Keystroke Biometric Studies with Short Numeric Input on Smartphones

Michael J. Coakley[1], John V. Monaco[2], and Charles C. Tappert[1]
[1]*Seidenberg School of CSIS, Pace University, Pleasantville, NY 10570*
[2]*U.S. Army Research Laboratory, Aberdeen, MD 21005*
{mcoakley@pace.edu, john.v.monaco2.ctr@mail.mil, ctappert@pace.edu}

Abstract

A keystroke biometric system was extended to accommodate touchscreen keystroke features. In addition to the usual key-press and key-release timing features available from mechanical keyboards, the touchscreen features also included pressure, screen location, accelerometer, and gyroscope information. Short numeric input data, ten-digit numeric strings, were collected from 52 participants on identical Android smartphones. Several user authentication biometric experiments were performed on these data to measure overall system performance and to quantify the biometric value of various feature subsets. Two validation procedures (repeated random sub-sampling and leave-one-out cross-validation) and two distance metrics (Euclidean and Manhattan) were compared in the study, and the best results were compared to previous keystroke biometric studies. The best equal-error-rate performances on the key-press/release timing features, the touchscreen features, and all features combined were 19.7%, 4.0%, and 3.9%, respectively. Of the various touchscreen feature subsets, the gyroscope features performed best with an equal error rate of 4.3%

1. Introduction

This study concerns keystroke biometrics (also referred to as keystroke dynamics) and how they can be leveraged to provide user authentication on a mobile phone. Keystroke dynamics are a behavioral biometric which utilize typing characteristics, rhythms, and cadence believed to be unique to individuals. These characteristics are also believed to be difficult to mimic or replicate [6, 9, 12].

There has been much research associated with keystroke biometrics on traditional mechanical keyboards. Killourhy and Maxion of Carnegie Mellon University conducted several short-string keystroke biometric studies on mechanical keyboards. In one of their studies, they utilized a ten character numeric string, similar to a phone number, to determine whether users could be authenticated based on the keystroke biometric data collected on a numeric keypad [7]. Using a random forest classifier, they achieved an equal error rate (EER) of 8.6%.

At Pace University, Bakelman et al. extended the research done by Maxion and Killourhy and conducted keystroke biometric experiments of their own utilizing a similar short sting numeric input in conjunction with a classification algorithm application created at Pace University [1]. Bakelman's results were not as robust as the results obtained by the team at Carnegie Mellon when used in conjunction with the features used in the Carnegie Mellon study, obtaining an EER of 10.5%. However, when Bakelman and team ran their experiments using their own feature data, they recorded an EER of 6.1%.

Whereas a significant amount of research has been devoted to keystroke biometrics on traditional hardware keyboards, utilizing mechanical-keyboard-like keystroke biometrics on mobile phones has only recently begun to garner attention. Buchoux and Clarke conducted several mechanical-keyboard-like keystroke biometric studies on Smartphones. In one study, they acknowledged the results they achieved were relatively poor, with EERs ranging from 38-42% (estimated from graphs in their paper). However they remained optimistic that performance would improve if the proper algorithms were used [3]. A subsequent study evaluated the use of different statistical classifiers on mechanical-keyboard-like keystroke biometric data extracted from a smartphone running the Microsoft Windows Mobile 5 Operating System [2].

Fleming leveraged long text input on a smartphone in an attempt to use mechanical-keyboard-like keystroke biometrics to accurately identify users. Using key press and release durations, as well as bigram and trigram transition features, he was able to accurately identify users with 70% accuracy using a *k*-nearest-neighbor (kNN) classifier [5].

Whereas mechanical-keyboard-like keystroke biometric features primarily consist of time intervals between key-press and key-release events, touchscreen keystroke biometric features can utilize the sensors associated with the touchscreen of the smartphone or tablet. These measurements can be captured and extracted via the many sensors that come bundled in today's smart devices and are not readily accessible using traditional hardware computer keyboards. Characteristics such as pressure used to strike a key, size of the finger used for key compression, finger

location, and finger orientation can be collected on a wide variety of modern smartphones and leveraged as a behavioral biometric [11].

Sougata Sen from Singapore Management University conducted experiments where they attempted to authenticate users by measuring the pressure of the key pressed on the screen, as well as the duration of each press. Experimenting with several different classification algorithms, the research team observed a top EER of 15.2% [10].

2. Methodology

2.1 Data Collection

Data samples were collected from 52 participants using five identical Android LG-D820 Nexus 5 mobile devices. The participants were selected as a convenience sample of computer users, roughly two-thirds from undergraduate students taking an introductory computing course and one-third from a department of government working professionals

All participants were tasked with entering the same 10-digit string, **914 193 7761**, as they would type a phone number on a mobile device. Similar to the study by Killourhy and Maxion [6], this 10-digit number was chosen for several reasons. It is longer than most passcodes, such as the 4-digit string commonly used for screen lock, yet short enough for easy entry by the participants. The same number was used by all participants to avoid bias and to allow each participant's data to be used as imposter data for the other participants. The particular digit sequence was chosen due to participant familiarity with the local telephone area code "914". The sequence was also designed to span the numeric touchscreen keypad, in particular the subsequence "1937".

On two separate occasions, separated by several weeks, each participant was tasked with entering the digit string 30 times after a brief practice session of 10 repetitions. Only correctly-entered sequences were retained for experiments, i.e., samples containing errors or more than 22 events (11 key-press + 11 key-release) were discarded. The resulting dataset contained 2,236 samples with 43 +/- 16 samples per user. After each data collection session, data on each of the Android devices were saved locally to the device, and then transmitted to a centralized server. Figure 1 provides an overview of the data collection process.

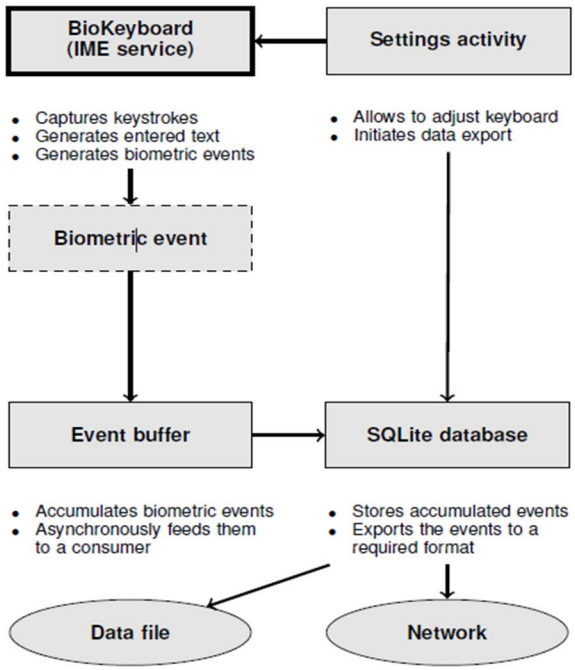

Figure 1. Data Collection and Transmission.

The default keyboard application that comes equipped on the Android devices does not provide for keystroke capturing, so a customized keyboard application had to be developed and installed on each of the five Android mobile devices utilized in this study. The BioKeyboard referred to in Figure 1 is a custom Android keyboard application that captures the key-press and key-release events and transmits the recorded data to a remote server. Android provides specific Application Program Interfaces (APIs) for creating Input Method Editors (IMEs), the mechanism through which text, numbers, or symbols are displayed and typed by the user [4]. A customized keyboard application is such a control. The IME used in this study provided the visual interface our subjects needed to see and enter the required string of characters (in the form of a numeric keypad), and additionally recorded the values of each smartphone sensor at the time of each key press or release event. The data associated with each event include:

1. The key that was pressed or released
2. Keyboard layout (character, numeric, or symbol)
3. Screen orientation (portrait or landscape)
4. Type of action (press or release)
5. Time of the event (milliseconds)
6. Screen coordinates of the event (pixels)
7. Touchscreen pressure applied by the finger
8. The size and orientation of an ellipse covering the area touched

In addition to the eight data elements listed above, the IME used in this study recorded accelerometer and gyroscopic sensor values at the time of each event. The accelerometer measures acceleration (m/s^2) along three orthogonal axes, and the gyroscope measures rotational velocity (rad/s) along three orthogonal axes. The gravity component was not removed from the accelerometer sensor values, therefore the orientation of the phone could be determined from acceleration along the 3 orthogonal axes. From these sensor values, we consider two types of features in addition to those described above:

9. Current acceleration/rotational velocity
10. Change in acceleration/rotational velocity since the last measurement

Since the accelerometer and gyroscopic sensors operate independently of the IME, updating at a rate of 5Hz under a normal delay, each event recorded by the BioKeyboard IME contains the sensor values sampled at the most recent time step before each press or release event. The BioKeyboard is available for external use at: https://bitbucket.org/pacebiometrics/android-biokeyboard, and the dataset collected in this study is available upon request.

2.2 Feature Extraction

Mobile device touchscreens are capable of capturing much more information than traditional mechanical keyboards. To evaluate whether the additional sensors provide greater verification capability, we define three sets of features: timing, touchscreen and combined timing and touchscreen.

The timing features consist of durations of each soft mechanical-keyboard-like keystroke, as well as the intervals between each key-press/key-release event, for a total of 31 total timing features. These are the same features used in [7].

The touchscreen (non-timing) features consist of the position, shape, accelerometer, and gyroscope sensor values recorded at each press and release event. Additionally, this feature set also contains the difference between each press/release event value for each key as well as the difference of each measurement between consecutive press-release and release-press events. Altogether there are 583 touchscreen features.

The combined feature set is simply the concatenation of timing and touchscreen features. All features were normalized into the range of 0-1 by restricting each feature at +/- 2 standard deviations. The complete feature set is summarized in Appendix A.

2.3 Authentication algorithm

Our study utilized a dichotomy model which transforms a multi-class problem into a two-class problem. This is accomplished by calculating the differences between the feature attributes collected and then utilizing those differences in order to classify a query sample as being "within-class" (genuine) or "between-class" (impostor) [8]. The dichotomy model implements a k-nearest-neighbor classifier that classifies the distances from a query sample to a template as being either within-class or between-class.

Two validation approaches were utilized in this study: repeated random subsampling (RRS) and leave-one-out cross-validation (LOOCV). The RRS is computationally efficient for larger datasets, as only random sets of samples are utilized in each fold. LOOCV is less efficient, however it maximizes the amount of training data.

The RRS is a Monte Carlo validation procedure and typically has lower variance and higher bias compared to LOOCV [11]. Due to the small size of each fold in LOOCV, a higher variance is typically observed [12].

In the RRS procedure, a random subset of 5 samples from each user is selected as the query samples, and the remainder of the dataset is used as the reference. The EER is obtained by the dichotomy classifier, which compares the distances between a query sample and each template with the known within-class and between-class distances. A decision is made using a linear-weighted kNN classifier by varying the distance threshold as described in [8]. This process is repeated 30 times to obtain a confidence interval for the EER. The LOOCV is a standard p-fold cross validation with $p = 1$, i.e., in every fold a single sample is used as the query and the remaining samples are used as the reference. In both procedures, k was set to 21 in the kNN classifier.

In addition to the two different validation procedures (RRS and LOOCV), two different distance metrics (Euclidean and Manhattan distances) were employed with the kNN classification algorithm.

3. Experimental results

The EER was obtained for seven different feature sets using the two validation procedure (LOOCV and RRS) and the two distance metrics (Euclidean and Manhattan), for a total of 28 experiments. The first set of experiments compared the three different feature types (keystroke, touchscreen, and combined) and the second set of experiments compared the four different sensor types (pressure, location, accelerometer, and gyroscope).

3.1 Feature Type

The feature types are broken down into keystroke (timing features only), touchscreen (mobile sensor features only), and combined (both timing and sensor features). These results are shown in Table 1. The timing features contain the key-hold durations, press-press latencies, and release-press latencies. The touchscreen features contain the non-timing features, including screen coordinates, pressure,

accelerometer, and gyroscopic features, as described in Appendix A. The combined features are simply the timing features and the touchscreen features concatenated into a single feature file. The ROC curves of each feature type obtained under each validation procedure and distance metric are shown in Figures 2 and 3, respectively.

	Euclidean			Manhattan	
	RRS	LOOCV		RRS	LOOCV
Keystroke	23.0%	20.0%		22.6%	19.7%
TouchScreen	13.8%	4.9%		11.8%	4.0%
Combined	14.9%	7.1%		11.5%	3.9%

Table 1- Feature Type EERs

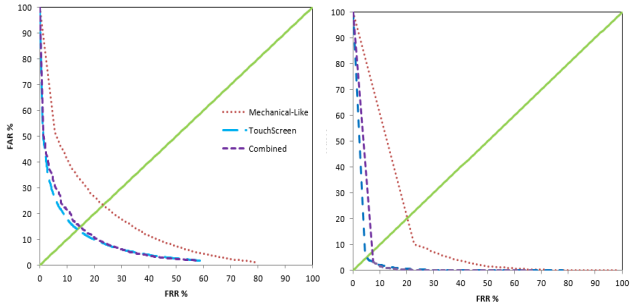

Figure 2. Feature type ROC curves using Euclidean distance, and RRS (left) and LOOCV (right) procedures.

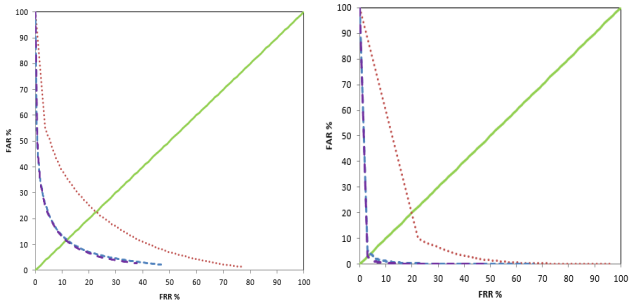

Figure 3. Feature type ROC curves using Manhattan distance, and RRS (left) and LOOCV (right) procedures.

As these results indicate, the traditional keystroke timing features performed poorly compared to both the touchscreen features and the combined features, regardless of the distance metric or cross validation procedures used. The results associated with the touchscreen and the combined features, however, were quite robust, particularly when used in conjunction with the Manhattan distance metric and the Leave-One-Out Cross-Validation procedure.

3.2 Sensor type

To compare the capability of each sensor type, the touchscreen features are broken down into their respective sensor categories. Table 2 illustrates the comparative results of the four touchscreen feature types - pressure, location, accelerometer, and gyroscopic. The Manhattan distance ROC curves of each sensor type are shown in Figure 4 (RRS) and Figure 5 (LOOCV).

	Euclidean			Manhattan	
	RRS	**LOOCV**		**RRS**	**LOOCV**
Pressure	27.0%	26.4%		24.9%	24.3%
Location	19.0%	18.3%		17.9%	15.0%
Accelerometer	13.0%	11.2%		11.3%	8.9%
Gyroscope	10.6%	8.6%		8.1%	4.3%

Table 2-Sensor type EERs

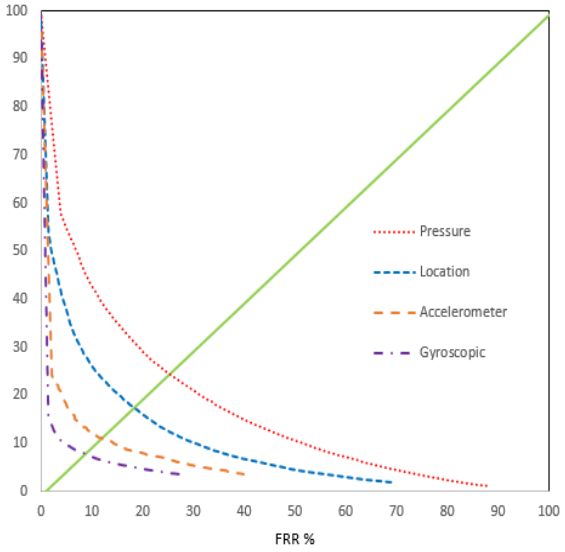

Figure 4. Sensor type ROC curves (RRS and Manhattan).

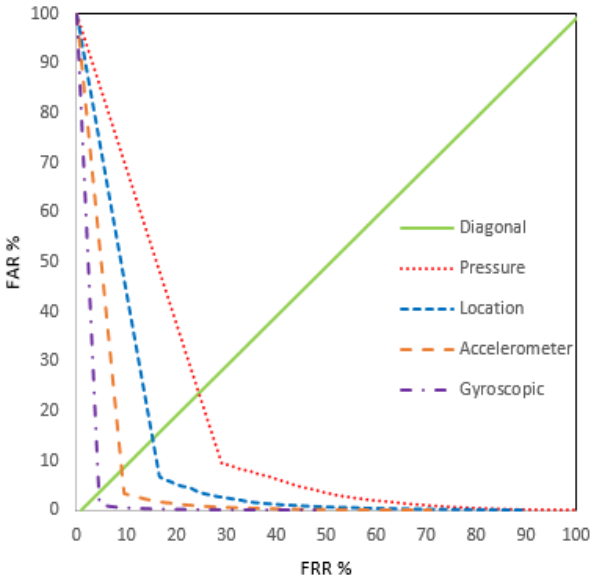

Figure 5. Sensor type ROC curves (LOOCV and Manhattan).

The results in Table 2 clearly indicate that the pressure features returned the weakest results, with a best EER of

24.3% with the Manhattan distance metric and the Leave-One-Out Cross Validation procedure. The results associated with gyroscopic features, on the other hand, were much more robust, returning a best EER of 4.3% when using the Manhattan distance metric in conjunction with LOOCV. In fact, when comparing the results in Tables 1 and 2, it is clear that the gyroscopic sensor features alone were comparable with the best results associated with the combined features in the first phase of the experiments.

4. Conclusions

The main goal of this study was to analyze the authentication performance of the various types of text-entry biometric information that can be extracted from today's smartphones. The performance of the traditional timing key-press/key-release features was considerably worse in this study (about 20% EER) compared to performances reported in earlier studies on mechanical keyboards (e.g., 8.6% EER by Killourhy and Maxion [6] and 6.1% EER by Bakelman et al. [1]). On the other hand, the excellent results achieved by the touchscreen features (4.0% EER) and the gyroscope only features (4.3% EER) appears to offer a compelling approach towards user authentication on smartphones.

Two different validation methods (RRS and LOOCV) were compared in the study and, as expected, the LOOCV method provided better results than RRS but at the expense of greater computation time. Of the two distance metrics compared, the Manhattan distance outperformed Euclidean distance consistently throughout all of the experiments, regardless of the feature set used.

It is interesting to observe that the traditional timing key-press/key-release features, those available on mechanical keyboards, provide weaker performance results as the form factor (size of the keys) decreases from desktops to laptops and now to the touchscreens of handheld devices. In addition to the form factor, the feel of individual keys on mechanical keyboards apparently fosters more consistent and repeatable keystroke dynamics compared to the flat, somewhat slippery feel of the virtual keyboards on touchscreens. This is an area meriting further investigation.

References

[1] N. Bakelman, J. V. Monaco, S.-H. Cha, and C. C. Tappert, "Keystroke Biometric Studies on Password and Numeric Keypad Input," *2013 Eur. Intell. Secur. Informatics Conf.*, pp. 204–207, Aug. 2013.

[2] A. Buchoux and N. L. Clarke, "Deployment of keystroke analysis on a Smartphone," *Proc. 6th Aust. Inf. Secur. Manag. Conf.*, Dec. 2006, pp. 29–39.

[3] A. Buchoux and N. L. Clarke, "Smartphone Deployment of Keystroke Analysis," *Adv. Network, Comput. Commun. 6*, pp. 190–197, 2008.

[4] "Creating an Input Method," *Android Developers Console*. [Online]. Available: http://developer.android.com/guide/topics/text/creating-input-method.html.

[5] S. Fleming, "Identification of a Smartphone User Via Keystroke Analysis", *Naval Postgraduate School Dissertation,* March 2014.

[6] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies," *Communications of the ACM*, vol. 33, no. 2. pp. 168–176, 1990.

[7] R. A. Maxion and K. S. Killourhy, "Keystroke biometrics with number-pad input," *Proc. Int. Conf. Dependable Syst. Networks*, pp. 201–210, 2010.

[8] J. V. Monaco, N. Bakelman, S.-H. Cha, and C. C. Tappert, "Recent Advances in the Development of a Long-Text-Input Keystroke Biometric Authentication System for Arbitrary Text Input," *2013 Eur. Intell. Secur. Informatics Conf.*, pp. 60–66, Aug. 2013.

[9] K. Revett, *Keystroke Dynamics*, Wiley, 2008.

[10] S. Sen and K. Muralidharan, "Putting 'pressure' on mobile authentication," *2014 7th Int. Conf. Mob. Comput. Ubiquitous Networking, ICMU*, pp. 56–61, Jan. 2014.

[11] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors," *Tech. Rep. WM-CS-2012-06*, 2012.

[12] Y. Zhong and Y. Deng, "A Survey on Keystroke Dynamics Biometrics: Approaches, Advances, and Evaluations," chapter 1 in *Recent Adv. User Authentication Using Keystroke Dyn. Biometrics*, vol. 2, pp. 1–22, 2015.

[13] W. Dubitzky, M. Granzow, and D. Berrar. *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media. p. 178, 2007.

[14] R. Kohavi. "A study of cross-validation and bootstrap for accuracy estimation and model selection." IJCAI. Vol. 14. No. 2, 1995.

Appendix A.

No.	Timing features
11	Durations (press to release of each key)
10	Press-press latencies
10	Release-press latencies

No.	Touchscreen features
11	Press pressure
22	Press {x, y} screen location
22	Press {major, minor} ellipse
33	Press {x, y, z} acceleration
33	Press {x, y, z} rotation
11	Release pressure
22	Release {x, y} screen location
22	Release {major, minor} ellipse
33	Release {x, y, z} acceleration
33	Release {x, y, z} rotation
11	Press-release (same key) pressure difference
22	Press-release (same key) {x, y} screen location differences
22	Press-release (same key) {major, minor} ellipse differences
33	Press-release (same key) {x, y, z} acceleration differences
33	Press-release (same key) {x, y, z} rotation differences
10	Press-press transition pressure difference
20	Press-press transition {x, y} screen location differences
20	Press-press transition {major, minor} ellipse differences
30	Press-press transition {x, y, z} acceleration differences
30	Press-press transition {x, y, z} rotation differences
10	Release-press transition pressure difference
20	Release-press transition {x, y} screen location differences
20	Release-press transition {major, minor} ellipse differences
30	Release-press transition {x, y, z} acceleration differences
30	Release-press transition {x, y, z} rotation differences

Table 3 –Mobile keystroke feature descriptions and counts. There are 33 timing features and 583 touchscreen features.