Inconspicuous Personal Computer Protection with Touch-Mouse

Ming-Chun Huang[1], Wenyao Xu[2], Jason J. Liu[1],
Yi Su[1], Lei He[2], and Majid Sarrafzadeh[1]

[1] Computer Science Department, University of California, Los Angeles
[2] Electrical Engineering Department, University of California, Los Angeles

Abstract. We present a hassle-free personal information protection design that continuously monitors user identity with a Microsoft touch-mouse [1] under a windows-based computer environment. This is the first design which investigates the relationship between time-indexed pressure map trajectories extracted from a touch-mouse and user behavior patterns categorized by common mouse action primitives. This design serves as an assistive method to enhance existing password and biometric based security mechanisms, enabling continuous and unobtrusive personal identity monitoring. Commercialized windows-based systems can be seamlessly integrated with the proposed system and this design can offer a convenient and lightweight solution for physical computer intrusion detection.

1 Introduction

Personal computer (PC) safety and security issues have received increasing public attention in recent years because more and more private information, ranging from work-related documents, emails to life-related family photos, social networking and chatting history, are stored electronically. The increased reliance on computers for daily personal activities makes the cost of losing computer information expensive and unacceptable. In order to enhance personal information safety, modern personal computers optionally enable built-in password managers, fingerprint scanners, voice recorders, and even vision pattern trackers shown in Fig. 1 to assist PC owners to be more aware about preserving their private information [2,3]. In general, the methods to maintain safety and security are categorized into three types. Biometrics is a commonly used solution for user identity verification, such as voice recognition [4], fingerprint recognition [5], iris recognition [6], and face recognition [7]. Biometrics-based identification methods are reliable and unique for individual users. However, hardware and dedicated biometric sensor setup is usually mandatory in order to sample special biometric features. Moreover, biometric identification process is often complicated and time consuming. Voice and fingerprint recognitions takes 10-30 seconds. Reliable iris and face recognitions takes a couple of minutes. Therefore, biometric checking is only requested at the login stage. On the other hand, for password-based protection, no extra hardware is required but a password-based system security

L. Marinos and I. Askoxylakis (Eds.): HAS/HCII 2013, LNCS 8030, pp. 29–38, 2013.

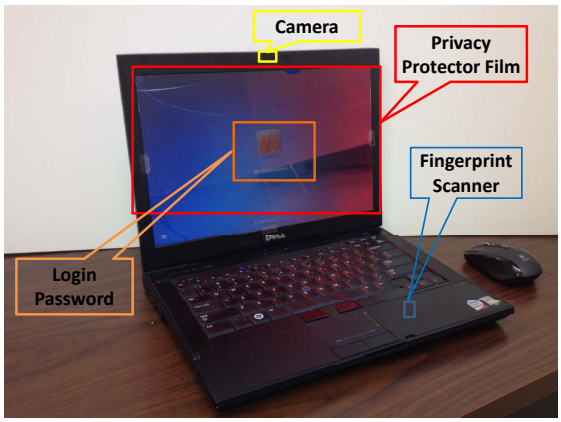


Fig. 1. Commonly used user identity verification tools

is highly-related to the complexity of the password. For example, whenever a user creates his/her own password, it is recommended to include a combination of numbers, letters and special characters such as $*$, !, @ with sufficiently long password length. Both biometric or password-oriented designs are usually exploited as one-time user identity verification methods only at the login stage, rather than serving as a continuous user identity monitoring method. These designs inherently interfere with user activities and require users' attention to complete the verification processes. Therefore, if a user forgets to logout his/her own PC, all these safety mechanisms fail to protect user information. To compensate this obvious drawback for both biometric and password based system, behavior pattern based methods, enabling the concept of continuously analyzing user inputs and correlating with user identities, have been proposed. Because these new methods do not require any user attention while monitoring, they are preferable to serve as a continuously monitoring system and are used to provide user identity re-authentication for reducing the risk of unauthorized PC abuse. Researchers have developed systems to analyze behavior patterns collected from keyboard, mouse, and memory usage to continuously monitor and detect abnormal PC access patterns. F. Monrose and A. Rubin [8] proposed an authorization method via keystroke dynamics. M. Rusara and C. Brodley [9] used mouse movement and flicks to re-authenticate the user identity on the fly. Furthermore, there were methods detecting indirect user behaviors such as audit logs [10], call-stack [11] and call-trace [12], to verify user identity. While these systems offered continuous re-authentication for intruders defense, the applicability of these systems was limited due to the fact that users tended to adjust their behaviors while using a variety of computer applications. To tackle this issue, instead of using behavior features which were application dependent, we found that the relationship between touch-mouse pressure maps and mouse events rarely changed under a variety of applications. By using these application independent features, our novel security add-on with a touch-mouse is capable of robust, continuous, and inconspicuous monitoring of user identity across a variety of applications.

2 Background

2.1 Microsoft Touch-Mouse

Our system adopts a novel re-authentication method by extracting user behavior patterns from a Microsoft touch-mouse. A photograph of the Microsoft touch-mouse and its pressure map is shown in Fig. 2. In addition to providing the same mouse event information as a traditional mouse (mouse displacement, flicking interval and frequency), the touch-mouse is equipped with a highly reliable and calibrated surface capacitive sensor array. Each Microsoft touch-mouse has 195 capacitive sensors in total (13×15) covering the mouse surface. Quantified capacitance values can be utilized to estimate finger pressure values via computing a function of contact area and the width of the interval between two sides of a capacitor. Pressure distributions on a palm and fingers can be easily sampled and recorded by the touch-mouse. Our testing results revealed that the harder a finger presses on the mouse surface, the higher the capacitance value returned. Different users generate different pressure maps shown in Fig. 2, because each user has his/her own habit of holding a PC mouse.

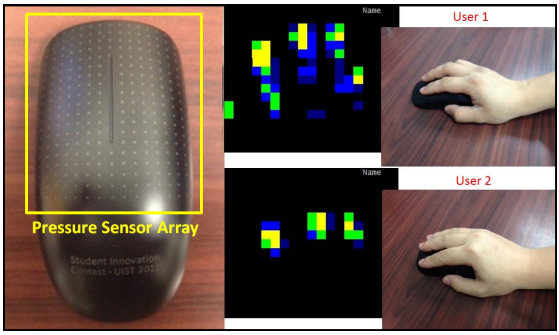


Fig. 2. Microsoft touch-mouse and user pressure map samples

2.2 Related Work

Under Graphical User Interface (GUI) environments, the most common mouse activities are moving and button flicking. The combinations of a sequence of moving and flicking generates plenty of compound mouse events, such as drag and drop, and batch selection. To better support navigation functionality in GUI environment, a mouse wheel was introduced for scrolling web pages and documents directly. Later, the touch-mouse was further enhanced to improve the user experience. The concept of mouse gestures, along with the invention of touch-mouses, are designed to provide users more power in controlling their computers, such as zoom in/out the current window and rotate a photo. In Fig. 3, commonly used mouse gestures provided by Microsoft touch-mouse are shown and these gestures can be viewed as features which are exploited in this design to model user behaviors for user identity verification.

Fig. 3. Microsoft touch-mouse events categories

By using the APIs provided by Microsoft, time stamps, button flicking events, and mouse displacements can be easily extracted and by grouping these simple mouse events together, compound features can be created accordingly. In the prior literature of mouse behavior pattern monitoring, only simple mouse events were extracted to perform user behavior identification. With the lack of palm and finger pressure images, previous experiments primarily utilized time stamps, relative mouse displacement information, and mouse button flicking events as features to train and test a variety of classifiers. Pusara and Brodley [9] showed a decision tree classifier which could discriminate multiple users with a false positive rate of 0.43% and a false negative rate of 1.75% in a well-controlled environment. Ahmed et al. [13] used neural networks to perform a series of classification experiments which resulted in low recognition error rate with limited and known actions among a small group of users. Although their experimental results looked very promising, they pointed out that analyzing mouse movements alone was still not sufficient for user re-authentication. This was because user behaviors in using a mouse could vary dramatically across different applications. Mouse event patterns collected from an exciting game can be far away from the patterns extracted by using a photo editing application. Therefore, a trained classifier by recognizing a user identity from one application might not be able to be applied to other applications. In reality, the range of the applications that a user might use may be unlimited.

3 Behavioral Model of Touch-Mouse

Our idea originates from the observation that mouse events are correlated with their corresponding pressure map sequences and their relationship does not significantly vary under different application scenarios. Instead of analyzing mouse

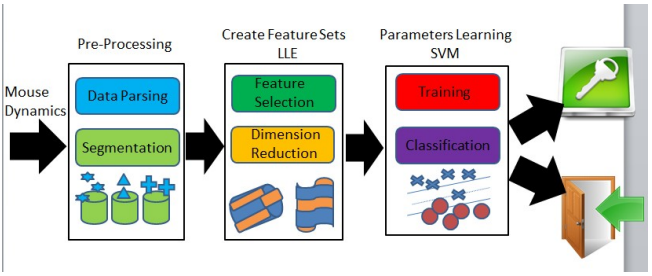


Fig. 4. Framework of behavior recognition

events and pressure map sequences separately, we propose to synchronize mouse events with the corresponding pressure map sequences via time stamps as indices and analyze both information together. This observation provides a fundamentally different basis from those existing mouse-based user behavior analysis methods. By creating a novel mapping between mouse events and a sequence of pressure maps, the dependence between the proposed re-authentication algorithm and the type of applications can be greatly reduced. Therefore, user behavior patterns are not unlimited anymore, but depend on the number of the action primitives supported by Microsoft touch-mouse. In addition, different users are likely to have distinct habits in using their touch-mouse, such as flicking different locations of a touch-mouse surface or scrolling with different forces; therefore, this method is able to effectively identify the inter-class differences and tolerate minor intra-class variations. In this paper, targeted action primitives are categorized by static holding, dynamic clicking/moving, and scrolling, which are commonly used in PC applications.

4 Algorithm for User Re-authentication

The user identification problem can be formulated into a standard learning and testing procedure as shown in Fig. 4. The key idea in this paper is to pair pressure map sequences with the available mouse action primitives and generate low dimension trajectories of the pressure maps as training and classification features. Our program starts when a target PC is powered on. It stays in memory and runs as a background process. Mouse pressure maps are provided by calling Microsoft touch-mouse APIs and mouse events are recorded via a regular mouse event library supported in C#.

Basic algorithm flow was described as follows:

1. Data Parsing: parse collected pressure maps and mouse event logs; After data parsing, a sequence of time-indexed pressure maps which contain 13*15 pixels and a sequence of time-indexed mouse event logs is generated.
2. Segmentation: data segmentation based on the mouse event logs; This stage tries to group a sequence of pressure maps and mouse events to generate

meaningful segmentations. The primary delimiters for segmentation are the timing of button press/release, mouse relative displacements, and the center of mass of the palm and finger pressure values. After partitioning, a sequence of pressure maps and mouse events are grouped into a series of time-indexed units, which are the smallest units used for feature selection stage.

3. Feature Selection: feature selection based on the common mouse action primitives; The simplest mouse action is static holding and the most complicated mouse action primitives in the proposed detection framework are drag and drop, and finger scrolling. Drag and drop is composed of an initial mouse button press, hold, mouse move, and ends with the mouse button release. Scrolling is composed of a similar process but instead of detecting mouse displacements, it exploits the center of mass of the palm and finger pressure to determine dragging. In total, eleven features are selected.
4. Dimension Reduction: Extracted features from the procedure above include a sequence of pressure maps and mouse action primitives; Locally Linear Embedded (LLE) dimension reduction algorithm [14] transforms pressure map information into dimension-reduced 2-D trajectories. These generated mouse trajectories and action primitives become the inputs for the cluster training and input classification.
5. Training and Classification: trajectory samples are trained and classified by the Supported Vector Machine (SVM) program; Basic binary training and classification procedures are applied on the trajectory and action primitive sample pairs from both training and testing data sets [15, 16].

The proposed framework provides continuous re-authentication based on the classification results. We utilize the built-in password function in PC to verify user identity if a re-authentication signal is triggered. Nevertheless, we did not limit the possibility of using other identity verification methods. With appropriate programming integrations, this framework should be able to seamlessly combine with most existing biometric or password-oriented verification mechanisms.

5 Evaluation

In this section, we discuss a series of experiments to evaluate the proposed re-authentication method. The first experiment tested the repeatability of our algorithm. The second one investigated the validity of our assumption: mouse events are highly correlated with their corresponding pressure map sequences and this relationship does not significantly vary under different application scenarios. The third experiment investigated the evaluation of defense against intruders. We would like to investigate how likely intruders could bypass our re-authentication system. We specially compared the performance of static mouse action primitives with the compound/dynamic action primitives to investigate if pressure maps could effectively assist the system to defend against intruders under a variety of applications.

5.1 Evaluation of Mouse Action Primitives Repeatability Off-Line Testing

This experiment primarily investigated the repeatability of the proposed design support in a well-controlled environment. Both training and testing processes were conducted off-line.

Experimental Setup and Description. 15 subjects participated in data collection, in which a total of 11 mouse action primitives were included. In each mouse action primitive, each subject was asked to repeat the experiment 100 times. In the experimental processes, we noticed that the user behaviors tended to be biased. For instance, if a tester had limited or no experience with a Microsoft touch-mouse, his/her behavioral patterns on touch-mouse was not significant. Sometimes he/she may even accidentally perform incorrect actions, such as doing scrolling when a zoom in action was expected. Therefore, an off-line training for each subject was given. Every subject was given specific instructions on how to use touch-mouse and they were given time to get used to the mouse before starting data collection processes. To minimize unnecessary interrupts, a background software was programmed to record the mouse actions without interrupting the experiment progress.

Evaluation Result Analysis. In the evaluation phase, the collected data set was divided into two categories. Half of them were used for training, and the remaining data was used for testing. On average, in more than 80% of the testing cases, our training data set could correctly recognize user identities. The highest accuracy was in the static hold primitive and the lowest accuracy was in three-finger scrolling primitive. Our result revealed that with more mouse dynamics came more variations in the data set. This result suggested using static mouse features over dynamic features in user identity monitoring under a fixed and controlled data collection environment.

5.2 Evaluation of Mouse Action Primitives Repeatability Testing Under Distinct Applications

This experiment was designed to test if the proposed system could survive under distinct applications. In other words, even though the behavior of a subject using a touch-mouse was different in distinct applications, the relationship between pressure maps and mouse action primitives should not significantly vary across distinct applications.

Experimental Setup and Description. The experimental setup was very similar to the first experiment. However, all data set gathered from the previous experiments were reused as a training data set only. Testing data was collected on-line for 10 minutes. In the 10 minutes, subjects were requested to play computer games, surf web, and editing photos randomly. These three totally

different applications were selected in order to verify if the proposed features had any dependence on the type of applications. During testing, the frequency of a re-authentication warning was recorded.

Evaluation Result Analysis. A promising result revealed in Fig. 5 that, compared with the first experiment, the accuracy was similar across different types of applications. The selected features apparently had little dependence on application types. Since three finger scrolling was not used in gaming, it was not recorded and compared in this experiment. Nevertheless, static mouse actions still on average had higher accuracy than compound/dynamic mouse action primitives.

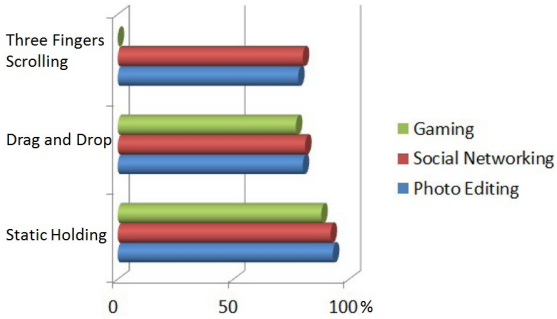
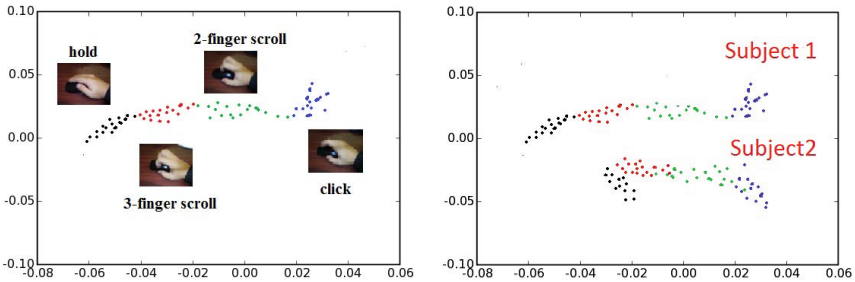


Fig. 5. Our design provides consistent accuracy under distinct applications. Three-finger scroll function is not used in gaming application.

5.3 Evaluation on Defense against Intruders

From the previous two experiments, static mouse action tended to result in a better accuracy. The following experiment attempted to understand if static mouse action primitives could effectively prevent unauthorized intruders as well.

Experimental Setup and Description. In this experiment, we investigated if the proposed algorithm could defend against intruders with unauthorized PC access by using static, dynamic action primitives, and pressure map trajectories. Different subjects have different behaviors in using their touch-mouse and the manifold trajectories were different for distinct action primitives as shown in Fig. 6. Three subjects were randomly picked as PC account owners. The data set of the selected subjects were trained and loaded separately in different user accounts, and then the remaining 12 people were asked to attack the trained system. To increase the success of attack, these three selected subjects showed how they used their touch-mouse to other participants.



(a) (b)

Fig. 6. Visualization of dimension reduced trajectories and hand postures. Visualization of two dimension reduced trajectories from two selected subjects.

Evaluation Result Analysis From the experimental result, static movement, 36% intrusion rate, could not effectively prevent intruders attack, because after few trial-and-errors, static mouse action primitives could be mimicked eventually. Nevertheless, we did not observe any complex dynamic mouse action primitives and pressure map trajectories pairs, such as drag and drop, or scrolling, that could be mimicked by any attacker. This observation provided a design trade-off between user-friendliness and security level. If a computer contained highly confidential information, it should utilize all mouse action primitives and pressure maps to increase data safety. Inevitably, false-alarms might happen more frequently and take more user attention to respond to the re-authentication requests.

6 Conclusion and Future Work

Our design can effectively integrate with existing security mechanisms to provide continuous and hassle-free re-authentication by using a commercially available product - Microsoft touch-mouse. This system can serve as a robust abnormal pattern detector, which delegates final verification tasks to the existing password or biometric-based verifiers. To demonstrate the validity of this system, we performed experiments with 15 subjects under three type of application scenarios: web page-browsing, gaming, and photo editing. The experimental results demonstrated the possibility of preventing physical computer intrusion with a touch-mouse based user identity monitoring system.

References

1. Microsoft touch-mouse - Microsoft hardware, `http://www.microsoft.com/hardware/en-us/products/touch-mouse/microsite/`
2. Liu, S., Silverman, M.: A practical guide to biometric security technology. IT Professional 3(1), 27–32 (2001)

3. Prabhakar, S., Pankanti, S., Jain, A.K.: Biometric recognition: Security and privacy concerns. IEEE Security Privacy Mag. 1(2), 33–42 (2003)
4. Gish, H., Siu, M.H., Rohlicek, R.: Segregation of Speakers for Speech Recognition and Speaker Identification. In: Proc. of Conference on Acoustics, Speech, and Signal Processing, pp. 14–17. ACM Press (1991)
5. Isenor, D.K., Zakey, S.G.: Fingerprint identification using graph matching. Pattern Recognition 19(19), 113–122 (1986)
6. Wildes, R.P.: Iris recognition: an emerging biometric technology. Proc. of the IEEE 85(18), 1348–1363 (1997)
7. Lanitis, A., Tayler, C.J., Cootes, T.F.: Automatic face identification system using flexible appearance models. Image and Vision Computing 13(13), 393–401 (1997)
8. Monrose, F., Rubin, A.: Authentication via Keystroke Dynamics. In: Proc. of Conference on Computer and Communications Security, pp. 48–56. ACM Press (1997)
9. Rusara, M., Brodley, C.: User Re-Authentication via Mouse Movements. In: Proc. of VizSEC/DMSEC, pp. 48–56. ACM Press (2004)
10. Igun, K., Kemmerer, R.A., Porras, P.A.: State Transition Analysis: A Rule-Based Instrusion Detection Approach. In: Proc. of Software Engineering, pp. 181–199. ACM Press (1995)
11. Feng, H.H., Kolesnikov, O.M., Fogla, P., Lee, W., Gong, W.: Anomaly detection using call stack information. In: Proc. of IEEE Symposium on Security and Privacy, pp. 62–78. IEEE Press (1998)
12. Denning, D.E.: Forgot username/password? Software Engineering 2(13), 222–232 (1987)
13. Ahmed, A.A., Traore, I.: A new biometric technology based on mouse dynamics. IEEE Transaction on Dependable and Secure Computing 4(3), 165–179 (2007)
14. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research 4, 119–155 (2003)
15. Liu, J.J., Xu, W., Huang, M.C., Alshurafa, N., Sarrafzadeh, M.: A dense pressure sensitive bedsheet design for unobtrusive sleep posture monitoring. In: IEEE International Conference on Pervasive Computing and Communications, IEEE (2013)
16. Kouropteva, O., Okun, O., Pietikainen, M.: Classification of handwritten digits using supervised locally linear embedding algorithm and support vector machine. In: Proc. of European Symposium on Artificial Neural Networks, pp. 229–234. ESANN (2003)