Dynamic User Verification Using Touch Keystroke Based on Medians Vector Proximity

Shatha J. Alghamdi
Faculty of Computing and Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia
sjalghamdi@kau.edu.sa

Lamiaa A. Elrefaei
Faculty of Computing and Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia
Faculty of Engineering, Shoubra
Benha University, Cairo, Egypt
lamiabdalah@yahoo.com

Abstract - In this paper a user verification system on mobile phones is proposed. This system is based on behavioral biometric traits which is a keystroke dynamics derived from a touchable keyboard. A mobile application is developed for collecting those touch keystroke dynamics. In contrast to other systems, no specific text or numbers are used to build our dataset. The Median Vector Proximity classifier is applied on the touch keystroke data (touchable keyboard) and the performance of the system is investigated using different number of features and we found that the system with 31 features gained an average EER=12.9%. While with an extra two features (average of finger size and pressure) the average EER=12.2%. This shows that the more features used results in more accurate systems. The proposed system is compared against other systems and shows promising results in dynamic authentication area.

Keywords - Touch Keystroke Dynamic; Biometric; Mobile Authentication; Soft Keyboard; Touch Gesture; Touch Screen.

I. INTRODUCTION

Mobile phone is one of the most essential things for people. These devices are not only used for calling or sending text messages, they are also used in many applications such as accessing internet (social networking, e-banking or e-commerce), sending and receiving emails and storing sensitive documents [1]. Many of these applications require the user to establish their identities on the phone. On a mobile device, the user authentication must act quickly, easily and user-friendly.

Biometrics meets these authentication criteria, so they become more preferable. Most of behavioral biometrics are collected implicitly and are cheaper than the biological ones [2]. Keystroke dynamics is a behavioral biometric that checks what you type and how you type it [3].

The literature has explored the feasibility of using keystroke dynamics and typing pattern behavior for user authentication for personal computers (PCs) [4]. Unlike PCs and old mobile phones, touch screen is the primary input medium on the latest mobile phones and tablets [4]. However, the finger gesture on the touch screen has additional features which could be detected through touch screen.

Al-Jarrah [5] proposed a personal computer anomaly detector for keystroke dynamics (hardware keyboard). His proposed system was based on statistical measure of proximity depending on median and standard deviation. The author evaluated his proposed system (Median Vector Proximity) using the CMU dataset [6]. In this paper, we propose applying Median Vector Proximity [5] on touch keystroke data derived from mobile’s touch screen (touchable keyboard). In this case, we built our own dataset since there is no available touch gesture dataset [3, 4, 7, 8].

The main contributions of this work are:

• Using mobile’s touch screen to detect touch keystroke dynamics.
• Building our own touch gesture dataset by using an Android application.
• Don’t specify what should the participants write during the experiment (dynamic text).
• Investigating the performance of the median vector proximity classifier using different numbers of touch gesture feature.
• Evaluate the proposed system and compare it with the previous work.

The rest of this paper is organized as follows: the related work is introduced in section II. Our proposed system is explained in section III. The system evaluation and results are discussed in section IV. Finally, a conclusion and a future work are shown in section V.

II. RELATED WORK

The literature showed that the keystroke dynamics and typing pattern behavior -detected from personal computer users- are feasible for user authentication [4]. Based on a proximity statistical measures, Al-Jarrah [5] proposed an anomaly detector for keystroke dynamics. The author evaluated his proposed system (Median Vector Proximity) using the CMU dataset which used the password (.tie5R0oan) for their experiments [6]. Using 31 features detected from hardware keyboard (personal computers), it has shown a lower error rates (EER≈8%) compared with other 14 detectors in [6].
In the other hand, several studies found that the features extracted from touch-screen could be used efficiently to differentiate between different users [3, 7, 8]. In [7], Julio Angulo and Erik Wastlund built an android application to collect the data about how users draw specific lock patterns on a touch-screen. Two main features were used: finger-in-dot time, which is the time interval from the moment the user’s finger touches a dot until the finger is moving outside the dot area. The other feature was finger-in-between-dots time, which is the finger’s speed when moving from one dot to another. By analyzing data from 32 users using different classifiers (Euclidean, Manhattan, Mahalanobis, Recursive Partitioning, Supportive Vector Machine and Random Forest), they found that a Random Forest classifier attained the best result by giving an average EER of 10.39%. This percentage approved that finger movements to draw the lock pattern on touch-screen could be used for identification purposes.

There are few researches that focus on touch keystroke features. Nan Zeng et al. [3] proposed a non-intrusive user verification mechanism using 12-key touchable keyboard. The acceleration, pressure, touch size, key-hold and inter-key time feature sets are collected from 80 users through an Android application. Their approach was based on removing the outlier information so a small amount of raw data is filtered out. The authors proved the uniqueness of each user pattern. They used a nearest neighbor algorithm to classify data. The EER for a 4-digit password is about 3.65%, while it is between 4.55% and 4.45% for 8-digit password. The authors also tested how the four sets of features contribute to the final accuracy, and they found that the combination of all feature sets always outperforms individual feature set.

Xuan Huang et al. [9] proposed a mobile authentication system which is based on specific username and password (username="abertaytest" password="abertay2011") but also combines the typing behavior recognition. The proposed system detects the keystroke latency and key hold-time features from a touchable QWERTY-keyboard. To classify these features, authors used statistical approach. This system has four alert levels (low, medium, high and very high). By testing the system on 40 users with different alert levels, they found that EER = 7.5%.

Margit Antal et al. [10] found that the classification and verification accuracy was improved by adding touch screen based features. They tested their system on 42 users using two types of android mobiles: Nexus7 tablet and Mobile LG Optimus L7 II P710 device. Each user types a specific password ("tie5Roanl"), they extracted 41 purely touch keystroke features and 71 features (mixed of keystroke and touch features). For identification, they used Naïve Bayes, Bayesian Networks, C4.5(J48), k-NN, SVM, Random Forest and MLP classifiers. The random forest showed 82.5% and 93% of accuracy by using 41 and 71 features respectively. For Verification, they used Euclidean, Manhattan and Mahalanobis classifiers. The Manhattan provided 15.3% and 12.9% of EER by using 41 and 71 features respectively.

All the work in [3, 4, 7-10] is done on a touchable keyboard and they specified a specific word or numbers to be entered by the participants. In this paper, we propose a user verification system on mobile phone using the touch keystroke data detected from the touch screen (touchable QWERTY-keyboard) as in [9] and applying Median Vector Proximity classifier [5] on it.

By exploring the literature [3, 4, 7, 8], there is no available touch gesture dataset. So, we developed a mobile application for collecting the required data and built our own dataset.

III. PROPOSED WORK

In this section, the proposed system structure is presented first, then the data collection method is explained, and then the computed features is discussed and finally the details of the user identification using median vector classifier is presented.

A. Proposed system

We propose a mobile user verification system which depends on mobile’s touch screen for detecting keystroke dynamics. Like any biometric system, this system consists of some modules [11] as show in Fig. 1.

The system has two phases: Enrolment and Authentication. Here the sensor that will be used for collecting data is the mobile’s touch screen. The extracted features are introduced in section III.C. In the Matching module, the median vector classifier [5] will be used and its details are discussed in section III.D.

B. Data Collection

In order to collect touch data, we built an android application –using Android Development Tools (ADT) with Eclipse- which runs as stand-alone application on android mobile. This application collects the touch data from a touchable keyboard.
We conduct the experiment on only one mobile device to make sure that the data is consistent. Here we used a HTC one M8 Android phone with a Super LCD3 capacitive touch screen, and 1440 x 2560 pixels (5.5 inches) to perform the experiment.

A group of 17 users were selected to participate in our experiments. The only requirement was that the participant is actually a user of a smart phone with a touch screen. Each user were asked to enter whatever message he wants (e.g. a description message about himself) in a single session. Five sessions were recorded for every user.

Actually, dealing with touchable keyboard is different than dealing with the hardware one. We found that a single touch gesture (tap, scroll, zoom … etc) on the touch screen contains many of touch events. When the user touches the touch screen, the gesture starts and a stream of events take place. The gesture ends when the user’s finger leaves the screen. Throughout this interaction, the android system tracks the position and other information of user’s finger until the gesture ends. A sample of raw data collected from the touch screen and recorded by the phone is shown in Table I. The table shows the information about three taps on the touch screen (tap touch gesture). A single tap starts with action down and ends with action up. During this interaction all finger’s touch information is detected under the action move mode. Those information are [12, 13]:

- The Down Time which is the time when the gesture was started in milliseconds (ms).
- The Event Time which is the time of the current touch event in milliseconds (ms).
- The size of the finger touch contacts the touch screen. The actual value in pixels is normalized and scaled to a value between 0 and 1.
- The finger pressure applied to the touch screen; or in other words, the strength of the finger touch contacts the touch screen. The pressure ranges from 0 (no pressure) to 1 (normal pressure).

We will extract two extra features which are average size and average pressure. Those features will be used later to investigate if they really have an effect on the authentication accuracy.

D. Median Vector Classifier

The user authentication method involves two phases [11]: Training and Testing phases. In the Training phase, a user profile is built to be used later. In the testing phase, the user’s typing behavior is compared to the stored user’s profile to make a decision whether accept or reject that user.

We used the legitimate-attacker procedure as in [5]. The dataset will be composed into:
- Legitimate user training data.
- Legitimate user testing data.
- Attacker testing data.

Through all 17 participants, one is selected as a legitimate user while the remaining 16 participants as attackers. All participants will do 5 sessions. For a selected legitimate user, 3 sessions are used in training phase to train the classifier and build a legitimate user profile (template) which will be used later in decision making in the testing phase. The remaining 2 sessions of the selected user are used as a legitimate user testing data. For the other 16 attackers, their 5 sessions are treated as attackers data (Fig. 2 illustrates this process).

In training phase, the medians vector (μ) for the training set will be calculated (5 medians). In addition, the standard deviations vector (σ) for the same training set is calculated [5].

Later, the legitimate user and attackers testing data are tested depending on these two vectors (median and standard deviation).

| TABLE I. SAMPLE OF TOUCH RAW DATA COLLECTED USING OUR ANDROID APPLICATION |
|-------------------------------------------------|-----------------|-----------------|-----------------|-----------------|
| **Action Type** | **Down Time** | **Event Time** | **Size** | **Pressure** |
| **First Tap**  | 96274361     | 96274361       | 0.24705884 | 0.26666668    |
| DOWN            | 96274361     | 96274424       | 0.27450982 | 0.3            |
| MOVE            | 96274361     | 96274458       | 0.19862746 | 0.33333334    |
| MOVE            | 96274361     | 96274464       | 0.19862746 | 0.33333334    |
| UP              | 96276665     | 96276665       | 0.27450982 | 0.3            |
| **Second Tap**  | 96276665     | 96276777       | 0.27450982 | 0.33333334    |
| MOVE            | 96276665     | 9627786        | 0.21960786 | 0.33333334    |
| UP              | 96277247     | 96277247       | 0.19862746 | 0.26666668    |
| **Third Tap**   | 96277247     | 96277341       | 0.19862746 | 0.26666668    |

C. Feature Extraction

We extracted the same features as in [5, 6]. Those features are:
- Hold time is the time between key-down and key-up of a single character.
- Up-Down time is the time between key-up of a character and key-down of the next character.
- Down-Down time is the time between key-down of a character and the key-down of the next character.

The CMU dataset [6] contains 31 features extracted from 10-character password .tie5Roant plus the return key. We extracted features for the first 11 characters typed by the participant. No specific text should be written by the participant (dynamic text).

For 11 characters, there are 11 of holds, 10 Up-Downs and 10 Down-Downs. Total of 31 features will be extracted. We will extract two extra features which are average size and average pressure. Those features will be used later to investigate if they really have an effect on the authentication accuracy.
In testing phase, the standard deviation ($\sigma$) of the feature element has been selected to be the proximity distance threshold from the median ($\mu$). For every feature element $i$ from the 31 features; if the feature is within the proximity distance from the feature’s median, mark the Feature Score (FS) as 1 otherwise mark it as 0 [5] as in (1).

$$FS_i = \begin{cases} 1, & f_i \in [\mu \pm \sigma] \\ 0, & f_i \notin [\mu \pm \sigma] \end{cases}, i = 1, 2, 3, \ldots F$$ (1)

Then, calculate the Test-Score (TS) as the sum of Feature-Scores (FS) as in (2).

$$TS = \sum_{i=1}^{F} FS_i \quad , i = 1, 2, 3 \ldots F$$ (2)

After that, by determining the minimum percentage of accepted features which is the Pass-Threshold (PT), we will find the Pass-Score (PS) which is the minimum number of accepted features of a user as in (3).

$$PS = PT \times F$$ (3)

Finally, if the Test-Score is greater than Pass-Score (TS $\geq$ PS) then mark the input test as legitimate (accepted input) and as attacker (rejected input) otherwise.

IV. SYSTEM EVALUATION

A. Evaluation Techniques

The performance of any biometric system is described by False Acceptance Rate (FAR) and False Rejection Rate (FRR) [11]. FAR is the probability of accepting an attacker while the FRR is the probability of rejecting a legitimate user. The point where FAR equals FRR is an Equal Error Rate (EER).

Systems that have lower FAR and FRR are more secure and more accurate in distinguishing users. FAR and FRR are calculated using the following formulas (4,5):

$$FAR = \frac{Number \ of \ accepted \ AA}{Total \ number \ of \ AA}$$ (4)

$$FRR = \frac{Number \ of \ rejected \ LA}{Total \ number \ of \ LA}$$ (5)

Were AA is an Attacker Attempts and LA is a Legitimate user Attempts.

In our proposed work, we test with different Pass-Thresholds (the minimum percentage of accepted features) ranging between 40% to 80% to find a suitable Pass-Threshold where that FAR is equal to FRR (EER). The value of PT can be tuned to reduce or increase the FAR and FRR depending on the access control priority. Reducing the FAR causes an increasing in rejecting a legitimate user (increase security and decrease usability). In the other hand, reducing the FRR would decrease the probability of rejecting a legitimate user and the probability of accepting an attacker is increased (decrease security and increase usability).

B. System Results

By implementing the Median Vector Proximity algorithm [5] on the 31 extracted features, the FAR and FRR for different Pass-Thresholds are shown in Table II. The Pass-Threshold that results on FAR=FRR is about 45.9%.

Also, the FAR and FRR for different Pass-Thresholds are calculated for the 33 features (adding the average of size and pressure features) as in Table III. We found that the EER (FAR and FRR are equal) when the Pass-Threshold = 46 %.

The FAR and FRR with different Pass-Thresholds for both 31 and 33 features based on the Median Vector Proximity are shown in Fig. 3, 4 and 5. We can see that when reducing Pass-Threshold, the FAR will be increased (less secure system). In the other hand, when increasing the Pass-Threshold, FRR will be increased (more secure system). As we can see in Fig. 3, the FAR and FRR for both systems are semi equal when the PT=50%.

Among all 17 typists, the EER average and standard deviation is calculated for the Median Vector Proximity when applied on 31 and 33 features as in Table IV.

As we can see from Fig. 3, 4, 5 and Table IV, when using 33 features the avgEER is decreased by 0.7% which brings better results. This shows that the accuracy of the system will be increased by the increasing in the features set. Also, by increasing the pass threshold over 50%, the system will become more secure but less usable. So depending on the user’s needs, we can tune the pass threshold to achieve a desirable result.

<table>
<thead>
<tr>
<th>Pass Threshold</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAR</td>
<td>0.5404</td>
<td>0.2206</td>
<td>0.0784</td>
<td>0.0123</td>
<td>0.0037</td>
</tr>
<tr>
<td>FRR</td>
<td>0.1176</td>
<td>0.2353</td>
<td>0.4118</td>
<td>0.6667</td>
<td>0.8431</td>
</tr>
</tbody>
</table>
TABLE III. FAR AND FRR FOR DIFFERENT PASS-THRESHOLDS USING MEDIAN VECTOR PROXIMITY (33 FEATURES)

<table>
<thead>
<tr>
<th>Pass Threshold</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAR</td>
<td>0.5025</td>
<td>0.1985</td>
<td>0.0735</td>
<td>0.0147</td>
<td>0.00037</td>
</tr>
<tr>
<td>FRR</td>
<td>0.0784</td>
<td>0.2353</td>
<td>0.4314</td>
<td>0.6667</td>
<td>0.8235</td>
</tr>
</tbody>
</table>

TABLE IV. THE AVERAGE EER AND STANDARD DEVIATION EER FOR OUR PROPOSED WORK

<table>
<thead>
<tr>
<th>Algorithm</th>
<th># Features</th>
<th>EER avg.</th>
<th>EER std.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our Proposed Work</strong> (Median Vector Proximity)</td>
<td>31 Touch Keystroke Features + 2 Touch Features</td>
<td>0.1219</td>
<td>0.1337</td>
</tr>
<tr>
<td></td>
<td>31 Touch Keystroke Features</td>
<td>0.1293</td>
<td>0.1361</td>
</tr>
</tbody>
</table>

Figure 3. FAR and FRR with different Pass-Thresholds for both 31 and 33 features

Figure 4. FAR and FRR for 31 touch keystroke features with different Pass Threshold

Figure 5. FAR and FRR for 33 touch keystroke features with different Pass Threshold

Author in [5] applied his experiments on a personal computers and had got EER=8%. The remaining literature used touch screen based features. In [3, 9] authors used long text which brings more features so they had the least EER (4.5% and 7.5%). Authors in [10] used the same password in [5] but they extract touch keystroke features besides some extra touch features. They had got an EER = 15.3% and EER=12.9% for 41 and 71 features respectively. Finally, our proposed work which did not depend on a specific text (dynamic text); had got EER=12.9% for purely 31 touch features and EER=12.2 for the same features set with two extra touch features (size and pressure). Table V summarize this comparison between our proposed work and the literature.

V. CONCLUSION AND FUTURE WORK

There is an increasing demand for more secure access control in many of today's mobile applications. Touch screen is the primary input medium on the latest mobile phones [4]. The keystroke dynamics provides a natural choice for secure mobile access. Also, it is cost-effective since no extra hardware is needed; only suitable software is needed to collect keystroke timing information.

In this paper, we proposed to apply the Median Vector Proximity [5] classifier on the touch keystroke data derived from mobile’s touch screen (touchable keyboard). We evaluated the system on no specific text (dynamic text) and extracted 31 and 33 touch features. The average EER were 12.9% and 12.2% respectively. We found that the average EER was reduced by about 0.7%. Therefore, the more features we used results in more accurate systems. As a future work, we will apply different classifiers on those features and find out which classifier would give better results.
### TABLE V. COMPARISON BETWEEN OUR PROPOSED WORK AND THE LITERATURE

<table>
<thead>
<tr>
<th>Literature (Algorithm)</th>
<th>Texta</th>
<th>Applied on</th>
<th>EER avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Al-Jarrah [5] (Median Vector Proximity)</td>
<td>“tie5Roanl”</td>
<td>31 Hardware Keystroke Features</td>
<td>0.08</td>
</tr>
<tr>
<td>Nan Zheng et al [3] (Nearest Neighbor Algorithm)</td>
<td>Some specified 4-digits and 8-digits PINs</td>
<td>Touch Keystroke + other Touch Features</td>
<td>0.0455</td>
</tr>
<tr>
<td>Xuan Huang et al [9] (Statistical Approach)</td>
<td>Username=“aberta ytest” password=“aberty2011”</td>
<td>42 Touch Keystroke Features</td>
<td>0.075</td>
</tr>
<tr>
<td>Margit Antal et al [10] (Manhattan)</td>
<td>“tie5Roanl”</td>
<td>41 (Only Touch Keystroke Features)</td>
<td>0.1530</td>
</tr>
<tr>
<td></td>
<td></td>
<td>71 (Touch Keystroke Features + other Touch Features)</td>
<td>0.1290</td>
</tr>
<tr>
<td>Our Proposed Work (Median Vector Proximity)</td>
<td>Not specified (any text could be used)</td>
<td>31 Touch Keystroke Features</td>
<td>0.1219</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31 Touch Keystroke Features + 2 Touch Features</td>
<td>0.1293</td>
</tr>
</tbody>
</table>

a. The TEXT that is used in the researcher’s experiments

### ACKNOWLEDGMENT

This research has been supported by King Abdulaziz City for Science and Technology (KACST), under grant number (PGP – 35 - 444). The authors, therefore, acknowledge with thanks KACST technical and financial support.

### REFERENCES