Hindawi
Security and Communication Networks
Volume 2018, Article ID 2567463, 10 pages
https://doi.org/10.1155/2018/2567463

Research Article

Understanding Keystroke Dynamics for Smartphone Users Authentication and Keystroke Dynamics on Smartphones Built-In Motion Sensors

Hyungu Lee,[1] **Jung Yeon Hwang**,[2] **Dong In Kim**,[1] **Shincheol Lee**,[1] **Sung-Hoon Lee**,[3] **and Ji Sun Shin**[1]

[1]*Department of Computer and Information Security, Sejong University, Seoul 05006, Republic of Korea*
[2]*Electronics and Telecommunications Research Institute, Daejeon 34113, Republic of Korea*
[3]*Information Security Engineering, University of Science and Technology, Daejeon 34113, Republic of Korea*

Correspondence should be addressed to Ji Sun Shin; jsshin.sejong@gmail.com

Received 3 November 2017; Revised 19 January 2018; Accepted 13 February 2018; Published 14 March 2018

Academic Editor: Amir Anees

Personal Identification Numbers (PINs) and pattern drawing have been used as common authentication methods especially on smartphones. Such methods, however, are very vulnerable to the shoulder surfing attack. Thus, keystroke dynamics that authenticate legitimate users based on their typing manner have been studied for years. However, many of the studies have focused on PC keyboard keystrokes. More studies on mobile and smartphones keystroke dynamics are warranted; as smartphones make progress in both hardware and software, features from smartphones have been diversified. In this paper, using various features including keystroke data such as time interval and motion data such as accelerometers and rotation values, we evaluate features with motion data and without motion data. We also compare 5 formulas for motion data, respectively. We also demonstrate that opposite gender match between a legitimate user and impostors has influence on authenticating by our experiment results.

1. Introduction

As we live in the smart era, the number of smartphone users grows every year [1, 2], whereas security measures to authenticate for an owner are standstill. Pattern drawing and PIN entering are most often used, and nowadays fingerprint scanning or other biometric data scanning is also often adopted as an authentication method [3, 4]. The latter are known to be safer than the former, since simple patterns and PINs can be leaked via shoulder surfing attacks. However, users often prefer using patterns and PINs rather than fingerprint scanning because fingerprint scanning sometimes fails and should be repeated. Therefore, to provide a moderate usability, devices providing biometric data-based authentication also provide backup authentication methods such as PIN or patterns.

Keystroke dynamics has appeared to complement such problems by checking not just the numbers or patterns but also how a user types (the time speed, touch size, and so on), the so-called keystroke dynamics. By combining PIN (or pattern) and keystroke dynamics as a multifactor authentication, keystroke dynamics strengthens user authentication. Clearly, biometric data can be multifactored with PIN and pattern as a multifactor authentication. However, the biggest problem with using biometric data is that the device should securely keep the biometric data in private: leakage of biometric data of a user invalidates lifetime use of the user's biometric data as a private key. On the other hand, keystroke dynamics change as PIN or secret pattern changes. Thus, using keystroke dynamics as an authentication factor is less risky upon compromise.

In this paper, we look into keystroke dynamics focusing on smartphones that have a touchscreen and on-board motion sensors. The keystroke dynamics authentication has been substantially researched particularly on personal computer keyboard keystroke dynamics. Keystroke dynamics on smartphones can have more diverse features since a smartphone has a touchscreen. Furthermore, since 2010, many smartphones are equipped with motion sensors such as

accelerometer and gyroscope. Since 2002, researches on mobile keystroke dynamics have been studied; however, still more researches are necessary to improve performances (error rates), characterize undiscovered important features, or find better classification methods. Therefore, in this paper, we focus on smartphone keystroke dynamics with 6-digit PIN with distance-based classification. We develop application, collect user keystroke data, experiment classifications, and show our results.

Our contributions are the following. We collected user data samples and experimented keystroke dynamics using the most simple classification algorithm, distance-based algorithm. We experimented with both of Euclidean distance and Manhattan distance and obtained better performance with Manhattan distance, 7.89% EER (equal error rate). Compared to the state of the art, ours give considerably good performance. There are three previous studies that performed better than ours. However, they are not with 6-digit PIN: one is with 4-digit or 8-digit PIN [5], one is with thumbnails (images) [6], and the other is with 300 characters [7]. Our experiment and results are most relevant to real-world applications of 6-digit PIN and keystroke dynamics multifactor authentications since to certain level of security and usability, 6-digit PIN is popularly used and no method other than the PIN itself (images or characters) is required. We also discovered an interesting aspect of keystroke dynamics. Our experiment result shows that keystroke dynamics are more effective in opposite gender imposters: FAR reduces when imposters are opposite gender compared to when imposters are the same gender. We also investigate which features have influence on reducing FAR and analyze feature characteristics. Our result can be useful to applications where gender authenticity is very important, for instance, online dating or online same gender competition exam/game. This result is not about gender classification, but one observation from keystroke dynamics based user authentication study. The contribution of this result is providing further understanding of keystroke dynamics characteristics.

The rest of the paper is organized as follows. In Section 2, we discuss related work and compare former studies. In Section 3, we discuss distance-based algorithms which are used to classify a legitimate user and how the classification works. In Section 4, we explain what features were extracted and the background of our experiment. We analyze our result in detail in Section 5. Finally, we conclude our study and propose future work in Section 6.

2. Related Work

Keystroke dynamics was proposed as a user authentication first in 1975 [19] and it was started from typing rhythms of users on the computer keyboard. Since then, two studies [20, 21] also showed the possibility with good results from experiment with few subjects in 1977 and 1980, respectively.

Keystroke dynamics experiment using the computer keyboard were conducted first by Umphress and Williams in 1985 [22]. In their study, they make a reference profile of a legitimate user using mean keystroke latency and mean time interval of consecutive characters so that it distinguishes the legitimate user from others. The result the study showed was 11.7% FAR (false acceptance rate) and 5.8% FRR (false rejection rate).

From 2002 to 2006, studies about keystroke dynamics on the mobile were studied first [8, 9, 23]. Latency between pressing and releasing a key and between pressing the first key and the last key (hold time) was used as features to authenticate [9].

In 2009, Saevanee and Bhattarakosol [24] suggested keystroke dynamics using finger pressure on the touchscreen first. It showed 99% accuracy with the Probabilistic Neural Network (PNN). As Android 1.6, called "Donut," was released on September 15, 2009, more various types of features such as a size of fingertip, orientation of a device, and angle of a device were available and, thus, more studies have made progress in reducing EER (equal error rate) using them [25].

Since December 6, 2010, Android 2.3 provided data from gyroscope, rotation vector, linear accelerometer, and gravity. Thus, more features can be extracted from them. Cai and Chen [26] first made use of orientation of a device as a feature of keystroke dynamics, and they guessed key numbers by angles of x-axis (azimuth), y-axis (pitch), and z-axis (roll) and showed 71.5% accuracy. In addition, studies using both orientation and accelerometers have been researched. Xu et al. [27] guessed the enter keys and showed 88.7% accuracy which is higher than the former study, and Wu and Chen [14] showed 0.556% EER using these two features and time, pressure, and size features.

Table 1 shows the mobile keystroke dynamics studies. Mobile keystroke dynamics compares results using PIN (Personal Identification Number) or characters in experiments. At the beginning in mobile keystroke dynamics study, EER for 4-digit PIN was 11.3% [8] and 8.5% [9]. Since then, studies have been conducted on various PINs such as 6-digit, 8-digit, 10-digit, and 16-digit PINs beside 4-digit PIN. Chang et al. [13] showed EER of 23%, 21%, and 16% for 6-digit, 8-digit, and 10-digit PINs. And also Teh et al. [18] showed EER of 7.57% for 4-digit PIN and 5.49% for 16-digit PIN. Experiments using characters as well as PIN have been conducted. Starting with EER 6.9% of the study using 6 characters [9], experiment using 6 to 8 characters showed EER of 21.02% [15], using 10 characters showed EER of 0.806% [16], and using 34 characters showed EER of 9.3% [17], respectively. Later, there was a new type of study that extracts pressure and time interval from image instead of keys [6].

Classifying users, there were different approaches using distance-based classifier beside the statistical and the neural network classifier. Among experiments on 4-digit PIN, the Euclidean distance classifier showed 20% of EER [10] and the nearest neighbor distance classifier showed 3.65% of EER [5]. In one study using characters instead of PIN, it utilized various distance-based classifiers and obtained 2.2% FAR and 4.6% FRR using both the weighted Euclidean distance classifier and the array disorder method [7]. However, their result is based on 300 characters. Study comparing various distance-based classifiers, respectively, evaluated that kNN Manhattan weighted distance and kNN Manhattan scaled weighted distance were the best as 8% EER [5]. Other study comparing three distance-based classifiers, Manhattan,

Table 1: Comparison of studies for keystroke dynamics. Motion data column indicates whether features from motion data are used or not.

Authors	Year	Methodology	Motion data	Number of subjects	Number of training samples	Classifier	EER (%)
Clarke et al. [8]	2003	4-digit PIN	X	30	30	Statistical	11.3
Clarke and Furnell [9]	2007	4-digit PIN	X	30	30	Neural network	8.5
		6 alphabetic characters				Neural network	15.2
Chang et al. [6]	2012	3–6 thumbnails	X	100	5	Statistical	6.9
De Mendizabal-Vázquez et al. [10]	2014	4-digit PIN	O	80	3–9	Euclidean distance	20
Zheng et al. [5]	2014	4-digit PIN/ 8-digit PIN	O	80	80	Nearest neighbor distance	3.65/ 4.45
Samura et al. [7]	2014	300 characters (approximately)	X	43	5	Weighted Euclidean distance + array disorder	2.2 FAR, 4.6 FRR
Giuffrida et al. [11]	2014	8-9 characters	O	20	40 (approximately)	kNN ($k = 1$) Manhattan weighted, kNN ($k = 1$) Manhattan scaled weighted	8
Antal and Szabó [12]	2015	10 characters	X	42	2/3 of data	Manhattan distance	12.9
Chang et al. [13]	2016	6-digit PIN 8-digit PIN 10-digit PIN	X	100	100	Statistical	23 21 16
Wu and Chen [14]	2015	8-digit PIN	O	100	500	SVM	0.556
Buschek et al. [15]	2015	6–8 characters	X	28	-	Probabilistic modeling	21.02
Dhage et al. [16]	2015	10 characters	X	15	10	Statistical	0.806
Bond and Awad [17]	2015	34 characters	X	25	-	Neural network	9.3
Teh et al. [18]	2016	4-digit PIN 16-digit PIN	X	50/150	7	Gaussian estimation, z-score matching function, standard deviation drift	7.57 5.49

Mahalanobis, and Euclidean distance, showed that the Manhattan distance classifier was the best being 12.9 EER and it was better than others by about 3% [12].

3. Distance-Based Classification Algorithms

As a behavioral biometric authentication, keystroke dynamics authentications make use of unique rhythms and behavior when a person types keys or characters on a keyboard. For authentication, first, a template of a legitimate user is created by feeding feature data into the template, and it is used to distinguish a legitimate user from imposters. Various classifiers are employed to decide the result such as support vector machine, multilayer perceptron, K-nearest neighbor, and distance-based classifiers.

In this paper, we use distance-based classifiers and classified if it is a legitimate user or not by computing a distance between a sample and mean point of user samples.

First of all, data samples are scaled to reduce the influences from different feature scales. For distance-based classifications, we need to define distance metric to use. In our experiment, we choose Euclidean distance and Manhattan distance. We review each definition in the next.

3.1. Scaling (Preprocessing). Since there are various on-board sensors in devices, units are different depending on a sensor

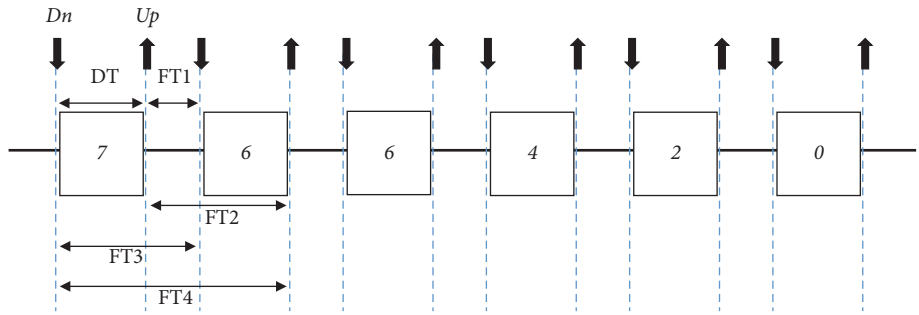


Figure 1: Feature configuration.

type and thus needed to be scaled within the same or fixed range to use as features; different units can be unexpected weighted values and cause different results. To scale various units, we use two scaling methods: the MinMax scaling and the standard scaling.

3.1.1. MinMax Scaling. The MinMax scaling is a scaling that scales data to a fixed range, 0 to 1, and it is calculated by the following equation:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{1}$$

where X is a set of data, X_{min} is the minimum value of the data, and X_{max} is the maximum of the data. X_{sc} represents the result, scaled X.

3.1.2. Standard Scaling. The standard scaling scales data, x, where the mean is 0 and the standard deviation is 1 so that the data are scaled around 0 with the standard deviation value, 1. The formula is given by

$$z = \frac{x - \mu}{\sigma} \quad (\text{where } \mu = 0,\ \sigma = 1), \tag{2}$$

where μ is the mean of x and σ is the standard deviation of x.

3.2. Distance Metrics

3.2.1. Euclidean Distance. The Euclidean distance is calculating the distance between two n-dimension vectors, $p(p_1, p_2, \ldots, p_n)$ and $q(q_1, q_2, \ldots, q_n)$, as a straight line and the formula is given by

$$\begin{aligned} d(p,q) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}. \end{aligned} \tag{3}$$

3.2.2. Manhattan Distance. The Manhattan distance calculates the distance between two n-dimension vectors, $p(p_1, p_2, \ldots, p_n)$ and $q(q_1, q_2, \ldots, q_n)$, by subtracting the values and then summing the absolute of them as follows:

$$\begin{aligned} d(p,q) &= |q_1 - p_1| + |q_2 - p_2| + \cdots + |q_n - p_n| \\ &= \sum_{i=1}^{n} |q_i - p_i|. \end{aligned} \tag{4}$$

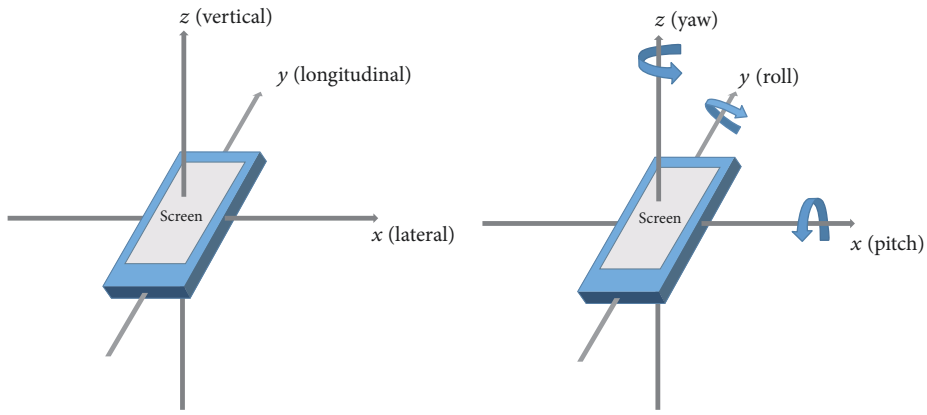
4. Features and Data Collection

Possible data extracted from smartphone are divided into two groups, keystroke data and motion data, in large. The keystroke data measured by gesture APIs that perceives touch inputs from keystroke data are "time," "size," "coordinate," and so on; "time" returns the time when events happen, "size" returns a size of a fingertip that pressed the touchscreen, "coordinate" returns coordinates of a point where a user touch. From motion sensor data, features related to movements can be extracted: accelerometer, gravity, rotation, and atmospheric pressure. In the following sections, we explain each feature.

4.1. Keystroke Data

4.1.1. Time. In "time," there are 4 types of down-time (DT) and a flight-time (FT). As Figure 1 shows, a DT is difference in time from the moment a user presses (or touches) a key (Dn) to the moment the user releases the key (Up). A FT is difference in time between pressing or releasing a key and another key; time interval between releasing a key and pressing the next key is called FT1; time interval between releasing a key and releasing the next key is called FT2; time interval between pressing a key and pressing the next key is called FT3; time interval between pressing a key and releasing the next key is called FT4. Time data, therefore, captured per 1 sample, 6-digit PIN, consist of 30-raw data in "time": 1 DT and 4 FT per 1 key.

4.1.2. Size. "Size" extracts sizes of user's fingertip each time pressing and releasing happen. Two data are captured per 1 key: one is size when pressing a key (sizeDn) and the other is size when releasing the key (sizeUp). There are thus 12-raw data in "size" for 1 sample, 6-digit PIN.



(a) Accelerometer (acc) (b) Gyroscope (gyr)

Figure 2: Axes of the motion sensors, (a) accelerometer and (b) gyroscope.

4.1.3. Coordinate. "Coordinate" extracts coordinate values for horizontal x-axis and vertical y-axis (x, y), where a key is pressed and released on the touch screen of a device. Coordinate values from pressing (xyDn) and releasing (xyUp) a key are 4-raw data so that 24-raw data of "coordinate" are extracted per 1 sample, 6-digit PIN.

4.2. Motion Sensor

4.2.1. Accelerometer (ACC). "Accelerometer" calculates device's accelerometer (m/s^2) of 3 axes, lateral x-axis, longitudinal y-axis, and vertical z-axis as Figure 2(a), by taking gravity values into account. Numbers of raw data of "accelerometer," therefore, are uneven by samples and are needed to be reshaped as regular form. Formulas and grouping the data to reshape are dealt with in Section 4.2.4 with the following "grot" and "gyr."

4.2.2. Game-Rotation (GROT). "Rotation" calculates device's angles with the geomagnetic field so that the values are influenced by the north. "Game-rotation," however, calculates the angles without any influence of the north and it means that "game-rotation" values are more accurate to measure the relative rotation. The "game-rotation" is more suitable than the "rotation" to tell person's behavior pattern, and it also returns values by 3 axes: x-axis, y-axis, and z-axis.

4.2.3. Gyroscope (GYR). "Gyroscope" measures the rate of rotation (rad/s) of a device by 3 axes, x-axis (pitch), y-axis (roll), and z-axis (yaw) as Figure 2(b). Filtering or corrections for any noise or drift are not applied in "gyroscope" data.

4.2.4. Formulas for Motion Data. As mentioned, the 3 types of motion data, "acc," "grot," and "gyr," are uneven and contain overfull data. They, therefore, need to be reshaped by some formulas. First of all, we group them by an interval between pressing and releasing a key and discard the rest of them. Then there are 5 formulas for the grouped data: average value (mean), root mean square (RMS), sum of positive values (pos), sum of positive values (neg), and standard deviation (std) [11].

In case of X array of n values, $X = \{x_1, x_2, \ldots, x_n\}$, mean ($x_{\text{mean}}$), RMS ($x_{\text{RMS}}$), pos ($x_{\text{pos}}$), neg ($x_{\text{neg}}$), and std ($x_{\text{std}}$) are defined as follows in sequence:

$$
\begin{aligned}
x_{\text{mean}} &= \frac{x_1 + x_2 + \cdots + x_n}{n}, \\
x_{\text{rms}} &= \sqrt{\frac{1}{n}\left(x_1^2 + x_2^2 + \cdots + x_n^2\right)}, \\
x_{\text{pos}} &= \sum_{i=i}^{n} x_i \quad (\text{where } x_i > 0), \\
x_{\text{neg}} &= \sum_{i=i}^{n} x_i \quad (\text{where } x_i < 0), \\
X_{\text{std}} &= \sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2}{n}} \quad (\text{where } \overline{x} \text{ is mean}).
\end{aligned}
\tag{5}
$$

4.3. Raw Data Collection and Extracted Features. As shown in Figure 3, we developed an Android application that collects raw data by entering 6-digit PIN, "766420"; the PINs were generated by considering various positions of numbers on the touchscreen following [18, 26]. We installed the app on Nexus 5X and 22 subjects (users) participated in the data collection. 100 samples were collected for each user where one sample is one time input of 6-digit PIN. As Table 2 shows, each sample consists of 6 sets of 20 features per one key; 5 "time" features (DT, FT1, FT2, FT3, and FT4), 2 "size" features (sizeDn and sizeUp), 2 "coordinate" features (XyDn and XyUp), 3 "acc" features (x, y, and z), 3 "grot" features (x, y, and z), and 3 "gyr" features (x, y, and z); thus, one sample has 120 features in total.

4.4. Error Rate. Let us first define error rates. In user authentication, there can be two types of errors, false acceptance error, and false rejection error. False acceptance error (FAR) indicates error rate of accepting an imposter user as a legitimate user. False reject error (FRR) means error rate of rejecting a legitimate user as considering him/her as an imposter.

Table 2: Types and numbers of features from the PIN, "766420."

Raw data 1 key																		
Type	Keystroke									Motion								
Feature	Time					Size		Coordinate		Acc			Grot			Gyr		
	DT	FT1	FT2	FT3	FT4	sizeDn	sizeUp	*Xy*Dn	*Xy*Up	*x*	*y*	*z*	*x*	*y*	*z*	*x*	*y*	*z*
Each #	1	1	1	1	1	1	1	2	2	Formula								
			5			2		4		1	1	1	1	1	1	1	1	1
	11 features									9 features								
Total #	20 features per 1 key 1 sample = 20 ∗ 6 = 120 features																	

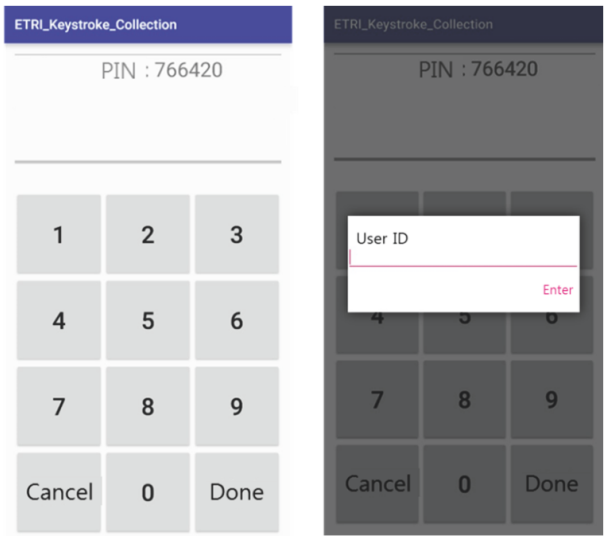


Figure 3: Application to collect user data. After entering PINs, "766420," users enter their ID on the pop-up box.

FAR tells the soundness of the system, whether it is a secure authentication mechanism or not. If FAR is high, imposter users including attackers can easily go through the authentication system. On the other hand, FRR tells the completeness of the system whether it is usable or not. If FRR is high, a user can fail in the authentication and has to retry the authentication procedure again and again. In a fuzzy data-based authentication system, there are no perfect completeness and perfect soundness since authentication factor has noise. In real applications, achieving good soundness is more important completeness because most users bear with 2-3 retrials as long as the authentication system provides expected level of security against imposters.

Depending on the threshold value distinguishing legitimate users from imposters, FAR and FRR move. Generally, reducing FAR increases FRR and vice versa. When they are equalized, we say it is EER (Equal Error Rate). EER is often used as a performance measure to show research results of fuzzy data identification/authentication. When FAR and FRR are close to EER, they approximately satisfy the relation (FAR + FRR) = 2 ∗ EER. Therefore, once EER is known, you can reduce a wanted error rate (for instance, FAR) and expect the other error rate from the relation (for instance, FRR ~ = 2 ∗ EER − FAR).

5. Experiment Results

In this section, we show our experimental results, one with distance-based classification and another with OCSVM. We present each result and their comparisons.

5.1. Distance-Based Classification. In our experiment, we use two scaling methods, MinMax scaling and standard scaling, and use two distance metrics, Euclidean distance and Manhattan distance (details are in Section 3). Our experiments are done with all four combinations of two scaling methods and two distance metrics and here we presented the best results.

We have two main results. First, our experimental result says that adding features from motion data in addition to features from keystroke data gives better performance (i.e., smaller EER). Second, our experimental result gives that keystroke authentication is more effective against opposite gender imposters. We further discuss two results in the next.

5.1.1. Adding Features from Motion Data. As discussed in Section 4.2.4, motion data are uneven and contain overfull data, while other keystroke data such as time, size, and coordinate are atomic and easily transformed into a feature. Thus, to shape motion data into a feature, there are five formula ways: "mean," "root mean square," "positive sum," "negative sum," and "standard deviation." First, we experiment with 5 different formulas featuring motion data. We all tried 4 combinations of MinMax scaling or standard scaling, and Euclidean distance or Manhattan distance. We obtained the best result with standard scaling and Manhattan distance and present the detailed result next.

Figure 4 shows the result of the averaged EER for 5 formulas, respectively, and EER of the "mean" formula, 12.63%, is the lowest rate. We also experimented with inclusion of all 5 formulas. Still, the experiment with including only "mean" formula gave the best result.

As seen in Figure 5, when "mean" formula of motion data is added, the error rate improved compared to the experiment using features only from keystroke data (time, size, and coordinate). Using the keystroke data only, we obtained 8.94% EER and 7.89% EER by adding the motion

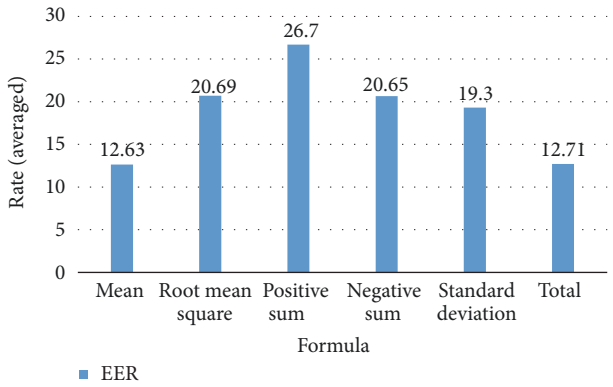


Figure 4: Comparison of formulas for motion data only. "Total" means using all of the 5 formulas as features.

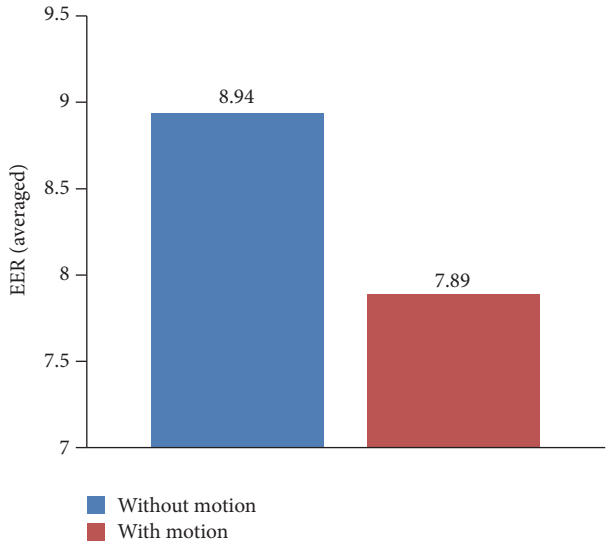


Figure 5: Rate change by adding motion data. Motion data are calculated by "mean" formula.

data. The EER decreased by 1.05% when the motion data were added.

5.1.2. FAR by Gender Match. Our experimental result shows that opposite gender's match influences the result, especially on FAR. If gender of a legitimate user and gender of impostors are different, the experiment result gives lower FAR.

In Figure 6, we draw a line between male and female by a legitimate user's gender and compare three cases, same gender, opposite gender, and total case, which is irrelevant to gender. In Figure 6, dotted bars show the test results in case of the keystroke data only and solid bars show the test results when motion data is also used. Regardless of the case, the result shows lower FAR if the gender between a legitimate user and imposters is not matched. FAR decrease by 4.07% and 0.64% in case of opposite gender for male legitimate users and female legitimate users, respectively. Figure 6 and the rest of the results, Figures 7 and 8, are based on the results from experiments with standard scaling and Manhattan distance, which gave the best result.

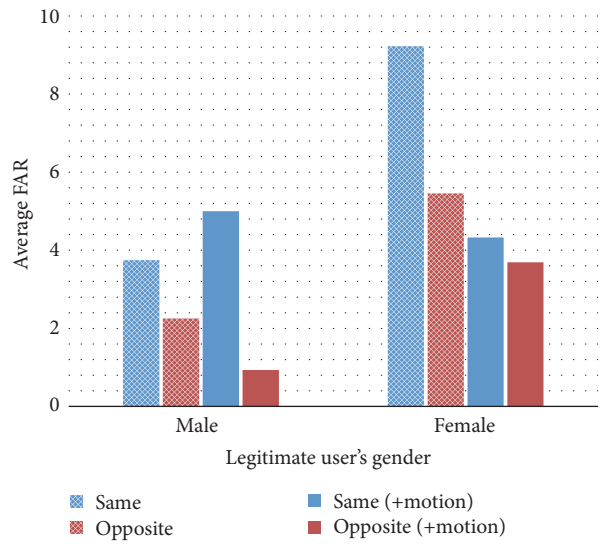


Figure 6: Average FAR by gender match. By gender match, opposite or the same, we compare keystroke data and keystroke data with motion data.

Figure 7 shows how much each feature influences FAR by gender's match. For better vision, we also provide Figure 8 showing differences of error rate between the cases with the same gender imposters and opposite gender imposters. From Figure 8, we can see that the feature "grot" ("game-rotation" that returns rotation values without the geomagnetic field) and the feature "sizeDn" (a size when pressing a key) reduce FAR the most upon standard scaling. The biggest decrease on FAR is 22.73% from "grot" and "sizeDn" and "acc" comes next as 7.08% (7.28%) and 4.98% (5.16%). This result determining features strong against opposite gender imposters can be very useful in applications where gender authenticity is critical, for instance, online dating, or online same gender competition exam/game.

5.2. One-Class Support Vector Machine (OCSVM). Support vector machines (SVMs) are one of the machine learning algorithms which is supervised and used as classification method, regression method, or outliers detection. Using labeled training data, SVMs find optimal hyperplane and it is used to differentiate unlabeled data. In SVMs, a kernel transforming data into higher dimensional data is used if data are not linearly separate. There are various kernels such as linear, polynomial, radial basis function, and sigmoid in kernel function [28].

Among SVMs, there are one-class support vector machines (OCSVMs) which are unsupervised learning and determine whether new data is outlier or not. Using the OCSVM, we compare with the same conditions that one is using keystroke data only and the other is adding motion data. As previous result with distance-based classifier shows, EER decreases by about 1.24% using the OCSVM as well as when the motion data were added as Figure 9 shows. Repeating the same experiment about gender match on the OCSVM, FAR decreasing is more obvious as shown in Figure 10. In case of male legitimate users, FARs decline by

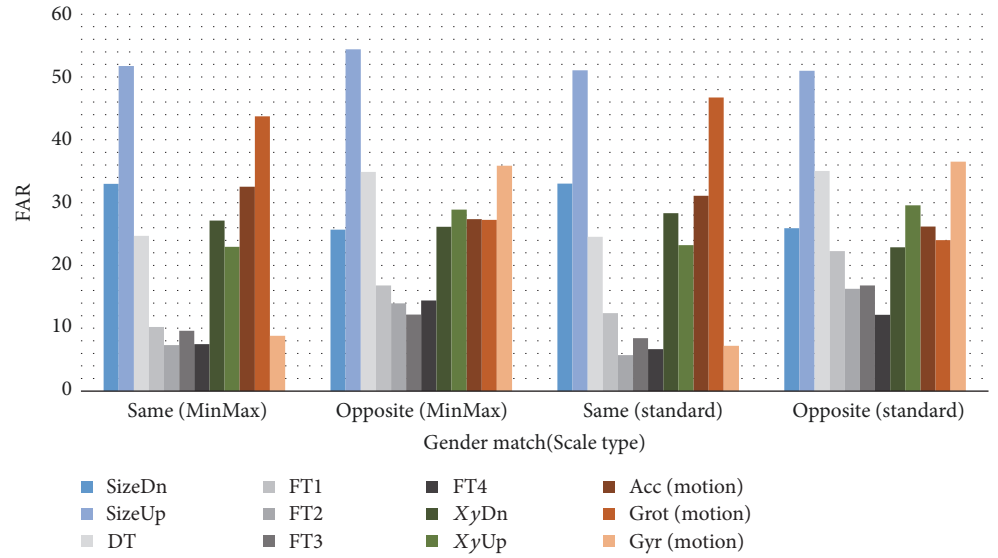


Figure 7: FAR change of single feature by gender match.

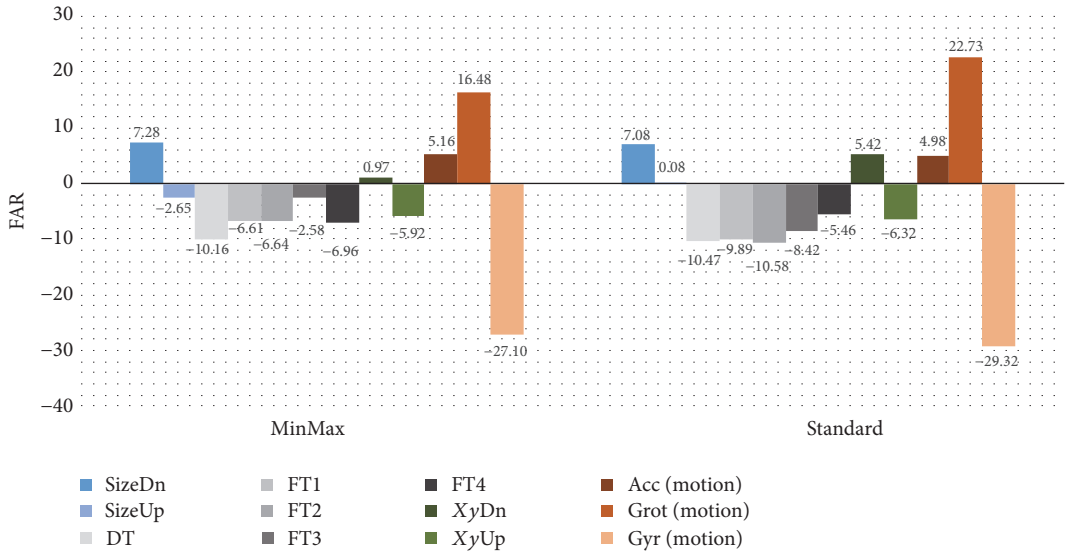


Figure 8: Visualization of differences between FAR upon experiment with the same gender and FAR upon experiment with the opposite gender imposters (i.e., the former minus the latter).

about 8% and the other case's FARs show about a 6% drop irrespective of data types.

6. Conclusion

Recently, new authentication methods on smartphones using biometrics information such as iris scan and face recognition are rising. However, the existing methods including PINs are still often used. Many studies for the keystroke dynamics are in progress to strengthen PIN-based authentication. In our study, we include features from motion data to see how they are effective in keystroke dynamics authentication. We show which formula is better to handle motion data and that keystroke authentication improves when features from motion data are added. We obtained the best result with motion data using "mean" formula and obtained 7.89% EER, which is the best performance so far upon 6-digit with distance-based classification. We also showed an interesting result that gender's match has influenced FAR and found features, "grot" ("game-rotation" that returns rotation values without the geomagnetic field) and "sizeDn" (a size when pressing a key) that influence the most. We believe our results will contribute to better understanding of keystroke dynamics authentication and to future study and development of 6-digit distance-based keystroke dynamics. In particular, our result by gender match can be adopted to applications where gender authenticity is crucial, for instance, online dating, or online same gender competition exam/game.

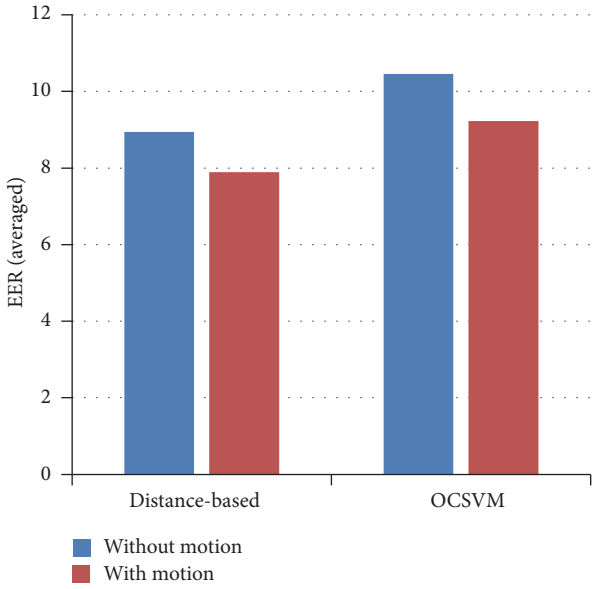


Figure 9: Rate change by adding motion data on SVMs.

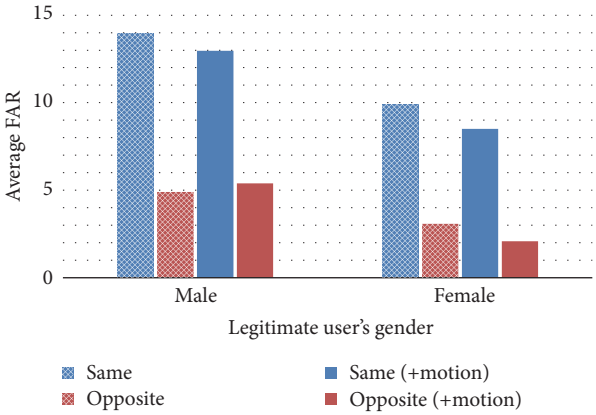


Figure 10: Average FAR by gender match. By gender match, opposite or the same, we compare keystroke data and keystroke data with motion data.

For the future work, studies for additional authentication step focusing on specific features that are strengthened to opposite gender are needed to be investigated. Also, we will continue keystroke dynamics research with different preprocessing approaches and classification methods and also to find more appropriate combinations of features reducing ERR.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (no. 2015-0-00168, Development of Universal Authentication Platform Technology with Context-Aware Multifactor Authentication and Digital Signature, and no. 2016-0-00097, Development of Biometrics-Based Key Infrastructure Technology for Online Identification).

References

[1] Statista, "Number of smartphone users in the U.S. 2010-2022," https://www.statista.com/statistics/201182/forecast-of-smartphone-users-in-the-us/.

[2] Statista, Number of smartphone users in South Korea from 2015 to 2022, https://www.statista.com/statistics/467171/forecast-of-smartphone-users-in-south-korea/.

[3] N. Ben-Asher, N. Kirschnick, H. Sieger, J. Meyer, A. Ben-Oved, and S. Möller, "On the need for different security methods on mobile phones," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 465–473, ACM, Stockholm, Sweden, September 2011.

[4] P. K. Sari, G. S. Ratnasari, and A. Prasetio, "An evaluation of authentication methods for smartphone based on users' preferences," in *Proceedings of the IOP Conference Series: Materials Science and Engineering*, vol. 128, IOP Publishing, Bristol, UK, 2016.

[5] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: user verification on smartphones via tapping behaviors," in *Proceedings of the IEEE 22nd International Conference on Network Protocols (ICNP '14)*, pp. 221–232, IEEE, North Carolina, NC, USA, October 2014.

[6] T.-Y. Chang, C.-J. Tsai, and J.-H. Lin, "A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices," *The Journal of Systems and Software*, vol. 85, no. 5, pp. 1157–1165, 2012.

[7] T. Samura, M. Izumi, and H. Nishimura, "Flick input authentication in Japanese free text entry on smartphones," in *Proceedings of the 53rd Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE '14*, pp. 1348–1353, 2014.

[8] N. L. Clarke, S. M. Furnell, B. M. Lines, and P. L. Reynolds, "Keystroke dynamics on a mobile handset: a feasibility study," *Information Management and Computer Security*, vol. 11, no. 4, pp. 161–166, 2003.

[9] N. L. Clarke and S. M. Furnell, "Authenticating mobile phone users using keystroke analysis," *International Journal of Information Security*, vol. 6, no. 1, pp. 1–14, 2007.

[10] I. De Mendizabal-Vázquez, D. De Santos-Sierra, J. Guerra-Casanova, and C. Sánchez-Ávila, "Supervised classification methods applied to keystroke dynamics through mobile devices," in *Proceedings of the 48th Annual IEEE International Carnahan Conference on Security Technology, ICCST '14*, pp. 1–6, October 2014.

[11] C. Giuffrida, K. Majdanik, M. Conti, and H. Bos, "I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics," in *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, vol. 8550, pp. 92–111, Springer, Cham, Switzerland, 2014.

[12] M. Antal and L. Z. Szabó, "Keystroke Dynamics on Android Platform," *Procedia Technology*, vol. 19, pp. 820–826, 2015.

[13] T.-Y. Chang, C.-J. Tsai, W.-J. Tsai, C.-C. Peng, and H.-S. Wu, "A changeable personal identification number-based keystroke

dynamics authentication system on smart phones," *Security and Communication Networks*, vol. 9, no. 15, pp. 2674–2685, 2016.

[14] J. Wu and Z. Chen, "An implicit identity authentication system considering changes of gesture based on keystroke behaviors," *International Journal of Distributed Sensor Networks*, vol. 11, no. 6, Article ID 470274, 2015.

[15] D. Buschek, A. De Luca, and F. Alt, "Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices," in *Proceedings of the 33rd Annual CHI Conference on Human Factors in Computing Systems, CHI '15*, pp. 1393–1402, April 2015.

[16] S. Dhage, P. Kundra, A. Kanchan, and P. Kap, "Mobile authentication using keystroke dynamics," in *Proceedings of the International Conference on Communication, Information and Computing Technology, ICCICT '15*, IEEE, Mumbai, India, January 2015.

[17] W. Bond and A. E. A. Awad, "Touch-based static authentication using a virtual grid," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pp. 129–134, 2015.

[18] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "TDAS: a touch dynamics based multi-factor authentication solution for mobile devices," *International Journal of Pervasive Computing and Communications*, vol. 12, no. 1, pp. 127–153, 2016.

[19] R. Spillane, "Keyboard apparatus for personal identification," IBM Technical Disclosure Bulletin, vol. 17, no. 3346, 1975.

[20] G. E. Forsen, M. R. Nelson, and R. J. Staron, "Personal attributes authentication techniques," Tech. Rep. RADC-TR-77-333, Rome Air Development Center, New York, NY, USA, 1977.

[21] R. S. Gaines, W. Lisowski, S. J. Press, and N. Shapiro, "Authentication by keystroke timing: some preliminary results," Tech. Rep. R-2526-NSF, RAND Corporation, California, Calif, USA, 1980.

[22] D. Umphress and G. Williams, "Identity verification through keyboard characteristics," *International Journal of Man-Machine Studies*, vol. 23, no. 3, pp. 263–273, 1985.

[23] N. L. Clarke, S. M. Furnell, B. M. Lines, and P. L. Reynolds, "Subscriber authentication for mobile phones using keystroke dynamics," in *Proceedings of the 3rd International Network Conference (INC '02)*, 2002.

[24] H. Saevanee and P. Bhattarakosol, "Authenticating user using keystroke dynamics and finger pressure," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference, CCNC '09*, IEEE, Nevada, Nev, USA, January 2009.

[25] M. Trojahn and F. Ortmeier, "Biometric authentication through a virtual keyboard for smartphones," *International Journal of Computer Science and Information Technology*, vol. 4, no. 5, 2012.

[26] S. Zahid, M. Shahzad, S. A. Khayam, and M. Farooq, *Keystroke-Based User Identification on Smart Phones*, Springer, Berlin, Germany, 2009.

[27] Z. Xu, K. Bai, and S. Zhu, "TapLogger: inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 113–124, ACM, Arizona, Ariz, USA, April 2012.

[28] Wikipedia, Support vector machine, https://en.wikipedia.org/wiki/Support_vector_machine.