Gesture Authentication for Mobile Devices

Shuja Shahzada
Carleton University
Ottawa, Canada
shujashahzada@gmail.com

Sonia Chiasson
Carleton University
Ottawa, Canada
chiasson@scs.carleton.ca

Robert Biddle
Carleton University
Ottawa, Canada
robert.biddle@carleton.ca

1. INTRODUCTION

The most common approach to authentication involves a password entered on a keyboard with letters, numbers and special symbols. Recently, however, touchscreen devices have become widespread. Text passwords are still most commonly used on these devices even though studies show that users generally perform better with text input using a physical keyboard than a virtual keyboard [3]. In this paper we present a design and evaluation of a new gesture based password scheme for mobile devices, GesturePass. We speculated that gestures would be familiar and make passwords easy to enter without burdensome visual attention.

2. DESIGN

GesturePass is a recall-based scheme that randomly assigns a set of four simple gestures as a password. The ten possible gestures included are: drag up, drag down, drag left, drag right, tap, hold, pinch in, pinch out, rotate clockwise and rotate anticlockwise. We choose these gestures because they are commonly used gestures in applications that recognize gestures, such as Google Maps. Tap, hold and drag are single touch gestures, whereas the turns and pinches are multi-touch gestures. Single touch gestures require one finger (one point of interaction). Multi-touch gestures require the use of multiple fingers (multiple points of interaction) but do not specify exactly which fingers should form the gesture. Any two fingers that allowed the motion of the gesture could be utilized to perform a pinch or turn. For example, a pinch could be done with a index finger and a thumb, an index finger and a middle finger, or both index fingers (figure 1).

GesturePass was deployed as a part of the existing MVP framework [1], a system specifically designed for conducting research studies of authentication schemes. GesturePass contains about 100 lines of PHP, and about 200 lines of JavaScript. The PHP code runs on the server, and generates JavaScript code that runs in the web browser. The code was adapted from the Hammer JavaScript event logger sample program [2].
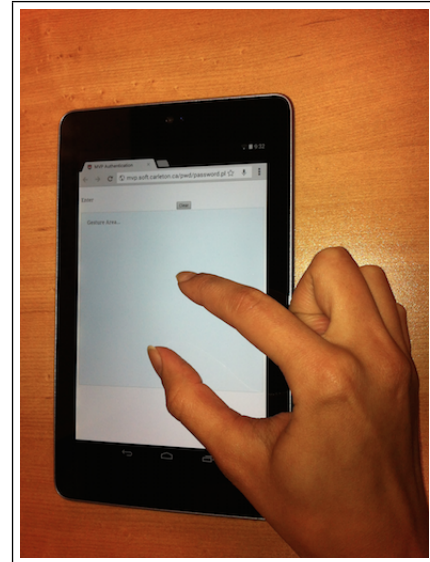
Figure 1: Example of Pinch In Gesture on GesturePass (with thumb and index finger)

3. EVALUATION

We compared GesturePass with a traditional PIN on a Nexus 4 phone and a Nexus 7 tablet. The main focus was investigating GesturePass and the different devices were chosen for exploratory comparison. We analyzed the number of password practices, login attempts, login times, gesture times, and a range of usability questionnaire responses for each device.

Our control condition used a system-assigned PIN password scheme that was implemented using JavaScript. Our system used a large on-screen keypad input rather than the standard built-in Android keyboard. To log in successfully using this scheme, participants had to remember a randomly generated four digit PIN password. The password spaces of the two condition were therefore identical.

There were 30 participants in this study. 15 participants were assigned the phone condition and 15 participants were assigned the tablet condition. Each condition had one gesture password and one PIN. Participants ranged in age from 18 to 33, with an average age of 24. The experiment consisted of three separate sessions, to test memorability and usability over time, totalling about an hour in a lab environ-
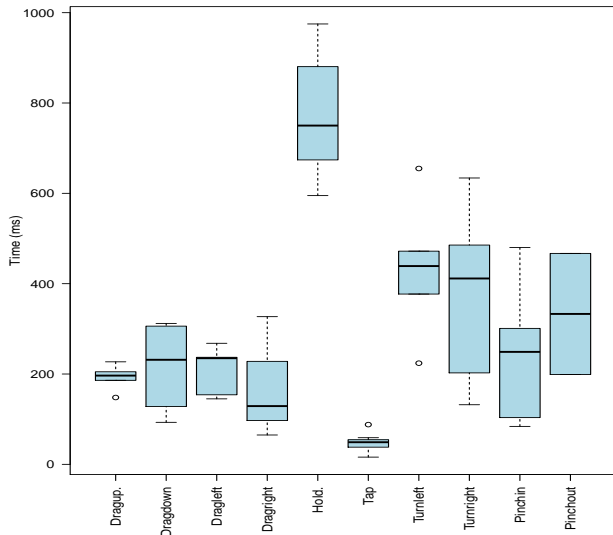
Figure 2: Phone Gesture Times

ment for each participant. The three sessions were separated by at least 24 hours.

In the paragraphs below we briefly outline the results of our study. The study was approved by our Research Ethics Board. Comparisons were made using appropriate statistical methods but to save space we do not report these details.

Number of Practices: On average participants took fewer practices on the tablet than the phone. In addition within each device, on average participants took more practices with the gesture password scheme than the PIN password scheme. This may be a result of the novelty of the password scheme and that participants felt that they needed more practices to get comfortable with the scheme.

Login Attempts: There were no significant differences in the number of login attempts at session 3 between any compared conditions. It appears that by session 3, participants had no more difficulty logging in with either scheme. As we conducted the experiments, we had noted that the participants had gained confidence with using the system. After the first session, they seemed more assured how it worked and clearly understood the scheme. As a result we decided that comparing the session 3 login attempts would be the most reliable way of assessing their ability. The number of login attempts were very similar for the tablet PIN, phone PIN and phone gesture conditions. By session 3, most participants required between 1 to 2 attempts per condition.

Login Time: Participants took longer to login using the gesture scheme than the PIN scheme for both devices. PINs were faster on both devices. We further see that the distributions were wider for the gesture passwords and that some users were much quicker than others at entering their gestures. PIN login times were roughly 5 seconds on each device and GesturePass login times were roughly 7–8 seconds on each device.

Gesture Differences: For login time we also looked into specific gesture times for both the tablet and phone conditions. Entry times for each gesture in the phone condition

can be seen in figure 2. As expected, tap was the quickest gesture and hold was the slowest gesture. By design the hold gesture require 500 milliseconds. The drag gestures, including **dragup**, **dragdown**, **dragleft** and **dragright**, were faster than the turn (**turnright**, **turnleft**) and pinch (**pinchin**, **pinchout**) gestures. The turn and pinch gestures also have very similar times for both devices. In addition to the longer entry times, the multi-touch gestures caused more errors.

Perception: Questionnaire responses suggest that in terms of the ease of use, accessibility and security components, GesturePass was preferred over PINs. In addition, the tablet gesture condition generally scored higher medians than the phone gesture condition. Our exploratory analysis suggest that most users preferred the tablet gesture condition. We speculate this is because users could more easily enter gestures (especially multi-finger gestures) on the tablet than the mobile phone. Participants also commented that they liked GesturePass because it was easy and fun to use, entertaining and more interactive. ("...Novel and pretty easy to use...", "...entertaining ...", "...fun new experience ..." ,"The gestures make it more interactive.")

4. CONCLUSIONS

We designed a novel gesture password system for mobile devices. Our results suggest that the new scheme might be a viable alternative to a traditional PIN password. We compared GesturePass and a PIN with comparable password spaces. We found no statistical differences in the number of login attempts as a PIN after three sessions, and GesturePass was favoured over PIN passwords in our usability analysis. In addition, our usability analysis show that GesturePass was preferred over PINs especially for tablets. However, our studies also show that participants did not favour using even simple multi-touch gestures.

We feel our initial study shows further research may be worthwhile. For example, our study was conducted in a controlled lab environment. Alternative tests could be done in different mobile situations such as in public areas while walking. Another limitation is that although the memorability of the passwords was tested, participants were not required to follow traditional procedures to reset their passwords nor were the password restricting access to resources of value.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] S. Chiasson, C. Deschamps, E. Stobert, M. Hlywa, B. F. Machado, A. Forget, N. Wright, G. Chan, and R. Biddle. The MVP Web-based Authentication Framework. In *Financial Cryptography and Data Security*, pages 16–24. Springer, 2012.

[2] Jorik Tangelder. Hammer JavaScript Library: A JavaScript library for multi-touch gestures, 2014. http://eightmedia.github.io/hammer.js/.

[3] A. Sears, D. Revis, J. Swatski, R. Crittenden, and B. Shneiderman. Investigating Touchscreen Typing: The Effect of Keyboard Size on Typing Speed. *Behaviour & Information Technology*, 12(1):17–22, 1993.