On the Privacy-Preserving HCI Issues*
(Extended Abstract)

Taekyoung Kwon[1], JongHyup Lee[2], and JooSeok Song[2]

[1] Dept. of Computer Engineering, Sejong University, Seoul, 143-747, Korea
[2] Dept. of Computer Science, Yonsei University, Seoul, 120-749, Korea
tkwon@sejong.ac.kr,
{jhlee,jssong}@emerald.yonsei.ac.kr

Abstract. Actual interactions between human users and computers occur at the user interface, which includes both hardware and software. When users attempt to input sensitive information to computers, a kind of *shoulder surfing* that might use direct observation techniques, such as looking over someone's shoulder, to get the information could be a great concern at the user interface. In this paper, we observe privacy-related issues at the user interface and then present an abstract model for privacy-preserving human-computer interactions. In such an abstract model, we also present two prototype methods which could work with traditional input devices.

1 Introduction

Human-Computer Interaction (HCI) is recently one of the most significant research topics in computer science. Since actual interactions between human users and computers occur at the user interface, which includes both hardware and software, privacy concerns should remain on that user interface unless all sort of possible intrusions and observations are eradicated. Needless to say, it is not trivial to remove them. When users attempt to input sensitive information to computers using a keyboard, keypad, mouse or touch screen, a kind of *shoulder surfing* that might use direct observation techniques, such as looking over someone's shoulder, to get the information could be a great concern at the user interface [7].

In this paper, we observe privacy concerns at the user interface and present an abstract model for the consideration of privacy-preserving HCI. In such an abstract model, we also present two prototype methods which could work with traditional input devices. One is a method with a single user interface, while the other is with more user interfaces, implying out-of-channels. More details of our study will be presented in the full paper version, with regard to the observation, model, and method.

* This research is made by the support of industry, university and research cooperation business of Seoul Metropolitan city (subject No. 10557).

C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2009, LNCS 5614, pp. 544–549, 2009.

The rest of this paper is organized as follows. In Section 2, we observe privacy concerns at the user interface. In Section 3, we present an abstract model for privacy-preserving HCI, and then two prototype methods in Section 4. This paper is concluded in Section 5.

2 Privacy Concerns at the User Interface

The user interface - also known as Human Computer Interface or Man-Machine Interface (MMI) - is the aggregate of means of input and output (I/O) between human users and computer (or electronic/mechanical) systems, for allowing the users to manipulate systems and the systems to indicate the effects of the users' manipulation. In computer science, a common understanding of the user interface is that it is a kind of general purpose I/O device along with its corresponding software. Thus, it is trivially considered to input sensitive as well as non-sensitive information at the same user interface, such as a keyboard, keypad, mouse or touch screen. Here the sensitive information implies user-private information including a password, Personal Identification Number (PIN), Social Security Number (SSN), and so on. When users attempt to input such information at Automated Teller Machines (ATMs), public pay phones, kiosks, or any traditional computer systems, the so-called shoulder surfing attack is a big concern. Note that the shoulder surfing attack is neither a kind of technical attack, for example, based on malicious software, nor a social engineering attack. It is done by direct observation techniques, such as looking over someone's shoulder, and also can be done at a distance using binoculars, Closed-Circuit TeleVision (CCTV) cameras or other vision-augmenting devices.

There have been a number of technical proposals to prevent shoulder surfing, for example, by a physical shield at the user interface, sophisticated display which grows darker beyond a certain viewing angle, keypad which alters the physical location of keys at each input trial, graphical password which is less trivial to guess, and eye-tracking technique which is less traceable by simple observations [1,2,4,5,6]. Also, there have been the policy-based or legal enforcement such that security cameras are not allowed to be placed directly above the user interface or other users are not allowed to get close to the active user at the user interface [7].

However, those schemes are eventually vulnerable to the overall attacks of an accurate shoulder surfer who also breaks the policy-based or legal enforcement. If the shoulder surfer is equipped with sufficient monitoring devices placed around the user interface and allowed to record the user's input transactions more accurately, the shoulder surfer can succeed in obtaining the user's sensitive information at the user interface.

3 Abstract Model for Privacy-Preserving HCI

The basic assumption of our approach is that the user interface is under the observation of a shoulder surfer, as like that the network is under the control

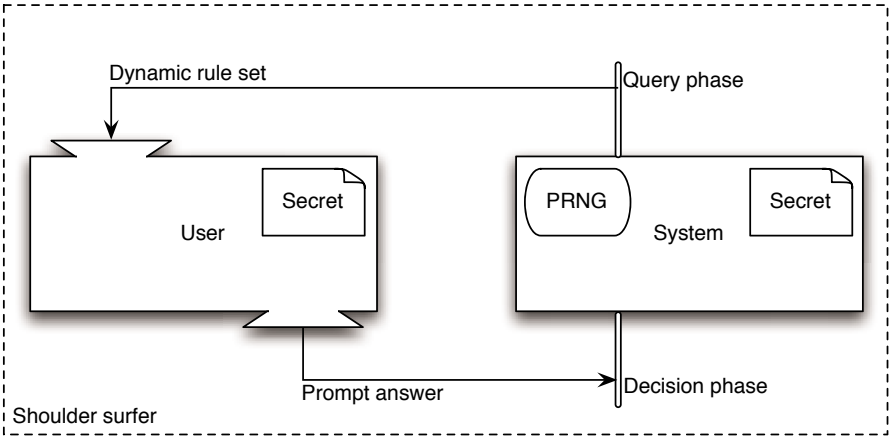

Fig. 1. Abstract model for privacy-preserving HCI

of an attacker in the famous Dolev-Yao model. Thus, both user and system are considered as black boxes but the user interface between them are not from the perspectives of the shoulder surfer. That means, the shoulder surfer can observe and log every conversation between the user and the server at the user interface, while (s)he cannot break into the memory of both user and server, in our model. Fig. 1 illustrates this model in very abstract levels. The user who memorizes a secret may be queried by the system which stores the same secret and is equipped with a Pseudo-Random Number Generator (PRNG), with regard to a set of dynamic rules or puzzles, in a *query phase*. The user prepares a prompt answer from the secret and the given dynamic rule set in a cognitive sense, and then inputs the answer at the user interface, which is followed by a decision of the system, in a *decision phase*. Those phases may be repeated for a sufficient amount of time, so that the system can make a correct decision with overwhelming probability. For achieving the goal of privacy-preserving, the followings should be attained in the query and decision phases at the user interface.

- The probability that the shoulder surfer gets the information about the secret from dynamic rule sets and prompt answers, should be negligible.
- The probability that an active attacker succeeds in forging prompt answers correctly on the given dynamic rule sets, should be negligible.
- The dynamic rule sets should be sufficiently random to prevent replay attacks.

In this abstract model, we could consider both singe and multiple user interface models between the user and the system. The single interface model means a trivial case that the user is given a I/O device within the same flow of control, while the multiple interface implies a out-of-band channel which means that the user is given another I/O device out of the flow above, saying, beyond the control

of the shoulder surfer. In the following section, we will present two prototype methods in those respective models.

4 Prototype Methods for Privacy-Preserving HCI

In the sense of aforementioned privacy-preserving model using human recognition, we present prototype methods in our abstract model, to secure authentication based on PINs against the shoulder-surfing attack. Usually customers using ATMs are required to enter their PINs at the user interface provided by the ATMs. However, the customers' input actions at the user interface could disclose the secret PINs to the shoulder surfer. Thus, we devise prototype systems, in which users enter perturbed numbers (as prompt answers) instead of the secret PINs themselves. Given a set of dynamic queries from ATM, users should be able to produce a correctly perturbed number, by a simple and intuitive method in a cognitive sense, using their knowledge of respective PINs.

In the proposed prototype methods, we assume the system has a keypad implemented on a touch screen, which is already wide-spread in modern ATMs. Fig. 2(a) shows the initial configuration of keypad in the prototype method. Like a normal ATM, ten digit keys ($0 \sim 9$) and two special keys ($*$ and $\sharp$) are shown but with modification of the shape of digit keys from 1 to 9 to contain nine sub-blocks in each key. For more flexible constructions, we color each key distinctly according to the digit. The assigned digit of the key appears at the

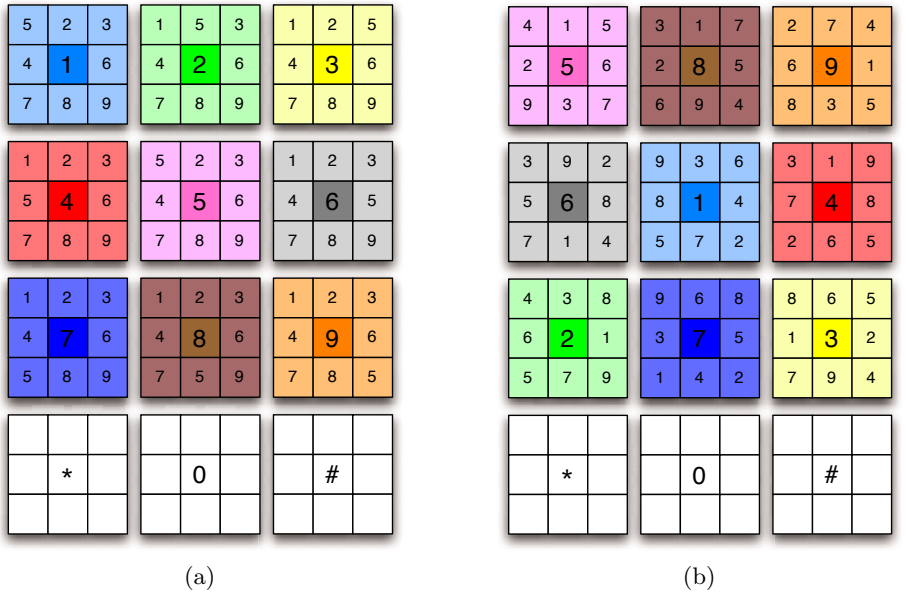

Fig. 2. Proposed prototype of shuffling keys. (a) before shuffling keys (b) after shuffling keys.

center block with thicker and larger font, and the other digits except zero and the assigned digit are shown in the surrounding sub-blocks, which indicate eight directions from the key. Then, in each query phase, both keys and surrounding sub-blocks are shuffled[1] randomly as shown in Fig. 2(b). In other words, the randomly shuffled keypad corresponds to the dynamic query set.

There are two options regarding the starting point. One is to start from the first digit of user's PIN. Otherwise, we could start from a color or number possibly indicated in each query. (See below for more flexible constructions.) Then, from the starting point (i.e., key) on a keypad, a user moves to the next key according to the direction indicated by subsequent digit of the secret PIN, *without* touching the keys actually. The user may only stare at the keypad and follow the direction on it. If a digit of user's PIN is 0, the user stays with the current digit key.

The user may stop following directions if (s)he finishes following all digits in the PIN or there is no way to move on the keypad. Then, finally the user enters the last digit key (s)he has followed on the keypad, into the system. The whole processes from the random shuffling above are repeated to ensure the possession of the secret PIN with overwhelming probability. For example, using the configuration of Fig. 2(b), if the PIN is 46013 and the first option of starting point is considered, then the user may start at the digit key '4' and move to '3', '7' and then reaches '2' as a final destination. Thus '2' is entered by the user. In the same case but with a different PIN such as 57852, the user may proceed with '5', '1' and '6' according to 5785, but there is no more digit key in the left side of key '6'. Thus, '6' is entered by the user in this case.

In the prototype system above, basically we assume the single interface model discussed in the previous section, but it can be extended to the multiple interface model, in which attackers hardly observe all interfaces at the same time. For example, we can consider a small handheld device providing another user interface for out-of-band channels, so that such a device can be used for notifying the color or number of keypad as a random starting point. In this case, since the starting point is notified to the user through a out-of-band channel and is more difficult to trace or guess, the shoulder surfer may have more difficulties in obtaining the secret information. With regard to the handheld device and the out-of-band channel, we con consider a PDA or cell phone such as i-phone, equipped with a bluetooth or further communication facility.

As for security, the probability that an attacker guesses a right answer in a single attempt is $\frac{1}{9}$ without any information related to the PIN itself. Since the authentication process is repeated, the probability that the attacker finally succeeds in authentication is reduced to $(\frac{1}{9})^t$ simply, where t is the number of iterations very related to the length of PIN. In addition, the attacker cannot succeed in guessing a right digit at a specific position of the PIN because each query-answer phase can be stopped at any digit of the PIN, feeding nonlinearity

[1] More specifically, we can consider two levels of random shuffling of those keys and sub-blocks. That is, firstly we shuffle the nine digit keys, and then respective surrounding keys.

to the prototype system. Hence, the only meaningful attack is that the attacker guesses all possible cases of digits based on the entered number from the observation and repeats this guessing through a huge number of shoulder-surfings on a specific target user. Then, the attacker tests all possible sequences using the statistics for candidate digits but such an attack is impractical.

To sum up, the prototype system is secure against shoulder-surfing attacks in the sense of practicality since even accurate surfing may not work.

5 Conclusion

In this paper, we observe privacy concerns at the user interface, with regard to more powerful (or less restrictive) shoulder surfers who may break the rules assumed in the previous schemes, and then we present a simple model in abstract levels, along with prototype methods for privacy-preserving HCI. The prototype methods do not require any arithmetic computation to the user [3] in a practical sense, but provide relatively stronger security against accurate shoulder surfers than the related schemes [1,2,4,5,6]. In the full paper version, we provide experimental results and analyses, along with more details of our study.

References

1. Hoanca, B., Mock, K.: Screen Oriented Technique for Reducing the Incidence of Shoulder Surfing. In: International Conference on Security and Management (SAM) (2005)
2. Hoanca, B., Mock, K.: Secure Graphical Password System for High Traffic Public Areas. In: ETRA - Eye Tracking Research and Applications Symposium (2006)
3. Hopper, N., Blum, M.: Secure human identification protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001)
4. Kumar, M., Garfinkel, T., Boneh, D., Winograd, T.: Reducing Shoulder-surfing by Using Gaze-based Password Entry. In: Symposium on Usable Privacy and Security (SOUPS) (2007)
5. Roth, V., Richter, K., Freidinger, R.: A PIN-Entry Method Resilient Against Shoulder Surfing. In: Conference on Computer and Communications Security (CCS) (2004)
6. Tan, D., Keyani, P., Czerwinski, M.: Spy-Resistant Keyboard: Towards More Secure Password Entry on Publicly Observable Touch Screens. In: Computer-Human Interaction Special Interest Group (CHISIG) of Australia (2005)
7. "Shoulder Surfing" in Wikipedia, the free encyclopedia, `http://www.wikipedia.org`