Typestroke : User authentication system via typing for touchscreen devices

Shruti Nair, Pratik Zambani
Stony Brook University, USA
{nshruti, pzambani}@cs.stonybrook.edu

ABSTRACT

Nowadays, smartphones store private information such as login credentials for various services and payment details. In spite of being security and privacy critical, smartphones are being widely safeguarded using traditional authentication systems including passwords and PINs which suffer from a severe disadvantage of intrusion once the phone is unlocked. This paper devices a novel algorithm to predict the user of the smartphone based on it's typing movements. The algorithm is based on a scoring function which takes user's error rate, time difference between typing two letters, force and typing coordinates as it's four primary features. Among all the features, time and force are selected as the most reliable features owing to each user's different typing behaviour and speed. Five models are then created by giving different weights to each feature. The model with the highest weights given to time and force performed the best in distinguishing between same and different users. Based on the model, a scoring function is developed which yielded access to the user if and only if the user's typing gesture matches 85% with the training data. In all the other cases, the intruder gets blocked.

Keywords **: {Authentication, Gesture movement, Smartphones}**

I. INTRODUCTION

[1] Smartphones and tablets have become ubiquitous in the lives of many people. The main reason for the success can be attributed to their ability to offer mobility, computing power, storage capacity and a smooth interface. This combined with innumerous mobile applications explains the huge popularity of such devices.

[2] Research on authentication for smartphone users has seen a concentrated work during the last few years of which many were based on behavioral biometrics which particularly looks into user behaviour in terms of the use of keystrokes and touchscreens, and the individual characteristics of hand waving and gait. However, smartphone authentication using typing gesture is still relatively an unexplored domain.

This paper presents a novel authentication scheme called Typestroke, which takes into account of two human behaviors: how the phone is held and how the user types in the password or text in general. Our experiments confirmed that every user has a unique phone movement behavior and a different way of touch-typing phrases on the smartphone. Typestroke computes the phone-holding behavior with 7 built-in smartphone sensors, for: the orientation, the gravity, the magnetometer, the gyroscope and 3 variants of the accelerometer [3][4][5]. Along with it, touch data records information related to the x-y coordinate for each letter and touch time between two letters.

Sensors are started at the time of the first touch-type and continued until each user writes at least 7 - 10 phrases in one trial. Users are allowed to input the standard phrases with their natural speed, hence they are expected to be able to use this authentication mechanism quite comfortably. We have extracted 4 different statistical features from two different data streams (TouchScreen Logger and Sensor data logger) and taken into account of all the physical sensors (a total of 16 from each sensor), from each typing pattern. In [6], authors show that these time-based features are the most widely used features in keystroke dynamics. In addition to that, we have taken the force applied by each user from touch point data, and their x-y coordinate typing areas for better estimation. In order to check the usability of our proposed method, we collected 30 observations from 10 users over a course of three days.

The remainder of the paper is as follows. Section 2 covers related work. In section 3, we present an initial assessment and methodology including the algorithm proposed. Section 4 presents the experimental setup, data collection and discusses obtained results. Section 6 and 7 presents planned future work and concludes the paper.

II. RELATED WORK

Touch stroke-based user authentication is now the most relevant and tested behavioral method used for user authentication both on laptops as well as on smartphones which uses either hardware/ touchscreen based keyboards. Since we have implemented text-independent touch-typing dynamics using Android touch screen -keyboards, we consider software keyboard-based work as our related work.[7][8]

A study conducted by Huang et al. [9] devised a touch-keyboard user authentication on mobile phones. The users were told to enter their username and password for 6 times as a part of training process. Based on the keystroke latency and keyhold-time features, the study reported an Equal Error Rate (EER) of 7.5%. On the same line, a recent study conducted by Saira et al. [10], on smartphones found out that that the keystroke force might not be unique and reported an EER of 8.4% when used simultaneously with classical keystroke features (timings).

Several researches have been conducted to study the utility of accelerometers and gyroscopes. For example, Giuffrida et al. [11] created UNAGI, a fixed-text and sensor-enhanced authentication mechanism for Android phones. The method was tested on 20 subjects and the project achieved an EER of 4.97% for passwords, and 0.08% for only sensor data.

Similarly, Aviv et al. [12] present a method that relies on accelerometer data and keystroke timings to infer 4-digit PINs for unlocking smartphones. Specifically, they demonstrated the use of accelerometer data for learning user tapping and gesture-based inputs as these methods are required to unlock smartphones using PIN/password and graphical password patterns. Additionally, they collected data in two situations, sitting and walking.

Li et al. [13] proposed a system similar to our approach, that uses a continuous authentication for smartphones by taking the user’s finger movement pattern into account for learning. In contrast to us, they did not consider entering text into a soft keyboard but only gestures like sliding towards a special direction or taps.

Similarly, we propose to fuse the phone movement patterns (before, while, and after swiping/typing) with the swiping or typing behaviors. However, our work differs from Sitovas on the following aspects: (i) our data collection was completely unconstrained, (ii) we apply feature-level fusion of modalities in addition to the score-level fusion, and we test the system under continuous authentication paradigm and report accuracy.

Typestrokes differs from prior research in terms of the data input as phrases rather than mere passwords. The data has been extensively tested based on not only pressure feature but also on the typing speed and their relative x-y coordinate position for each letter. It also differs in the scoring strategies, number of sensors, sensor-data-acquisition and constraints on the input.

The fusion not only improves the classification accuracy but also provides more complex feature space. It may be possible for adversaries to train a robot [14] or a human imitator [15] to imitate swipe/type or phone movement patterns separately. However, we believe that it will be extremely difficult to imitate both (swiping/typing patterns and the corresponding phone movement patterns) simultaneously, especially when the features extracted from both of the modalities are less correlated.

III. METHODOLOGY

3.1 Data Information

Modern smartphones are equipped with multiple sensors with the capability to detect and compute device/user movement. Accelerometer and orientation sensors are the most used sensors for movement recognition.

Thus, in order to conduct the experiment, we chose Tap_Typing_Data that included two kinds of files.

- TouchPoint Logger data
- Sensor Logger data

3.1.1 Sensor Logger data

Sensor Logger data primarily contains information related to Accelerometer and Input mode which gives us the necessary knowledge for calculating force and pressure applied by each user. The other features include InputFinger, UserID, KeyboardType, TaskType, BlockNum, TrialNum, Phrase,curSysTime ,upNowTime, eventTime, sensorType, value0, value1, value2, isGestureExpert, Device and InputMode.

3.1.2 TouchPoint Logger data

TouchPoint Logger data is stored in the form of an XML file. The sensor data provides user information like UserID, Inputfinger etc. Along with it, it also provides adequate data related to the X-Y coordinate position which the user typed for each letter. In addition to some of the features of the sensor data, touch point also provides start_time for each typed letter by a user. All this information were stored in the form of an IME data. Some of the other features include Trial Information, Device Name, IsGestureExpert, TaskType, isCanceled and editTextContent.

3.1.3 Experiment Setup

The data was collected over a period of three days. Each user was instructed to type the same set of phrases on the Google keyboard.

The second digit of the ID indicates the date the data was collected. The last three digits of the ID can be used to identify a user, e.g., 40002, 41002, 42002 are from the same user.

0	Day 1
1	Day 2
2	Day 3

Table No. 1 Experimental Setup

3.2 Algorithm

Out of all the features taken into consideration, we have chosen force, time, error rate and position as our primary features. As discussed in Section 2, time plays a key role in determining the authenticity of the user. In addition to that we have also taken into account of the forces applied by each user.

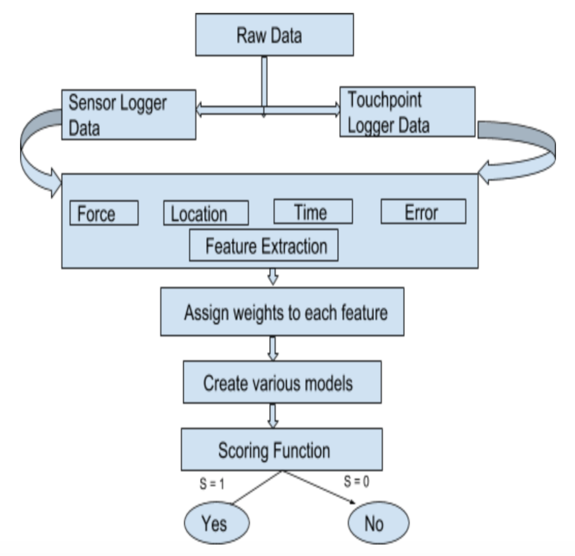


Figure No. 1 Scoring Algorithm

3.2.1 Force

To calculate the force, sensor logger provides three features which are the forces recorded by the accelerometer in three dimensions i.e X,Y and Z. We also added a fourth dimension to all of these sensors and named it magnitude,

e.g. this dimension for the accelerometer is calculated as follows:

$$\text{SM} = (a_x^2 + a_y^2 + a_z^2) \qquad (1)$$

where a_x, a_y and a_z are the readings from the accelerometer sensor along the X, Y, Z dimensions, respectively. [16]

For evaluating force, the mean and standard deviation for both the training and testing data were recorded. Based on the values provided, the common area shared between both the curves divided by the area of test data was taken as a measure to find out the authentic user. The figure 3 above shows the green area as the force obtained from the training data of 9th user and the blue area shows the force obtained from the testing data of 10th user.

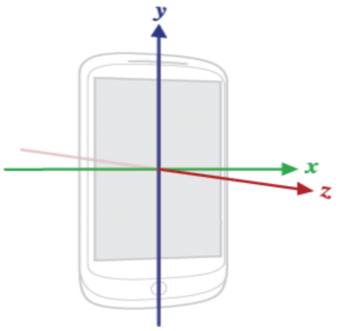


Figure No. 2 Definition of the coordinate system

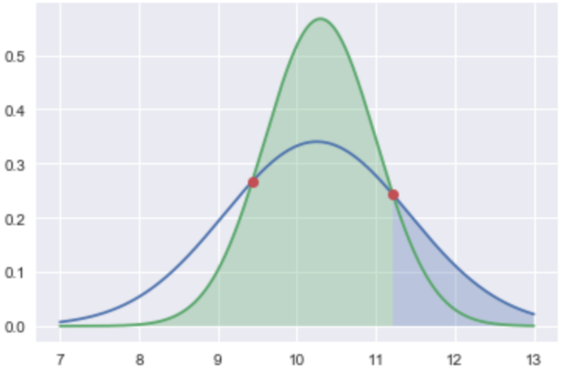


Figure No. 3 Force Area between training and testing data.

3.2.2 Location

From the touch_logger_data, we have calculated location information from the following features : typed letter, x coordinate for the typed letter, y coordinate for the typed letter, current typed string.

In order to store location points, we maintained a dictionary which stored 26 keys corresponding to 26 letters, and recorded x,y values for each letter. This data structure allowed us to perform statistical analysis on location data.

As when we constructed the dictionary, there were many missing values, owing to the lesser number of phrases entered. In order to fill those missing values, we utilized the coordinate information for each letter in a standard Android phone. For eg, if the matrix value for a-x is empty, it can be substituted by looking for the value s-c from another dictionary which stores the manhattan distance between every two letters.

	a	b	c	...
a	182, 190 ...	301, 281 ...	258, 252 ...	
b				
c				
...				

Table No. 2 26 X 26 matrix for storing position coordinates

3.2.3 Time

The time taken by user to type each letter is calculated using start_time information for each letter from touch_logger_data. For example, if we need to find the time taken to type AND by a particular user, we calculated it as

Time taken to type AND = (Startime of N - Startime of A) + (Startime of D - Starttime of N) + (Starttime of Space - Startime of N) As when we constructed the matrix, there were many missing values, owing to the lesser number of phrases entered. In order to fill those missing values, we utilized the coordinate information for each letter using a standard Android phone. As shown in Figure No 4, the time calculated for S-C has been utilized using the information of A-X or D-V and so on.

3.2.4 Error Rate

The error rate for a particular user was calculated using an algorithm well used in genomics called edit distance. The edit distance takes two strings string1 and string2 and their respective lengths and returns the number of characters which needs to be change to generate string 2 from string 1. The pseudo code has been described in Figure No 5.

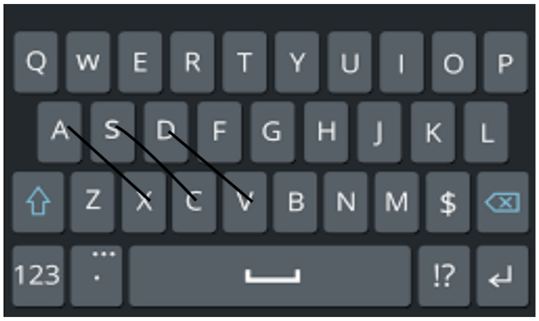
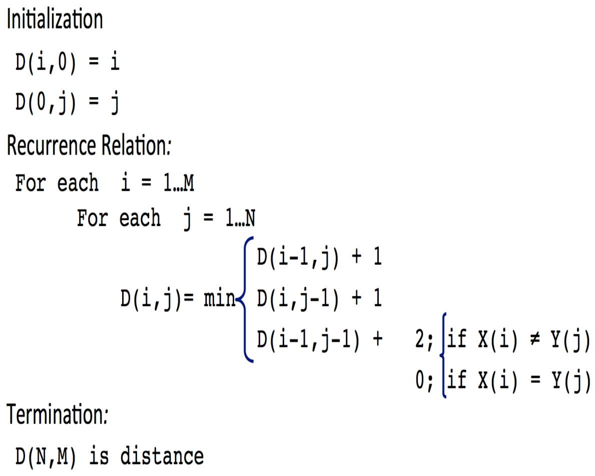


Figure No. 4 A smartphone touchpad showing movement time similarities



```
Initialization
 D(i,0) = i
 D(0,j) = j
Recurrence Relation:
 For each  i = 1…M
      For each  j = 1…N
                        ⎧D(i-1,j) + 1
              D(i,j)= min⎨D(i,j-1) + 1
                        ⎩D(i-1,j-1) +   2;⎧if X(i) ≠ Y(j)
                                        0;⎩if X(i) = Y(j)
Termination:
 D(N,M) is distance
```

Figure No. 5 Edit Distance Algorithm

IV. RESULTS AND ANALYSIS

The training data was taken as the mix of Day 1 + Day 2 + (1/2) of Day 3 for each user. The rest of the Day 3 data is taken for testing for each user. Considering the features described above, we created multiple models by assigning different weights to each feature in each of the models. The total sum of all weights was equal to 100. We train the model using training data of a particular user and calculate mean and standard deviation values of time to type two subsequent letters, force applied on the touchscreen, errors made while typing phrases and x and y touch coordinates of letters pressed. The weights of features in various models are

1. Touch - 15, Time - 50, Error - 5, Force - 30
2. Touch - 10, Time - 60, Error - 10, Force - 20
3. Touch - 60, Time - 10,

Error - 20, Force - 10

4. Touch - 20, Time - 40,
   Error - 10, Force - 30
5. Touch - 0, Time - 50,
   Error - 5, Force - 45

After training all the models with one user’s train data, we ran test data of all users on these models with the aim to clearly distinguish actual user from other users. The scoring function takes the score of each test phrase for a feature (between 0.0 to 1.0) and multiplies it with the weight of that particular feature and sums it up for all features.

The score of a feature depends on where the test phrase feature values lie on the bayesian curve. If the value is near mean, the feature gets high value of 1.0 and if the value is more than 3 standard deviation away from mean, the we give it 0 value and other values get a score between 0 and 1 depending on how far it is from the mean. The final score of a phrase adding all feature scores falls between 0 to 100. The higher the score the higher probability that the phrase has been typed by the same user and the lower the score the higher the probability that the phrase has been typed by a different user. We took 6 test phrases which have been typed by majority of users as test data and calculated the scores for each phrase for all the users.

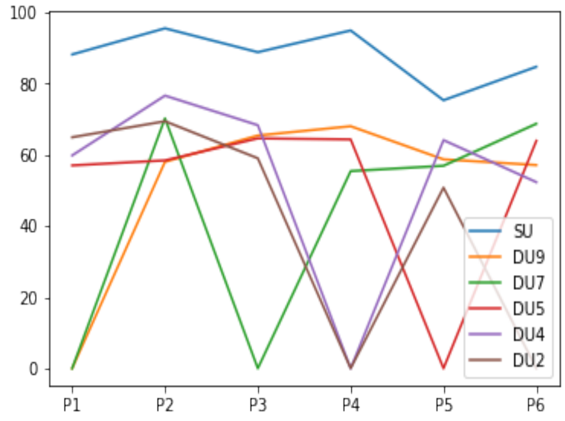


Figure No. 6 (Weights - 5, 15, 50, 30)

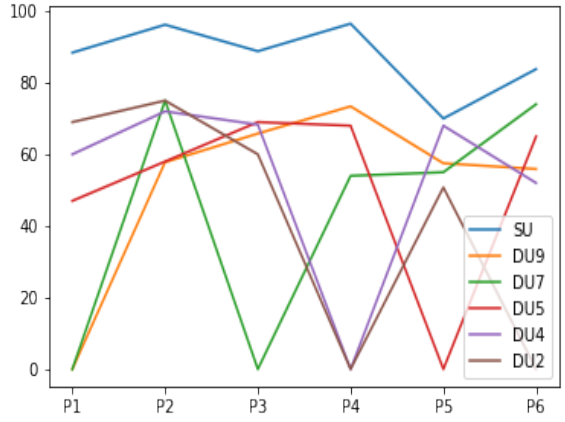


Figure No. 7 (Weights - 10, 60, 10, 20)

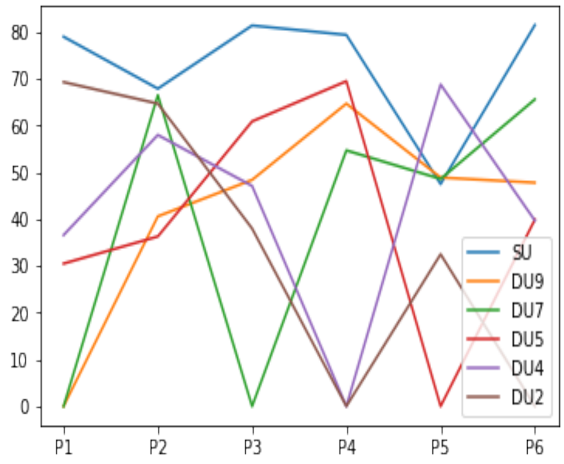


Figure No. 8 (Weights - 60, 10, 20, 10)

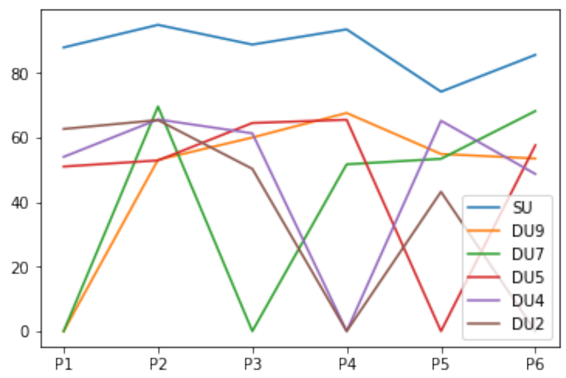


Figure No. 9 (Weights - 20, 40, 10, 30)

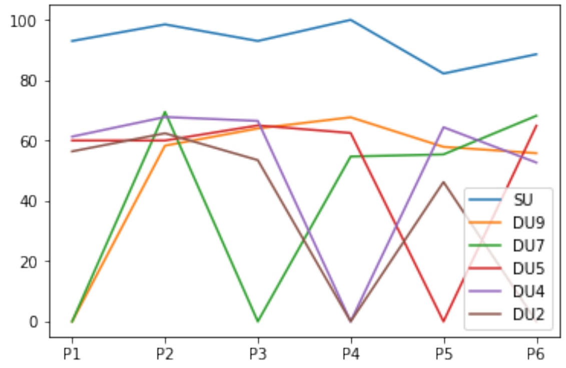


Figure No. 10 (Weights - 0, 50, 5, 45)

From the above results graphs, we observe that time is the most important factor to distinguish between different users. Users have different typing behaviour and speed and the model with higher weight to time, distinguishes well between various users. Force used to type on the touchscreen keyboard is another dominant factor. High weight for force leads to better differentiation across users. We can notice from Fig 9 where weights for time and force are high, there is a clear distinction between different users and same user for a particular user's training data. X and Y touch coordinates do not have much variance in data and the mean and standard deviation values for various users are fairly similar. Hence, giving high weight to this feature will not help in differentiating users as can be seen from Fig. 8. Error rates are calculated using edit distance algorithm and we observed on our dataset that many users have made numerous mistakes in their test data. Due to this, the error scores for different users were fairly nearby.

V. LIMITATIONS

In general, the model performed fairly well and reached as high accuracy as 85% for the same user to make it inaccessible for the intruder to access the smartphone. However there were some major limitations encountered during the project which needs to be resolved in the future.

Setting Weights : From the previous literature, it was quite clear that , time was one of the most dominant features among all. But the correct estimation of the weights to be alloted to each feature was not fixed. The brute force involved setting weights from 1 to 100, but it involved a high computation capacity computer to correctly estimate the best result.

Error rate : Since, the experimentation involved long hours of typing by the subject, most of the subjects didn't maintain their typing accuracy and made a lot of errors in between, making it difficult to estimate the importance of error rate among all the features.

VI. FUTURE WORK

We believe that the model has been fairly developed to prevent the intruder from accessing the phone. We are looking forward to conduct more user study to get a better estimate of the weights for the feature.

Once the user data is abundant, we are planning to perform ANOVA test to clearly determine the results better.

In addition to that, we are also planning to execue the model in both iOS and Android to check it's practical applicability in the near future.

VII. CONCLUSION

The four features - touch coordinates, force applied on touchscreen keyboard, time to type and error rates can be effectively used to distinguish intended phone user from potential adversaries. Due to errors in the train dataset, we could not establish the importance of error rates compared to other features. We confirm time and force are crucial and lead to clear distinction in the score values for different users. Touch coordinates are very similar and can be ignored or given low weight in the models. The algorithm described in the paper can be implemented easily and included as part of in built software keyboard in smartphone devices. It is important to protect the privacy of personal data such as messages, emails, photos etc and a continuous authentication such as typestroke is an effort in that direction.

ACKNOWLEDGMENT

The authors wish to thank the Dr. Xiaojun Bi for their helpful and valuable comments that greatly contributed to improving the idea and the manuscript of the paper.

REFERENCES

[1] Buriro A., Crispo B., Del Frari F., Wrona K. (2015) Touchstroke: Smartphone User Authentication Based on Touch-Typing Biometrics. In: Murino V., Puppo E., Sona D., Cristani M., Sansone C. (eds) New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops. ICIAP 2015.

[2]Saevanee, H., Bhatarakosol, P.: User authentication using combination of behavioral biometrics over the touchpad acting like touch screen of mobile device. In: Proceeding of the International Conference on Computer and Electrical Engineering (ICCEE 2008), pp. 82–86. IEEE, Phuket (2008)

[3] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing, " IEEE Commun. Mag., Vol. 48, No. 9, pp. 140-150, Sep. 2010

[4] A. Z. Rakhman, L. E. Nugroho, Widyawan and Kurnianingsih, "Fall detection system using accelerometer and gyroscope based on smartphone," *2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering*, Semarang, 2014, pp. 99-104.

[5] V. Douangphachanh and H. Oneyama, "Exploring the use of smartphone accelerometer and gyroscope to study on the estimation of road surface roughness condition," *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Vienna, 2014, pp. 783-787.

[6] A. Darabseh and A. Siami Namin, "On Accuracy of Keystroke Authentications Based on Commonly Used English Words,"

2015 International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, 2015, pp. 1-8.

[7] Md Liakat Ali1John V. Monaco1 · Charles C. Tappert1 · Meikang Qiu , “Keystroke Biometric Systems for User Authentication” 1 Accepted: 14 February 2016 Springer Science+Business Media New York 2016

[8] Frank et al., "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication," IEEE Transactions on Information Forensics and Security, Vol. 8, No. 1, Pg. 136-148, Jan., 2013.

[9] Huang, X., Lund, G., Sapeluk, A.: Development of a typing behavior recognition mechanism on android. In: Proceeding of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1342–1347. IEEE, Bradford (2012)

[10] Zahid, Saira and Shahzad, Muhammad and Khayam, Syed Ali and Farooq, Muddassar: Keystroke-based user identification on smart phones Recent Advances in Intrusion Detection, pp.224–243. Springer,(2009)

[11] Giuffrida, Cristiano and Majdanik, Kamil and Conti, Mauro and Bos, Herbert: I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 92–111, Springer (2014)

[12] Aviv, Adam J and Sapp, Benjamin and Blaze, Matt and Smith, Jonathan M: Practicality of accelerometer side channels on smartphones In:Proceedings of the 28th Annual Computer Security Applications Conference. , pp.41–50. ACM, 2012

[13] L. Li, X. Zhao, and G. Xue, “Unobservable re-authentication for smartphones,” in Proceedings of the 20th Network and Distributed System Security Symposium, NDSS, vol. 13, 2013, pp. 1–16.

[14] A. Serwadda, V. V. Phoha, Z. Wang, R. Kumar, andD. Shukla, “Toward robotic robbery on the touch screen,” ACM TISSEC , vol. 18, pp. 14:1–14:25, May 2016.

[15] C. M. Tey, P. Gupta, and D. Gao, “I can be you: Questioningthe use of keystroke dynamics as biometrics.,” in NDSS , TheInternet Society, 2013.

[16]https://developer.android.com/reference/android/hardware/SensorEvent