Statistical Keystroke Dynamics System on Mobile Devices for Experimental Data Collection and User…

Article · September 2016

CITATIONS
0

READS
117

2 authors:

Noor Mahmood
middle east unviersity
3 PUBLICATIONS **1** CITATION
SEE PROFILE

Mudhafar M Al-Jarrah
15 PUBLICATIONS **22** CITATIONS
SEE PROFILE

Statistical Keystroke Dynamics System on Mobile Devices for Experimental Data Collection and User Authentication

Noor Mahmood Al-Obaidi
Dept. of Computer Science
Middle East University
Amman, Jordan
e-mail: noor_mah89@yahoo.com

Mudhafar M. Al-Jarrah
Dept. of Computer Information Systems
Middle East University
Amman, Jordan
e-mail: maljarrah@yahoo.com

***Abstract*—This paper presents a keystroke dynamics system for mobile devices that employs a statistical distance-to-median anomaly detector. The selected feature set combines the keystroke timing features of hold and latency and the touch screen features of pressure and finger area. The proposed system consists of two modules: training and testing. The aim of the system is to be a research tool to serve two purposes: (i) the generation of a model-independent dataset of keystroke data on mobile devices, for comparison of keystroke dynamics anomaly detectors; (ii) to be used in the evaluation of the authentication performance of the implemented distance-to-median anomaly detector. The system works in the Android environment on Nexus smartphones and tablets. The experimental work has generated a dataset of 2856 records from 56 subjects, 51 records per subject, where each record represents 71 feature elements resulting from the typing of a standard 10-character password. Statistical analysis of the collected dataset showed an equal-error-rate (EER) of 0.049 when using a different pass-mark per subject, and 0.054 when using a global pass-mark for all subjects. The EER results are much lower than previously published results using three distance-based verification models. Also, the false-acceptance-rate at 5% false-rejection-rate is 5.6%, which is much lower than previously published results, but it is still high and needs to be reduced. Evaluation of the testing (authentication) part of the system was carried out through test runs where a genuine user enters his user-id and password as a login attempt, and the resulting test vector of feature elements are matched against the stored template of the user. The login attempt is classified as genuine or impostor based on a preset pass-mark. Conclusions and suggestions for future work are presented.**

Keywords— behavioral biometrics; keystroke dynamics; EER; FAR; FRR; distance-to-median; anomaly detector; pass-mark, test-score.

I. INTRODUCTION

Recently, digital devices such as smartphones and tablets, have overtaken PCs and laptops as the main communication and computing tools for most people, for carrying out personal and business tasks. This trend has highlighted the need to protect the private and business data stored on these devices [1], especially that mobile devices have additional security risks, they can be lost or stolen, and are more vulnerable to attacks due to their use in social interactions. Access control through user authentication is one of the primary technologies for data protection on mobile devices.

User authentication for access control has traditionally relied on passwords, an approach that is no longer considered satisfactory because passwords can be compromised by hackers using keyloggers, or through shoulder surfing or network sniffing. Two-factor authentication has been offered as a safer solution, such as sending a one-time-password (OTP) over a secondary communication channel, but this approach is not risk-free if the OTP is picked-up by a hacker who manages to access the secondary channel, or if the mobile device is infected by malware.

Alternative authentication methods for mobile devices have been considered, using biometric features (physiological and behavioral). In general, the biometric systems offer several advantages over password-based authentication schemes, and can provide a much more accurate and reliable security protection, because it relies on unique features for identity verification. The behavioral biometrics method has the added advantage that it can be used for continuous authentication [2].

Keystroke dynamics is a behavioral biometrics-based authentication method that identifies the typing rhythm of an individual by observing the timing of keystrokes and the latency between keypresses. The keystroke dynamics method was initially studied on mechanical keyboards [3], but recently the focus has shifted to touch mobile devices. Ali et al. [4] present a state of the art review of research on keystroke biometrics on mobile devices.

The research work on behavioral biometrics on mobile devices, requires a methodology that involves steps below:

- Selection of measurable features or attributes.
- Selection of an anomaly detector model.
- A public dataset for reproducible evaluation.
- Implementation of the anomaly detector, for dynamic evaluation of the selected model.

This paper presents a keystroke dynamics system that implements a statistical anomaly detector, for dataset collection, and the dynamic authentication of users on mobile devices, using timing and touch screen features.

II. RELATED WORK

Keystroke dynamics (KSD) research has progressed over the past few years into a scientific discipline where models and systems are evaluated using public datasets, and experimental results are repeatable by others. The Ph.D. thesis of Killourhy [5] and the paper by Killourhy and Maxion [6] represent an important milestone in KSD research; that was carried out at Carnegie Mellon University (CMU). The work presented a comprehensive comparative study of KSD anomaly detectors, using an experimental approach in which a KSD dataset was collected and utilized in the comparison. The aim of the study was to evaluate most published anomaly detectors on a unified dataset, using the same typing text, to arrive at a fair and scientifically-based comparison. The work was motivated by the fact that published results of some classifiers could not be reproduced, so when evaluations were replicated, the results were often extremely different. Therefore, an independent evaluation was needed in which different algorithms are compared on equal grounds. The work involved implementing 14 known anomaly detection algorithms, which helped to provide an unbiased implementation platform for all algorithms.

The authors collected data from 51 subjects, each typing the same password 400 times, over eight sessions. The unified password which was typed by all subjects is a complex password of mixed characters (“tie5Roanl”). The 400 records of each subject were divided into 200 records for training and 200 for genuine user testing, while an impostor set of 250 records was used, which contained 5 records from each subject. The measured features were Hold, UD (Latency) and DD (Hold + UD). In the comparison process, the work identified which detectors had the lowest equal-error-rate (EER). The dataset was made available online so that other researchers can assess new detectors and report comparative results.

The CMU dataset was used by other researchers in the development of anomaly detectors that produced lower error rates. In [7] a statistical anomaly detector, the median vector proximity model, was presented which measured the distance from the median rather than the mean as an indicator of the acceptance of a typing feature. The median was preferred over the mean because of the nature of the typing rhythm data which have a lot of variations and outliers. The acceptable distance from the median in this study was the standard deviation. The reported average of EER for the 51 subjects of the CMU dataset was 0.08, which is lower than the best performing detector in the CMU study (0.096). Another median-based model was presented in [8], in which the distance to the median was calculated as the product of the median multiplied a constant factor of 0.7. The reported average of EER for the CMU dataset was 0.07. A multi-model based on the median model was investigated in [9] which resulted in EER value of 0.062 using the CMU dataset.

The work of Antal, et al. [10] at Sapientia University (SU) presented the experimental results of collecting a KSD dataset on touch mobile devices, using the same password as in the CMU study, for consistency with former studies. The study added touch screen features of pressure and finger area, and the data collection software was implemented on Nexus mobile devices running the Android operating system. The measured feature set included the timing features in the CMU study (Hold, UD, DD), and the touch features of pressure (P) and finger area (FA). The collected dataset included typing records of 42 subjects where each subject made 51 typing attempts, 34 for training and 17 for testing. In this study. The EER metric was calculated for three verification models (Euclidean, Manhattan, and Mahalanobis) that were used in the CMU study, for comparison purposes. These models relied on the distance to the mean to measure anomaly or acceptance. The experimental results showed reduced EER error rates despite the fact the SU dataset was much smaller than the CMU dataset, which can be attributed to the effect of adding the touch features into the KSD model. The dataset has been made available online.

The paper in [11] presented an enhanced statistical median-based anomaly detector, the Med-Min-Diff (MMD) model, which is a binary classifier that compares a testing vector of feature elements resulting from the typing of a password, with the pre-

stored template of upper and lower thresholds. The MMD model was evaluated using the mobile SU public dataset, which consisted of 41 timing features only, and 71 timing and touch screen features. The evaluation gave an EER value of 0.067, which is much lower than the EER values of three distance-based verification models as presented in [10]. The MMD model followed previous work [7, 8, 9] in which the anomaly or acceptance of a feature element is determined based on the distance from the median of the set values of the feature element. The MMD model has been chosen as the anomaly detector for the proposed keystroke dynamics system of the present work.

III. THE PROPOSED KSDMob SYSTEM

A. Background

The proposed KSDMob system is designed to work on mobile devices (Nexus smartphones and tablets) that use the Android operating system. The system provides two phases: training, for keystrokes data collection, and testing, for live authentication, based on the collected data. The system deploys a statistical anomaly detector (binary classifier) to identify a login attempt of the user as genuine or impostor.

B. Feature Selection in KSDMob

In previous research on keystroke dynamics that were based on the CMU comparative study, the selected features were based on desktop keyboards, which included timing features only (Hold, UD, DD). Mobile devices have additional features that can be measured, including pressure, finger area and sensor readings. In this research, we are adopting a feature set that includes timing as well as pressure and finger area, which was proposed in [10]. Details of the selected feature set elements are as follows:

1. Hold: key-press duration.
2. Latency or Up-Down (UD): time difference between two key events.
3. Down-Down (DD): time between key-down of the first key and key-down of the second key.
4. Pressure (P): the maximum value of finger pressure during key-press on a particular key area.
5. Finger Area (FA): the maximum value of finger area during key-press on a particular key area.

C. Training Phase

The KSDMob system provides a training phase module which deals with the following tasks:

- User registration, in which user-id and password are entered.
- Settings, in which the number of repetitions of password entry, and the pass-mark are set.
- Training password entry, in which the password is typed a number of times according to the setting. The collected data records of each typing attempt are stored in a CSV file for a subsequent statistical analysis.
- Template generation, in which the user-id and a template of the upper and lower thresholds for each feature element are generated and stored in a database.

Fig. 1 shows the user registration screen of the training module. The registration password is used in the training phase as the reference text.

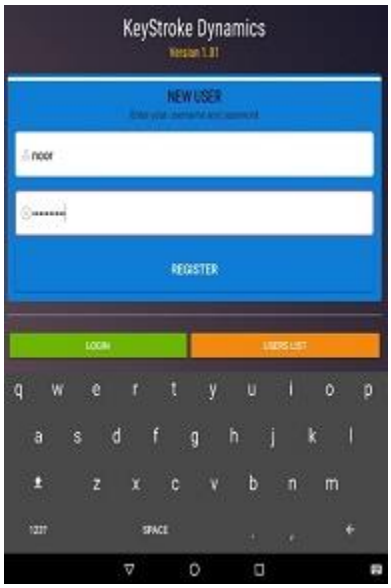

Figure 1: New User Registration

Fig. 2 shows the training password entry screen, where the password is entered a pre-determined number of times, and any entered password that does not match the registration password is rejected.

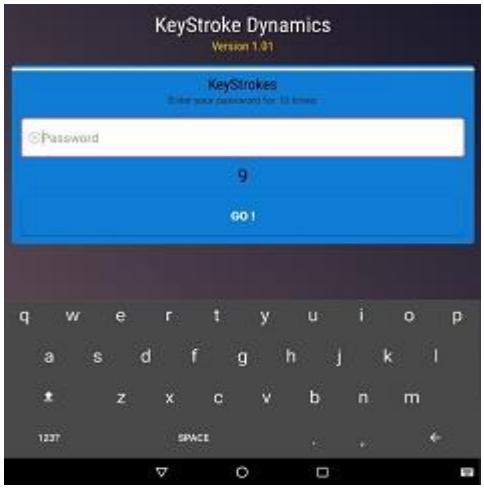

Figure 2: Training Phase Password Entry

The collected data records from the training phase can be used for training and template generation of a user's input in an authentication context. Also, the training data for each user can be used with the genuine (positive) testing data for the same user, collected using the same tool, and the impostor (negative) testing data, for statistical error metrics analysis (ERR, FAR, FRR) of a group of subjects.

D. The MMD Anomaly Detector Model

The KSDMob system employs a statistical median-based anomaly detector, the MMD model, in the authentication phase, which was proposed in [13]. The EER detection performance of the proposed model was evaluated using the SU dataset.

The anomaly detection process of the MMD model is based on the following criteria:

- The feature set elements are the Hold, Pressure and Finger Area for each pressed key, and the latencies (UD and DD) between any two keys. The "Enter" key is included as part of the typed text.
- The central value for each feature set element is the median.
- The Lower Threshold (LT) = Minimum value of a feature set element.
- The Distance to Median (DTM) = Median – Minimum of a feature set element.
- The Upper Threshold (UT) = Median + DTM x C, where C is a constant factor that allows the upper threshold to cover a wider area from the median than the lower threshold. The value of C was chosen to be 1.1, derived through experimental tuning to get the lowest EER.
- The Test-Score is the number of feature set elements in an authentication test vector that are classified as genuine. The test vector is a set of 71 feature elements derived from a typing attempt during the testing phase.
- The Pass-Mark is the metric used by the anomaly detector to compare the Test-Score with, to decide on the classification outcome (genuine or impostor).

E. Testing Phase

In the dynamic testing (authentication) phase, the stored template of the user is retrieved from the database using his user-id. The test vector that is generated from the password login is verified against the stored template, to determine if the user is genuine or an impostor. The following steps are carried out:

- Loading the training template using the user-id
- Reading the testing password and generating the test vector, which contains the same number of feature elements as in the template.
- Matching the test vector with the template's upper and lower thresholds. Each feature element of the test vector is set to 1 if the feature element is within the thresholds, otherwise 0.
- The Test-Score is calculated as the sum of features' scores.
- The test outcome is genuine if the Test-Score is greater than or equal to the Pass-Mark.

Fig. 3 shows the testing phase login screen.

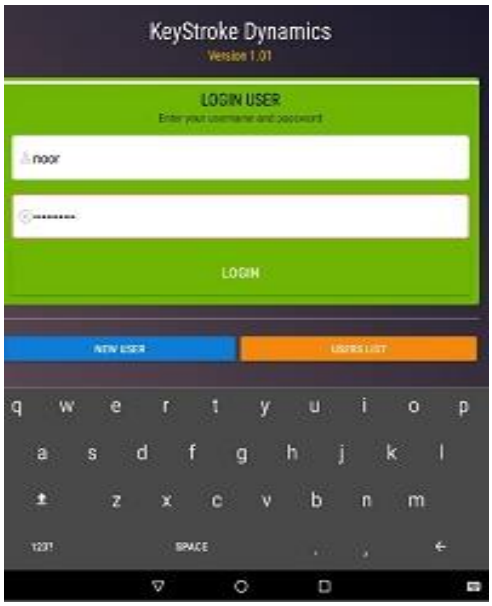

Figure 3: User Login

IV. RESULTS and DISCUSSIONS

A. Data Collection

The data collection was executed on Nexus 7 tablet running Android. In our experiment we used a second Nexus model (Nexus 9) for verification of results. As in [11], the collected data consisted of two subsets, a 41 features subset (14 Hold, 13 DD, 13 UD, Avg. of Hold) and a 71 features subset (14 Hold, 13 DD, 13 UD, 14 Pressure, 14 Finger Area, Avg. of Hold, Avg. of Pressure, Avg. of Finger Area).

The keystroke data were collected in two sessions for each subject, the first session was for training which consisted of 34 typing attempts, and the second session was for genuine user testing which consisted of 17 typing attempts. In our experiment, each subject was allowed a pre-training rehearsal session of 10 typing attempts, to familiarize the subject with the password, so as to reduce the effect of a new password on a user's [typing profile. Analyses of the collected dataset of 2,856 typing attempts are presented in sections below. The dataset for the 71 features subset has been made available online at [12].

B. Coefficient of Variation Analysis

The first analysis of the KSD-Mobile dataset is the coefficient of variation (CV), which is the ratio of standard deviation to average, of each feature element. Table I shows the average of the coefficient of variation of each of the feature categories (Hold, UD, DD, P, FA) for the 71 timing and touch features subset. It can be seen that the latency features (UD and DD) have higher CV than Hold, therefore will have a more distinguishing effect among different users. The pressure's CV is also high, close to the latencies, which suggests that it is also sensitive to variations in the typing pressure among different users. The size of finger area has similar low CV as the Hold feature; therefore, a weak indicator of variation among users.

Table I.: Analysis of the Coefficient of Variation According to Features

Feature	**Average of Coefficient of Variation**
Hold	0.2468
DD	1.2315
UD	1.4482
Pressure	1.1187
Finger Area	0.2975

C. Analysis Results Using the MMD Model

The analyses reported in the following sections are produced through a statistical analysis of the MEU-Mobile dataset. For each subject, 34 records are used as training data to calculate the upper and lower thresholds (the template), and 17 records are used as testing data. Also, an impostor data subset is created against each subject, which included five records from each subject other than the subject under attack.

D. EER Analysis Using Variable Pass-Marks

The dataset is analyzed using the MMD model which calculates the EER value, where the Pass-Mark is variable, i.e. it is determined separately for each subject. The analysis is done on both the 41 features timing data only, and the 71 features which include timing and touch screen features.

The EER results in Table II show that the EER value for timing and touch features is much lower than the timing only EER value, which supports the idea that adding more non-timing features will enhance the detection performance. Also, it is noticeable that both EER values are lower than the EER values obtained using the same model on the SU dataset as reported in [13], which can be attributed to the effect of pre-training rehearsal in our experiment. The obtained EER values are much lower than the EER values of the three verification models (Euclidean, Manhattan, and Mahalanobis) that were reported in [10].

Table II. ERR Analysis Using a Variable Pass-Mark

Anomaly Detector Model	**EER result using 41 timing feature elements**	**EER result using 71 timing and touch feature elements**
MMD	0.084	0.049

E. EER Analysis Using a Global Pass-Mark

A global (fixed) Pass-Mark is determined for the entire population, which represents the average of all subjects' Pass-Marks. The analysis is done on both the 41 features timing data, and the 71 features which include timing and touch screen features. The results in Table III show that for the global Pass-Mark analysis, the EER value is lower when using timing and touch features. Also, the results demonstrate that using one Pass-Mark for all subjects gave close results to the case of tuned, variable Pass-Marks.

Table III. EER Analysis Using a Global Pass-Mark

Anomaly Detector Model	EER using 41 timing feature elements	EER using 71 timing and touch feature elements
MMD	0.091	0.054

F. FAR Analysis at 5% FRR

This analysis is achieved by tuning the Pass-Mark for each subject to the point of 5% FRR, to get the FAR value at that point. The assumption here is that a 5% rejection rate of genuine users is acceptable [5]. The analysis is carried out for the 71 timing and touch features subset. Table IV shows that the FAR value of 5.79% is very close to the FRR value. However, FAR is much more serious than FRR. Therefore, more reduction in the FAR value is needed for a stronger authentication model.

Table IV. FAR Analysis at 5% FRR

Anomaly Detector Model	FAR result at 5% FRR using 71 timing and touch features
MMD	FAR 5.79% FRR 5.04%

IV. CONCLUSION and FUTURE WORK

This work has shown that touch screen features have enhanced the authentication performance of the statistical MMD anomaly detector. The following points highlight the main findings and contributions of this research:

1. The EER values (0.0494 for the 71 features data and 0.0845 for the 41 features data), using the MMD anomaly detector and the MEU-Mobile dataset, were lower than equivalent values that were obtained using the same anomaly detector with the SU dataset. The difference between results of using the same model on the two datasets can be attributed to the effect of doing a rehearsal, in the KSDMob experiment.

2. The EER values with a global (fixed) Pass-Mark has resulted in a value that is very close to the variable Pass-Mark case (5.48 vs. 4.94), using the 71 features data. This suggests that the MMD model can be used with a pre-determined global Pass-Mark for all users.

3. The false-acceptance-rate (FAR) at 5% false-rejection-rate (FRR) was 5.79%, which is very close to the FRR result (5.04%). The 5% FRR can be accepted as a rejection rate of genuine users, but the FAR value needs to be further reduced.

Suggestions for future work, for possible enhancement of the authentication process using keystroke dynamics on mobile devices are:

- Enhancing the median-based anomaly detector, by adding other timing and sensor features.
- Investigating the effect of the training sample size on the authentication outcome, and that of the size positive and negative samples on the accuracy of error rates calculations.
- Collecting and analyzing data from dynamic authentication runs using the proposed system, for verification of the performance of anomaly detectors as practical authentication tools.

REFERENCES

[1] S. Ryan, "Mobile keystroke dynamics: assessment and implementation", Masters thesis, California State University, Northridge, 2015.

[2] S. Mondal & P. Bours, "Combining keystroke and mouse dynamics for continuous user authentication and identification", IEEE International Conference on Identity, Security and Behavior Analysis (ISBA), 2016.

[3] P. S. Teh, A. B. J. Teoh, & S. Yue, "A survey of keystroke dynamics biometrics", The Scientific World Journal, 2013.

[4] L. Ali, J. V. Monaco, C. C. Tappert, & M. Qiu, "Keystroke Biometric Systems for User Authentication", Journal of Signal Processing Systems, Springer US, pp 1-16, First online, 07 March 2016.

[5] K. S. Killourhy, "A scientific understanding of keystroke dynamics", Carnegie Mellon University, PhD Thesis, No. CMU-CS-12-100, Department of Computer Science, 2012.

[6] K. Killourhy, and R. Maxion, "Comparing anomaly detectors for keystroke dynamics", Proceedings of the 39th Annual International Conference on Dependable Systems and Networks (DSN2009), June29-July2, pp.125–134, IEEE Computer Society Press, Los Alamitos, 2009.

[7] M. M. Al-Jarrah, "An anomaly detector for keystroke dynamics based on medians vector proximity", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 6, pp. 988-993, 2012.

[8] A. O. Al-Rahmani, "An Enhanced Classifier for Authentication in Keystroke Dynamics Using Experimental Data", Master thesis, Middle East University, Jordan, 2014.

[9] S. A. Al-Robayei, "A Multi-Model Keystroke Dynamics Anomaly Detector for User

Authentication", Master thesis, Middle East University, Jordan, 2016.
[10] M. Antal, L. Z. Szabó, & I. László, "Keystroke dynamics on android platform" Procedia Technology, Vol. 19, pp.820-826. 2015.
[11] N. M. Al-Obaidi & M. M. Al-Jarrah, "Statistical Median-Based Classifier Model for Keystroke Dynamics on Mobile Devices", 6th International Conference on Digital Information Processing and Communications, Lebanese University, ISBN: 978-1-4673-7504-7, IEEE, 2016.
[12] N. M. Al-Obaidi & M. M. Al-Jarrah, "MEU-Mobile Keystroke Dynamics Dataset, available online on: https://sites.google.com/site/keystrokedatasets/, viewed on 1-July-2016.