Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 19 (2015) 820 – 826

Procedia Technology

8th International Conference Interdisciplinarity in Engineering, INTER-ENG 2014, 9-10 October 2014, Tirgu-Mures, Romania

Keystroke dynamics on Android platform

Margit Antal*, László Zsolt Szabó, Izabella László

Sapientia University, Faculty of Technical and Human Sciences, Soseaua Sighisoarei 1C, Tirgu Mures (Corunca) 540485, Romania

Abstract

Currently people store more and more sensitive data on their mobile devices. Therefore it is highly important to strengthen the existing authentication mechanisms. The analysis of typing patterns, formally known as keystroke dynamics is useful to enhance the security of password-based authentication. Moreover, touchscreen allows adding features ranging from pressure of the screen or finger area to the classical time-based features used for keystroke dynamics. In this paper we examine the effect of these additional touchscreen features to the identification and verification performance through our dataset of 42 users. Results show that these additional features enhance the accuracy of both processes.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of "Petru Maior" University of Tirgu Mures, Faculty of Engineering

Keywords: Security; User Authentication; Behavioral Biometric; Keystroke Dynamics; Touch Features

1. Introduction

At present more and more people store private and sensitive data on their smartphones. Consequently, the demand is growing for secure mobile authentication methods. Setting a password-based authentication is the most frequently used method to protect data from intruders. However, people tend to use passwords, which can be easily remembered, hence easy to crack. Therefore, additional mechanisms are needed to enhance the security of password based authentication. One such complementary method is to use the typing pattern of the user, known as keystroke dynamics.

Keystroke dynamics is an active research topic and has been researched mainly on desktop computers. There are very few studies conducted on mobile phones, even fewer on smartphones with touchscreen. The main research

* Corresponding author. Tel.: +40 265 250 620; fax: +40 265 206 211.
E-mail address: manyi@ms.sapientia.ro.

2212-0173 © 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of "Petru Maior" University of Tirgu Mures, Faculty of Engineering
doi:10.1016/j.protcy.2015.02.118

question of this is study is whether new features provided by touchscreens - such as pressure or finger area - can improve the accuracy of a keystroke based authentication system.

The next section briefly presents the research field of keystroke dynamics, reviewing the research studies conducted on devices with touchscreens. Then we present the research methodology, including data collection and its evaluation through identification and verification measurements. The final section presents some conclusions and future directions.

2. Keystroke Dynamics

Keystroke dynamics is a heavily researched field. One of the most important advantages is low implementation and deployment cost [11]. In contrast to other biometric methods, this method does not require any dedicated hardware device. As the capture of keystroke pattern is implemented using a backend software, it makes this method transparent and noninvasive for the user [11]. Keystroke dynamics can be used both for strengthening entry point based authentication and as a continuous authentication mechanism [2]. Compared to other methods, the main disadvantage of this type of biometrics is low accuracy [11].

Keystroke dynamics studies reported data acquisition using various input devices, ranging from normal to pressure sensitive keyboards [7]. The most commonly used time-based features are dwell time and flight time. Dwell time is the time interval between key press and key release (sometimes called hold time) whereas flight time is the time interval between releasing one key and pressing the next one. Sometimes three or more consecutive key time events are used as features (n-graph), but the majority of papers used digraph features (two consecutive keys). Most of the existing pattern recognition approaches were tested for keystroke recognition, including statistical and machine learning approaches. The simplest method is to construct a reference template for the respective user and compute the distance between the current typing pattern and the reference template in the authentication stage. This method is known as template matching and can be combined with different metrics, ranging from simple Euclidean metric to Mahalanobis metric. Neural networks and support vector machine (SVM) were the best [11].

Biometric systems can have two distinct functions: verification and identification. Verification is a binary decision problem, in which the system accepts or rejects the identity claimed by the user. Identification, also called recognition, is a classification problem: the system classifies the input pattern into one of the N known classes.

The quality of biometric systems is usually characterized by three kinds of errors: FAR, FRR and EER. False Acceptance Rate (FAR) is the rate at which a biometric system accepts a sample as one belonging to the claimed identity when the sample belongs to an impostor. False Rejection Rate (FRR) is the rate at which a biometric system incorrectly rejects a sample provided by the genuine user. EER is the rate at which FAR is equal to FRR.

The following is an overview of studies that used touchscreen based devices for data collection.

Saevanee and Bhattarakosol presented the first study [9] using keystroke dynamics combined with finger pressure. Through a dataset collected from 10 users they demonstrated that users can be identified with 99% accuracy by using only finger pressure information. However, data were collected using a notebook with touchpad acting as a touchscreen. Participants had to enter their 10 digits long cell phone numbers. Since each user has a different phone number, only FRR type error can be measured on the dataset. For FAR error measurement impostor data must be collected. The lack of impostor data can be considered the main limitation of this study.

Another study related to keystroke dynamics using touchscreen features is presented in the master thesis of Johansen [5]. The purpose of this study was to compare keystroke dynamics on personal computer to smartphones. A total of 42 persons took part in the experiment. Some of them completed both the personal computer and the smartphone experiment. The main finding of the study is that using only the timing features, the performance on smartphones is worse than on standard keyboard. However, using the extra smartphone features besides the timing features, the performance is significantly better than on a standard keyboard. The study proposed to answer the question how hard is to imitate someone’s typing rhythm. The results show that it is easier to imitate someone's typing on a standard keyboard than on a smartphone. The main limitation of this study is that the data collection process used a numerical password on a 12-key mobile phone keyboard.

The main goal of the study presented by Trojahn et al. in the paper [12] is to demonstrate that pressure and size of the finger as additional features reduce the error rate of a keystroke-based authentication system. The test required each of the 152 data providers to introduce a 17-digit passphrase. Each participant typed the password ten times in a

single session. The best FAR+FRR combination was obtained by using duration (hold time) combined with digraph and trigraph timing information. The error rates were further lowered by using touchscreen-based additional features. The main limitation of this study is similar to Johansen's; in addition, data collection was performed in a single session.

We have found only one study using mobile soft keyboard for user authentication [4]. Typing data was collected from 13 users during a period of 3 weeks. The developed soft keyboard collected key press information in all contexts requiring a soft keyboard. Key press length, drift, pressure, finger area and device orientation were used as features. User authentication results are reported using FAR and FRR errors. The data collection mechanism is not defined very clearly – explaining differences between the concepts of touch and key press would have proven valuable in understanding Draffin's method.

A recent study using pressure as feature for user authentication on mobile devices is presented by Sen and Muralidharan [10]. Similar to other studies this one is based on a 4-digit password. Besides the presented verification results using FAR and FRR type errors, EER is also reported based on a special impostor model.

Table 1 summarizes the results obtained by these recent studies. Unfortunately, the different types of errors reported by the studies make the comparison difficult.

Table 1. Error rates obtained by recent studies conducted on touchscreen (Feature notations are explained in Table 2)

Paper	#participants	#samples/user	Password	Features	Result(s)
[10]	10	100	4 digits	$H+P_{min}+P_{max}$	EER: 15.2%
[12]	152	10	17digits	H+P+FA+UD	FAR: 4.19%; FRR: 4.59%
[4]	13	NA	NA	H+ P+FA+ Keypress location+ drift+orientation	FAR: 14.0%; FRR: 2.2%
[9]	10	30	10 digits	P	EER: 1%

In conclusion, we can state that no study has been designed to measure the performance of keystroke dynamics on smartphones using a real-life password. Moreover, the effect of touchscreen features – pressure and finger area – has not been studied in such a realistic environment.

3. Methodology

3.1. Data Collection

An Android application having its own software keyboard was developed for data collection (see Fig.1. (b)-(d)). Users had to introduce some personal data, such as gender, birth date and their experience level regarding smartphone usage in the registration phase (see Fig.1. (a)). Because typing pattern can be influenced by several factors, data should be collected in several sessions. The majority of participants completed 2 sessions in a period of two weeks. In each session users had to enter the same password (**.tie5Roanl**) for 30 times. This is considered to be a strong password and was also used in the keystroke dynamics experiment designed by Killourhy [6]. A total number of 42 people took part in this study, 24 male and 18 female participants, aged 20-46 (with the average of 22.2 years). All the participants were students, except for one female teacher. We excluded from the collected data input patterns containing deletions and created a dataset for measurements containing 51 input patterns from each user. We decided to use the same password for each user so that each participant data may be used in the measurements both as an impostor and as a legitimate user.

For data collection two types of Android devices were used, a Nexus 7 tablet and a Mobil LG Optimus L7 II P710 device. All in all 37 tablet users and 5 mobile phone users supplied data.

Typing the chosen password on our software keyboard required pressing 14 keys: 8 letters, a digit, a period character, twice the Shift key in order to switch to/from capital letters and twice the numerical keyboard switch key. Touching the screen triggers a touch down event in the system that saves timestamp, pressure and finger area. When releasing the screen the timestamp is saved. The components of the feature vector are shown in Table 2.

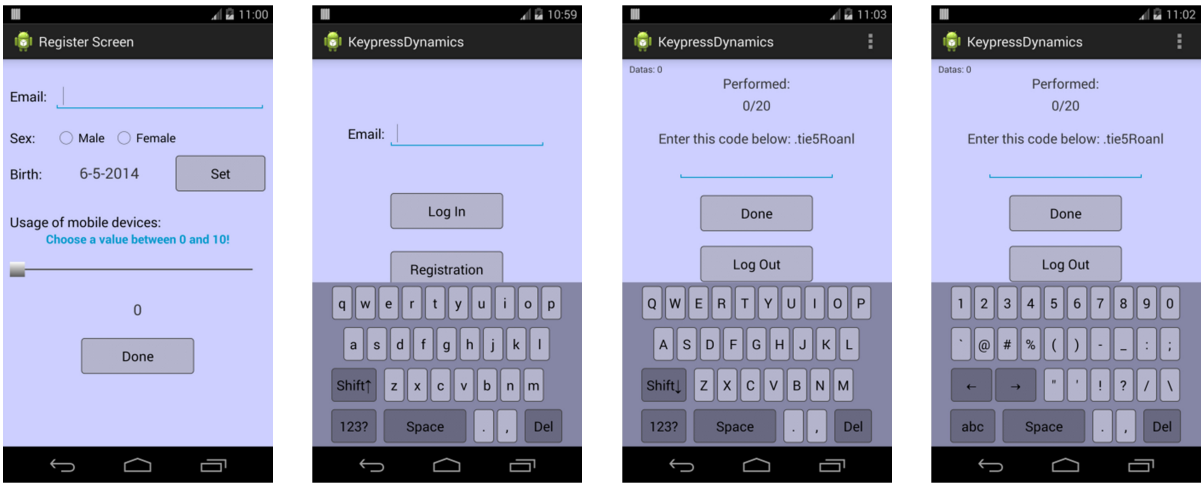

Fig. 1. (a) Registration screen; (b) Login screen; (c) Data collection – letter keyboard; (d) Data collection – numerical keyboard.

Table 2. The elements of a feature vector

Feature name	Explanation	Number of features
Key hold time (H)	Time between key press and release	14
Down-down time (DD)	Time between consecutive key presses	13
Up-down time (UD)	The time between key release and next key press	13
Key hold pressure (P)	Pressure at the moment of key press	14
Finger area (FA)	Finger area at the moment of key press	14
Average hold time (AH)	Average of key hold times	1
Average finger area(AFA)	Average of key finger areas	1
Average pressure (AP)	Average of key pressures	1
Total		**71**

3.2. Measurements

User identification measurements were performed using WEKA (version 3.6.11) [13], a popular machine learning software. Significant differences in results were determined using corrected paired t-test at 0.05 significance level. Some of the default parameters of the classifiers were optimized with Weka's search methods (these are always mentioned).

Various classifiers were used previously on keystroke dynamics datasets: statistical methods, decision trees, neural networks, fuzzy methods, support vectors are some in the list, for review see [11]. For this paper we selected some well-known algorithms implemented in Weka, covering various machine learning methods.

Naive Bayes is a probabilistic classifier based on Bayes' theorem. This classifier assumes that all features of an instance are independent which is usually not true. In spite of this naive approach, this classifier works well in many real-world applications.

Bayesian network is a probabilistic model that represents a set of random variables and their conditional dependencies using a directed acyclic graph. The nodes of the graph are the random variables and the edges represent conditionally dependent variables [1].

Nearest neighbors (k-NN, IBk in Weka) is an instance based classification algorithm where a new instance label is decided by the K closest neighbors (K=1 was used, giving the best results in our tests).

Decision trees are extremely popular methods based on tree like graphs, and appeared between the algorithms used for keystroke dynamics in recent years. Their training and testing time is fast, and classification results are among the best methods for many application areas. We used Weka's J48 implementation of the C4.5 algorithm (confidence factor 0.2, minimum 4 instances per leaf). The Random forest classifier [3] is an ensemble learning method, introducing randomization in the evaluation of a set of decision tree structures (we used 100 trees).

Support vector machines build a linear discriminant function that separates the instances of classes. If no linear separation is possible, a kernel maps the instances into a high-dimensional feature space. We used the LibSVM implementation through Weka with radial basis kernel. The C and γ kernel parameters were optimized by a grid search algorithm distinctly for the two datasets (C=10.55, γ =1.86 for the 41 features set and C=7.46, γ =0.25, for the 71 feature set respectively) and all input features were normalized (0-1).

Multilayer perceptrons (MLP) are artificial neural networks trained by the backpropagation algorithm. We only illustrate here the good result obtained by the Weka implementation with default settings (number of hidden layers was Weka's default setting [number of attributes +number of classes] / 2).

4. Results

4.1. Identification results

In order to show the difference in classification accuracy between keystroke data with and without touchscreen based features (pressure and finger area), two datasets were used. The first one contained 41 features (H+DD+UD+AH), whereas the second one 71 features (H+DD+UD+P+FA+AH+AP+AFA).

We used the collected data without any transformation, secondary feature calculation or feature selection (excepting normalization in case of support vector method). No boosting or other tuning methods were used, except for the case of random forests classifier, which is based on internal randomization. The data used in this work can be accessed at http://www.ms.sapientia.ro/~manyi/keystroke.html.

Table 3. Classification accuracies measured for the two datasets. The second column shows the accuracies for the first dataset: 41 features. The third column shows the accuracies for the second dataset: 71 features.

Classifier	Accuracy using time based features H+DD+UD+AH (41 features)	Accuracy using time and touchscreen based features H+DD+UD+P+FA+AH+AP+AFA (71 features)
Naïve Bayes	50.15% (2.86)	78.93% (2.63)
Bayesian Networks	75.95% (2.65)	91.94% (1.73)
C4.5(J48)	54.79% (3.84)	69.02% (3.32)
k-NN (IBk)	41.07% (2.83)	72.98% (2.25)
SVM(LibSVM)	61.71% (3.22)	88.33% (1.87)
Random forest	82.53% (2.53)	93.04% (1.65)
MLP	53.01% (3.39)	86.26% (2.19)

Table 3 presents the mean of accuracies for these methods. We report classification accuracies as mean of 10 runs of 10-fold stratified cross-validation on the whole dataset (total of 100 runs, standard deviation in parentheses). In all cases the methods performed significantly better on the 71 features dataset (at 0.05 significance level).

4.2. Verification results

Verification measurements were performed using the R [8] script provided by Killourhy & Maxion [6]. This script provides EER computations for three anomaly-detection methods based on Euclidean, Manhattan, and Mahalanobis metrics. The data was normalized and then partitioned into three equal parts, each containing 17

password data from each user. Two thirds of the data were used for creating user's template and the remaining one third for testing FRR. The first five password data from each user, except for the tested one, were used as impostor data (FAR testing). The measurement was repeated 3 times, each time using a different fold for testing and training. Table 4 summarizes the verification results, the average EER obtained by the three measurements. The lowest error (12.9%) was obtained by the Manhattan metric using both time and touchscreen based features. The lowest EER using time based features is 15.3%, provided by the Manhattan metric.

Table 4 Equal error rates for anomaly detectors

Detector	H+DD+UD+AH (41 features)	H+DD+UD+P+FA+AH+AP+AFA (71 features)
Euclidean	17.5%	15.7%
Manhattan	15.3%	12.9%
Mahalanobis	23.3%	16.6%

In conclusion, not only classification accuracy was improved by touchscreen based features, but also verification accuracy.

5. Conclusions

In this paper we demonstrated experimentally that touchscreen based features improve keystroke dynamics based identification and verification. A dataset was collected using Android devices with touchscreens. Both time and touchscreen based features were studied. Identification measurements were performed using several machine learning classification algorithms, of which the best performers were Random forests, Bayesian nets and SVM, in this order. Not only identification, but also verification measurements were performed on the same datasets. In this case EER were computed using three different distance metrics: Euclidean, Manhattan and Mahalanobis. Manhattan was the best performing distance function.

In case of identification measurements, the addition of touchscreen based features to the default feature set induced an increase of over 10% in accuracy for each classifier. This improvement is harder to notice in the case of verification measurements where the equal error rate was reduced by 2.4% (Manhattan metric).

In the data preprocessing stage, we observed that several typing patterns contained deletions and these were eliminated from the dataset. However, these errors can be considered a separate feature of the user and can be studied in the future.

Acknowledgements

This research has been supported by Sapientia Foundation - Institute for Scientific Research.

References

[1] Bouckaert R R. Bayesian Network Classifiers in Weka. 2008. http://www.cs.waikato.ac.nz/~remco/weka.bn.pdf

[2] Bours P, Continuous keystroke dynamics: A different perspective towards biometric evaluation, Information Security Technical Report, Volume 17, Issues 1–2, February 2012, p. 36-43

[3] Breiman. L. Random forests, Machine Learning 45(1) , 2001, p. 5-32

[4] Draffin B, Zhu J, Zhang J, KeySens. Passive User Authentication through Micro-behavior Modeling of Soft Keyboard Interaction, In: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering Volume 130, 2014, p. 184-201

[5] Johansen UA. Keystroke Dynamics on a Device with Touch Screen. Master's Thesis, Computer Science and Media Technology Department, Gjovik University, 2012.

[6] Killourhy K S, Maxion R A. Comparing anomaly detection algorithms for keystroke dynamics, in Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '09), 2009, p. 125–134

[7] Nonaka H, Kurihara M. Sensing pressure for authentication system using keystroke dynamics, in Proceedings of the International Conference on Computational Intelligence, Istanbul, Turkey, 2004, p. 19–22

[8] http://www.r-project.org/

[9] Saevanee H, Bhattarakosol P, Authenticating user using keystroke dynamics and finger pressure, 6th IEEE Consumer Communications and Networking Conference, 2009, p. 1-2.
[10] Sen S, Muralidharan K. Putting Pressure on Mobile Authentication, Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU), 2014, p. 56-61.
[11] Teh P S, Teoh A B J, Yue S. A Survey of Keystroke Dynamics Biometrics, The Scientific World Journal, vol. 2013, Article ID 408280, 2013, 24 pages.
[12] 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, 2013, p. 114-119
[13] Witten I H, Frank E, Hall M, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann Publishers, 2011.