Mateusz TYBURA, Antoni SZCZEPAŃSKI
RZESZOW UNIVERSITY OF TECHNOLOGY, DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
2 Wincentego Pola Str., 35-959 Rzeszów

Touch screen based user identification

Abstract

The main subject of this work was to find a new way of user identification based on his interactions with a touch screen of a detecting device such as smartphone, tablet or operating panel. The entire identification process was performed in implemented mobile software and the identification was based on certain mathematical criteria. No additional external device was connected to the portable device for user recognition process.

Keywords: security, authorization, touch screen, user recognition.

1. Introduction

A lot of modern devices let the user interact with them using a touch screen. These are not only smartphones or tablets but also some laptops with touchable screens. Touch extends the possibility of communication between man and machine, but it is not often used during user authentication process. There exists the authentication method called unlock pattern, but it is still a kind of a password. For most people it is probably much easier to remember unlock pattern than a string of alphanumeric characters, but recognition of the user is not yet performed by his individual manner of interaction with a screen, for example, by the reaction time.

2. Authorization basics

Passing the user login and password is the most common way to check both identity and permissions for access in an operating system or any other application. Everything goes around alphanumeric data stored in human and machine memory. It could be encrypted but still there are many limitations making them less random than it could be. There are no connections with a person so anyone could simply try to guess it. Only limitation of attempts make it harder to break. Passwords are also hardened in the case of eyesight attack which basically consists in looking on a screen to read it. All modern systems use symbols such as stars or dots to cover the original text. So as long as someone has no access to the device, it is secure. Otherwise there are some ways to just copy it and paste to some text editor for making it human readable. Another big issue is connected with human memory. The user can forget the password. Many systems give opportunity for recovering it. As long as it is based on correct answers for questions such as “Who was your favorite teacher?” it is trivial to guess by someone knowing the user.

Nowadays there are more and more systems secured using additional data given to the user in every attempt to login. It can be sent by an SMS or an email message.

The user may have to answer a phone call made by some automatic software. Also there are some specialized devices called tokens, which simply show randomly generated numbers connected with the numbers generated on the second one.

Mobile devices use simpler kinds of authorization such as PIN codes or unlock patterns. The first one is just a four elements set of decimal digits which must be entered in a correct way.

The second one is more complicated. On Android smartphones it is just 9 dots which can be connected with themselves in a finite number of ways. Some researches have already shown the weakness of this solution because of human limitation. In short, it is all about the usage of a letter and shape like patterns which are well known for everyone.

Another method of authorization was introduced by Windows 8. The user can simply choose any image file stored in the internal device memory (flash memory or hard drive) and then make a password using three gestures [1]. It is an interesting but still limited method. It can be noted that people must have some way to repeat the authorization process over and over again. First of all it decreases the number of images which are possible to use. The user must login without problems, so the pictures should have something very characteristic, like a dog’s eye or a flag on the mountain peak.

The latest solutions for authorization are built on biometrics. Things such as the eye iris or the fingerprint are individual but devices such as computers, mobile phones or ATMs cannot use such data without specialized hardware and software. There are also less specialized methods like e.g. gait pattern analysis which can be done on a video image with quite good results [2]. Another way of detection is gathering data about finger veins. This technique is still in development which makes it better and better like the local line binary pattern method [3].

3. The concept of user identification

All analyses showed that any known type of non-biometric authorization had limitations and there was no wide spread, easily usable method for doing it with the usage of individual user’s characteristics. There is a lot of place for something different. This solution must be easy, secure, possible to implement on many devices and easily configurable. Computational complexity should also be taken into account because not all devices have a lot of power and memory to perform very complex calculations.

The authorization should consist of a few steps (Fig. 1). It should be as simple as possible. The first step is to read some data from a sensor or some sensors. Then there must be carried out some computations in order to perform the analysis based on their results. The last step is to recognise the user after the analysis. Wide availability of devices with touch interactions available was the core of our concept of user identification.

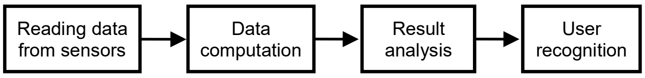

Fig. 1. Basic concept of user identification

Nowadays there are some methods based on touch interaction. One of the solutions is MTi which is a method for user identification on a multitouch display [4]. A high accuracy made it one of the considered methods for development but it has also a big limitation. It must have multitouch detection which is not available in more embedded devices. Also for smartphones it is not something natural, when thinking of how users use their phones for writing SMS messages or browsing the Internet.

Some other researches show the usage of variety of techniques such as bodyprint, which is a scanning of body parts based on placing some of them on a mobile phone screen [5] or keystroke biometrics – observation of how the user interacts with an on-screen keyboard [6]. Data like this is even used in the case of detecting abnormality in behaviour for preventing intrusion and identify some potential malware [7].

Because those methods were not designed especially for the login process and being aware of all devices, even with only single touch detection, we decided to create a new method. The first step is building a mathematical model, i.e. define some basic symbols and definitions. *RF* stands for the function used for recognition. Its input is a matrix named *Tc* which contains all measured data.

$$RF(T_c) \rightarrow output$$

There are at least three ways to think of the function *output*. It can return the vector of recognition Rm which can contain the data in binary or fuzzy logic values

$$R_m = [c_1 \quad c_2 \quad \cdots \quad c_n]$$
$$\text{where: } n \in N_+ \quad c \in \{0;\ 1\} \text{ or } c \in \langle 0; 1\rangle$$

or it can give the most similar user variable *msu* for database with an additional value of the difference *diff* between Tc and Uc . The second one stands for users characterises, which are connected with the users data

$$R_m = [msu \quad diff]$$

There was also a concern of what characteristics would be used for recognition. It was very crucial to choose some which were simple an available to measure on all the considered devices. For the first attempt there were used: the user reaction time and the accuracy of interaction with a touch screen. The values of these two were divided into three parts for storing minimum, average and maximum value.

$$U_c = [T_r \quad A]$$
$$T_r = [T_{min} \quad T_{avg} \quad T_{max}]$$
$$A = [A_{min} \quad A_{avg} \quad A_{max}]$$
$$T_{min}, T_{avg}, T_{max}, A_{min}, A_{avg}, A_{max} \in R$$

The last problem was to choose the right way for calculating the difference. Because of wide mathematical usage – the Euclidean distance was chosen for that purpose. All characteristics were used separately and then summed into one value.

After creating the mathematical model, the conception was extended to more practical aspects. There were a few things to do. First of all, implementation had to be very simple and useful. For this reason the recognition system was presented on the whole screen with full coloured, contrasting circles having sufficient width and height to be easily seen, and not to big because of the screen size and resolution limitations. The reaction time was simply measured from the moment when a circle appeared on the screen till the user touched any region of the screen. The accuracy was defined as a distance between the touching point and the centre of a circle.

Only one problem had to be solved. Because the system had to be simple, fast and for security reasons also fully reliable, it was necessary to make some compromise in the number of interactions in both learning and working modes, i.e. when adding new user characteristics to the internal database and when recognizing someone in the login process. 30 times for learning and 3 times for working mode were chosen for building a prototype and making research of this method of authorization. There was full availability of any change in order to make it better after figuring out it would be necessary.

4. Tests

The ready to use mobile application was started for gathering data in order to make sure that this method was at least useful for basic implementation of the biometric based recognition system. All such systems are focused on making sure that touch recognition can be useful when identifying users in the process of authorization in applications or in an operating system on a device with a built-in touch sensor (usually screen).

The first test consisted of two basic steps. Firstly, the input data about two distinct users were stored in the application internal database (tab. 1). Then every user tried to log 9 times to make sure that it worked.

As Table 1 shows, the reaction time seems to be much more unique than the accuracy. This can depend on the calculation accuracy and the size of the circles on the screen, but the analysis on such small amount of data could give some false conclusions.

Tab. 1. Two users data collected in the first test

User	Reaction time, ms			Accuracy, px		
	Min	Avg	Max	Min	Avg	Max
1	420	505	668	57.901	80.124	115.066
2	558	648	1030	58.949	86.299	107.446
Difference	138	143	362	1.049	6.175	7.620

Obviously, the number of users was very low but it was considered as necessary before testing on more users. The failure in this test would led to designing the whole theoretical model and the prototype from the scratch.

All the login attempts were correct. This test showed that the difference value was not a constant (Tab. 2). As mentioned before, the difference was calculated as the sum of Euclidean distances between all values representing the user. The output is just a real number.

Tab. 2. Differences calculated while login attempts

User	Differences (real numbers)								
1	249.710	332.442	312.043	300.093	148.528	476.064	181.560	248.258	196.977
2	360.814	393.632	230.622	272.049	443.275	442.718	380.917	375.181	479.645

Because of the unstable value, there is no way to minimize the size of the data vector for the user to only one variable. Also there is no possibility to compare the calculated values of the difference with the stored ones. This difference between the data collected when adding users and the data collected on login attempts seems to be a problem which will be solved after more considerations about the method.

The second test was performed after collecting more data. Ten distinct users were stored in the system and then two of them tried to login.

Data were captured on every attempt and after collecting they were marked as correct or incorrect by comparison of the user login and the recognised user login which was stored in the login attempt logs cleared after every user finished the test.

Unfortunately, in this test there were instances of false recognitions. However, the gathered data showed that there were more positive than negative recognitions. At least 5 recognitions were incorrect among 20 attempts, so it was about 25% of all the attempts.

The test showed that there must be some type of a filter designed for limiting false recognition. One of the simplest way to do that is just to calculate some value of the difference which will be as similar as it can be to the user data and as different as it can be to the other user data.

The analysed data was plotted. All the attempts are represented by distinct points in a 2-D figure. Numbers on the X axis denote the number of attempt with its maximum equal to the sum of positive attempts, because the positive and negative attempts were divided into two sets. The Y axis is labelled by the output values of the difference function. The positive recognitions are in blue, and the negative are in red.

The first user had 14 correct and 6 incorrect recognitions (Fig. 2).

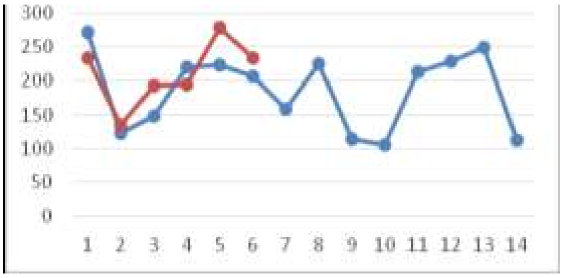

Fig. 2. Plot of the first user login attempts

The points connected with the lines showed that there was no way to present them using a simple linear function $y = a\,x + b$. Both of data sets are rising and decreasing, with one common thing. The lines seems to have the same direction.

With just one user it seems to be impossible to just cut off some values for better recognition. It can be easily seen that in some cases, like for the second attempt, "good" and "bad" points are almost the same. There is also a range between the first and second point where these two lines are collinear. And there are two places on the plot where they have some points in common. All of them are seated near the fourth point of lines.

Even worse lack of floating point operations in some devices would make it harder to recognise. Operations on integer numbers are less precise, so it is quite obvious that lines of correct and incorrect attempts would more likely be the same as for calculations on floating point numbers.

The second user had 15 correct and 5 incorrect recognitions (Fig. 3), which was a little better than for the first user in this test.

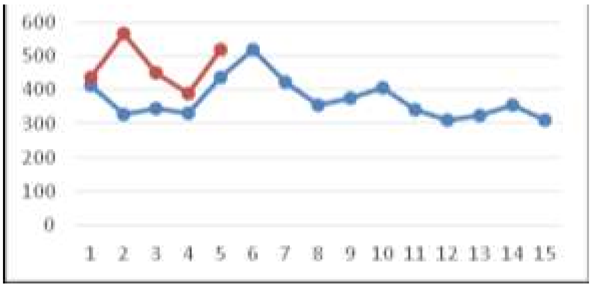

Fig. 3. Plot of the second user login attempts

First of all this data seemed to be more likely to be presented by a basic linear function. There are also differences in the direction. For example, from the 1st to 2nd attempt the correct line goes down while the incorrect line goes up on the Y axis. The lines are more unique without any point in common. Only one (first) point seems to be almost the same for both lines. For this (second) user it looks like there is some value for which there would be about 50% correct guesses without any incorrect. This simple analysis showed that there must be something quite different than one value filtering.

There is also one more problem to consider. What to do when the user cannot login after implementing this filtering part? One way is to just show a failed login attempt message. Other way is to give everyone availability to login every time. It can be achieved by defining some kind of 0 element in the set of users. That element would be returned if after filtering there were no similar user. Of course, this user must have no permission to any element of the secured system.

This secure scenario of 0-user is based on one thing. Giving someone feedback about the failure makes him easier to break through all security. This encourages someone not entitled to more and more targeted attempts to logon failure. It seems to be easy to implement the 0-user scenario, but it could lead to the backfire effect. Any system or device would be easily paralysed by just putting wrong authorization data. After that, even one mistake from someone who is fully authorized to login, there is no simple way to avoid waiting for too much time. If this system is part of life rescuing process, effects of paralyse are quite obvious and very bad.

5. Conclusions

Touch based recognition is something that can be done. The presented attempt of authorization of users seems to be very easy to model, analyse and implement on many touchable devices.

There are many existing methods of touch-based user detection but they are not universal or designed exactly to replace any other type of authorization.

There is still a lot to do in developing a more precise and widely available method of touch-based authorization.

Further development will start after gathering more data. There will be more usage of statistics and methods like SVM or kNN in order to increase the efficiency of this authorization method.

The user recognition method, proposed in this paper, is based on the assumption that the response time and precision of touching the screen are largely unique attributes of each user as a human being. As shown by the preliminary tests, this assumption may be true.

6. References

[1] Pogue D.: Windows 8 the missing manual, O'Reilly, 2013.
[2] Jay Prakash Gupta, Nishant Singh, Pushkar Dixit, Vijay Bhaskar Semwal, Shiv Ram Dubey: Human Activity Recognition Using Gait Pattern. International journal of computer vision and image processing, vol. 3, issue 3, 2013.
[3] Bakhtiar Affendi Rosdi, Chai Wuh Shing, Shahrel Azmin Suandi: Finger Vein Recognition Using Local Line Binary Pattern. Sensors 11(12), 2011.
[4] Blažica B.: The Inherent Context Awareness of Natural User Interfaces: a Case Study on Multitouch Displays, 2014.
[5] Holz C., Buthpitiya S., Knaust M.: Bodyprint: Biometric User Identification on Mobile Devices Using the Capacitive Touchscreen to Scan Body Parts, Yahoo Labs.
[6] Buschek D., De Luca A., Alt F.: Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices.
[7] Damopoulos D.: Anomaly-Based Intrusion Detection and Prevention Systems for Mobile Devices: Design and Development, University of the Aegean, 2013.

Received: 16.09.2015 *Paper reviewed* *Accepted: 03.11.2015*

Mateusz TYBURA, MSc

Mateusz Tybura was graduated in IT (MSc 2015) from Rzeszow University of Technology. Currently he is studying for PhD degree on the same university. He is member of KNEiTI scientific circle. His main research areas are security and mobile technologies.

e-mail: tyburam@hotmail.com

Antoni SZCZEPAŃSKI, PhD, eng

Antoni Szczepański earned his PhD in 2005 in electrical engineering. He works at the Rzeszow University of Technology, in the area of computer simulation of electrical circuits and application of numerical methods to the electromagnetic field calculation. Besides, he is interested in programming languages and algorithms. He has also experience in mobile applications developing.

e-mail: aszczep@prz.edu.pl