Using Fake Cursors to Secure On-Screen Password Entry

Alexander De Luca, Emanuel von Zezschwitz, Laurent Pichler, Heinrich Hussmann
Media Informatics Group, University of Munich (LMU)
Amalienstr. 17, 80333 Munich, Germany
{alexander.de.luca, emanuel.von.zezschwitz, hussmann}@ifi.lmu.de, laurent.pichler@campus.lmu.de

ABSTRACT
In this paper, we present a concept using fake cursors to disguise on-screen password entry. We performed two user studies with different amounts of dummy cursors and differently colored cursors. The results show that dummy cursors significantly improve security. At the same time, decrease in performance is kept within an acceptable range. Depending on the required degree of security, the studies favor 8 or 16 differently colored cursors as the best trade-off between security and usability.

Author Keywords
Ninja Cursors; Authentication; Security

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: User Interfaces - Input devices and strategies, evaluation

General Terms
Human Factors; Performance; Security

INTRODUCTION
Password entry is a ubiquitous task. In many instances, the user has to authenticate in a public or semi-public setting like internet cafés or office environments, exposing the password to onlookers. On-screen keyboards are often used to minimize the possibility of losing the password due to keyloggers and other malicious software. For instance, this is commonly used by online banking websites. They enforce the use of virtual keyboards or keypads to input the secret credentials. While being more secure against keyloggers and the like, this approach is highly vulnerable to shoulder surfing attacks, that is, an attacker observing the input from a nearby position. It is almost impossible to hide the input as this would mean covering a large portion of the screen space.

One of the most common solutions to this problem is adding overhead to the input to make it hard to follow. A famous example is the spy-resistant keyboard by Tan et al. [5] which uses an indirect input method to make on-screen keyboard use more secure. Unfortunately, indirect input makes the interaction with such a system quite slow. Other researchers add overhead in the form of fake input (e.g. adding additional digits to a password) like Vibrapass by De Luca et al. [1]. Graphical authentication systems like the one presented by Wiedenbeck et al. [7] use indirect input in the form of distracting icons or images to hide the input. The randomness introduced with these systems makes them slower and significantly reduces memorability. Finally, some research focuses on using additional hardware to make the input invisible to an attacker [3] or dislocate the input from the terminal [4].

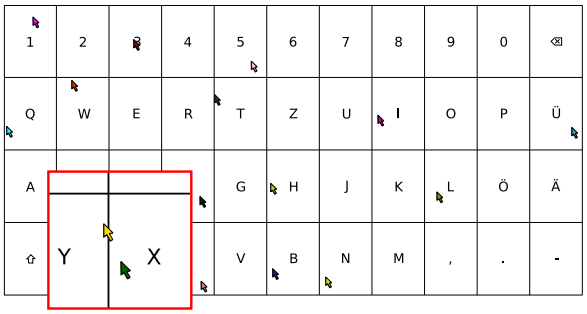


Figure 1. 16 colored mouse cursors in the user study. The red outlined area was created for the screenshot for better visibility.

We propose a shoulder surfing resistant input method using multiple fake cursors. The idea is inspired by Ninja cursors [2]. In their work, the authors propose using several concurrent cursors that move in the exact same way to quickly reach objects on big screen spaces. As opposed to Ninja cursors, in our system, only one cursor performs the actual input while the other cursors act as distraction for an attacker. That is, they do not move in line with the genuine cursor.

Quite recently, Watanabe et al. [6] presented a demo of a similar approach to secure PIN-entry. The main differences to their work are that we use an intelligent cursor algorithm instead of pre-recorded cursor movements; that we introduced coloring to improve the usability of the system; and that we performed extensive evaluations of the approach to prove its appropriateness to secure on-screen password entry.

The results of the studies show that, depending on the required level of security, using 8 or 16 differently colored cursors provides a good trade-off between security and usability. In either case, the approach significantly improves shoulder surfing resistance of on-screen password entry.

CONCEPT
The main idea is using a specific number of fake cursors that hide the input on the on-screen keyboard from onlookers. Since the fake cursors move differently from the active

CHI 2013, April 27–May 2, 2013, Paris, France.

cursor, users can identify it while attackers have difficulties to do so. Figure 1 shows an on-screen keyboard with 16 colored mouse cursors, 15 of them fake cursors. The active cursor has a random color.

The fake cursors move with respect to several rules: a) They use bezier-like paths to a randomly selected target on the keyboard as straight movements would give them away. b) When the active cursor changes its direction by more than $90°$, the distraction cursors pick new targets using similar angles. c) They never leave the keyboard and are always located at a target when the active cursor is. Off-keyboard cursors would easily be identified as fake.

By itself, on-screen keyboard-entry is already slow. Adding fake cursors does further reduce the input speed but at the same time significantly improves its resistance to shoulder surfing attacks. Such an approach is not meant for contexts with several authentication sessions per day but for systems with high security demands like online banking.

Threat Model and Theoretical Security

The approach was designed to be resistant against shoulder surfing attacks. For a successful attack, the attacker needs to observe all cursors or immediately identify the active cursor. No visual feedback of clicked buttons is given. The most promising attack on the system is video-based (combination of screen and mouse recording). This would enable comparing the mouse movement to the movement of the cursors on screen and therewith identify the active cursor.

Attack-resistance is highly depending on the strategy that the user employs to identify the active cursor as described later in this paper. If fast unnatural movements are used, the current algorithm of the fake cursors does not keep pace and the active cursor becomes obvious.

PRE-STUDY

We conducted a pre-study with a simple pointing task using a repeated measures factorial design with two independent variables: *Cursors* (4, 8, 16, 24) and *Color* (differently colored cursors or all white). We did this to test whether color makes it easier to keep track of the active cursor. To minimize learning effects, we used a 2x4 Latin square design resulting in eight cases. The main goals of the pre-study were: 1) First insights on the usability using different numbers of cursors. 2) Identify search strategies for the active cursor.

Procedure and Participants

The study took place in an isolated room at our premises. For each combination of the independent variables, the participant had to find the active cursor and then use it to click two targets, located on the screen, in a predefined order. The location of the targets was not varied. Three trials were allowed to correctly perform each task. If successful or after failing three times, the next task started. After each trial, the cursors (including the active one) were randomly arranged on the screen. That is, the active cursor had to be found again.

The task order was automatically assigned by the prototype based on the user ID. Each participant performed each possible subtask twice resulting in 16 tasks per user. At the end of the study, the participants were asked to fill out a questionnaire collecting demographics and qualitative data.

We recruited 16 participants for the study. That is, the Latin square design was applied two times. The average age was 25 (range: 19-51). Five participants were female. Participants received a 5 Euro voucher for an online shop.

Results and Discussion

Search strategies are not only interesting from a point-of-view of usability but they are also related to security. A bad strategy can influence whether an attacker can identify the active cursor or not. The analysis of the questionnaire and the video material revealed two main strategies: 1) Eight participants moved the mouse cursor to the border of the interaction area to identify it. This strategy can influence the security since the fake cursors do not behave this way. 2) Five participants moved the mouse in small shapes (e.g. waves). This is not duplicated by the algorithm and thus, based on good hand-eye coordination, the active cursor can be identified.

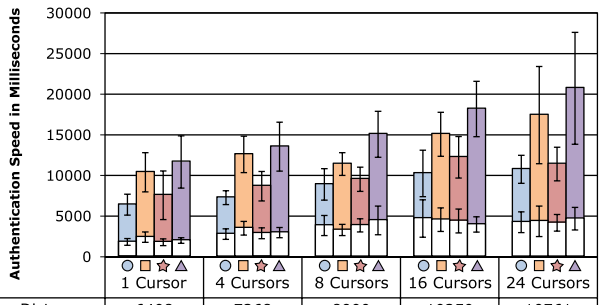
Performance-wise, the most interesting results are on the time required to finish the task, to identify the active cursor and to keep track of it. Keeping track refers to the fact that the user has to be able to follow the cursor and not loose track of it. The fastest task completion time was achieved with four colored cursors (M=4.9s), the slowest one with 24 white cursors (M=10.7s). A 4 x 2 (*Cursors* x *Color*) within participants analysis of variance of task completion time revealed a highly significant main effect for *Cursors* ($F_{3,45} = 7.739, p < .001$). Post-hoc tests revealed a significant difference between Cursors levels 4 (M=5.1s) and 16 (M=8.9s) as well as 4 and 24 (M=10.7s; both $p < .05$).

We considered the time to hit the first target as the time required to find the active cursor and the time from the first to the second (last) target as the performance of keeping track of the active cursor. Analyzing the data based on these assumptions revealed an interesting finding. While the color did neither significantly influence the overall performance nor the time to find the active cursor, it significantly influenced the tracking performance. Using the same ANOVA, we could identify a significant main effect of *Color* ($F_{1.0,15.0} = 6.608, p < .05$). Keeping track of a colored active cursor (M=1.9s) was significantly easier than of a white cursor (M=2.5s). In addition, the positive influence of *Color* increases with the amount of cursors.

MAIN STUDY

The main study was firstly conducted with 39 people and then repeated with 20 of those participants. The first part will not be presented here and was only done to have access to trained users for the second iteration.

We used a repeated measures factorial design with three independent variables: *Cursors* (1, 4, 8, 16, 24), *Color* (yes, no) and *Password* (dictionary, non-dictionary). Passwords consisted of six characters. Non-dictionary passwords were randomly created containing upper case and lower case letters as well as digits and special characters. Dictionary passwords were selected from a list of most commonly used words in



	1 Cursor	4 Cursors	8 Cursors	16 Cursors	24 Cursors
Color Dict (search)	6408 (1830)	7268 (2791)	8900 (3858)	10250 (4722)	10761 (4262)
Color No Dict (search)	10392 (2421)	12592 (3522)	11408 (3308)	15071 (4566)	17435 (4371)
No Color Dict (search)	7568 (1795)	8685 (2905)	9540 (3874)	12237 (4414)	11407 (4182)
No Color No Dict (search)	11666 (2000)	13545 (2979)	15070 (4480)	18184 (3985)	20728 (4693)

Figure 2. Average overall authentication speed. White bars and brackets indicate the time required to find the active cursor.

German, the mother language of all participants. Please note that a new level was introduced to *Cursors*. Level 1 was required to have a baseline to compare the performance of the system. To minimize learning effects, we used a 2x5 Latin square design resulting in ten cases. *Password* was randomized. An example with 16 colored cursors is shown in figure 1. Having 20 participants allowed for two repetitions of the Latin square design. All participants were familiar with the system from the first study iteration.

Procedure and Participants

The study took place in an isolated room at our premises. It was filmed with a high definition camera from the right side of the user, recording both the mouse and the screen. The purpose of the camera was: 1) To identify strategies and usability issues like in the pre-study. 2) To use the video material for a security analysis after the study.

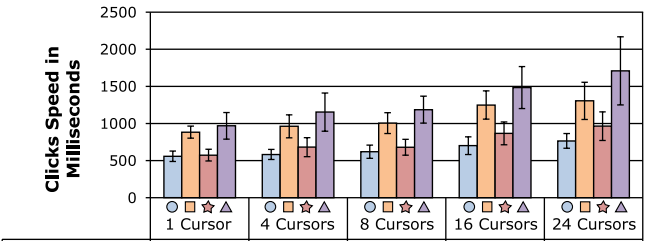
At the beginning, the password task was explained to the participants. For each combination of the independent variables, they had to input two passwords (dictionary and random). When it was not correct, the next task did not start and the user had to fix it by deleting the input pressing "backspace" on the virtual keyboard. The start position of the cursors was randomized after each task.

The participants received a unique list from the experimenter containing their passwords in the order of the experiment. It should be noted here that the users did not get the same passwords that they used in the first iteration of the study. At the end of the study, the participants were asked to fill out a questionnaire collecting demographics and qualitative data.

The 20 participants had an average age of 25 (range: 16 - 32). Nine of them were female. Again, 5 Euro vouchers for an online shop were given out as incentives.

Results and Discussion

All statistical results are based on a 5 x 2 x 2 (*Cursors* x *Color* x *Password*) within participants ANOVA. Errors were very low across all conditions and thus error rates will not be discussed in this paper.



	1 Cursor	4 Cursors	8 Cursors	16 Cursors	24 Cursors
Color Dict	558	582	620	701	765
Color No Dict	883	962	1005	1248	1305
No Color Dict	573	681	680	867	964
No Color No Dict	968	1153	1186	1484	1709

Figure 3. Average time needed between two button presses.

Authentication Speed

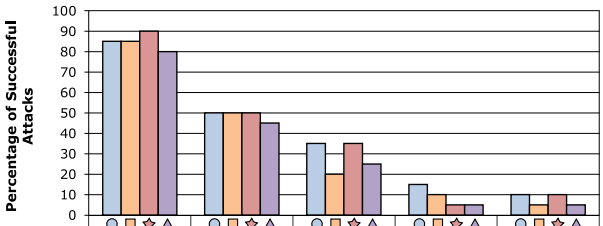
We differentiated three interaction times: overall (time required for the whole authentication), search (time to find the active cursor, i.e. before the first click) and click (average time between all button clicks). "Overall time" presents a realistic measure of how long authentication takes. "Search time" provides an approximate value of how long it takes to find the active cursor. Finally, "click time" indicates how easy it is to keep track of the active cursor.

The results of overall and search time are depicted in figure 2. It shows that while it takes longer to find the active cursor with more cursors, *Color* and *Password* did not influence that time but it is much lower for one cursor. It is interesting to note that search time only slightly rises with the amount of cursors. There was a highly significant main effect for *Cursors* ($F_{2.017,38.325} = 16.858, p < .001$). Post-hoc tests showed significant differences between 16/24 cursors and all other levels of *Cursors* (all $p < .05$) and no significant difference between 16 and 24 cursors.

For overall time, effects of all three independent variables can be observed. One cursor using dictionary passwords was the fastest input method (M=6.4s). 24 cursors without color and random passwords was the slowest (M=20.7s). We found highly significant main effects for *Cursors* ($F_{2.557,48.589} = 15.794, p < .001$) and *Password* ($F_{1.0,19.0} = 80.05, p < .001$) and a significant main effect for *Color* ($F_{1.0,19.0} = 9.093, p < .05$). Post-hoc tests showed that colored cursors were faster ($p < .05$) and dictionary passwords were faster than random passwords ($p < .001$). Additionally, there were significant differences between 16/24 cursors and all other numbers and no significant differences between 16 and 24 cursors.

The results for click time are shown in figure 3. We found highly significant main effects for *Cursors* ($F_{2.502,47.546} = 26.183, p < .001$), *Password* ($F_{1.0,19.0} = 15.946, p < .001$) and *Color* ($F_{1.0,19.0} = 91.231, p < .001$). Post-hoc analysis results are similar to overall time.

We again asked the participants which strategies they used to find the active cursor. 13 of them used the strategy of moving the cursor to one of the borders of the interaction area. Two users based finding the cursors solely on performing specific shapes with the cursors. The final five used both strategies depending on the situation. No new strategies were found.



	1 Cursor	4 Cursors	8 Cursors	16 Cursors	24 Cursors
Color Dict	85	50	35	15	10
Color No Dict	85	50	20	10	5
No Color Dict	90	50	35	5	10
No Color No Dict	80	45	25	5	5

Figure 4. Percentage of successful shoulder surfing attacks on the system. While 16 and 24 cursors are significantly more secure than 1, 4 and 8, there is no apparent difference between them.

Security

The video material was used to perform shoulder surfing attacks. The attacker was highly familiar with the system. He was allowed to watch the input once to steal the password. The result was compared to the original password using the Levenshtein distance, a measure of similarity between two strings (operations necessary to get from one string to the other). Distance "0" indicates a correct guess. Overall, the analysis of all video material took the attacker one full day.

Figure 4 shows the results for the security analysis. In the worst case (one cursor, no color, dictionary), 90% of passwords were successfully shoulder-surfed. The most secure instances were 16 cursors (no color, dictionary and random) and 24 cursors (color, random and no color, dictionary) with 5% success rate. There is no obvious improvement using 24 cursors compared to 16 cursors. Additionally, color does not seem to influence the security of the system as well.

The Levenshtein distance produces parametric data (ratio) and thus allows for using parametric significance tests. We found a highly significant main effect for *Cursors* ($F_{4,76} = 72.863, p < .001$). No other main effects and no interaction effects could be identified. Post-hoc tests revealed (highly) significant differences between all levels of *Cursors* with the exception of 16 and 24 cursors ($p = 1.0$).

There was no influence of strategies to find the active cursor on the amount of stolen passwords per user. This was most probably since the number of users that did not use the border strategy, at least to some extent, was quite low. However, we could observe awareness of the fact that the strategies might influence the security. For instance, one user stated that "*I avoided moving the cursor to the border as I think this would negatively influence the security of the system ...*".

Best Combination

The results show that the advantage of colored cursors kicks in after the active cursor has been identified. That is, it is easier to keep track of the active cursor if all cursors are differently colored. On the other hand, security was not influenced by colors. We argue that depending on the required degree of security, 8 or 16 colored cursors are the best trade-off between security and usability. For high security contexts like online banking, 16 cursors present the best solution. The very low error rates across all conditions support this claim.

This is supported by the results of the questionnaire, in which we asked the participants to rank the different combinations with respect to the best security-usability trade-off. Summed up, 13 participants ranked 8 and 16 colored cursors as the best trade-off (6*16, 7*8). One participant chose 24 colored cursors. In addition, on Likert scales from 1 (no agreement) to 5 (full agreement), 19 participants either fully (14) or partially agreed (5) that the colors improve ease-of-use. 13 participants either fully (6) or partially agreed (7) that the approach improves security (no one disagreed).

CONCLUSION AND FUTURE WORK

We presented a system using fake cursors to hide password entry on on-screen keyboards. Two user studies showed good usability properties and a significant increase in security. The best trade-off between usability and security was achieved with 8 and 16 differently colored cursors respectively.

Even though the participants were trained, they cannot be considered experts. We could not test for learning effects but believe that there is room for improvement. For instance, one study participant mentioned that he "*did it several times now and it gets easier every time*". An open question is therefore how the system performs at long-term use. We plan to conduct a long-term web-based study to answer this question. For instance, it will be interesting to find out if users become significantly faster after long-term use and if they develop more advanced search strategies.

We have a few ideas on how to avoid the border strategy and improve authentication speed. The most promising one is to assign a fixed color or a fixed start location to the active cursor. This way, the cursor can be identified by simply looking for the cursor with that color or at the specific key. We will conduct further studies to find out a) if this approach has the potential to improve search time and b) if this way, other strategies for finding the cursor become obsolete. We argue that this will at the same time reduce authentication speed and make the current algorithm more efficient as unusual behavior is not anymore necessary to identify the active cursor.

REFERENCES

1. De Luca, A., von Zezschwitz, E., and Hussmann, H. Vibrapass: secure authentication based on shared lies. In *Proc CHI '09* (2009), 913–916.
2. Kobayashi, M., and Igarashi, T. Ninja cursors: using multiple cursors to assist target acquisition on large screens. In *Proc. CHI '08* (2008), 949–958.
3. Sasamoto, H., Christin, N., and Hayashi, E. Undercover: authentication usable in front of prying eyes. In *Proc. CHI '08* (2008), 183–192.
4. Sharp, R., Scott, J., and Beresford, A. Secure mobile computing via public terminals. In *Proc. Pervasive '06* (2006), 238–253.
5. Tan, D. S., Keyani, P., and Czerwinski, M. Spy-resistant keyboard: more secure password entry on public touch screen displays. In *Proc. OzCHI '05* (2005), 1–10.
6. Watanabe, K., Higuchi, F., Inami, M., and Igarashi, T. CursorCamouflage: multiple dummy cursors as a defense against shoulder surfing. In *SIGGRAPH Asia '12 Emerging Technologies* (2012), 6.
7. Wiedenbeck, S., Waters, J., Sobrado, L., and Birget, J.-C. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proc. AVI '06* (2006), 177–184.