Keystroke Dynamics

Jason Ligon and Abhishek Verma

CONTENTS

15.1 EXISTING BENCHMARK DATASETS FOR KEYSTROKE DYNAMICS

Unlike other biometric modalities, there are only a few public datasets available for keystroke dynamics, some of which are composed of several subdatasets. Giot et al. [1] compiled a list of current public databases for keystroke dynamics using a keyboard in collecting the data. The data provided by these, though not mobile, can be extended into mobile in the future.

15.1.1 GREYC

GREYC is an important public dataset due to its large number of users. The dataset contains 133 users with 100 of them providing samples of at least five distinct sessions. The majority of these sessions were spaced out for at least 1 week. All of the users typed the password "greyc laboratory"

12 times on two distinct keyboards per session, thus giving 60 samples for the 100 users that participated in each session. Both the extracted features (hold time and latencies) and raw data are available and allow computing of other extracted features. This data are stored in a SQLite dataset file.

15.1.2 WEBGREYC{A,B}

Another important public dataset is WEBGREYC due to its number and length of sessions. A total of 118 users' keystroke samples were acquired once a week for 18 months. The maximum number of sessions per user was 47 sessions. The importance of this dataset is that it contains two kinds of samples:

1. An imposed login and password (typical of the usual keystroke dynamic dataset samples) (WEBGREYCA).

2. Login and password information were chosen by the user (WEBGREYCB). It is worth nothing that several imposters were also asked to type this.

This is the first public dataset where each user designates their own password. It, therefore, provides the most realistic scenario. The dataset is stored in a set of text files containing both raw and extracted features.

15.1.3 DSN2009

A significant dataset in terms of *number of samples per user* is the DSN2009. The dataset contains 51 users, each providing 400 samples captured in eight sessions (50 inputs per session). Each session has a 1 day delay. Though this dataset has a large number of samples per user, the samples have been captured in a relatively short-time period. Each data has been captured typing the password ".tie5Roan." The dataset contains other extracted features such as: hold time, interval between two pressures, interval between the release of a key, and pressure of the next one. The dataset is stored in a raw text, CSV, and Excel files.

15.1.4 PRESSURE{0,1,2}

A public keystroke dynamic dataset called PRESSURE{0,1,2} uses a pressure sensitive keyboard to collect data. The keyboard embeds the following raw data: key code, time when pressed, time when released, and pressure force. There were a total of 104 users on the dataset, but only seven provided a significant amount of data (89–504 samples). The other users

have only provided 3–15 samples. Three different keystroke samples were used: "pr7q1z" (PRESSURE2), "jeffrey allen" (PRESSURE1), and "drizzle" (PESSURE3). The dataset is available in a CSV or SQL file.

15.1.5 BIOCHAVES{A,B,C}

Created by the biochaves team, this database consists of three subdatasets (A, B, C) for static-text keystrokes and one dataset (D) for free text. The maximum number of 15 users in a dataset provided 10 samples each. Each dataset contained a unique group of users. The dataset is composed of a couple of ASCII codes of the pressed key and elapsed time since the last key down event. The release of a key, however, is not tracked. Each dataset is stored in raw text files.

15.1.6 Keystroke 100

Keystroke 100 contains keystroke latency and keystroke pressure. The dataset also contains keystroke patterns of users typing the password "try4-mbs." There are a total of 100 users providing 10 samples each.

15.1.7 GREYC-NISLAB{A,B,C,D,E}

This dataset created by GREYC labs revolved around the study of recognizing soft biometric traits for keystroke dynamics. The dataset can also be used for classical keystroke dynamics authentication. There are a total of 110 users, each providing 10 inputs of 5 different passwords in several sessions typed with one hand and another 10 using both hands. The data were obtained using the same software as the GREYC database. The dataset is available in an Excel file.

15.2 RELEVANT WORKS ON MOBILE KEYSTROKE DYNAMICS

In today's market, smartphones usually rely on a four digit PIN that the user must remember and a fingerprint biometric authentication system to protect mobile phones from intruders. As mobile phones become more integrated into consumers' lives, there is an ever increasing need to secure the data in the devices used every day.

CASE STUDY 1

In order to better protect user data, the keyboard on mobile devices can help better secure sensitive information. Luca et al. [2] examined the best

method of keystroke behavior as a method for authentication and the resilience of the system against attacks. An application was developed for data gathering in this study. It involved four different unlock methods: horizontal, vertical, diagonal, and two finger vertical. The application kept track of four factors: pressure (how hard the finger presses), size (area of the finger touching the screen), X- and Y-coordinates, and time. The two-finger-unlock method differed from the other methods because it provided two sets of X- and Y-coordinates, pressure, and size. There were 48 participants in this study and each was tasked to unlock the device 160 times over a period of 2 days. The method of unlocking the screen was counterbalanced in order to minimize learning effects.

In this study, it was assumed that the attacker had the device at hand, presumably stolen, and knowledge of the user's password pattern (shape for unlocking the phone). This meant that the first security barrier was already compromised. As with other available commercial systems, the attacker then has three tries until the device is blocked.

The methods presented in this work were designed to provide security against such attacks. Furthermore, even after a user loses the mobile device and authentication system, the mobile device should still be protected.

On the first day, the participants were asked to unlock the screens 80 times with each method. They were also instructed to unlock the mobile device using the same finger. After 20 tries, the users were asked to perform a different task, such as typing a text message, then resumed unlocking the device again using the same finger they used before. After 2 days, they were asked to perform the same task again, which was done in order to reflect more realistic data by observing how performance would change over time. In addition, each unlock that was provided was used as an attack against other users.

The algorithm used to analyze the data was the dynamic time warping (DTW) algorithm. This allowed for the comparison of two sets of data. The algorithm looks for similarities between the given sets and calculates the costs match of one set to the other. The result is a value called *warp distance*. The smaller the warp distance value is, the more similar the two sets are (i.e., if warp distance is 0, the two sets are identical). The larger the values are, the sets differ from each other. A set was composed of the time series of a touch screen (combinations of X- and Y-coordinates, pressure, size, and time).

A reference set was also implemented to represent the baseline for comparison and act as an ID for users. This was captured by taking the first 20 unlocks for each user. Each unlock was compared to the 19 other unlocks using DTW. The average warp distance for each respective unlock was then calculated. The unlock method with the lowest average warp distance was then chosen to be the reference set for the user.

The unlock attempts that were not used in creating the reference sets were compared to the reference set using DTW. In order to check the resistance of the system to attacks, the unlock attempts of other participants were also

compared to the reference set. Furthermore, the parameters used to determine the success of the system were true positive (TP): correctly accepted users, true negative (TN): correctly rejected attackers, false positive (FP): wrongly accepted attackers, and false negative (FN): wrongly rejected users. Each value was calculated per method (i.e., vertical unlock have TP, TN, FP, and FN values) which was then used to determine which unlock method was the most efficient.

The results of the study were based on 30,720 unlocks (640 per user). The ideal result would show that the method(s) would have very high TP and TN values and low FP and FN values. The results of the study have shown that the diagonal unlock method provided the highest accuracy in determining keystroke biometrics compared to other methods. It produced the highest accuracy values in FP and FN and also contained the lowest FP and FN values. The two-finger vertical unlock method performed the poorest with the lowest TN value, meaning to say that the majority of the attacks were successful.

Overall, the result of this study has shown that the developed system could identify users well (high TP values), regardless of the method used. Despite these results, however, there is a drawback to using this approach for keystroke biometrics. In the best case (diagonal unlock method), the TN value was relatively high. Meaning to say, attacks are still possible (~4 out of 10 attacks would be successful). In terms of security, the result of this statistic does not provide a satisfying result. Despite the room for improvement for the unlock method, the more promising way to go is using a method that allows collecting more significant data per dataset.

CASE STUDY 2

Jeanjaitrong and Bhattarakosol [3] determined the feasibility of applying keystroke dynamics on mobile phones in their case study. They developed a web-application where the participants were granted passwords based on symbols. The symbols used for this study were heart, spade, club, and diamond with each symbol having different color: black, red, blue, and green. The web-application had three pages; registration, data collecting, and forgot password page. The registration page asks each user personal information and remembers the granted password. The data collection page is where the user performs the unlocking task. The page shows 16 symbols arranged as a 4×4 matrix. The password forgot page shows the users of their granted password.

There were a total of 10 random iPhone users that were selected for this study. The web-application provides the users a password that they would have to enter. A password consists of 4 out of the 16 symbols. The length of the password is based off an ATM machine and iPhone passcode length. Each user was asked to perform the unlock procedure 10 times per round and

there was a total of 10 rounds. The data collected in this experiment were button-pressed time, button-released time, and an ordered pair of screen position (*X*-, *Y*-coordinate). Using this data, the authors were able to calculate dwell time, interval time, and distance.

The dwell time refers to the amount of time the participant took while pressing down a button. This was calculated using the difference between the time a button was pressed and the release time. Interval time ratio refers to the ratio of the interval time between button presses and the overall time used to press the whole password. The distance value is the distance between two different points on the screen that was pressed by the user.

The values used as performance metrics in this study were the false acceptance rate (FAR) and the false rejection rate (FRR). The authors determined the FAR, FRR, and accuracy for a single factor authentication and multifactor authentication. Their study has shown that the triple-factor used to authenticate a user yielded a higher accuracy rate compared to dual and single factor authentication. Using the dwell time, interval time, and distance factors to authenticate a user yielded the best values FAR, FRR, and accuracy.

In order to show that keystroke dynamics in mobile touch screen devices have similarities to their keyboard counterpart, the authors generated FAR, FRR, and accuracy percentages for a keyboard. Their results have shown a close relationship in terms of FAR, FRR, and accuracy values using dwell time and interval as factors.

Based on the results of this study, the keystroke dynamic mechanism has shown that it can be just as effective as the keyboard. Furthermore, this proves to be a cost efficient method in securing data in mobile phones.

CASE STUDY 3

In Reference 4, a software prototype was implemented on a Microsoft Windows phone. Each user was instructed to create two passwords, a simple four-digit PIN and an alphanumeric password. The password textbox implemented in the software captures key events and interkeystroke latencies. These values were evaluated based on three classifiers: Euclidean distance, Mahalanobis distance, and feed-forward multilayered perceptron (FF MLP) neural network. Both the Euclidean and Mahalanobis distance are statistical-based methods that have low processing requirements which is an important factor to consider on mobile platforms. FF MLP, on the other hand requires high processing requirements in exchange for better performance rate.

There were 20 people who participated in this study. And in a single session, each participant was asked enroll by entering a password they had chosen 20 times and authenticate it by entering it 10 more times.

The results have shown the importance of limited process capacity of mobile devices. While neural network-based approaches have outperformed the statistical-based methods, it has exceeded the process capability of the

device. Due to the poor performance of the neural network algorithm, it was not possible to calculate the performance rates of the system (both FAR and FRR).

The two statistical analysis methods have little difference in performance in obtaining either the PIN or alphanumeric password. However, the results do show that the performance of the classifiers on the alphanumeric password is stronger than the PIN. This suggests that a short PIN is an ineffective tool for keystroke analysis. The alphanumeric password yielded much lower FRR and FAR values than the PIN password.

After the study, the authors also took a survey asking the participants which authentication method they would use on a mobile device. The vast majority of the participants preferred fingerprint-based solutions, with speech recognition solutions coming in second, and keystroke analysis third.

Half of the participants thought that entering 20 samples to be time consuming. However, 18 out of the 20 participants said that they would use the keystroke solution if it were available and thought that it would provide more security.

15.3 CONCLUSION

Due to the fast evolution of smartphones over the last few years, keystroke dynamics authentication used in computers could also be implemented in the near future. There are significant advantages in using this method, making it a viable method in securing phones. One of the advantages of this method is the lack of need for hardware. All the hardware required to implement this system is inherently built into the system. Another benefit to keystroke dynamics is the small amount of data that is needed to train a recognition system. Compared to other biometric techniques such as facial recognition the amount of data needed in sufficiently implementing keystroke dynamics requires a much smaller sample size. This leads to a shorter processing time. External environmental conditions also do not affect the verification process making the authentication process easier for the user [5]. Based on these advantages, keystroke dynamics provides a reliable method in securing sensitive data in mobile devices.

Despite the advantages this method has over other biometric techniques, there are still some drawbacks using keystroke dynamics. In Reference 4, the case study using neural networks, which can produce optimal results, exceeded the processing capability of the mobile device used. Another disadvantage is despite having a small sample size requirement, users in his study found providing multiple samples troublesome. Depending on the position of the user (sitting down, walking, or standing), the performance

of the verification process can also be affected. This could possibly lead to false rejection by the system if the user tries to login in a different position and the system was trained when the user was standing up.

Considering the advantages and disadvantages of applying keystroke dynamics on mobile phones, it is within reason to think that this can be applied to mobile devices in the near future [5]. As mobile phone technology rapidly advances, obtaining performance that rivals desktop computers is within reach.

REFERENCES

1. Giot, R., B. Dorizzi, and C. Rosenberger. A review on the public benchmark databases for static keystroke dynamics. *Computers and Security*, Elsevier, 2015, Vol. 55, pp. 46–61.
2. Luca, A. D., A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you! In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems—CHI '12*, Austin, Texas, 2012. n. pag. Web.
3. Jeanjaitrong, N. and P. Bhattarakosol. Feasibility study on authentication based keystroke dynamic over touch-screen devices. In *2013 13th International Symposium on Communications and Information Technologies (ISCIT)*, Surat Thani, Thailand, 2013. n. pag. Web.
4. Buchoux, A. and N. L. Clarke. Deployment of keystroke analysis on a smartphone. *Research Online*. Edith Cowan University, 2008, p. 48. Web. February 22, 2016, http://ro.ecu.edu.au/ism/48
5. Avila, C. S., J. G. Casanova, F. Ballesteros, L. J. M. Garcia, M. F. A. Gomez, D. De Santos Sierra, and G. B. Del Pozo. PCAS—Personalised Centralized Authentication System. *European Union's Seventh Framework Programme*, January 31, 2014. Web. February 22, 2016, https://www.pcas-project.eu/