The MOBIKEY Keystroke Dynamics Password Database: Benchmark Results

Margit Antal and Lehel Nemes

Abstract In this paper we study keystroke dynamics as an authentication mechanism for touchscreen based devices. A data collection application was designed and implemented for Android devices in order to collect several types of password. Besides easy and strong passwords we propose a new type of password—logical strong—which is a strong password, but easy to remember due to the logic behind the password's characters. Three main types of feature were used in the evaluation: time-based, touch-based and accelerometer-based. We propose a novel feature set—secondorder—which is independent of the length of the password. The preliminary results show that the lowest equal error rate (EER) is achieved by the logical strong password, followed by the strong password. The worst performance was achieved by the easy password; suggesting that the strong password is the best choice even in the case of keystroke dynamics based authentication systems.

Keywords Keystroke dynamics · Password difficulty · Mobile authentication · Performance evaluation · Sensors

1 Introduction

The pervasive presence of mobile devices equipped with many powerful sensors has led to new authentication mechanisms. One of them is user-authentication based on keystroke dynamics, an active research topic with remarkable results in the case of computers with hardware keyboards. Keystroke dynamics is a behavioural biometric which adds a second level security to alphanumerical passwords, by modelling the users' typing rhythms. Attempts to access the device by impostors, who have illegally

M. Antal (✉) · L. Nemes
Faculty of Technical and Human Sciences, Sapientia University,
Soseaua Sighisoarei 1C, 540485 Tirgu Mures/Corunca, Romania
e-mail: manyi@ms.sapientia.ro

L. Nemes
e-mail: nemes_lehel@yahoo.com

R. Silhavy et al. (eds.), *Software Engineering Perspectives and Application in Intelligent Systems*, Advances in Intelligent Systems and Computing 465,
DOI 10.1007/978-3-319-33622-0_4

obtained the user's password (through smudge-attack or shoulder surfing), can be detected based on the fact that they do not type the password in the same rhythm or that they handle the mobile device differently (device holding position, touchscreen usage).

In this paper we propose to investigate the influence of password difficulty on the authentication system's performance. The analysis is performed on our new dataset collected using mobile devices. This allows investigation not only of the effect of password difficulty, but also the influence of new features provided by the sensors of mobile devices.

Our work makes several contributions. One concerns the collected data, which contain the password typing patterns of three types of password i.e. easy, strong and logical strong. Data was collected using mobile devices therefore; besides time-based raw data we obtained additional data from sensors such as touchscreen and accelerometer. We have already made this data publicly available, hence it can be used by other researchers. Another contribution is the proposed secondorder feature set, independent of the length of the password and with equal error rates close to those obtained from the full feature set. The final contributions concern the evaluation results and the software used for the evaluation. Overall, we hope that our work will help focus attention on the opportunities provided by mobile device sensors in user identity verification.

The remainder of this paper is organised as follows. The next section (Sect. 2) presents related work with an emphasis on studies conducted on touchscreen-based mobile devices. Section 3 addresses research methods such as data collection, feature extraction and the different feature sets used in the evaluation. Section 4 offers evaluation results including two-class classifiers and anomaly detectors. The final section concludes our study and its findings.

2 Related Work

Keystroke dynamics is a well researched area. Several survey papers have been published to date [1, 4, 9, 17]. Most of this research has been carried out on computers or older mobile devices that utilise hardware keyboards. Less work has been carried out on touchscreen equipped mobile devices. However, the influence of key press pressure has been studied before the touchscreen smartphone era [8, 12, 14, 16]. In these studies special pressure-sensitive hardware keyboards were built. All these studies came to the conclusion that using key pressure as an addition feature increased the keystroke dynamic authentication system's performance.

In very recent years a few studies have been conducted on touchscreen-based mobile devices [2, 3, 6, 7, 10, 19, 21]. Except for Draffin et al.'s study [7], the other papers present results related to password-based authentication using keystroke dynamics. The most important aspects for the purpose of comparison are the datasets, the features, the methods and the results. Table 1 presents the characteristics of the datasets used in the aforementioned studies. It is important to note that

Table 1 Characteristics of keystroke datasets collected on touchscreen-based mobile devices

Study	# Users	Password	Raw data	Available	Best result(s) (%)
[19]	152	17-digit	Time	NO	FAR: 6.61 FRR: 8.03
[21]	80	4–8 digit	Time Touch Accelerometer Gyroscope	NO	EER: 3.75
[10]	20	7q56n5ll44 phrase	Time Space	NO	EER: 13.6
[3]	42	.tie5Roanl	Time Space Touch	YES	EER: 12.9
[6]	28	6–8 character	Time Space Touch	NO	EER: 13.74

not all studies saved the touch related raw data in the same way. Zheng et al. [21] and Buschek et al. [6] saved pressure and size (finger area) both at the moment of touch down and touch up. Conversely Antal et al. [2] saved this raw data only at the moment of key press. There are several differences between spatial raw data too. While Antal et al. saved the x, y coordinates only at the key press moment, Buschek et al. saved both the coordinates of the touch point at the moment of touch down and touch up. The differences between raw data imply different features for the analysed studies. Only Zheng et al. used raw data obtained from the accelerometer and the gyroscope sensors.

We have found only three papers which have studied the influence of password difficulty on the performance of keystroke dynamics system. Bartlow and Cukic [5] conducted the first study in this direction. Besides common short 8-lowercase letter passwords, such as `computer` and `swimming`, they used long 12-character length randomly generated passwords the typing of which required the usage of the Shift key. Example of such passwords include `+AL4lfav8TB=` and `UC8gkum5WH`. In almost every EER performance measurement they observed a notable increase (at least 2 %) from short to long password, indicating that the usage of the shift key in a password plays a significant role. In feature ranking the shift key related features proved to be very discriminating.

Meng et al. [18] questioned the use of keystroke dynamics as biometrics. They built a training interface which allows intruders to train themselves in imitating another person's password typing rhythm. For this study they used two 8-character length passwords, an easy and a difficult one. They concluded that passwords that are easier to type are also easier to imitate.

Mondal et al. [15] introduced complexity measurement related to the typing of a password after which several performance measurements were conducted. In contrast to the previous two studies, they concluded that easier passwords are better choice for keystroke dynamics biometrics.

3 Methods

3.1 Data Collection

An Android application was designed and implemented with the aim of collecting typing data for different passwords. Users had to type in three different fixed passwords. The following passwords were used: easy—`kicsikutyatarka`; logical strong—`Kktsf2!2014`; strong—`.tie5Roanl`. The easy password contained only lowercase letters and was formed by the first three words of a Hungarian saying. Our proposal utilises the logical strong type and is based also on the same Hungarian saying, but in this case we took the first letters of the words and used `sf2!` for `sfsf` (two occurences of sf) followed by the year of data collection. The logic behind the logical strong password was explained to subjects before the data collection experiment. The strong password was used in the keystroke dataset collected by Killourhy [11].

54 volunteers took part in the experiment, 5 women and 49 male, with an average age of 20.61 years (range: 19–26). At the registration stage they stated their experience with touchscreen devices as follows: 2—inexperienced, 6—beginners, 17—intermediate and 29 advanced touchscreen users. Among them 4 users were left handed the others right handed. Data was collected in three sessions one week apart. In each session they typed at least 60 passwords, at least 20 passwords from each type. At the end of data collection each user had provided at least 60 samples from each type of password (easy: 3323 samples, strong: 3303, logical strong: 3308). The data was collected using 13 identical Nexus 7 tablets. Typos were not allowed, instead, the subjects had to retype the password. Each password had to be typed in the same way: the same keys had to be typed in the same order.

3.2 Feature Extraction

The application implemented a custom keyboard in order to store the time, touch and accelerometer related raw data during each user's typing. Raw data was saved at touch events initiated by the user for example, at the point of touch down and touch up. Touch down events were generated by the system when the user touched a key on the software keyboard, and touch up at the point of key release. Table 2 shows the raw data saved during the data collection process.

Table 2 The most important raw data saved during data collection

Data	Explanation
Key	The pressed key
Downtime	The timestamp at touch down event
Uptime	The timestamp at touch up event
Pressure	The pressure exerted on the screen at touch down event
Finger area	Touch area at touch down event
x, y	The x and y coordinate at touch down event
ax, ay, az	Acceleration measured along x, y, z axes

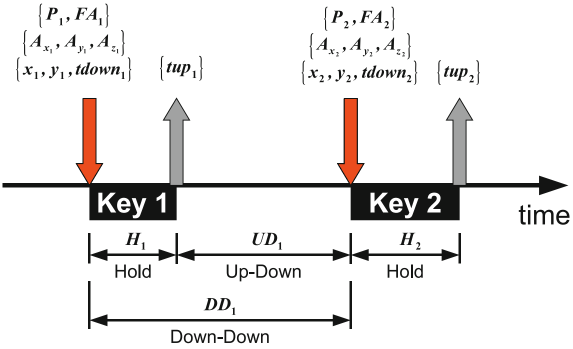

Fig. 1 Data collection. Raw data: *x*, *y*—coordinates; *tdown*, *tup*—timestamps; *Ax, Ay, Az*—directional accelerations; *P*—pressure; *FA*—finger area. Time-based features: *H*—hold time; *UD*—up-down time; *DD*—down-down time

Figure 1 shows the data saved at the moment of touch down and also the time-based features that can be extracted from these data such as hold time—the time between key press and release, down-down time—the time between consecutive key presses, and up-down time—the time between key release and next key press. The Nexus 7 tablet contains an embedded accelerometer with range $-2g$ and $+2g$ and measures the accelerations along three axes (the axes are device related). Its fastest sampling rate on sensor readings is about 50 Hz. During data collection these values were saved at the moment the user touched the screen. Using these directional accelerations we could characterise the device holding preferences of the users.

3.3 Feature Sets

Table 3 shows the full feature sets for each type of password. Because these feature sets contain features related to each key in a password, some feature types contain a different number of features for each password. Mean hold time (MHT) feature represents the average of key hold time values. The other mean values were computed similarly. The total distance feature (TD) was calculated as the sum of the distances (in pixels) between two consecutive buttons on the virtual keyboard. Total time (TT)

Table 3 Full feature sets for each type of password

Mnemonic	Feature type	Easy	Strong	Logical strong
HT	Hold time	15	13	13
DD	Down-down time	14	12	12
UD	Up-down time	14	12	12
P	Pressure	15	13	13
FA	Finger area	15	13	13
MHT	Mean hold time	1	1	1
MP	Mean pressure	1	1	1
MFA	Mean finger area	1	1	1
MAX	Mean X acceleration	1	1	1
MAY	Mean Y acceleration	1	1	1
MAZ	Mean Z acceleration	1	1	1
TD	Total distance	1	1	1
TT	Total time	1	1	1
V	Velocity	1	1	1
Total		82	72	72

represents the time needed to type in the password. Velocity (V) was computed as the quotient of the distance and the total time. Before evaluation data was normalized into the range [0, 1].

Besides the full feature sets presented in Table 3 some evaluations were performed on a so called—secondorder—feature set. This feature set contains 9 features: mean hold time, mean pressure, mean finger area, mean x acceleration, mean y acceleration, mean z acceleration, velocity, total time and total distance. The most important characteristic of this feature set is that the number of features is password-independent. All information related to this research is available at http://www.ms.sapientia.ro/~manyi/mobikey.html.

4 Evaluation and Results

Keystroke dynamics based authentication is a typical outlier detection problem. Given the keystroke data of a typed password the system has to decide whether the data belong to the genuine user. This problem can be formulated as a classification and as an anomaly detection problem. In the case of classification we typically employ a two-class classification algorithm, where the positive samples belong to the genuine user and negatives are selected from the others. Classifiers are more powerful since they yield information about the impostors (negative samples), whereas anomaly detectors can only check the deviation from the genuine user (positive samples). We should mention that in a real-world authentication system only the anomaly detection method is viable because of the lack of negative samples. However for comparison purposes, we present the evaluation of two-class classifiers too.

4.1 Two-Class Classification

In the case of two-class classification we call the data from the legitimate user positive samples and that from impostors we call negative samples. As our dataset contains data from several users and as each user typed the same password, one can easily select negative data for each user.

The general algorithm used for two-class classification measurements is depicted in Fig. 2. First we select positive and negative samples for a given user (*userData*). As negative samples we used two randomly selected samples from each other user. Then we repeat *nRuns* times the randomization of the data followed by n-fold cross-validation evaluation for the given user. The above two steps were repeated for each user.

Scores for positive and negative test samples were computed so as to form two sets, one for genuine users the other for impostors. Then a user-independent threshold was scanned through the two sets of scores and the False Negative (FN) and False Positive (FP) rates computed for each threshold. Plotted as error curves, these values show the system performance (see Fig. 3).

Besides Random Forests algorithm we chose to evaluate the k-nearest neighbours (kNN) and Bayes Net algorithms. All classification algorithms were used from the Weka Data Mining toolkit [20].

```
procedure MEASUREMENT($data, nFolds, nRuns$)
    for $user \leftarrow 1, numUsers$ do
        $userData \leftarrow selectPositiveAndNegativeSamples(data, user)$
        for $run \leftarrow 1, nRuns$ do
            $userData \leftarrow randomize(userData)$
            for $n \leftarrow 1, nFolds$ do
                $trainUserData \leftarrow trainCV(userData, n)$
                $testUserData \leftarrow testCV(userData, n)$
                train two-class classifier for trainUserData
                evaluate the trained classifiers using testUserData
            end for
        end for
    end for
end procedure
```

Fig. 2 Two-class classification measurement algorithm using n-fold cross-validation

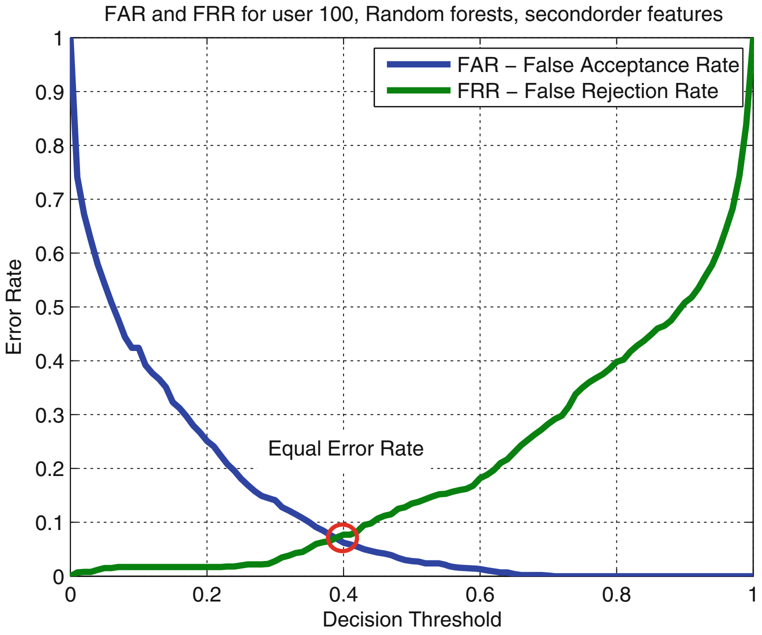

Fig. 3 EER computation for user 100 (Random forests classifier, secondorder features). EER for individual users were estimated as the intersection of FAR (False Acceptance Rate) and FRR (False Rejection Rate) curves

4.2 Anomaly Detection

In the case of anomaly detectors we used five detectors implemented in the R script provided by Killourhy and Maxion [11]. The detectors used were: Euclidean, Manhattan, Mahalanobis, Outlier count and Kmeans. This script works as follows: (i) it splits the data into three equal parts, each containing 20 samples from each user (in our case each part contained data from a single data-collection session) (ii) detectors are trained separately for each user using two-thirds of the data; evaluation was performed on the remaining third positive samples and two negative samples selected from each of the other users (20 positive + 53 * 2 negative); (iii) step (ii) is then repeated three times (threefold cross-validation), and the mean EER and its standard deviation computed.

4.3 Results

Results for classifiers and anomaly detectors are presented in Table 4. EER values were estimated for each user (see Fig. 3), then the mean and standard deviation were computed for each classifier or anomaly detector and each dataset.

Table 4 EER results for different methods and feature sets

Method	Features	Easy	Logical strong	Strong
Bayes net	Secondorder	0.074 (0.046)	0.058 (0.040)	0.067 (0.047)
kNN (k = 1)	Secondorder	0.056 (0.032)	0.048 (0.026)	0.054 (0.036)
Random forests (T = 100)	Secondorder	0.052 (0.029)	**0.045 (0.025)**	0.051 (0.032)
Bayes net	All	0.053 (0.039)	0.046 (0.037)	0.049 (0.038)
kNN (k = 1)	All	0.073 (0.036)	0.068 (0.033)	0.071 (0.043)
Random forests (T = 100)	All	0.032 (0.021)	**0.033 (0.025)**	0.033 (0.022)
Euclidean	Secondorder	0.208 (0.174)	0.149 (0.141)	0.181 (0.145)
Manhattan	Secondorder	0.202 (0.169)	0.144 (0.140)	0.169 (0.146)
Mahalanobis	Secondorder	0.191 (0.182)	0.154 (0.171)	0.159 (0.159)
Outlier count (th = 1.96)	Secondorder	0.208 (0.147)	0.164 (0.140)	0.178 (0.146)
Kmeans (k = 3)	Secondorder	0.177 (0.155)	**0.136 (0.132)**	0.143 (0.137)
Euclidean	All	0.238 (0.186)	0.183 (0.149)	0.195 (0.163)
Manhattan	All	0.203 (0.183)	0.154 (0.140)	0.167 (0.153)
Mahalanobis	All	0.256 (0.140)	0.193 (0.114)	0.210 (0.137)
Outlier count (th = 1.96)	All	0.160 (0.140)	**0.129 (0.126)**	0.143 (0.137)
Kmeans (k = 3)	All	0.173 (0.136)	0.128 (0.097)	0.131 (0.106)

The standard deviations are shown in parenthesis

We used 100 trees for the Random Forests classifier, $k = 1$ for the kNN classifier and the default Weka settings for the Bayes Net classifier. In the case of anomaly detectors the following settings were used: $k = 3$ clusters, at most 20 iterations for the kmeans detector; the $threshold = 1.96$ for the outlier count detector (used to count how many z-scores exceed a threshold) [11].

It can be seen that very low EER values were obtained by the classification algorithms, because these used the negative samples for building the user's model. However in real systems negative samples are not available (in the enrolment stage samples are collected only from the genuine user).

For the error curve we chose the DET error curve (Detection Error Tradeoff) [13], which is the most important error curve for biometric systems. Figure 4a, b show these error curves obtained for the Random Forests classifier (number of trees: 100) and Manhattan detector.

The best equal error rates were obtained by the Random Forests classifier, around 5 % for the secondorder feature set and around 3 % for the full feature set. We mention again that these classifiers use negative samples for building the user's typing model, which is not available in case of real systems. No significant differences were found in this evaluation between different types of password.

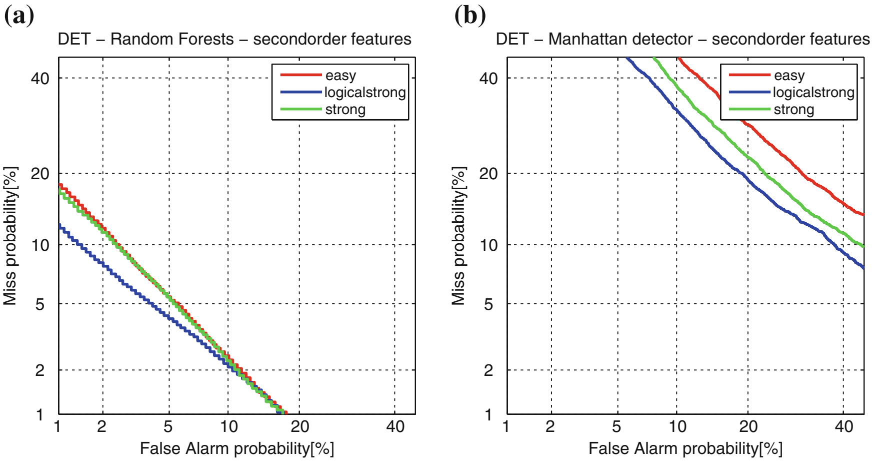

Fig. 4 DET curves—secondorder features. **a** Random Forests (T = 100). **b** Manhattan detector

In the case of anomaly detectors, where the user's model is based only on positive samples (the case of real systems), the equal error rates are always lower for logical strong and strong types of password.

5 Conclusions

Our objective in this work was to collect a dataset on mobile devices containing different types of password and to evaluate the influence of password difficulty on the performance of keystroke dynamics authentication. We provide both the datasets and evaluation methodology to the research community. The main contribution of this paper concerns the datasets, which not only contain three types of password, but contain raw data collected from mobile sensors too. Another contribution is the secondorder feature set which has the same number of features regardless of the password type. Measurements show the effectiveness of this novel feature set as very close to or sometimes better than the results obtained using the full feature set. Evaluations show that in the case of anomaly detectors the lowest equal error rates are obtained for the logical strong password, followed by the strong and the easy one. This is in concordance with the results obtained by Bartlow and Cukic [5] and Meng et al. [18].

Acknowledgments The research has been supported by the European Union and Hungary and co-financed by the European Social Fund through the project TAMOP–4.2.2.C–11/1/KONV–2012–0004–National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies. The authors would like to thank all volunteers who participated in the data collection experiment.

References

1. Ahmad, N., Szymkowiak, A., Campbell, P.A.: Keystroke dynamics in the pre-touchscreen era. Front. Human Neurosci. **7** (2013)
2. Antal, M., Szabó, L.: An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices. In: 2015 20th International Conference on Control Systems and Computer Science (CSCS), pp. 343–350 (2015)
3. Antal, M., Szabó, L., Laszló, I.: Keystroke dynamics on android platform. Procedia Technol. **19**, 820–826 (2015). In: 8th International Conference Interdisciplinarity in Engineering, INTER-ENG 2014, 9–10 Oct 2014, Tirgu Mures, Romania
4. Banerjee, S.P., Woodard, D.L.: Biometric authentication and identification using keystroke dynamics: a survey. J. Pattern Recogn. Res. **7**(1), 116–139 (2012)
5. Bartlow, N., Cukic, B.: Evaluating the reliability of credential hardening through keystroke dynamics. In: 17th International Symposium on Software Reliability Engineering, 2006. ISSRE'06, pp. 117–126 (2006)
6. Buschek, D., De Luca, A., Alt, F.: Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. CHI'15, pp. 1393–1402. ACM (2015)
7. Draffin, B., Zhu, J., Zhang, J.: Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction. In: Mobile Computing, Applications, and Services, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 130, pp. 184–201. Springer (2014)
8. Eltahir, W., Salami, M.J.E., Ismail, A., Lai, W.: Design and evaluation of a pressure-based typing biometric authentication system. EURASIP J. Inf. Secur. **2008**(1) (2008)
9. Giot, R., El-Abed, M., Rosenberger, C., et al.: Keystroke dynamics authentication. Biometrics (2011)
10. Kambourakis, G., Damopoulos, D., Papamartzivanos, D., Pavlidakis, E.: Introducing touchstroke: keystroke-based authentication system for smartphones. Secur. Commun. Netw. (2014)
11. Killourhy, K., Maxion, R.: Comparing anomaly-detection algorithms for keystroke dynamics. In: IEEE/IFIP International Conference on Dependable Systems Networks, 2009. DSN'09, pp. 125–134 (2009)
12. Loy, C., Lai, W.K., Lim, C.: Keystroke patterns classification using the ARTMAP-FD neural network. In: Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007, vol. 1, pp. 61–64 (2007)
13. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The DET curve in assessment of detection task performance. Technical report, DTIC Document (1997)
14. Martono, W., Ali, H., Salami, M.: Keystroke pressure-based typing biometrics authentication system using support vector machines. In: Computational Science and Its Applications ICCSA 2007, vol. 4706, pp. 85–93 (2007)
15. Mondal, S., Bours, P., Idrus, S.Z.S.: Complexity measurement of a password for keystroke dynamics: preliminary study. In: Proceedings of the 6th International Conference on Security of Information and Networks, pp. 301–305 (2013)
16. Nonaka, H., Kurihara, M.: Sensing pressure for authentication system using keystroke dynamics. In: International Computational Intelligence Society International Conference on Computational Intelligence. pp. 19–22 (2004)
17. Teh, P.S., Teoh, A.B.J., Yue, S.: A survey of keystroke dynamics biometrics. Sci. World J. **2013** (2013)
18. Tey, C.M., Gupta, P., Gao, D.: I can be you: Questioning the use of keystroke dynamics as biometrics. In: NDSS. The Internet Society (2013)
19. Trojahn, M., Arndt, F., Ortmeier, F.: Authentication with keystroke dynamics on touchscreen keypads-effect of different n-graph combinations. In: MOBILITY 2013, The Third International Conference on Mobile Services, Resources, and Users, pp. 114–119 (2013)

20. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2011)
21. Zheng, N., Bai, K., Huang, H., Wang, H.: You are how you touch: user verification on smartphones via tapping behaviors. In: 2014 IEEE 22nd International Conference on Network Protocols (ICNP), pp. 221–232 (2014)