. Syst.*, 2016.

[23] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, “Incremental support vector learning for ordinal regression,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1403–1416, Jul. 2015.

[24] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[25] S. S. Keerthi and C.-J. Lin, “Asymptotic behaviors of support vector machines with Gaussian kernel,” *Neural Comput.*, vol. 15, no. 7, pp. 1667–1689, 2003.

[26] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[27] M. Pielot, K. Church, and R. de Oliveira, “An in-situ study of mobile phone notifications,” in *Proc. 16th ACM Int. Conf. Human-Comput. Interaction Mobile Devices Services*, 2014, pp. 233–242.

[28] O. Wiese and V. Roth, “Pitfalls of shoulder surfing studies,” in *Proc. Workshop Usable Secur.*, 2015, pp. 1–6.

[29] B. Gu, X. Sun, and V. S. Sheng, “Structural minimax probability machine,” *IEEE Trans. Neural Netw. Learn. Syst.*, 2016.

Hongzi Zhu received the BS and MS degrees from Jilin University, in 2001 and 2004, respectively, and the PhD degree in computer science from Shanghai Jiao Tong University, in 2009. He was a post-doctoral fellow in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, and the Department of Electrical and Computer Engineering, University of Waterloo, in 2009 and 2010, respectively. He is now an associate professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include vehicular networks, mobile computing, and smart computing. He is a member of the IEEE Computer Society, IEEE, and the Communication Society.

Jingmei Hu received the BS degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, in 2016. She is currently working toward the PhD degree in the Department of Computer Science, Harvard University. Her research interests include operating system, mobile computing, and smartphone applications.

Shan Chang received the BS degree in computer science and technology from Xián Jiaotong University, in 2004 and the PhD degree in computer software and theory from Xián Jiaotong University, in 2013. From 2009 to 2010, she was a visiting scholar in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. She was also a visiting scholar with BBCR Research Lab, Electrical and Computer Engineering Department, University of Waterloo, from 2010 to 2011. Since 2013, she has been an assistant professor in the Department of Computer Science and Technology, Donghua University. Her research interests include security and privacy in mobile networks and wireless sensor networks. She is a member of the IEEE, the IEEE Computer Society, and the Communication Society.

Li Lu received the BS and MS degrees from Zhejiang University, in 2000 and 2003, respectively, and the PhD degree in information security from the Chinese Academy of Sciences, in 2007. He was a post-doctoral fellow in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, from 2008 to 2010. He is an associate professor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include applied cryptography, network security, pervasive computing, and sensor networks. He is a member of the IEEE Computer Society.