SocialAuth: Designing Touch Behavioral Smartphone User Authentication Based on Social Networking Applications

Weizhi Meng[1](✉), Wenjuan Li[1,2], Lijun Jiang[3], and Jianying Zhou[4]

[1] Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Lyngby, Denmark
weme@dtu.dk

[2] Department of Computer Science, City University of Hong Kong,
Kowloon, Hong Kong

[3] CyberTree Research Institute, Hong Kong, Hong Kong

[4] Singapore University of Technology and Design, Singapore, Singapore
jianying_zhou@sutd.edu.sg

Abstract. Modern smartphones expressed an exponential growth and have become a personal assistant in people's daily lives, i.e., keeping connected with peers. Users are willing to store their personal data even sensitive information on the phones, making these devices an attractive target for cyber-criminals. Due to the limitations of traditional authentication methods like Personal Identification Number (PIN), research has been moved to the design of touch behavioral authentication on smartphones. However, how to design a robust behavioral authentication in a long-term period remains a challenge due to behavioral inconsistency. In this work, we advocate that touch gestures could become more consistent when users interact with specific applications. In this work, we focus on social networking applications and design a touch behavioral authentication scheme called *SocialAuth*. In the evaluation, we conduct a user study with 50 participants and demonstrate that touch behavioral deviation under our scheme could be significantly decreased and kept relatively stable even after a long-term period, i.e., a single SVM classifier could achieve an average error rate of about 3.1% and 3.7% before and after two weeks, respectively.

Keywords: Behavioral user authentication · Touch gestures · Usable security · Smartphone security · Social networking · Machine learning

1 Introduction

Due to the capabilities and convenience, smartphones have been widely adopted by individuals. International Data Corporation (IDC) reported that up to 344.3 million smartphones have been shipped around the world in the first quarter of

© IFIP International Federation for Information Processing 2019
Published by Springer Nature Switzerland AG 2019
G. Dhillon et al. (Eds.): SEC 2019, IFIP AICT 562, pp. 180–193, 2019.
https://doi.org/10.1007/978-3-030-22312-0_13

2017, which achieved a growth rate of around 3.4% over the last year [4]. These devices have become a personal assistant, i.e., working as a social connection and work facilitator. A survey showed that nearly 40% of respondents play with their phones for three hours or more each day [1]. As modern smartphones can work like a mini-computer, users are willing to store personal data and complete sensitive tasks on the phones [7], such as personal photos, credit card information, transactions, etc. For example, 62% of phone users in Denmark were using their phone for viewing bank account and online payment [3].

As compared with PCs or laptops, smartphones are becoming a more private device (i.e., few people would like to share their phones) [8]. For profit purposes, cyber-criminals are always trying to exploit the stored data on smartphones. As long as having a victim's phone, cyber-criminals can launch various attacks, i.e., they can steal the identity of phone users and conduct impersonation attacks to threaten the whole networks, especially online social networks. As a result, designing appropriate user authentication mechanisms becomes very important to protect phones from unauthorized access.

Most smartphones adopt traditional password-based authentication mechanisms like PINs. However, this kind of authentication is known to be insecure, i.e., passwords are easily to be stolen via "shoulder surfing" [17], smudge attacks [2] and phone charging attacks (e.g., JFC attack [12]). To address this problem, research has been focused on behavioral authentication, which uses measurements from human actions to re-authenticate a user. Behavioral authentication is believed to complement the existing authentication mechanisms. Generally, behavioral authentication needs to build a normal profile at first and then detect an anomaly by identifying any great deviations between the current profile and the pre-defined normal profile. For instance, Frank *et al.* [6] proposed a behavioral authentication scheme with 30 features, which achieved a median equal error rate of nearly 4% using an SVM classifier.

Contributions. Up to now, there are many touch behavioral authentication schemes available in the literature, but how to design a behavioral authentication scheme for a long-term period still remains a challenge. Previous work ever showed that users' touch behavioral would become more stable after more trials [11]. Motivated by this observation, we advocate that the deviation of users' touch behavior would be reduced when they played some specific tasks. In this work, we focus on social networking applications due to their frequent usage by phone users [1], and design a touch gesture-based authentication scheme called *SocialAuth*. Our contributions in this work can be summarized as below.

- We revise and design a touch gesture-based authentication scheme with 22 features to authenticate a phone user, when they are playing a social networking application. As compared with some conventional tasks, i.e., inputting a PIN code, social networking applications allow users to perform more diverse touch gestures, like touch movement and multi-touch.
- To investigate the scheme performance, we performed a user study with a total of 50 Android phone users, who were required to use the phones in the same way as they would do in their daily lives. We mainly consider two

situations for data analysis. For the first situation, our scheme analyzes all touch gestures during the phone usage, while for the second situation, our scheme only considers the touch gestures when the users were playing with social networking applications.

- Experimental results with five popular classifiers demonstrated that the deviation of users' touch actions could be reduced when phone users were interacting with social networking applications, where an SVM could achieve a better average error rate of approximately 3.1% than other classifiers. Our study also verified the authentication performance after two weeks (as long-term period), and it is found that the SVM classifier could still reach an average error rate of nearly 3.7%.

Road Map. The reminder of this paper is organized as follows. Section 2 introduces related studies on touch behavioral authentication on mobile devices. We describe the authentication scheme, touch features, data collection and session identification in Sect. 3. In Sect. 4, we present a user study with 50 participants and analyze the scheme performance like authentication accuracy and long-term performance. We conclude our work in Sect. 5.

2 Related Work

Thanks to the rapid development of smartphones, touchscreens are becoming quite common and popular. Touch dynamics has thus received more attention worldwide. Feng *et al.* [5] designed a touchscreen-based authentication system called *FAST*, in which users utilize a digital sensor glove for authentication. Their approach could achieve a false acceptance rate (FAR) of 4.66% and a false rejection rate (FRR) of 0.13% using a random forest classifier. Meanwhile, Meng *et al.* [9] developed a behavioral authentication scheme with 21 features and performed a study with 20 participants. An average error rate of nearly 3% was reported by means of a PSO-RBFN classifier. Then, Frank *et al.* [6] developed *Touchalytics*, a touch behavioral authentication scheme with a total of 30 touch features. In the study, their system showed a median equal error rate of nearly 4%. Based on the observations obtained in their study, they claimed that *Touchalytics* could only be deployed as an optional rather than a stand-alone authentication mechanism. Later, Sae-Bae *et al.* [14] focused on multi-touch behavior and proposed to authenticate a user based on up to 22 multi-touch gestures, which could be extracted from both hand and finger actions.

Recent studies started combining behavioral authentication with other biometrics. For instance, Smith-Creasey and Rajarajan [15] described an authentication scheme by combining face and touch gestures based on a dataset with 50 users, and reported an equal error rate of 3.77% with a stacked classifier. Shahzad *et al.* [16] proposed an authentication scheme based on users' particular behavior when they perform a touch gesture and a signature. Nguyen *et al.* [13] proposed an authentication scheme called *DRAW-A-PIN*, which required users to draw a PIN on touchscreen instead of typing. Their system particularly employed a Content Analyzer and a Drawing Behaviour Analyzer to identify imposters. Meng *et al.* [11] proposed *TMGuard*, a touch movement-based authentication scheme with a combination of Android unlock patterns. Their study with 75 participants demonstrated that the security of Android unlock patterns can be enhanced without degrading its usability, and that users' touch behavior can become relatively stable after more trials.

3 Touch Gesture-Based User Authentication

3.1 Authentication Architecture

To secure a smartphone from unauthorized access, an ideal touch behavioral authentication scheme has to continuously monitor the behaviors and make an alert (or lock the phone) when any anomalies are detected. The high-level architecture of touch gesture-based authentication system is presented in Fig. 1.

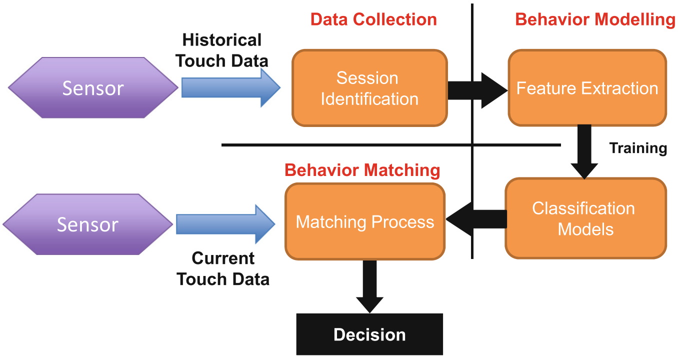

Fig. 1. The architecture of touch gesture-based authentication system.

A behavioral authentication system often contains three major phases: data collection, behavior modelling and behavior matching. The purpose of the first phase is to gather behavioral data from screen sensors and store them based on the particular session identification method. The second phase then refines the raw data and extracts features to build a normal behavioral profile for legitimate users. Various machine learning algorithms can be applied here. These two phases can help prepare the system for detecting behavioral anomalies. The last phase takes the current behavioral data from sensors and makes a decision by conducting a comparison with the pre-defined normal profile.

3.2 Touch Gesture Types and Features

Modern smartphones can provide a wide range of touch gestures, such as tap, swipe left or right, swipe up and down, and so on. Generally, these gestures on touchscreen can be categorized into the following types:

- **Single-Touch (ST):** this touch event starts with a touch-press down, and ends with a touch-press up without any touch movement in-between, like single-finger tap.
- **Touch-Movement (TM):** this touch event starts with a touch-press down, followed by a touch movement, and ends by a touch-press up, like swipe up and down.
- **Multi-Touch (MT):** this touch input starts with two or more simultaneous and distinct touch-press down events at different coordinates of a touchscreen, either with or without any touch movement before a touch press up event, like zoom, pinch and rotate.

To facilitate the comparison, in this work, we adopt and revise a touch behavioral authentication scheme on smartphones with up to 22 features, based on the work by Meng *et al.* [9]. These features can also be extracted when users interact with social networking applications, including *average touch movement speed per direction* (eight directions), *the fraction of touch movements per direction* (eight directions), *average single-touch time*, *average multi-touch time*, *the fraction of touch movements per session*, *the fraction of single-touch events per session*, and *the fraction of multi-touch events per session*. We further add one extra touch feature, namely touch pressure into the scheme, as many studies have proven its effectiveness [6,14].

Average Touch Movement Speed per Direction. Figure 2 shows how to define each direction; thus, a touch movement can be divided into different features. If we assume there are two points ($x1$, $y1$) and ($x2$, $y2$) in a touch movement' trajectory with relevant system time $S1$ and $S2$ (suppose $S1 < S2$). Then the features of *touch movement speed* (TMS) and *touch movement angle* between these two points can be calculated as follows:

$$TMS = \frac{\sqrt{(x2-x1)^2+(y2-y1)^2}}{S2-S1}$$

$$Touch\,movement\,angle\!: \quad \theta = \arctan\frac{y2-y1}{x2-x1}, \theta \in [0, 360^{\circ}]$$

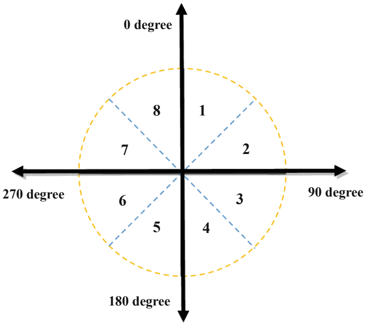

Fig. 2. Different directions for a touch action.

Let *ATMS* denote *average touch movement speed*. It is easy to calculate each feature based on the angles, i.e., *ATMS2* describes an average touch movement speed in direction 2, and *ATMS5* describes this feature in direction 5.

Fraction of Touch Movements per Direction. Intuitively, users may perform a touch movement more often in some certain directions. Therefore, the fraction of touch movements per direction varies among users and can be used for user authentication.

Average Single-Touch and Multi-touch Time. Single-touch and multi-touch are two types of touch gestures when users interact with their phones. Let *AST* denote *average single-touch time* and *MTT* denote *average multi-touch time*. The touch duration would be different between a single-touch and a multi-touch action.

Fraction of Touch Action Events. It is observed that users could have their own habit when interacting with the phone. For instance, some users would like to use single-touch more often than multi-touch, while some may prefer using multi-touch actions more, i.e., during web browsing. As a result, the fraction of touch events can be used for authenticating users. Three relevant features can be derived: *the fraction of touch movements per session* (denoted *FTM*), *the fraction of single-touch events per session* (denoted *FSTE*), and *the fraction of multi-touch events per session* (denoted *FMTE*).

Touch Pressure. With the development of modern smartphones, sensors are becoming more accurate and sensitive. In this case, average touch pressure (denoted *ATP*) has become one of the promising features for validating users. It is worth noting that all these features would be validated in Sect. 4.

3.3 Data Collection

Similar to [9,10], we employ an Android phone - Google/HTC *Nexus One* for data collection, which has a capacitive touchscreen of 480 × 800 px. This type of phone is selected because its OS can be replaced with a modified OS version. In this work, we updated the phone with a modified Android OS version 2.2 based on *CyanogenMod*[1]. The changes were mostly on its application framework layer by inserting system level command to record raw data from the touchscreen, such as the timing of touch inputs, the coordinates x and y, and the touch pressure and various gestures like single-touch, multi-touch and touch movement.[2] A separate logcat application was installed to help extract and record the captured data from the phone.

A sample of collected raw data from the phone is depicted in Table 1. Each record contains five major items: *input type*, *x-coordinate*, *y-coordinate*, *touch*

[1] http://www.cyanogenmod.com/.

[2] We inserted *Slog.v* command to two java source files (*InputDevice.java* and *KeyInputQueue.java*) regarding the *Application framework layer*, and then recompiled the whole source codes of *Froyo* operating system to generate our demanded experimental platform.

Table 1. A sample of raw data collected from touchscreen on the Android platform.

Input type	X-coordinate	Y-coordinate	Touch pressure	Time (ms)
Press down	478.5686	658.6726	0.090196080	1870785
Press move	473.5593	660.5503	0.101960786	1870807
Press move	471.2780	660.9001	0.101960786	1870814
Press move	468.7645	662.0188	0.125686300	1870852
Press move	470.5872	660.5211	0.125686300	1870898
Press move	472.8723	658.5432	0.125686300	1870910
Press up	470.6778	660.6223	0.125686300	1870933

pressure, and *system time (S-time)*. The system time is relevant to the last start-up of the phone and is managed by the phone itself, while the duration of each touch gesture can be computed by measuring the difference in system-time between touch press down and up. As a complementary item to the system time, the deployed logcat application can record regular timing information (e.g., 06–29 22:08:48.080) for later potential data verification. This kind of data collection does not need any special hardware on phone's side. It is worth noting that additional information can be collected by updating certain parts of the Android application framework.

3.4 Session Identification

To build a behavioral profile, session identification is an important factor that could affect authentication performance. The purpose of session identification is to help decide the length of a session. To ensure the collection of enough touch gestures, in this work, we adopted an event-based session identification includes a total of 120 touch gestures in each session [10]. A session ends if the number of touch gestures reached the pre-define value and then a new session starts. For implementation, session start and end can be easily determined by checking the raw data record.

4 User Study

4.1 Study Methodology

In the study, we recruited a total of 50 regular Android phone users (including 26 female and 24 male), who were aged from 18 to 61 years. Participants have a diverse background including students, senior citizens, researchers and business people. Table 2 details the background information of participants.

During the study, each participant was provided with an Android phone (a Google/HTC Nexus One) equipped with our modified OS version. The main purpose is to ensure that all data were collected under the same settings. Before the study, we described our research objective to all participants, introduced how

Table 2. Background of participants in the user study.

Occupation	Male	Female	Age	Male	Female
Students	14	16	18–30	14	16
Business people	2	3	31–40	5	5
Researchers	7	5	40–50	2	3
Senior citizen	1	2	Above	3	2

to perform data collection, and explained what kind of data would be collected, i.e., we emphasized that no personal data would be collected during the study. Further, we seek approval from each participant for gathering and analyzing the data, before they started the experiment.

More specifically, all participants were required to use the Android phones freely as the same way they would use the phones in their daily lives. By considering the limitations of a lab study, we allowed participants to do the actual data collection out of the lab, motivating them to have enough time to get familiar with the phone. They could decide when to start the collection process, according to our provided manual with detailed steps and explanations.

In this study, we mainly consider two situations for data analysis. For the first situation (*S1*), our scheme analyzes all recorded touch gestures when participants use the phones, whereas for the second situation (*S2*), our scheme only considers the touch gestures when participants play with any social networking applications. Each participant was required to complete 15 sessions for each situation (each session contains 120 touch gesture events) within 3 days. As a result, we could collect up to 1500 sessions of raw data, that is, 750 sessions for each situation. All participants could get a $20 gift card.

4.2 Machine Learning Classifiers and Metrics

As a study, we mainly employed five popular machine learning classifiers in the comparison: namely, Decision tree (J48), Naive Bayes, Radial Basis Function Network (RBFN), Back Propagation Neural Network (BPNN) and Support Vector Machine (SVM). To avoid any unexpected implementation bias, we extracted the above classifiers from WEKA [18] (using default settings), which is an open-source collection of machine learning algorithms.

Intuitively, a machine learning classifier is expected to achieve high classification accuracy. However, it is not easy in practice due to the behavioral dynamics. There is a need to balance false acceptance rate and false rejection rate in real-world applications. In practice, a desirable user authentication system is expected to achieve both a low FAR and FRR.

- False Acceptance Rate (FAR): indicates the probability that an impostor is categorized as a legitimate user.
- False Rejection Rate (FRR): indicates the probability that a legitimate user is classified as an intruder.

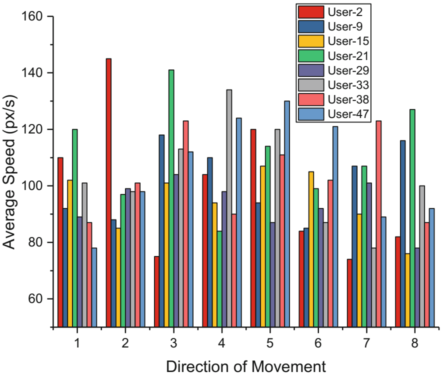

Fig. 3. The average touch movement speed per direction for eight different users.

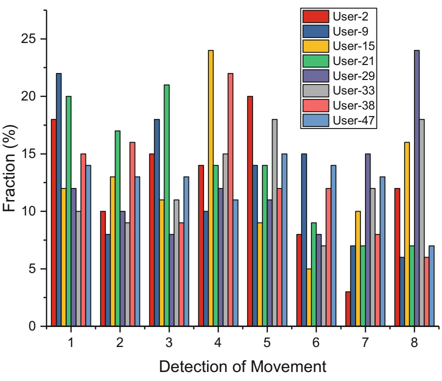

Fig. 4. The fraction of touch movements per direction for eight different users.

4.3 Result Analysis

As stated above, our authentication scheme is comprised of 22 touch features such as *ATMS1*, *ATMS2*, *ATMS3*, *ATMS4*, *ATMS5*, *ATMS6*, *ATMS7*, *ATMS8*, *FTM1*, *FTM2*, *FTM3*, *FTM4*, *FTM5*, *FTM6*, *FTM7*, *FTM8*, *AST*, *MTT*, *FTM*, *FSTE*, *FMTE* and *ATP*. In this part, we analyze the collected data regarding the effectiveness of features, touch behavioral deviation between two groups, authentication accuracy, and long-term performance after two weeks.

The Effectiveness of Features. Based on the collected 1500 sessions of touch gesture events, we calculate the touch features for each participant and randomly present 1/3 participants (about *eight* individuals) to validate the effectiveness of each feature in distinguishing users.

Figure 3 describes the average touch movement speed for different directions. It is found that the distributions varied with different users. For example, User-2 performed a higher speed in direction 1, 2 and 4; User-9 performed a higher speed in direction 3, 4 and 8; User-15 performed a higher speed in direction 1, 3, 5 and 6; User-21 achieved a higher speed in direction 1, 3, 5 and 8; and User-29 achieved a higher speed in direction 2, 3 and 7. The results prove that the use of *ATMS* per direction could help distinguish different phone users.

The fraction of touch movements for different directions is shown in Fig. 4. It is observed that User-2 conducted relatively more touch movements in direction 1, 3 and 5; User-9 performed more touch movements in direction 1, 3, 5, and 6; User-15 achieved a higher rate in direction 1, 2, 4, and 8; User-21 performed more touch movements in direction 1, 2 and 3; and User-29 had a higher rate in direction 1, 4, 7 and 8. The results validate that *FTM* in different directions can be used to characterize a user's touch behavior.

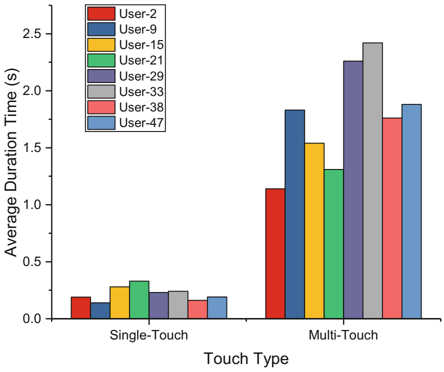

Fig. 5. The average duration time regarding single-touch and multi-touch for eight different users.

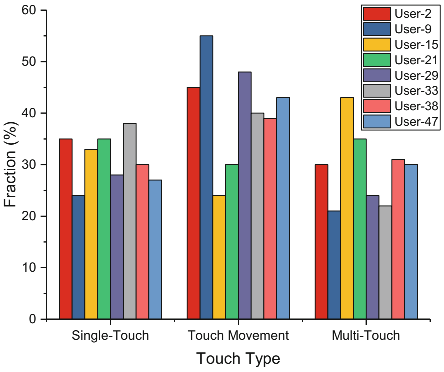

Fig. 6. The fraction of single-touch, touch movement and multi-touch for eight different users.

Figure 5 presents the average duration time regarding single-touch and multi-touch action. It is visible that time consumption could vary with different users. For single-touch action, User-21 consumed more time than the others while User-9 could finish the gesture with the minimum time among these users. For multi-touch action, User-33 and User-2 required the longest and the shortest time to finish the action. Based on our data, there is no direct relationship identified between single-touch and multi-touch. In this case, these features can be used to distinguish different users.

Figure 6 describes the fraction of single-touch, touch movement and multi-touch for eight users. It is found that User-2, User-21 and User-33 performed more single-touch actions than others. For touch movement, User-9 achieved a much higher rate than others, whereas User-15 achieved a higher rate than others regarding multi-touch. These results prove that these features can be used to model phone users' touch habits. It is similar to the feature of average touch pressure (ATP), it is found that User-9 achieved the biggest touch pressure of 2.012, while User-33 had the smallest touch pressure of 0.8892. The values of ATP for other users mainly ranged from 1 to 2.

Overall, our data analysis validates that our adopted 22 features could be effective in distinguishing phone users. This observation is in-line with the results in many previous studies like [6,9].

Touch Behavioral Deviation. Under *S1*, we considered all touch behavioral events when participants used the phone, while under *S2*, we only considered the touch gestures when they were using social networking applications. A total of four social networking applications were selected in the study: WeChat, Facebook, Twitter and Instagram. Our major purpose is to investigate the touch behavioral deviation between the two situations. Intuitively, a smaller deviation is desirable, indicating that users' touch actions are more stable. Figure 7

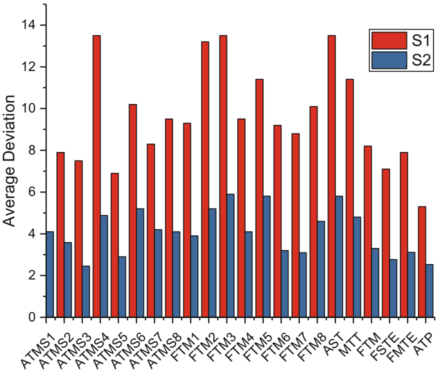

Fig. 7. The average behavioral deviation regarding all features under two situations.

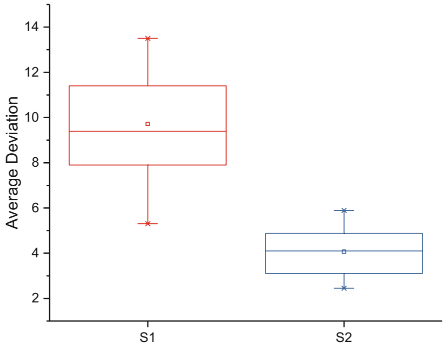

Fig. 8. The distribution of average deviation under two situations.

depicts the average behavioral deviation regarding all features and Fig. 8 shows the distribution of behavioral deviation under two situations.

It is visible that the average deviation for each touch feature under *S2* is much smaller than that under *S1*. For example, participants under *S1* made a deviation above 10 for ATMS1, ATMS4, ATMS6, FTM2, FTM3, FTM5, FTM8, AST and MTT, while the corresponding deviation was only ranged from 4.1 to 5.2 under *S2*. Figure 8 indicates that the deviations made under *S2* are mostly half or less than those made under *S1*. Intuitively, a higher deviation means that participants' touch gestures are more unstable, which may increase the difficulty of behavioral modelling. In contrast, a smaller deviation makes it easier to build a robust touch behavioral authentication scheme.

Further, we informally interviewed all the participants about their habits of phone usage. Based on their feedback, most participants reflected that their touch behavior would be quite dynamic when they freely used the phone without a task, whereas their touch actions would become focused when they were using a particular application, like social networking application. The feedback validated the observation that users' touch actions could become relatively stable under certain scenarios.

Authentication Accuracy. To investigate the authentication performance, we applied 18 sessions (up to 60% of the total sessions) as training data to help each classifier build a touch behavioral profile for each participant. Then we used the remaining sessions for testing. The test was run in 10-fold mode provided by the WEKA platform. The false acceptance rate (*FAR*), false rejection rate (*FRR*), and average error rate (*AER*) are presented in Table 3.

It is found that under *S1*, the single classifier of SVM could reach a better error rate than other classifiers, i.e., SVM achieved an AER of 6.02% while the others could only reach a rate of nearly 10%. Under *S2*, it is visible that

Table 3. Authentication performance for different classifiers under two situations.

	J48	NBayes	RBFN	BPNN	SVM
S1					
FAR (%)	22.55	18.66	9.72	9.12	5.22
FRR (%)	23.78	20.73	10.45	10.34	6.82
AER (%)	23.17	19.70	10.09	9.73	6.02
S2					
FAR (%)	15.13	11.56	6.88	6.42	2.89
FRR (%)	16.55	13.23	7.11	7.88	3.24
AER (%)	15.84	12.40	7.00	7.15	3.07

the performance was much better than that under *S1*. For example, SVM still achieved the best performance among single classifiers, but could offer an AER of 3.07% under *S2* vs. 6.02% under *S1*. For other classifiers like J48 and NBayes, their AER could be reduced by around 7% under *S2*.

Overall, these results demonstrate that with a smaller deviation, it is easier for a classifier to model phone users' touch behavior and to provide desirable authentication accuracy. In addition, users' touch behavior can become relatively stable under our scheme of *SocialAuth*, when they play with certain phone applications like a social networking application, as compared to the situation by considering all touches during the phone usage.

Long-Term Authentication. In the study, up to 16 participants (seven males) chosen to attend our task on long-term authentication, in which they could keep using our provided phone and returned to our lab after two weeks. They then required to complete 5 sessions for each *S1* and *S2* within two days. After the experiment, they could get a $30 gift card.

Our goal is to investigate the behavioral deviation after two weeks. Similarly, Figs. 9 and Fig. 10 shows the average behavioral deviation regarding all features and the distribution of behavioral deviation after two weeks, respectively. Encouragingly, it is found that after two weeks, the behavioral deviation under *S2* is much smaller than those under *S1*, i.e., some features' deviations are smaller than 2. In other words, users' touch gestures were much more stable under *S2* than those under *S1*.

For authentication accuracy, we applied the same five classifiers on the new sessions without re-training. That is, we used the already built behavioral model (before two weeks) for each classifier. It is found that SVM still could achieve a smaller AER under two situations, but the rate is much different, i.e., it reached a rate of 3.68% and 9.82% under *S2* and *S1*, respectively. The results validated that users' touch behavior could become relatively stable when they play with social networking applications, making it easier to build a robust authentication scheme for a long-term period.

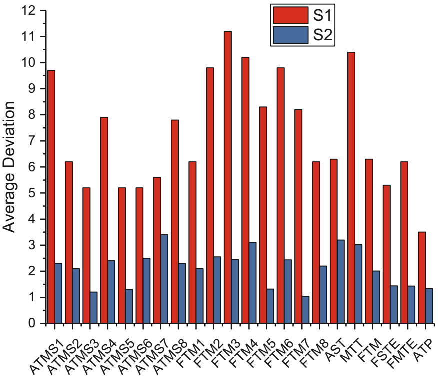

Fig. 9. The average behavioral deviation regarding all features after two weeks.

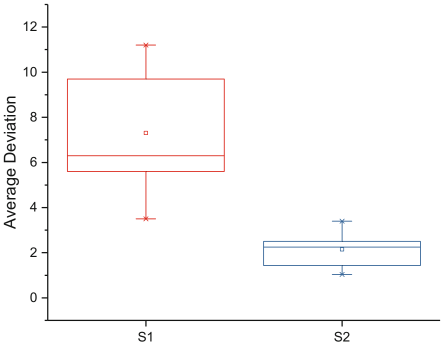

Fig. 10. The distribution of average deviation after two weeks.

5 Conclusion

How to design a robust scheme for a long-term period remains a challenge due to the inconsistent behavior. In this work, we advocate that users' touch behavior would become relatively stable when they interact with particular applications, and design a touch behavioral authentication scheme called *SocialAuth*. In the evaluation, we conducted a user study with 50 common Android phone users and considered two major situations for data analysis. We consider all recorded touch behavioral events under the first situation, whereas only consider the touch gestures when they use social networking applications under the second situation. It is found that an SVM classifier could reach an average error rate of 3.07% and 3.68% (before and after two weeks) under the second situation, versus a rate of 6.02% and 9.82% (before and after two weeks) under the first situation. The results demonstrated that with our scheme, users could achieve a much smaller behavioral deviation (more stable behavior) even after two weeks.

Acknowledgments. We would like to thank all anonymous reviewers for their helpful comments, and Jianying Zhou was supported by SUTD start-up research grant SRG-ISTD-2017-124.

References

1. Andrews, K.: Smartphone survey: the fascinating differences in the way we use our phones, 13 October 2017. http://www.abc.net.au/news/science/2017-10-13/smartphone-survey-results-show-fascinating-differences-in-usage/9042184
2. Aviv, A.J., Gibson, K., Mossop, E., Blaze, M., Smith, J.M.: Smudge attacks on smartphone touch screens. In: Proceedings of the 4th USENIX Conference on Offensive Technologies (WOOT), pp. 1–10 (2010)

3. Global Mobile Consumer Survey 2017 - Luxembourg, Accessed 12 Dec 2017. www.deloitte.com/lu/mobilesurvey
4. IDC: Smartphone vendor market share (2017). https://www.idc.com/promo/smartphone-market-share/vendor
5. Feng, T., et al.: Continuous mobile authentication using touchscreen gestures. In: Proceedings of the 2012 IEEE Conference on Technologies for Homeland Security (HST), pp. 451–456 (2012)
6. Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication. IEEE Trans. Inf. Forensics Secur. **8**(1), 136–148 (2013)
7. Karlson, A.K., Brush, A.B., Schechter, S.: Can i borrow your phone? Understanding concerns when sharing mobile phones. In: Proceedings of the 27th CHI, pp. 1647–1650 (2009)
8. Li, L., Zhao, X., Xue, G.: Unobservable re-authentication for smartphones. In: Proceedings of NDSS (2013)
9. Meng, Y., Wong, D.S., Schlegel, R., Kwok, L.: Touch gestures based biometric authentication scheme for touchscreen mobile phones. In: Kutyłowski, M., Yung, M. (eds.) Inscrypt 2012. LNCS, vol. 7763, pp. 331–350. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38519-3_21
10. Meng, Y., Wong, D.S., Kwok, L.-F.: Design of touch dynamics based user authentication with an adaptive mechanism on mobile phones. In: Proceedings of the ACM Symposium on Applied Computing (SAC), pp. 1680–1687 (2014)
11. Meng, W., Li, W., Wong, D.S., Zhou, J.: TMGuard: a touch movement-based security mechanism for screen unlock patterns on smartphones. In: Proceedings of the 14th International Conference on Applied Cryptography and Network Security (ACNS), pp. 629–647 (2016)
12. Meng, W., Fei, F., Li, W., Au, M.H.: Harvesting smartphone privacy through enhanced juice filming charging attacks. In: Nguyen, P., Zhou, J. (eds.) ISC 2017. LNCS, vol. 10599, pp. 291–308. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69659-1_16
13. Nguyen, T.V., Sae-Bae, N., Memon, N.: DRAW-A-PIN: authentication using finger-drawn PIN on touch devices. Comput. Secur. **66**, 115–128 (2017)
14. Sae-Bae, N., Memon, N., Isbister, K., Ahmed, K.: Multitouch gesture-based authentication. IEEE Trans. Inf. Forensics Secur. **9**(4), 568–582 (2014)
15. Smith-Creasey, M., Rajarajan, M.: A continuous user authentication scheme for mobile devices. In: Proceedings of the 14th Annual Conference on Privacy, Security and Trust (PST), pp. 104–113 (2016)
16. Shahzad, M., Liu, A.X., Samuel, A.: Behavior based human authentication on touch screen devices using gestures and signatures. IEEE Trans. Mob. Comput. **1**(10), 2726–2741 (2017)
17. Tari, F., Ozok, A.A., Holden, S.H.: A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In: Proceedings of the 2nd Symposium on Usable Privacy and Security (SOUPS), pp. 56–66 (2006). ACM, New York
18. Data Mining Software in Java: WEKA-waikato environment for knowledge analysis. http://www.cs.waikato.ac.nz/ml/weka/