The Seventeenth International Symposium on Artificial Life and Robotics 2012 (AROB 17th '12),
B-Con Plaza, Beppu, Oita, Japan, January 19-21, 2012

Extraction of Operational Behavior for User Identification on Smart Phone

Yuji Watanabe and Shunta Ichikawa

Nagoya City University, Nagoya 467-8501, Japan
(Tel: 81-52-872-5037, Fax: 81-52-872-5037)

yuji@nsc.nagoya-cu.ac.jp

Abstract: A smart phone has a large amount of private information, so that user authentication and identification are important to prevent attacks by illegal users who are not the owner of the smart phone. Both password authentication and biometrics can be applied only at the beginning of use. After the authentication is passed, not only the legal owner but also illegal users freely use the phone. For the second protection, the behavior-based user identification can continuously check the user activities after login. In this paper, we investigate operational behaviors at the first stage for user identification on smart phone. We make a text browsing application to record fingers history on smart phone. From the recorded fingers history, we extract and compare characteristic operational behaviors, for instance, the speed and the acceleration of fingers, the distance between fingers, and the distribution of touched region.

Keywords: Smart phone, Behavior-based user identification, Operational behavior, Security

1 INTRODUCTION

In recent years, smart phones have been exponentially popularized for many convenient applications. Because a smart phone has as a large amount of private information as a personal computer, user authentication and/or user identification are important to prevent attacks by illegal users who are not the owner of the smart phone. At first, user authentication can be classified into two types, namely *password authentication* and *biometrics*. Although the password authentication requires that the owner only memorizes a short password and inputs the password for login, the password is probably lost or forgotten and then it is illegally shared and used by attackers. On the other hand, the biometrics verifies the owner based on the intrinsic physiological characteristics, for example, fingerprint, face and voice [1]. The biometric characteristics cannot be lost or forgotten, so that it is hard to copy and share them. However, the user authentication of both password authentication and biometrics can be applied only at the beginning of use. After the authentication is passed, not only the legal owner but also illegal users freely use the computer or the phone.

For the second protection, many user identification systems based on *user behavior* have been proposed since the 1990s in personal computer environment. As user behavior which is difficult to be imitated, commands sequence [2-4], keystroke pattern [5-9], or mouse operation [10] has been employed. The identification systems initially create normal profiles of a legitimate user behavior. If the systems observe the remarkable difference between the profiles and the current user activities, then they give an alarm. The behavior-based user identification can continuously check the user activities after login.

Isohara et al [11] have proposed a simple and easy-to-use user identification system on mobile phone environment for the first time. The system records keystrokes in the background process, calculates the frequency of keys or key groups from the recorded keystrokes to make a normal user profile, and then estimates a similarity score by comparing the frequency-based profile with the latest keystrokes to detect illegal users. A prototype system was implemented on the BREW emulator and error rates were evaluated. Because the system was proposed for mobile phone with severely limited calculation resources, it used only the key frequency for user identification. However, since smart phone has more powerful calculation resources than mobile phone, identification systems based on other complicated characteristics such intervals of time between keys can be applied on smart phone. In addition, unlike mobile phone where only key buttons are pushed, there are also operational behaviors peculiar to smart phone with touch-screen panel, for example, *swipe*, *tap*, *flick*, and *pinch*.

In this paper, we investigate operational behaviors at the first stage for user identification on smart phone. We make a simple text browsing application to record fingers history on smart phone, that is, iPhone, iPod touch and iPad. From the recorded fingers history, we extract and compare characteristic operational behaviors, for instance, the speed and the acceleration of fingers, the distance between fingers, and the distribution of touched region. From the experiment

results, the distribution of region touched by fingers for each subject is interestingly different.

2 HISTORY RECORD APPLICATION

2.1 Platform for Development

To create an application to record fingers history on smart phone, we should firstly select a platform for developing the application. Although Google's Android became the world's leading smart phone platform in the last quarter of 2010 [12] (also in Japan [13]), we have developed some applications on iOS for a year (as a matter of course we start to develop on Android OS). iOS is Apple's mobile operating system for iPhone, iPod touch, iPad and Apple TV [14]. The user interface of iOS is based on the concept of direct manipulation, using multi-touch gestures. As interface control elements, there are sliders, switches, and buttons. Interaction with the OS includes gestures such as *swipe*, *tap*, *flick*, and *pinch*, all of which have specific definitions within the context of iOS and its multi-touch interface.

The iOS SDK (Software Development Kit) allows developers to make native applications for iPhone, iPod touch and iPad, as well as test them in an iPhone simulator. Xcode is the integrated development environment for the iOS SDK. Like iOS and Mac OS X, iPhone applications are written in Objective-C. iOS has four abstraction layers: the Core OS layer, the Core Services layer, the Media layer, and the Cocoa Touch layer. The developers mainly use the Cocoa Touch layer at the highest level because many sophisticated services and technologies can be implemented easily. However, lower level layers may still be accessed by the developer when needed.

2.2 Text Browsing Application

Like the previous studies on behavior-based user identification such as mouse operation [10] and mobile phone keystrokes [11], it is preferable that user behavior is recorded in the background process. Because multitasking is supported for iOS through only seven background APIs: background audio, voice over IP, background location, push notifications, local notifications, task completion, and fast app switching [14], it is difficult to store operational behaviors in the background process on iOS.

Because browsing is one of basic functions equipped with many applications on smart phone, as the first stage for user identification on smart phone, we make a simple text browsing application to record fingers history on iPhone and iPod touch as shown in Fig.1 (a). The display size is 480 pixels high by 320 pixels wide (iPhone 4 and 4th generation iPod touch has 960 x 640 pixels, and iPad has 1024 x 768 pixels). So the text browser has only vertical scroll. A text is scrolled by the following fingers operations.

- *Flick*: users place a finger on the screen and quickly swipe it in the desired direction.
- *Drag*: users place a finger on the screen and move it in the desired direction without lifting it from the screen.

Fig. 1 (b) illustrates that the origin is located at the top left corner of the screen, and the X and Y coordinates are specified. The UITextView Class which implements the behavior for a scrollable, multiline text region is used to display multiple lines of text, such as when displaying the body of a large text document. The class can get touched points not on screen but on text, so that the range of Y varies depending on the text document while the range of X is from 0 to 320. The Japanese text document in Fig. 1 (b) has Y ranging from 0 to 20000.

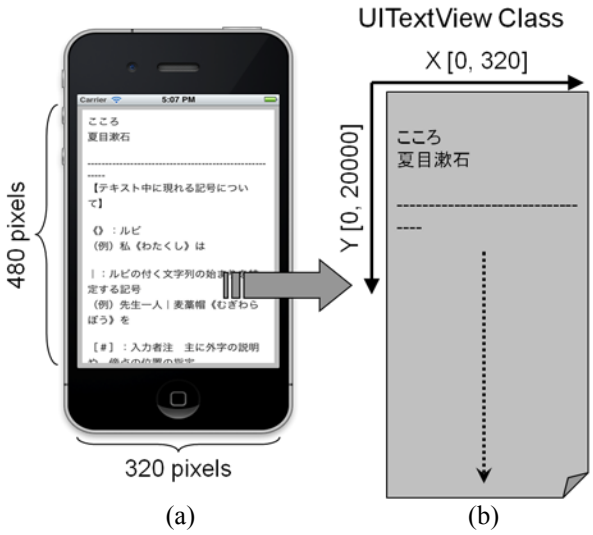


Fig. 1. (a) screenshot of the simple text browsing appli cation to record fingers history on iPhone and iPod to uch. (b) the origin and the X and Y coordinates on U ITextView Class

2.3 Characteristic Operational Behaviors

The fingers history continuously recorded by the text browsing application is in the form of $\{(x, y), t, event\}$, where (x, y) is the coordinates of point touched by fingers, t is the touch event time. And *event* means one of 3 touch events as follows.

1. A finger is placed on the text.
2. The finger is moving without lifting from the text.
3. The finger is released.

From the recorded fingers history, we should extract characteristic operational behaviors suitable for user identification. This study is propably closer to the previous research on mouse operation [10] than on mobile phone keystrokes [11]. The research on mouse operation [10] has checked the following features and then used the last one for user identification.

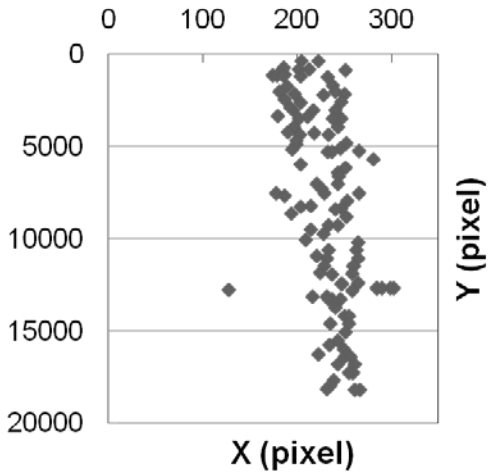
1. The region of mouse used by each user
2. The moving route of mouse from the starting point to the ending point
3. The velocity of mouse
4. The acceleration of mouse
5. The movement direction near the starting point and the ending point
6. The pattern of mouse movement

We also extract the same features, that is, the distribution of region touched by fingers, the speed and the acceleration of fingers, the movement direction, and the operations pattern of fingers.

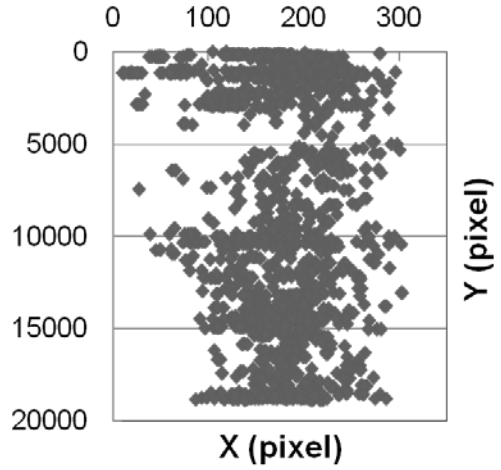
3 EXPERIMENTAL RESULTS

We carried out experiments using the text browsing application on iPod touch to extract characteristic operational behaviors. 5 subjects participated in this experiment. After the subjects were explained how to use iPod touch and the application, they were free to read the text document. When they finished reading, they returned iPod touch. We can get each recorded fingers history through iTunes.

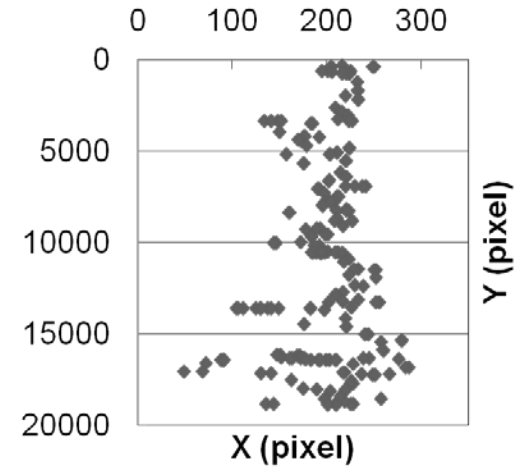
Fig. 2 shows the distribution of region touched by fingers for 3 subjects, A, B, and C. From the results, subject A and C are accustomed to operation on smart phone while subject B may be not used to reading the text document on smart phone. The distribution can be one of promising characteristic operational behaviors suitable for user identification. Table 3 depicts average, minimum and maximum speeds of fingers for 3 subjects. The speed is calculated by dividing the distance between *event* 1 and the corresponding *event* 3 by the taken time. Although subject C can manipulate smart phone more quickly than the others, the feature is invalid for user identification because the minimum and maximum speeds are overlapped. The acceleration of fingers is also inappropriate. We are now extracting the other features for all the subjects including the remaining subjects, so that we will show the results at conference site.



(a) subject A



(b) subject B



(c) subject C

Fig. 2. The distribution of region touched by fingers for 3 subjects

Table 1. Average, minimum and maximum speeds of fingers for 3 subjects

	A	B	C
Ave. speed	104.0	119.4	86.9
Min. speed	8.2	0	0
Max. speed	307.7	9749.3	608

4 CONCLUSION

In this paper, we made a text browsing application to record fingers history on smart phone and then extracted characteristic operational behaviors for user identification on smart phone. From the experiment results, the distribution of region touched by fingers for each subject was interestingly different. In future work, we will continue to extract and compare characteristic operational behaviors. We then plan to construct user identification system on smart phone based operational behavior.

REFERENCES

[1] Jain A, Bolle R, Panakanti S (1999), Biometrics: Personal Identification in Network Society. Kluwer Academic Publishers

[2] Shirai H, Nishino J, Odaka T, Ogura H (1999), An Intrusion Detection Technique Using Characteristics of Command Chains in an Interactive Computer Environment. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, J82-A (10):1602-1611

[3] Schonlau M, DuMouchel W, Ju W, Karr A, Theus M, Vardi Y (2001), Computer Intrusion: Detecting Masquerades. Statistical Science, 16(1):58-74

[4] Okamoto T, Ishida Y (2009), An Immunity-Based Anomaly Detection System with Sensor Agents. Sensors, 9(11): 9175-9195

[5] Joyce R, Gupta G (1990), Identity Authentication Based on Keystroke Latencies. Communications of the ACM, 33(2):168-176

[6] Leggett J, Williams G, Usnick M, Longnecker M (1991), Dynamic Identity Verification via Keystroke Characteristics. International Journal of Man-Machine Studies, 35(6):859-870

[7] Monrose F, Rubin A (1997), Authentication via Keystroke Dynamics. Proceedings of the 4th ACM conference on Computer and communications security, 48-56

[8] Cho S, Han C, Han D, Kim H (2000), Web-Based Keystroke Dynamics Identity Verification Using Neural Network. Journal of Organizational Computing and Electronic Commerce, 10(4):295-307

[9] Bergadano F, Gunetti D, Picardi C (2002), User Authentication through Keystroke Dynamics. ACM Transactions on Information and System Security, 5(4):367-397

[10] Izumi M, Nagao W, Miyamoto T, Fukunaga K (2004), User Identification System Using Feature of Mouse Operation. IEICE Transactions on Communications, J87-B(2):305-308

[11] Isohara T, Takemori K, Sasase I (2008), Anomaly Detection on Mobile Phone Based Operational Behavior. IPSJ Journal, 49(1):436-444

[12] Alto P (2011), Google's Android Becomes the World's Leading Smart Phone Platform. Canalys research release 2011/013, http://www.canalys.com/static/press_release/2011/r2011013.pdf

[13] http://www.m2ri.jp/newsreleases/main.php?id=010120110510500

[14] iOS: http://en.wikipedia.org/wiki/IOS