Game Authentication Based on Behavior Pattern

Noureldin Ali
School of Engineering, University of Portsmouth, Portsmouth, United Kingdom,
Noureldin.ali@myport.ac.uk

Yanyan Yang
School of Engineering, University of Portsmouth, Portsmouth, United Kingdom,
Linda.yang@port.ac.uk

ABSTRACT

Nowadays smartphones have gained a huge popularity, ranging from using social applications to play video games. Smartphones also became more private than before. Although security mechanisms are provided from different smartphone providers, still smartphones are vulnerable to getting attacks regularly. Therefore, a security mechanism can help in protecting the smartphone from attackers. We propose a security mechanism that can help in protecting smartphone while at the same time enjoy using it such as playing a video game. The authentication system is designed in a way that can read raw data directly from the touch sensor so, it's not bound to be used in a particular application. Our system authenticates the user in the background without interrupting the user interaction with the smartphone. We tested the system in a real environment with two popular games. We managed to get 70% accuracy in one game and 90% in the other game.

CCS CONCEPTS

• **Software and its engineering**---Software notations and tools---General programming languages---Language features ---Inheritance**;** •**Security and privacy**---Security services---Authentication---Biometrics;

KEYWORDS

Mobile devices; Authentication; Android; Grid; Sensor; Gesture; Biometric.

ACM Reference format:

Noureldin Ali and Yanyan Yang. 2017. Game Authentication Based on Behavior Pattern. In *Proceedings of The 15th International Conference on Advances in Mobile Computing & Multimedia, Salzburg, Austria, December 2017 (MoMM '17), 6 pages.*
DOI: 10.1145/3151848.3151878

INTRODUCTION

According to recent research, it suggests that by 2018 two-thirds of Britons will own a smartphone [1]. Smartphones have the ability to store private data like contacts, personal media, banking information, and applications. In comparison between smartphones and PCs, PCs can be shared with other users while in smartphones it rarely happens.

The current authentication system on smartphones may have some usability problems, a lot of smartphone users do not use the full extent of the current security mechanisms. One of the most popular authentication mechanisms is password authentication, but it has some flaws due to human memory limitation [2]. In order to prevent guessing attack passwords, the password must be either contain an unusual character or long [3]. A more appropriate authentication method for smartphones that make use of the touchscreen is the graphical password but, research shows that this will result in low entropy of the password [4].

Stopping illegitimate use of smartphones requires a new authentication mechanism. Normally, the system comes with an authentication mechanism that asks the user to enter a pin or a password when the system is first booted or logged on, so the user can have access to the system. If the owner forgot to lock the phone, an attacker could easily have access to the phone. It has been reported that even locked phones can still be accessible by using defects in the operating systems, such flaws in iOS [5] and Android [6].

A new study shows that pin numbers are vulnerable to thermal attacks [7], and a complex password is hard to remember. For that reason, a good authentication system should depend on a special kind of password that is not difficult to remember and hard to forge. For such system that would fulfil these needs is the owner biological data. Several studies have been done on authentication based on the biological data, such as iris recognition [8,9], fingerprint recognition [10,11], face recognition [12]. Nevertheless, the above-mentioned methods are not suitable for all of the smartphones since they require a special hardware to work and needs the user assistant.

In behavioural biometric, it is primarily supported via software implementation. Thus, it is easy to implement [13, 14]. The data can be collected in behavioural biometrics without the knowledge or notice of the user. Behavioral biometric does not need special hardware to collect data, and it is very economical. A recent study states that the human pattern in behavioural biometrics is estimated to be stable with time [15]. For that reason, it is convenient to

authenticate mobile devices or portable computers. These devices can gather plenty of useful data, such as motion, location, application usage, and touch screen pattern. Besides, each user usage of the mobile device is different from one person to another. The gathered data can be further used to create a comprehensive profile for each user.

Behavioral and physiological biometrics must fulfil some characteristics despite their differences, both need to meet the practical and theoretical specification so that they can be used for authentication purposes. The first characteristic is acceptability; the user must get familiarised with the authentication system and accept to use it on a daily basis. Second is performance; it's the system degree of achieving the best speed and accuracy. The third is collectability; the system data must be measurable. Fourth is distinctiveness; the system characteristics must be able to differentiate between two users. Fifth is circumvention; it is the system security level of preventing unauthorised usage. Sixth is permanence; the biometric system characteristics should not change over time. Last characteristics is universality; every user biometric characteristics should be common in the system.

A recent study shows that one of the most popular activity on smartphones is playing games, on average every month smartphone owners tend to play more than 10 hours of games [16].

Till this date, large companies like Samsung and Apple are always in a debate on what is the best authentication method. Apple debated in the past that the best authentication method is the fingerprint recognition and started implementing it on iPhone 5s in 2013. However, Apple had ditched fingerprint recognition on its latest iPhone release in 2017 and improved the functions of face recognition.

The new authentication mechanism aims to authenticate the user in the background while the system is running and without interrupting the user interaction with the system thus, asking the user to enter a password or pin number directly will result in bad user experience.

In this paper, we present a security mechanism that uses the pattern the user makes while playing a game on the smartphone. By using the motion of the finger on the touchscreen and compares it with the owner to check whether if the current user is the owner or unauthorised user. The system works in the background and does not need assistance from the user. First, the system learns the owner finger motion. Then, compare the current user pattern against the owner pattern, if the system detects that the current user is not the owner then the system will execute a command like a shutdown or ask the user for a password.

The smartphone senses the user's finger motions, these motions are clarified as different touch gestures. Gestures are used to interact with the smartphone.

The system was executed on LG K8 running Android Marshmallow. To assess the effectiveness of the system and test the system performance a significant amount of experiments were performed.

RELATED WORK

Authentication based on the biometric data is simple to perform, easy to use, and hard to forge. Rather than using the normal authentication which uses passwords as its main approach, the biometric authentication uses the individual features of each person, like iris recognition and fingerprint recognition.

In biometric authentication, the system runs 2 phases. The first phase is to train the system on the given biological data, like giving the fingerprint several times in the beginning so the system can learn and interpret the data, or stay still in front of the camera so the system can read the iris. The second phase is when the system uses the stored data of the first phase and compares it to the current biological data.

Behavioral features or physiological features were the main concentration on previous studies [14]. Several physiological features can be used for biometric authentication, like iris recognition [8,9], fingerprint recognition [10,11], and face recognition [12]. Nevertheless, physiological biometric authentication depends on special hardware on smartphones, which may not be available on all smartphones. Furthermore, the majority of these methods require the user support to operate, such as face recognition which requires the user to remain at a specific angle and with at least certain amount of light, so it can operate and authenticate the user.

In behavioural biometrics authentication, it uses the user behaviour pattern for authentication. Two main operations are used in personal computers; studies show that mouse movement [17] and keystroke [18]. However, keystroke in smartphone happens rarely since users barely use the keyboard for heavy typing. Thus, keystroke authentication on a smartphone will not suffice.

Behavioral biometrics aid in distinguishing between users by how each user uses applications or touchscreen, this has been revealed in a recent study [19]. Games were able to support users in performing better at security tasks that are available to use, like recognising and generating randomness [20].

In spite of the fact that smartphones do not have mouse devices but in previous studies [21] shows that finger motion on the touchscreen in similar to mouse movements thus, help us in understand how it works.

The OS receives from the hardware the movement of the mouse as the mouse moves, this contains the raw data of the mouse movement coordinates and button events, like button down and button up. Then, these data are translated by the OS into a sequence of point data in which later create the mouse movement.

Smartphones have different sensors that each sensor can perform tasks to the user; these sensors distinguish smartphones from personal computers. The most common sensors in smartphones are an accelerometer, multi-touch screen and magnetometer.

Nowadays, nearly all smartphones support multitouch screen. The multi-touch screen has the ability to acknowledge more than one touch at a time; each smartphone varies from one another, for example, some advanced smartphones can support more than ten touch points at a time while some basic smartphones can merely support two touches points at a time. Each touch event contains several attributes like, touch area, touch pressure, and touch

position; then these events are sent to the OS. Touch gestures such as double tap, sliding, or tap are a sequence of touch events attached together.

The accelerometer is used in the smartphone to measure the three-axis acceleration, *x*,*y*, and *z* [22]. This sensor is able to capture the position of the smartphone in a three-dimensional space.

The magnetometer is a sensor that gives the smartphone basic orientation with respect to Earth magnetic field so, the smartphone can always know the direction of the North.

Gyro is a sensor in smartphones that can notice the current orientation of the phone. It is used to know whether if the smartphone is held on a horizontal surface or not.

Smartphone sensors contribute in providing lots of biological data of the user; these data can be used later for another purpose like the biometric authentication. In previous studies, sensors such as accelerometer were used to feel the user's shaking data to connect two phones [23, 24].

Sensors were used to record the user's behaviour patterns to authenticate the user based on the recorded data [25]. Their idea was to use gyro sensor and accelerometer to sense the walking pattern of the user's. According to the study, the success rate was between 60% and 85% in recognising the user.

Another study was made to authenticate the user by using the accelerometer sensor to record the data when the user swings his/her harm [26]. This study was able to achieve 5% equal error rate. However, another study was made with accelerometer and gyro sensor to authenticate the user using the patterns of the arm movement as the user is making a call [27]. The result of this study was 9.33% false rejection rate and 4.44% false acceptance rate.

In a recent study, games have assisted in the progression of securing device pairing [28]. Security tasks take the benefit of the security factor that the games provide and the popularity of the games. Games do not face the issue of a possible leak of sensitive information, which some social applications may have.

DESIGN

The new authentication system aims to keep authenticating the user as long as the smartphone is in use, the new system would not interrupt the user or need any assistance in its operation, and the new system would not have extensive computational requirements.

The concept of the new authentication system is to monitor the user finger movement interaction with the smartphone touchscreen because each user is unique on how they interact with the smartphone. A continuous movement on the touch screen is called a touch gesture. Each user touch gesture is unique and is believed to carry their individual behavioural features.

We use these features to authenticate the user. The system has two modules, the training module and the authentication module. The authentication module is dependent on the training module. In Fig. 1, demonstrates the system architecture.

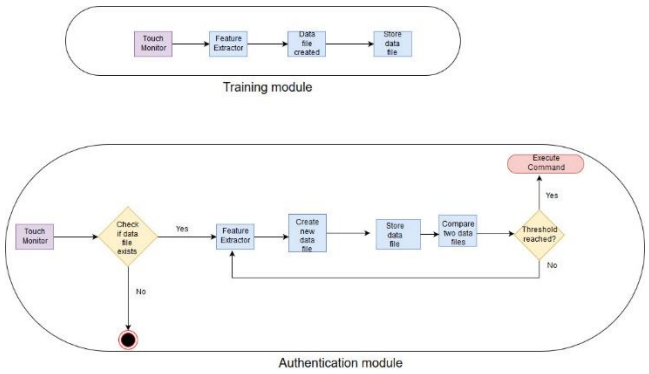

Figure 1: System architecture.

The training module observes the owner raw touch data and creates owner data file, which gathers the raw data into touch gestures. These touch gestures form the owner distinct behavioural features. The authentication module first checks whether the owner file data exists, if it does not exist then the authentication module ends. If the owner file data exists, then the authentication module creates a current data file of the current user and then compares the two data files.

The touch component works as follows: First, divide the screen into a grid, then check if a screen touch was made and if the answer is yes then start gesture recognition. Second, initialise the counter to zero and get the event responsible for the touchscreen and get the coordinates of the touch. Third, divide the vertical and horizontal lines of the coordinates with the grid values. Last, if the screen is not touched anymore, then reset the count and start new gesture.

The feature extractor work as follows: First, determine what gesture is made such as swipe, tapping, or double tapping. Second, configure which cell in the grid was touched as defined previously from the touch component. Third, increment the cell or cells that were touched to create the pattern. Last, if the user's finger is released from touching the screen, the gesture will end and will reset the touched cells values back to zeros.

The data file component work as follows: First, as the user finish every swipe or every tap the system creates and overwrite a text file that has all the touch gestures that were made or being made. Second, the system initializes the user pattern and makes another text file that has the user touch gesture placed in order. Last, the system creates another text file that will be used specifically for the comparison later in the authentication module. This last text file has all required data in reference to the previous text file.

Store data file component, basically it stores all of the collected data in a location on the smartphone that is permitted and accessible to the owner and the system, so if the owner decided to check the data file, it would not cause any problems.

The comparison component works as follows: First, it reads the data text file that was created from the training module and checks how many gestures were made in the training module. Then, depending on the number of gestures an array is created with that number. Second, the system starts by filling in that array with the training module data text file for later use. Third, the system starts

by doing the previous two steps into one, but this time the system uses the newly created text file that was created in the authentication module.

A threshold will be set to determine if the current user is the owner or not, if the difference between the two data files exceeded the threshold, then a command will be called to alert the OS that the current user is not the owner.

IMPLEMENTATION

Since Android is an open-source system, it was perfect for the implementation of the new authentication system and google play store is in constant growth and contains many games to test on the new system. The new system was executed on android 6.0 Marshmallow. First, we need to read the user finger movement on the touch screen but, unfortunately for the protection of the Android system, only uppermost applications can acquire the touch events. Therefore, we cannot benefit from the resources that Android API provides to developers.

As mentioned earlier, the Android API will not give us the privilege of reading the raw data directly from the touch sensor due to security reasons unless we root the Android OS. We decided to run the system from the Linux kernel to overcome this problem and at the same time, we will not need to root the OS. Fig. 2, demonstrates the Android architecture.

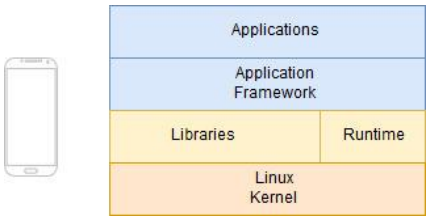

Figure 2: Android architecture.

The first layer is the Linux Kernel; it supplies the functionalities of the system like Power Management, Memory Management, Hardware, and Security Settings. The second layer is the Libraries and Android Runtime, it has the framework common functions, like data storage, graphics rendering, and web browsing. The third layer is the Application Framework, which is the API that authorizes interactions with the Android system. The last layer is the Application layer, which holds the applications, like Gallery, Browser, Camera, Phone, and Music. When the user touches the screen, the screen reads the information in raw data and then send this information to the Linux Kernel. When the Linux Kernel receives the information, it sends it to the Android libraries in the upper layer.

Linux Kernel is the one responsible for managing the sensors in the smartphone and are revealed to us as a device driver. In the smartphone folder, they are placed in the root */dev/*.

Android Debugger Bridge can provide some information about the smartphone, such as the sensors that are installed in the device.

Our concern for the moment is only the touch sensor, and the touch sensor on the test smartphone is recognized under *event6*. Now we will try to read the events from the touchscreen. The good thing for us is that the input devices in the smartphone use a familiar format. The format is available at kernel.org. The event structure contains a value, a code, a timestamp, and a type. Now we need to have all the information about the events in order to use it. We can use "evtest.c" it is available on elinux.org. It is a GNU/Linux application that helps in developing the sensor monitor application.

First, we ran "evtest" and called the event we want to monitor, which in this case is *event6* that is responsible for the touch sensor. The next step we will touch the screen and see the output.

```
Event: time 1503417334.129915, type 3 (Absolute), code 53 (?), value 210
Event: time 1503417334.129915, type 3 (Absolute), code 54 (?), value 397
Event: time 1503417334.129915, ------------- Report Sync ------------
Event: time 1503417334.138511, type 3 (Absolute), code 53 (?), value 178
Event: time 1503417334.138511, type 3 (Absolute), code 54 (?), value 398
Event: time 1503417334.138511, type 3 (Absolute), code 58 (?), value 9
Event: time 1503417334.138511, ------------- Report Sync ------------
Event: time 1503417334.146478, type 3 (Absolute), code 53 (?), value 146
Event: time 1503417334.146478, type 3 (Absolute), code 54 (?), value 399
Event: time 1503417334.146478, ------------- Report Sync ------------
Event: time 1503417334.154908, type 3 (Absolute), code 53 (?), value 112
Event: time 1503417334.154908, type 3 (Absolute), code 54 (?), value 397
Event: time 1503417334.154908, type 3 (Absolute), code 58 (?), value 50
Event: time 1503417334.154908, ------------- Report Sync ------------
Event: time 1503417334.162859, type 3 (Absolute), code 53 (?), value 78
Event: time 1503417334.162859, type 3 (Absolute), code 54 (?), value 396
Event: time 1503417334.162859, ------------- Report Sync ------------
Event: time 1503417334.187932, type 3 (Absolute), code 57 (?), value -1
Event: time 1503417334.187932, ------------- Report Sync ------------
```

Figure 3: Raw data.

After touching the screen, we were able to see the raw data as shown in Fig. 3, which has the format we mentioned earlier. Each code captures different data, for example, code 53 is capturing the *x* coordinates, code 54 is capturing the *y* coordinates, and code 57 is the tap counter.

Our data gathering was conducted using LG K8; it has Quad-Core 1.3 GHz CPU, 1.5 GB ram, 720x1280 pixels screen dimension, and running Android 6.0 Marshmallow. We conducted our experiment on two different games, on fifteen users, and each user played each game for 20 times.

The two games we chose are Pac-man and Angry Birds. Pacman was chosen because it's one of the popular games on google play market, it was downloaded more than 50 million times, and the user will make extensive amount of touch gestures while playing the game. However, Angry birds do not have the same amount of touch gestures like Pac-man but, it's one of the most famous games on google play market, and it has been downloaded more than 100 million times.

TESTING AND EVALUATION

We conducted several experiments to assess the effectiveness of the authentication system in the real environment. As stated before, the authentication system will use games that are available for free to the consumers on google play store.

In the beginning, the owner of the smartphone will take part in the experiment like any other candidate by using the training module. After documenting the owner data, the next step was to determine what is the threshold that would be beneficial for the owner. The last step is to run the authentication module on every user and document the results.

The first game is Pac-man, it is a maze game, where the user is required to develop the best pattern possible to finish the game by collecting all of the points without getting detected. Pac-man was considered as a good candidate for testing the system because it's rich with touch gestures and many different patterns that can be used to finish the game.

The second game is Angry Birds, it is a shooting game, where the game requires the user to destroy the obstacles. Angry Birds was used due to its different nature from the previous game, and it can provide different touch gestures such as tapping.

The threshold of Angry Birds was set to 20% and Pac-man was set to 30% based on the data collected from the training module, which is the maximum difference of touch gestures the owner of the smartphone makes in every try.

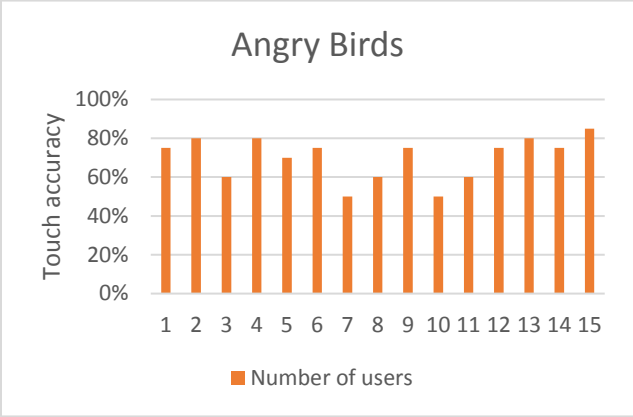

Figure 4: Angry Birds users.

As shown in Fig. 4, the overall touch accuracy of the 15 users playing Angry Birds is 70%.

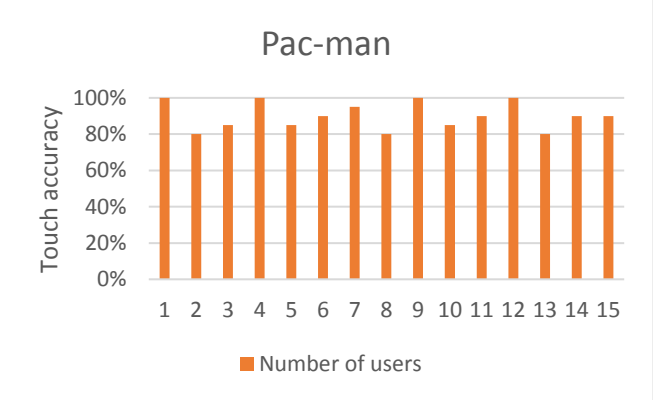

Figure 5: Pac-man users.

As shown in Fig. 5, the overall touch accuracy of the 15 users playing Pac-man is 90%.

CONCLUSION

Limited work has been done with the touch sensor, where most of the previous studies used other sensors and different approaches. The approach in this project takes advantage of the interaction of the users with their smartphones.

In this paper, we introduced an authentication system based on the user behavioural pattern while playing a smartphone game. Pac-man gave the best result of accuracy as it provides an accuracy of 90% among all the users. This is because Pac-man is a maze game that requires the user to collect all points and at the same time not get detected so, there are many different patterns can be followed to finish the game. We managed to authenticate users based on their distinct behaviour pattern. The design of the system is unique; the system can detect different touch gestures and reads its corresponding data directly from the touch sensor. This gives the proposed system the edge over other systems because other systems were designed at the API level, while the proposed system is designed at the Linux kernel level. This concludes that the system can work on any Android device.

Future work of the system, we can use Weka machine software to utilise the raw data and classify it in more specific details. Weka will help in utilising touch gestures in games that have limited touch gestures. In that way, a game like angry birds that has 70% accuracy will increase, and the use of Weka will widen the utilisation of any smartphone game regardless of how many touch gestures the game provides.

REFERENCES

[1] e - learning Best Practice Guide: 2016. *http://www.sme-elearning.eu/pdf/best-practice-guide/SMEELEARN%20-%20Best%20Practice%20Guide.pdf*. Accessed: 2017- 10- 27.

[2] Scindia, P. and Voris, J. 2016. Can mobile device users be identified by how they play a game?.*2016 IEEE 37th Sarnoff Symposium*. (2016). DOI: http://dx.doi.org/10.1109/sarnof.2016.7846774

[3] Yan, J., Blackwell, A., Anderson, R. and Grant, A. 2004. Password memorability and security: empirical results. *IEEE Security & Privacy Magazine*. 2, 5 (2004), 25-31. DOI: http://dx.doi.org/10.1109/msp.2004.81

[4] Uellenbeck, S., Dürmuth, M., Wolf, C. and Holz, T. 2013. Quantifying the security of graphical passwords. *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13*. (2013). DOI: http://dx.doi.org/10.1145/2508859.2516700

[5] iOS 5 iPhone Lockscreen Glitch To Bypass Password Protection: 2012. *http://www.ijailbreak.com/iphone/ios-5-iphone-lockscreen-glitch/*. Accessed: 2017- 10- 27.

[6] How to reset your Android lock screen password: 2012. *http://droidlessons.com/how-to-reset-your-android-lock-screen-password/*. Accessed: 2017- 10- 27.

[7] Abdelrahman, Y., Khamis, M., Schneegass, S. and Alt, F. 2017. Stay Cool! Understanding Thermal Attacks on Mobile-based User Authentication. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. (2017). DOI: https://doi.org/10.1145/3025453.3025461

[8] Qi, M., Lu, Y., Li, J., Li, X. and Kong, J. 2008. User-Specific Iris Authentication Based on Feature Selection.*2008 International Conference on Computer Science and Software Engineering*. 1, (2008), 1040–1043. DOI: https://doi.org/10.1109/csse.2008.1060

[9] Thavalengal, S., Bigioi, P. and Corcoran, P. 2015. Evaluation of combined visible/NIR camera for iris authentication on smartphones.*2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. (2015). DOI: https://doi.org/10.1109/cvprw.2015.7301318

[10] Clancy, T., Kiyavash, N. and Lin, D. 2003. Secure smartcardbased fingerprint authentication. *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications - WBMA '03*. (2003), 45–52. DOI: https://doi.org/10.1145/982507.982516

[11] 2016. Researchers publish method of hacking fingerprint authentication on smartphones. *Biometric Technology Today*. 2016, 4 (2016), 3. DOI: https://doi.org/10.1016/s0969-4765(16)30062-5

[12] Duc, B., Fischer, S. and Bigun, J. 1999. Face authentication with Gabor information on deformable graphs. *IEEE Transactions on Image Processing*. 8, 4 (1999), 504-516. DOI: https://doi.org/10.1109/83.753738

[13] Bhagavatula, C., Ur, B., Iacovino, K., Kywe, S., Cranor, L. and Savvides, M. 2015. Biometric Authentication on iPhone and Android: Usability, Perceptions, and Influences on Adoption. *Proceedings 2015 Workshop on Usable Security*. (2015). DOI: https://doi.org/10.14722/usec.2015.23003

[14] Yampolskiy, R. and Govindaraju, V. 2008. Behavioural biometrics: a survey and classification. *International Journal of Biometrics*. 1, 1 (2008), 81. DOI: https://doi.org/10.1504/ijbm.2008.018665

[15] Wang, X., Guo, F. and Ma, J. 2012. User authentication via keystroke dynamics based on difference subspace and slope correlation degree. *Digital*

Signal Processing. 22, 5 (2012), 707-712. DOI: https://doi.org/10.1016/j.dsp.2012.04.012

[16] An Upper Limit For Apps? New Data Suggests Consumers Only Use Around Two Dozen Apps Per Month: 2014. *https://techcrunch.com/2014/07/01/an-upper-limit-for-apps-new-data-suggests-consumers-only-use-around-two-dozen-apps-per-month/*. Accessed: 2017- 10- 27.

[17] Jorgensen, Z. and Yu, T. 2011. On mouse dynamics as a behavioral biometric for authentication. *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security - ASIACCS '11*. (2011). DOI: https://doi.org/10.1145/1966913.1966983

[18] Bergadano, F., Gunetti, D. and Picardi, C. 2002. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security*. 5, 4 (2002), 367-397. DOI: https://doi.org/10.1145/581271.581272

[19] Frank, M., Biedert, R., Ma, E., Martinovic, I. and Song, D. 2013. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *IEEE Transactions on Information Forensics and Security*. 8, 1 (2013), 136-148. DOI: https://doi.org/10.1109/tifs.2012.2225048

[20] Halprin, R. and Naor, M. 2009. Games for extracting randomness. *Proceedings of the 5th Symposium on Usable Privacy and Security - SOUPS '09*. (2009). DOI: https://doi.org/10.1145/1572532.1572548

[21] Ahmed, A. and Traore, I. 2007. A New Biometric Technology Based on Mouse Dynamics. *IEEE Transactions on Dependable and Secure Computing*. 4, 3 (2007), 165-179. DOI: https://doi.org/10.1109/tdsc.2007.70207

[22] SensorManager, Android Developers: *https://developer.android.com/reference/android/hardware/SensorManager.html*. Accessed: 2017- 10- 27.

[23] Castelluccia, C. and Mutaf, P. 2005. Shake them up!. Proceedings of the 3rd international conference on Mobile systems, applications, and services - MobiSys '05. (2005). DOI: https://doi.org/10.1145/1067170.1067177

[24] Mayrhofer, R. and Gellersen, H. 2009. Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions on Mobile Computing*. 8, 6 (2009), 792-806. DOI: https://doi.org/10.1109/tmc.2009.51

[25] Mantyjarvi, J., Lindholm, M., Vildjiounaite, E., Makela, S. and Ailisto, H. Identifying Users of Portable Devices from Gait Pattern with Accelerometers. *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005*. DOI: https://doi.org/10.1109/icassp.2005.1415569

[26] Okumura, F., Kubota, A., Hatori, Y., Matsuo, K., Hashimoto, M. and Koike, A. 2006. A Study on Biometric Authentication based on Arm Sweep Action with Acceleration Sensor.*2006 International Symposium on Intelligent Signal Processing and Communications*. (2006). DOI: https://doi.org/10.1109/ispacs.2006.364871

[27] Conti, M., Zachia-Zlatea, I. and Crispo, B. 2011. Mind how you answer me!. *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security - ASIACCS '11*. (2011), 249–259. DOI: https://doi.org/10.1145/1966913.1966945

[28] Gallego, A., Saxena, N. and Voris, J. 2013. Exploring Extrinsic Motivation for Better Security: A Usability Study of Scoring-Enhanced Device Pairing. *Financial Cryptography and Data Security*. (2013), 60-68. DOI: https://doi.org/10.1007/978-3-642-39884-1_6