Spy-Resistant Keyboard: Towards More Secure Password Entry on Publicly Observable Touch Screens

Desney S. Tan, Pedram Keyani, Mary Czerwinski
Microsoft Research
One Microsoft Way, Redmond, WA 98052, USA
desney@cs.cmu.edu, pkeyani@cs.stanford.edu, marycz@microsoft.com

ABSTRACT
Current software interfaces for entering text on touch screen devices mimic existing mechanisms such as keyboard typing or handwriting. Unfortunately, these techniques are poor for entering private text such as passwords since they allow observers to figure out what has been typed just by watching. In this paper, we present the Spy-Resistant Keyboard, a novel interface that allows users to enter private text without revealing it to an observer. We describe a user study we ran to explore the usability of the interface as well as additional security provided by it. Results indicate that although users took longer to enter their passwords, using the Spy-Resistant Keyboard rather than a standard onscreen soft keyboard resulted in a drastic increase in their ability to protect their passwords from a watchful observer. We discuss future extensions to these ideas.

Categories and Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces - Screen design, User-centered design, Graphical user interfaces; K.6.5 [Management of Computing and Information Systems]: Security and Protection.
General Terms: Human Factors, Performance, Security.
Keywords: Touch screen, keyboard, input technique, visual search, selective attention, security, password.

INTRODUCTION

Touch screens are becoming increasingly common, appearing on devices such as digital whiteboards, tablet PCs, as well as ATM and debit card machines. Many of these devices assume that the touch screen is the primary input mechanism and make using traditional mechanisms such as a keyboard or mouse inconvenient. As a result, these devices often employ alternative mechanisms for text input, including soft keyboards and handwriting recognition.

The soft keyboard functions like a hardware keyboard except that users touch an onscreen image map to type. With handwriting recognition, users enter text by writing on the touch screen. Unfortunately, these input interfaces are intrinsically observable. That means that someone watching the typist use these interfaces can fairly easily reconstruct text that has been entered, an activity known as shoulder surfing. This is undesirable when typing private text, such as passwords. Additionally, since most handwriting recognition programs use dictionaries to resolve ambiguous characters, recognition rates for passwords remain fairly low.

In this paper, we present the Spy-Resistant Keyboard, a novel interface that makes it hard for an observer to determine the text string typed or to use a replay attack to forge the string by repeating gestures. We present results from a user study evaluating both the usability as well as the additional security offered by this interface. Finally, we discuss future work that will extend these ideas.

Figure 1: Typing on publicly observable touch screen

RELATED WORK

In many systems, users have to authenticate themselves to access sensitive data and services. Currently, they have three basic methods to do this: tokens, biometrics, and knowledge. Token-based methods utilize something a user possesses, such as an identification card, to verify their identity [1]. Such methods often require costly construction and distribution of tokens, as well as installation of specialized sensing hardware. Additionally, possession of a token does not necessarily imply ownership, and theft or forgery remains a serious threat to these systems.

Biometric methods identify individuals based on distinguishing physiological or behavioral characteristics. These methods include signature, keystroke pattern recognition, voice, vein geometry, as well as eye-based, facial, finger, and palm imaging [for detailed review, see 3]. Just as with token-based methods, biometric methods involve costly hardware and characteristics can be stolen or forged. Furthermore, since these characteristics cannot be easily replaced, theft is more costly than it is with other methods.

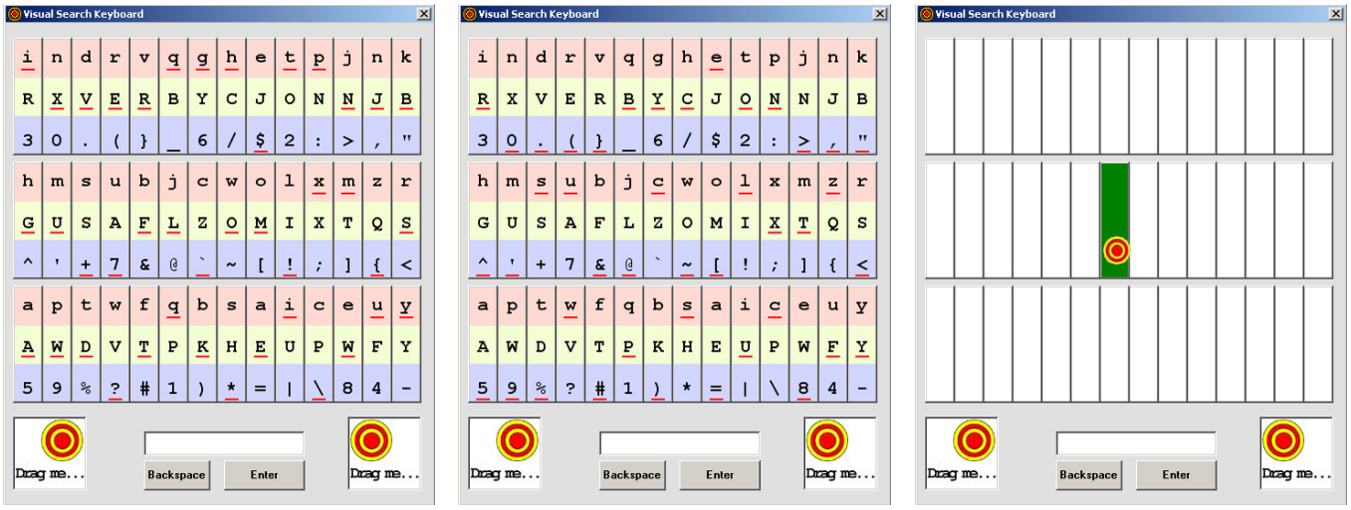

Figure 2: For example, user trying to type a "c" (*left*) searches Spy-Resistant Keyboard for the letter, finds it on the seventh Tile in the second row, (*center*) taps the Interactor once to shift the red underline to the "c" (all keys change shift state with each tap), and then (*right*) drags the Interactor to the appropriate Tile. It is very hard for an observer to reconstruct what has been typed.

The third class of methods, which remains dominant on many computing systems, verifies access privileges with passwords known only to the user. Historically, the choice of passwords has been such a prevalent problem that the National Institute of Standards and Technology has published a document advising users of proper password selection and use [4]. They recommend picking random strings of characters and keeping different passwords for different accounts. Unfortunately, this places a large strain on users, who have to remember an increasing number of passwords. To alleviate this problem, various researchers have proposed alternatives and augmentations to standard text passwords [for examples, see 1, 6].

With the introduction of large touch screen displays that utilize onscreen soft keyboards, learning someone's password has become as easy as watching them type it in. This is a serious threat to security since these methods are only as secure as the user's ability to keep the password secret. In fact, even apart from adversarial observers, people are generally more likely to peek at private content on large public displays, making unintentional viewing of password entry more likely than before [5]. The use of one-time passwords [2] is the closest method we have found that might protect against such attacks. However, this method usually requires that users constantly learn new passwords. Also, it cannot be applied to generic private text entry.

SPY-RESISTANT KEYBOARD

We designed an interface called the Spy-Resistant Keyboard that protects typists working on publicly observable touch screens from revealing private text to observers. This interface uses a level of indirection that allows typists to focus their attention on a particular part of the keyboard, while observers have to pay attention to and memorize the layout of the entire keyboard.

The Spy-Resistant Keyboard is composed of 42 Character Tiles, two Interactor Tiles, a textbox for feedback, a backspace button, and an enter button (see Figure 2). Each Character Tile is randomly assigned a lowercase letter, an uppercase letter, and either a number or a symbol. Lowercase letters are always on the top row of each tile and have a red background; uppercase letters are always in the middle and have a green background; numbers and symbols are always at the bottom and have a blue background. Since there are exactly 42 numbers and symbols combined, but only 26 letters, some letters are repeated. Just as each button on a standard keyboard represents two characters, depending on the state of the caps lock or shift keys, each Character Tile represents three characters, depending on the state of shifting. Rather than having a fixed shift state for the entire keyboard, as traditionally done, each tile has a randomly assigned shift state, indicated by the red line under the active character.

In order to type a character on the Spy-Resistant Keyboard, the typist first locates the tile that contains the character to be typed. Next, the typist clicks on one of the Interactors at the bottom of the keyboard to cycle through shift states and move the red underline to the desired character (see Figure 2). Clicking on the Interactor moves the underline to the next character on each tile. Note that since the underlines start on different types of characters on each tile, knowing that the typist has clicked on the Interactor but not knowing which tile the typist is focused on gives the observer no useful information about the kind of character being typed.

Finally, the typist drags the Interactor towards the Character Tile on which the desired character resides. Upon the start of the drag interaction, the system knows that the user has visually located the Character and blanks the Character Tiles (see Figure 2). Hence, without knowing where the Typist is going to drop the Interactor, adversarial observers have to memorize the location of all characters on the keyboard so that they can reconstruct the typed character from the location of the drop. Each tile highlights as the typist drags over it. The typist drops the Interactor on the desired

tile and the character is typed. The keyboard re-randomizes characters and the typist repeats the process to type the next character. After beginning the drag, the typist may also drop the Interactor on anything other than a Character Tile to reset the board and get a new set of characters, in case they lose track of their target.

USER STUDY

We compared the Spy-Resistant Keyboard to a standard soft keyboard in order to examine usability as well as additional security it provides. To ensure equivalent visibility, we used the same font for characters in each interface.

Participants and Setup

Six pairs (8 males, 4 females) of Microsoft employees volunteered to participate in the study. All users had normal or corrected-to-normal eyesight, and all were right-handed. The average age of users was 28.8, ranging from 21 to 38 years old. Users received a small gratuity for participating.

We ran the study on a SMART Board™ 3000i, which provides a physically large rear-projected touch screen display. The display was approximately 53″ tall by 40″ wide and ran at a resolution of 1024 x 768. Users stood in front of the display and interacted with the interfaces by touching them with their fingers (see Figure 1).

Task and Procedure

Before beginning the test, we gave users paper-based instructions on how to type with the soft keyboard as well as with the Spy-Resistant Keyboard. Both users took turns practicing each interface by typing in a password we provided. All users were able to complete each practice password in less than two and a half minutes.

For each trial in the test, one user played the role of Typist while the other was the Observer. The Typist used one of the two interfaces to type in passwords. The Observer watched the Typist to discover the passwords. Typists were allowed to use any technique they wished to prevent the Observer from figuring out the password. However, in order to simulate public visibility of the display, they were not allowed to explicitly physically obstruct the Observer's view of the keyboard.

Observers were also allowed to use any technique they wished to watch the Typist and figure out the password. For example, they could move around to get the best view of the screen and many took notes to help them reconstruct the passwords. After each entry, the Observer recorded what they thought the password was. The pair performed each entry twice for each password.

Design

We assigned each Typist one easy, one moderate, and one difficult password for each interface. All passwords were 8 characters long. We randomly chose the easy passwords from the set of English words having Kucera-Francis familiarity and concreteness ratings between 300 and 700 (e.g. contract). We chose the moderate passwords to contain 3 to 5 letter English words surrounded by random characters (e.g. #back$Jr). The difficult passwords were completely random sequences of 8 characters (e.g. s%g7^Lp=).

We used a 2 (Interface: Soft Keyboard vs. Spy-Resistant Keyboard) x 3 (Password: Easy vs. Moderate vs. Difficult) within-subjects dyadic design. Each user performed each of the 6 conditions twice, once as the Typist and once as the Observer. We balanced the order of Interface across pairs, with each member of a pair using the interfaces in the same order. We randomized the order of Password.

We collected the following dependent measures from the Typist in order to compare usability of the two interfaces: completion time, number of backspaces, and error rates for each password entry. In order to determine the level of security provided by the interfaces against watchful observers, we collected the Observer's guesses from each password entry. Finally, users filled out a post-test questionnaire indicating their preference for each of the interfaces.

Results

Typist Performance: Usability

We analyzed the average completion time required to enter each password with a 2 (Interface: Virtual Keyboard vs. Spy-Resistant Keyboard) x 3 (Password: Easy vs. Moderate vs. Difficult) repeated measures analysis of variance (RM-ANOVA). We found a significant main effect of Interface ($F(1,11)=114.11$, $p<0.0001$), with the Soft Keyboard resulting in faster completion times on average (see Figure 3).

We found no significant difference in the number of backspaces hit for each of the conditions. In fact, Typists seemed to hardly ever use the backspace key (average of about 1 backspace hit every 20 passwords typed). We also found no significant difference in the error rate of entering passwords. In fact, only 9 out of a total 144 passwords were entered incorrectly, and most were off by a single character.

Observer Performance: Security

We compared each guess made by the Observer to the password typed and generated two metrics representing the level of security offered by the interface: a strict metric, the number of characters in each guess that did not match its typed counterpart exactly; and a loose metric, the Levenshtein distance, or number of deletions, insertions, and substitutions required to transform the guess into the typed password. This loose metric accounted for characters that were shifted in position. Both these metrics produced ratings on a scale of 0 to 8, with 0 indicating poor level of security and 8 indicating strong level of security.

We performed similar 2 x 3 RM-ANOVAs for the level of security offered by the interfaces. This analysis revealed a significant main effect of Interface for both the strict metric ($F(1,11)=641.47$, $p<0.0001$) as well as the loose one ($F(1,11)=1250.68$, $p<0.0001$), with the Soft Keyboard resulting in far poorer security, on average. Additionally, the

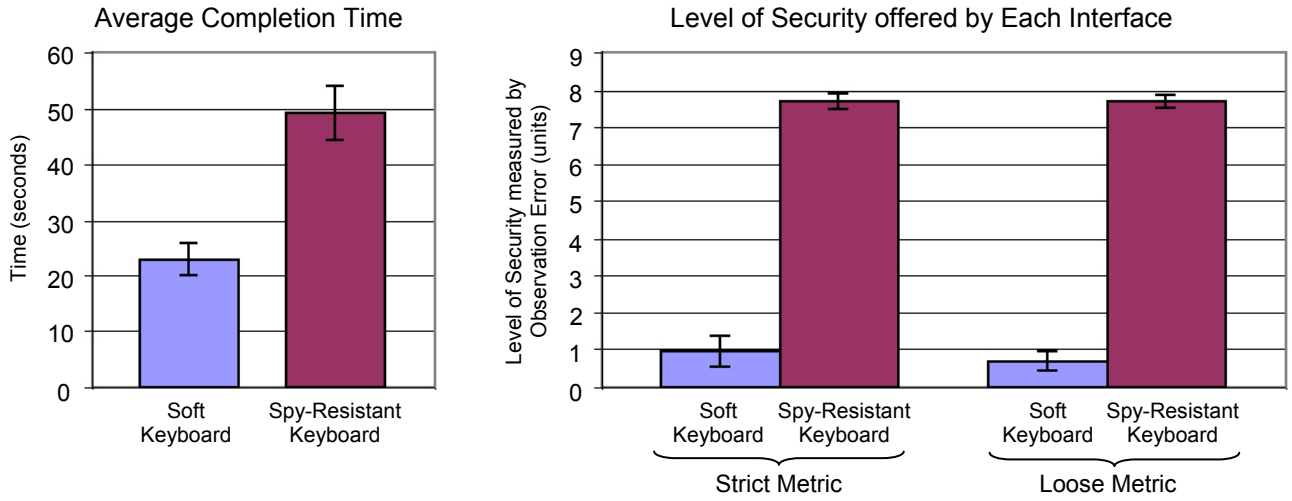

Figure 3: (*left*) Main effect of Interface for average time to type each password. (*right*) Main effects of Interface for the level of security, measured by errors in guessing the password.

loose metric revealed a significant main effect of Password ($F(1,11)=7.31$, $p=0.004$), with progressively higher security with the more difficult passwords (3.85 vs. 4.31 vs. 4.44). These results, illustrated in Figure 3, indicate the drastically improved level of security offered by the Spy-Resistant Keyboard against shoulder surfers.

Subjective Ratings

In addition to performance data, we gathered user preference data on 5-point Likert scales after the study. Users found the Soft Keyboard (M=4.92) significantly easier to use than the Spy-Resistant Keyboard (M=2.42), ($t(11)=16.58$, $p<0.0001$). However, users were also significantly less comfortable with using it to enter their passwords ($t(11)=-13.01$, $p<0.0001$, M=1.17 vs. M=4.50). This sentiment was further supported by users feeling like they had much more difficulty acquiring useful information when observing the someone using the Spy-Resistant Keyboard (M=4.50) as opposed to the Soft Keyboard (M=1.67), ($t(11)=-10.47$, $p<0.0001$). Additionally, most users agreed that the extra security was worth the extra effort.

DISCUSSION AND FUTURE WORK

Results from the study show that the Spy-Resistant Keyboard imposes a tradeoff between efficiency of entering text and the security of text entered. Using the Spy-Resistant Keyboard takes about twice as long as a soft keyboard, but distinctly makes the perceived as well as actual level of security provided against observers much stronger.

In future work, we will explore schemes to make the visual search and typing task easier, while still maintaining similar levels of security against an observer. One promising idea involves completely eliminating the visual search task by not randomizing the characters on the keyboard. Instead, when the user starts the drag, we would hide the characters and then animate each key into a new position. The typist would have to watch the changing location of one key, but the observer would have to know where all keys started and ended in order to reconstruct what has been typed.

We found that observers who devised strategies either tried to monitor the typist's gaze or concentrated on only a small region of the display hoping that the desired character lay there. Although this may be more effective than other strategies, it would still take many observations before gaining access to the full password. In future work, we will explore schemes that provide feedback on the remaining safe lifetime of a password based on the number of times it has been entered as well as the types of interfaces used.

Finally, we must stress that the Spy-Resistant Keyboard does not do well to protect against observation that may be rewound and replayed, for example from an observer recording with a camera. In future work, we plan to further explore techniques that protect against this kind of attack.

ACKNOWLEDGEMENTS

We are grateful to Patrick Baudisch, George Robertson, Brian Meyers, Greg Smith, Sumit Basu, Darko Kirovski, and Jeffrey Nichols for their discussion of this work.

REFERENCES

1. Brostoff, S., Sasse, M.A. (2000). Are Passfaces more usable than passwords? A field trial investigation. *Proceedings of HCI 2000*, 405-424.
2. Haller, N., Metz, C., Nesser, P., Straw M. (1998). A one-time password system. *Network Working Group Request for Comments 2289*.
3. Jain, A., Hong, L., Pankanti, S. (2000). Biometric identification. *Communications of the ACM*, 43(2), 90-98.
4. National Institute of Standards and Technology. (1995). Password Usage. *NIST FIPS PUB 112*.
5. Tan, D.S., Czerwinski, M. (2003). Information voyeurism: Social impact of physically large displays on information privacy. *Proceedings of CHI 2003*, 748-749.
6. Zviran, M., Haga, W.J. (1990). Cognitive passwords: The key to easy access control. *Computers and Security*, 9(8), 723-736.