International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Touch Screen Based Authentication

Shafeena C P

Dept. of Electronics and Communication Engineering, MIT, Anjarakandy, Kannur, India

Abstract: ***This paper investigates multi-touch gestures for user authentication on touch sensitive devices. A multi-touch gesture matching algorithm robust to orientation and translation was developed. Authentication has become an essential component in daily life. With the booming of smart phone and high-speed wireless networks in recent years, applications and data have been shifting from desktop to mobile devices at a vigorous pace. Although mobile computing provides great convenience in daily life, it becomes vulnerable to various types of emerging attacks. User authentication plays an indispensable role in protecting computer systems and applications, but the development of touchscreen hardware and user habit change post requirements for new authentication methods for mobile and tablets devices. In this paper, I present a robust user authentication scheme using both static and dynamic features of touch gestures. I take advantage of the pressure sensitivity of multi-touch screens to obtain irreproducible biometric patterns. Discriminative features such as distance, angle, and pressure are extracted from the touch-point data, and used in statistical analysis for verification. Based on these results, I conclude that the proposed scheme overcomes the limitations of the existing user authentication methods, and shows great potential to provide robust protection against unauthorized access.***

Keywords: multi-touch gestures; biometric authentication; password; pressure sensitivity

1. Introduction

As one of the predominant ways of communication, mobile phones have become an indispensable part of our daily life. Smartphones are increasingly prevalent and versatile in handling more tasks such as web-browsing, emailing, entertainment, navigation, trading, and online shopping. However, the popularity of smartphones and the vast number of applications also make these platforms more attractive to attackers. Currently, various types of attacks have been conceived and achieved to gain unauthorized access on different smartphone platforms. Protecting a user's personal data on a mobile device is becoming more crucial than ever. As the gateway to computer systems and applications, user authentication is a widely adopted and effective security measure for protecting confidential data. Without a reliable method of user authentication, data access may be granted to unauthorized individuals or groups that steal confidential information for malicious purposes, or in the case of businesses, use the information to their financial advantage. A user's identity can be verified using: (1) something the user has, such as driver's licenses, passport, and etc.; (2) something the user knows, such as password and pin; (3) something the user is, such as fingerprint, iris, and vein patterns; (4) something the user does, such as keystroke, and touch gesture. In consideration of cost, convenience, and feasibility, password has been the most widely used method for user authentication on mobile devices. There are three popular password schemes used in mobile device: text based passwords, shape-based passwords, and biometric passwords. A text-based password is the traditional scheme where a user chooses a string of digits, letters, or symbols as their password. A shape-based password is a more advanced scheme developed for devices with touch screens, which allows the user to perform a single-touch shape-based gesture as the password. A biometric password uses additional biometrical sensors to collect biometric features of the user, and uses them as the password. Biometric passwords eliminate the need for users to memorize potentially many passwords, and at same time increase the security of the authentication. The problem of text-based and shape-based passwords is that they can be easily disclosed and reproduced using traces left on the screen or through shoulder-surfing attack. Biometric password has much higher security, but it may require an additional device or sensor to read the biometric information. The development of touchscreen has greatly reformed human-computer interaction. The two main types of technology used to sense touch are resistive and capacitive sensors. Capacitive touchscreens currently dominate the smartphone and portable electronics markets. Unlike a resistive touchscreen, there is no moving part in a capacitive touchscreen panel, so the risk of screen damage is much lower. The latest generation of touchscreen is equipped with even more sensors, such as a pressure sensor which measures the pressure of the object that touches the screen. The growing understanding of security has increased the demand for convenient and robust user authentication methods. The new method should aim at overcoming the drawbacks of the existing methods, and achieve higher security and reliability. These desired features of user authentication may be accomplished by the fusion of different password schemes. Advancements in multi-touch screen technologies enable the collection of more information related to touch, such as position, pressure and size of multiple touch points. The data can derive some static and dynamic features of a touch gesture. Using the comprehensive feature set, the new approach will be able to enhance the security of user authentication, while still providing a simple user interface. This topic proposes a new approach for user authentication for mobile devices equipped with multi-touch screen. This extracts static and dynamic features from the raw touch-point data, such as distance, angle, and pressure, which reflects the unique characteristics of each gesture patterns and the user's hand geometry. Through statistical analysis, I found that the features such as distance and angle primarily describe the physical shape of the gesture and the features such as pressure and touch-point size mainly represent attributes of the user. In other words, some features tend to distinguish gestures and the others tend to distinguish persons. Thus, here combined different

Licensed Under Creative Commons Attribution CC BY

features to enhance the accuracy of authentication. As a proof of concept, here designed an algorithm for the proposed authentication scheme and implemented a prototype app on Android platform. The app can register multiple users and verify users' identities. Experimental results show that this scheme is promising for accurate user authentication on mobile and tablet devices. User authentication decides whether or not to give a specific user access to a system, an application, or data. Therefore, a robust and efficient user authentication is almost necessary for every computer system. A user can be authenticated by four different ways. The first type of user authentication is based on "something you have", which means a physical token or device that the user has and is only authorized to this user. The second type is based on "something you know", which is some secret information only the user knows, such as a password. The third type is based on "something you are", which usually refers to a user's biometric features, such as fingerprint, iris, and etc. The last type is based on "something you do", which characterizes a user's behavior, such as keystroke and touch gesture. Due to various limitations in design, production, and usage, there are three password schemes primarily used for user authentication in mobile devices: text-based passwords, shape-based passwords, and biometric passwords.

2. Methodology

In recognition of the high-demand for user authentication and the limitations of the existing approaches, I propose a touch screen-based user authentication method using static and dynamic features for mobile devices. It adopts the advantages from shape-based and a biometric password. I take advantage of the advanced multi-touch interface, and authenticate a user with a five-finger gesture. I collect the position and pressure data for the touch-points while a user performs a gesture on the touchscreen multiple times, and then extract static features, such as distance and angle, and dynamic features, such as shape of the palm, and pressure for each fingertip. Through similarity check, a stable pattern is selected and used as the template for later verification.

A. Raw Data Acquisition

Each time a touchscreen senses that an object has touched the surface of the screen, it will capture the x-y coordinates of the touched points, along with time-stamps, labels and the corresponding pressures. But the raw data cannot be directly used to recognize the user. First we extract information about the tracks made on the screen by the user moving his or her fingers, the average pressures, and the angle changes from the raw data. Using these processed features can make finding a match more efficient and accurate.

B. Thumb Detection and Finger Numbering

The numeric labels given to touch-points by Android are assigned, in increasing order, according to the order in which a finger first touches the screen, which means the touch generated by any finger can be assigned any label from 1 to 5. Each time the same gesture is performed, fingers do not necessarily touch the screen in the same order, which will result in the same finger being assigned a different label in different attempts of the same gesture. To verify the multitouch gesture input by comparing its features with the stored pattern, all touch-point data needs to be correctly labeled and ordered in a consistent manner. To do this need to re-order the touch sequences to produce consistent labeling. Due to differences in the palm shapes of individuals and the large diversity of the multitouch gestures chosen by users, the numbering process needs to adaptively and stably give correct labels to fingers. I construct a simple polygon that connects the starting point of each touch sequence. Index 1 is assigned to the touch sequence with the largest total distance between it and the other four vertices. Then I number the remaining touch sequences in a clockwise order starting from the position of the touch sequence with index 1. After the re-order process, the data can be re-labeled in a consistent form, so that we are able to compare touch sequences from different attempts. Figure 1 illustrates the detected thumb in red circle and the assigned finger labels.

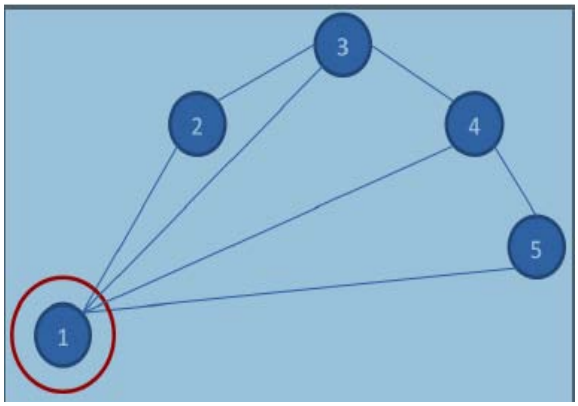

Figure 1: Thumb detection and finger numbering.

C. Distance Features

The trails of the fingertips indicate the basic shape of a gesture. The length of the fingertips trails is one of the features I use to characterize a gesture during the process. The distance feature is the accumulation of all distances between adjacent touch-points. The distance feature of each touch sequence is calculated as follows:

$$\text{distance}_i = \sum_{j=0}^{n} \sqrt{(x_{i,j+1} - x_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2} \quad (i = 1,2,3,4,5) \quad (1)$$

Where x_{ij} is the x coordinate of the j^{th} point of the sequence with index i,y_{ij} is the y coordinate of the j^{th} point of the sequence with index i.

D. Angle Features

Angle is also used to describe the shape of a gesture. The angle feature we extracted is the accumulation of all angle changes between adjacent segments of the selected touchpoint sequence. Other than just the shape of the gesture, angle can also represent the movement of a gesture, which helps increase the accuracy of pattern matching. Figure 2 illustrates the process to compute angle features.

Licensed Under Creative Commons Attribution CC BY

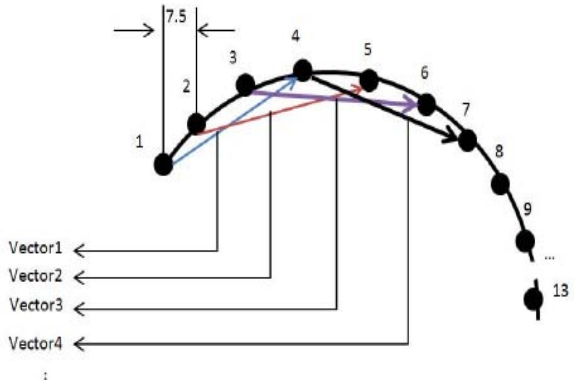

Figure 2: Angle feature extraction.

The locations of the points captured by a device have different densities in different areas of the touch sequence. It is inefficient and inaccurate to calculate angles between every pair of adjacent segments, because the number of short segments is large and the oscillation in short distances is frequent and unpredictable. Therefore, we pick points for the angle calculation based on the length of the entire trail of each touch sequence. We select the points that are the closest to the cumulative distances of 7.5%, 15%, 22.5%... of the total distance of the sequence. After point selection, we need to calculate the angles of the vectors between them. Due to the different sensitivities of the devices, it is possible that there are some mis-captured points. Moreover, the angle is very sensitive to sensor noise and small finger oscillation. To reduce the errors caused by these effects, the angle is measured for every vector connecting two points with a 2-point interval in between. In other words, we calculate the angle of the vector of the points 3 indexes away from each other. For example, angle $_{1,1}$ denotes the angle of points with index 1 and 4 from the touch sequence 1; angle $_{1,2}$ is the angle of points with index 2 and 5 from the touch sequence 1, so in total each touch sequence can generate ten angles between these selected points. The angle is calculated as follows:

$$angle_{i,j} = \arctan\left(y_{i,j+3}-y_{i,j}/\ x_{i,j+3}-x_{i,j}\right) \quad (2)$$
$$(i=1,2,3,4,5)$$

Where y_{ij} is the y coordinate of the j^{th} point of the sequence with index i; x_{ij} is the x coordinate of the j^{th} point of the sequence with index i.

Finally, the accumulated angle is calculated using the following formula.

$$acc_angle_i = \sum_{i=0}^{n} |angle_{i,j} - angle_{i,j-1}| \quad (3)$$
$$(\ i=1,2,3,4,5;\ n=10)$$

E. Pressure Features

Pressures of fingertips are typical biometric features of a multi-touch gesture because they are related to natural characteristics of the human hand and the habits of each individual. In other words, the pressures of the fingertips of each user while performing the same gesture tend to be different from each other. The touch-screen device can capture the pressure of each touch-point in a touch sequence, and the pressure feature of a sequence is the average pressure of all points. The pressure feature of each touch sequence is calculated as follows.

$$AvgPressurei = \frac{1}{n} * \sum_{j=0}^{n} pressure_{i,j} \quad (4)$$
$$(i=1,2,3,4,5)$$

Where $pressure_{ij}$ is the pressure of the j^{th} point of the sequence with index i.

F. Similarity Measure

As mentioned in the previous section, feature data is captured with noise and distortions. Therefore, in the enrollment stage, we require users to perform the same multi-touch gesture at least three times. The program will evaluate the stability of the three attempts and if any of the three attempts is similar enough to the other two, the program will stored that set of data in the database as the template data. Otherwise, the user will be asked to keep performing the same multi-touch gesture until it is accepted. Euclidean distance is used as the matching cost between two touch sequences when checking the similarity between two sets of distance features or two sets of pressure features. A set of inequalities is applied when comparing angle features as described below. The selection of template to be stored in the database is based on the similarity check of these features.

Distance similarity check: Compare the Euclidean distance between the new distance features and the stored distance features with a selected threshold.

$$Euclidean_{D\,(sequence_1, sequence_2)} = \sqrt{\sum_{i=1}^{n}\left(distance_{1,i} - distance_{2,i}\right)^2} \quad (5)$$

Where $distance_{1,i}$ is the computed i^{th} distance feature of the sequence 1 and $distance_{2,i}$ is the computed i^{th} distance feature of the sequence 2.

Angle similarity check: Compare the new angle features with the stored angle features. If the features of more than three fingers are similar, the two sets of features are similar.

$$acc_angle_{1,I} < acc_angle_{2,I} + (acc_angle_{2,I} * 0.1 + e) \quad (6)$$
$$acc_angle_{1,I} > acc_angle_{2,I} - (acc_angle_{2,I} * 0.1 + e) \quad (7)$$

Where $acc_angle_{1,i}$ is the i^{th} angle of touch sequence 1 and $acc_angle_{2,i}$ is the i^{th} angle of touch sequence 2, and e is a constant used to increase the accuracy of the measurement of the angle. When both inequalities are true, then we assume $acc_angle_{1,i}$ and $acc_angle_{2,i}$ are similar to each other. Considering the sensitivity of the angle calculation, it is hard to match all five angles. Based on our tests, we concluded that when more than three pairs of angle are similar, the two sets of angle are deemed to be similar.

Pressure similarity check: Compare the Euclidean distance between the new pressure features and the stored pressure features with a selected threshold.

$$Euclidean_{p\,(sequence_1, sequence_2)} = \sqrt{\sum_{i=1}^{n}\left(avg\ pressure_{1,i} - avgpressure_{2,i}\right)^2} \quad (8)$$

Licensed Under Creative Commons Attribution CC BY

Where $AvgPressure_{1,i}$ is the i^{th} pressure feature of the sequence 1 and $AvgPressure_{2,i}$ is the i^{th} pressure feature of the sequence 2.

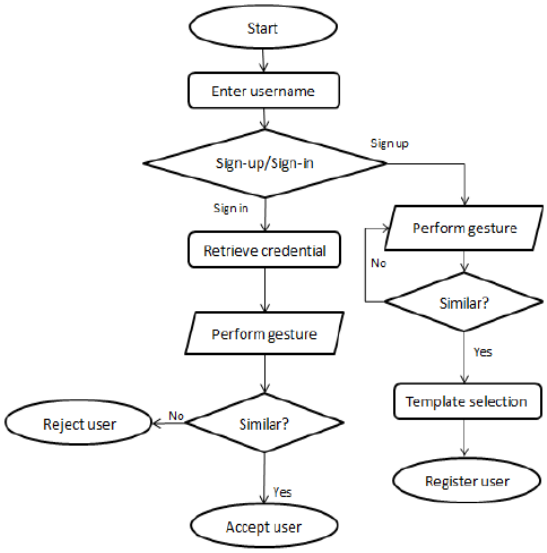

Figure 3: The flowchart of the application implementing the proposed user authentication scheme.

To test this scheme for user authentication on real touchscreen devices, I implement an Android app that implements the aforementioned algorithm. The flowchart illustrates the process of the app. The application starts with a launch screen in which the user is asked to type a username in the text field. Then the user can choose to sign-in or sign-up. The user name check will be performed for both choices. When the user chooses sign-up, the program will check the database for the input User name. If taken, the user will be asked to choose another User name; otherwise the user name will be accepted and the program will switch to the sign-up screen. If the user chooses sign-in, the program will check if the user name exists. If not, the program will pop up a window to inform the user that the user name is not found. When the input username matches a username in the database, the user will be forwarded to the sign-in screen. The sign-up interface is used for user registration. This interface will appear to the user when the username does not match any username in the database or the button Sign-up is tapped. The data collection begins with the user performing a multi-touch gesture. All x-y coordinates, time-stamps, labels, and pressures of 5 touch sequences from 5 fingers are sequentially captured by the device . After obtaining the data for the touch sequences of 5 fingers, the reorder function identifies the thumb and gives all fingers new labels. Then, these reordered data are used to calculate the distance, the average pressure, and the angle of each touch sequence. Finally, the similarity check described previously is applied after the gesture has been performed three times, and the date which passes the check is stored in the database. The screenshot below demonstrates the re-order function. The big circle represents the sequence that is re-labeled as index 1 and the rest of the sequences are re-labeled as 2,3,4,5 in clockwise order. In the application, as a visualization aid, the size of the circle represents the pressure of the touch points.

The sign-in screen handles the verification function. It will be activated when the button Sign-in is tapped and the input username matches a stored username in the database. The data collection process used in this stage is similar to the enrollment stage. After the user performs a gesture, the program extracts distance, angle and pressure features. Then the program runs a similarity check between the input feature data and the stored template feature data. Based on the result of the similarity check, the program decides whether to accept or reject the user.

3. Acknowledgement

First and foremost I thank the Almighty God for all the blessings. Special thanks to my college principal, dean and management for providing me with excellent library which were absolutely necessary for completion of my work. I extend my heartfelt gratitude to Asst. Prof. Neethu.M for her whole hearted support, advice and suggestions as a guide to the work.

4. Conclusion

In this paper, I present a new scheme for robust user authentication. It utilizes multi-touch screen and pressure sensor, and adopts the advantages of shape-based password and biometric password without requiring additional devices. I extract static and dynamic features from raw data. The fusion of the features helps enhance the reliability and stability of the scheme over any single feature approach. The experimental results show that the proposed authentication scheme achieves high accuracy, and the user feedbacks indicate that the users are willing to adopt it for everyday use. I found that the proposed scheme is an important improvement to the existing user authentication methods and has great potential to enhance the security of mobile computing.

References

[1] N. Sae-Bae, N. Memon, and K. Isbister, “Investigating multi-touch gestures as a novel biometric modality,” in *Proc. IEEE 5th Int. Conf. Biometrics, Theory, Appl. Syst.*, Sep. 2012, pp. 156–161.
[2] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, “Biometric-rich gestures: A novel approach to authentication on multi-touch devices,” in *Proc. ACM Annu. Conf. Human Factors Comput. Syst.*, 2012, pp. 977–986.
[3] P. Tome-Gonzalez, F. Alonso-Fernandez, and J. Ortega-Garcia, “On the effects of time variability in iris recognition,” in *Proc. 2nd IEEE Int. Conf. Biometrics, Theory, Appl. Syst.*, Sep./Oct. 2008, pp. 1–6.
[4] D. Rankin, B. W. Scotney, P. J. Morrow, and B. Pierscionek, “Iris recognition failure over time: The effects of texture,” *Pattern Recognit.*, vol. 45, no. 1, pp. 145–150, 2012.
[5] U. Park, R. R. Jillela, A. Ross, and A. K. Jain, “Periocular biometrics in the visible spectrum,” *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 96–106, Mar. 2011.

Licensed Under Creative Commons Attribution CC BY