Swypin: Using swipe-select to enter 4-digit PIN for Authentication in a Smartphone

Aditi Phatak
Computer Science Department
Dartmouth College
Aditi.Phatak.Gr@Dartmouth.edu

ABSTRACT

The 4-digit PIN authentication is pretty ubiquitous and can be seen as the preferred mechanism for authentication especially in ATMs and smartphones. The difference is that smartphones offer a much richer touchpad than the buttons of an ATM machine and we have not tried to exploit it yet to improve user experience.

This paper argues that the pressing of buttons is less intuitive[16] and has better memorability [17] than drawing a shape, but gesture-based authentication gives too many false negative. Also, while a 3x3 grid pattern lock seemingly fulfils the gap in the design space and has great theoretical complexity, the trade-off between creating a strong enough password and ease of input allows for human reluctance to be a hindrance to security.

This paper presents a new interaction mechanism, Swypin, for the user to enter a 4-digit PIN to unlock their smartphone in an intuitive way aiming to enhance the usability aspect of authentication.

Youtube Video Link: https://youtu.be/chbdl-dluhg

INTRODUCTION

Most smartphones today are protected by an authentication layer of some form. But more often than not, the authentication mechanism is one that is less focused on robustness of security and more focused on usability. User attempts to unlock phone between 10-200 times a day. [8] For this reason it is so important to not just be thinking of secure authentication mechanism if we do not focus on their usability and cannot get users to actually use them.

Within the framework of not using additional devices or hardware, a lot of the new research in authentication security focuses biometrics using face, voice or gesture, or a combination of them[18], which are all computationally expensive, have a learning curve and can be frustrating to use so many times a day [1]; or involves complicated body-gestures [19,20]. As on-screen textual input of passwords go, users have the tendency of selecting short and easy-to-guess textual passwords.[9,21] This seems to apply for both alpha-numeric passwords and for 3x3 grid based pattern passwords. Thereby reducing the effective complexity of the passwords whatever their theoretical complexity may be. Swipin offers this alternative interface to existing 4-digit PIN unlocking mechanism in a way that has a better flow and increased

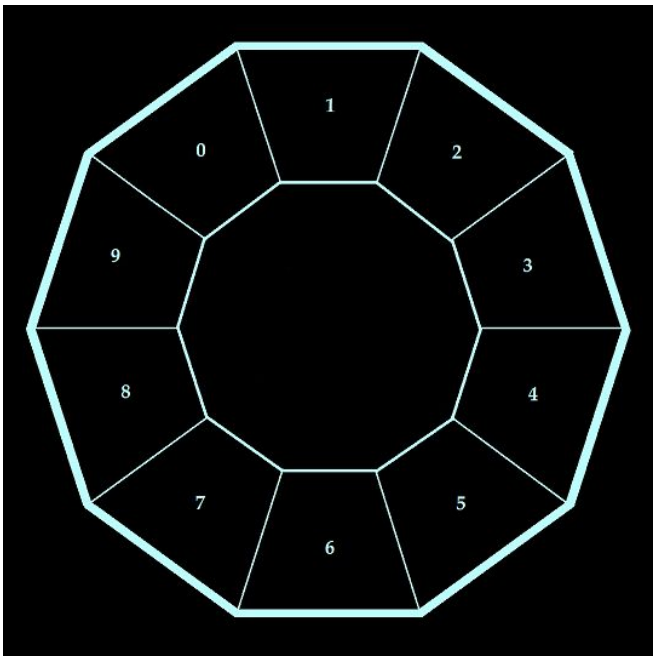

Figure 1. Swipin Decagonal Dial Design for swipe-select entering of 4 Digit PINs

speed and easier backtracking mechanism, as opposed to having to press backspace 4 times.

This paper proposes using a gesture-based input for entering a 4-digit PIN by creating an interface which allows users to swipe over the digits which are laid out in a doughnut-pie chart manner on the lock screen of

the phone. To select a number twice you would just have to swipe over into the inner concentric circle space and the digit would be selected. It is expected that this would allow the user to enter their existing 4-digit PIN in a faster and more usable way than typical tap-tap entry. It is also expected that cancelling the input and starting afresh will also be less frustrating and just swiping away from the Swipin dial would cancel all selections and you could just start drawing again. This is expected to be much easier for the users to interact with compared to the existing method of pressing backspace multiple times to clear the password entry field. The implementation application for this project will be built for an Android smartphone.

It has been observed that about 88% of smartphone users use digit based password to unlock their phones [22]. Using Swipin people can continue using their 4-digit PINs thereby reducing reluctance of the user in transitioning, keeping all the benefits that a 4-digit PIN provides with added ease and speed of input. The advantages of using 4-digit PIN include 10000 possible combinations of PINSs, easy transition from existing password input mechanism because of the sheer number of people currently using 4-digit PIN specifically for smartphone authentication

TECHNIQUE

The proposed technique for Swipin involves a dial design with a gap in the center with number 0-9 written around the edge of the dial. The user can swipe over multiple numbers to input the 4-Digit PIN.

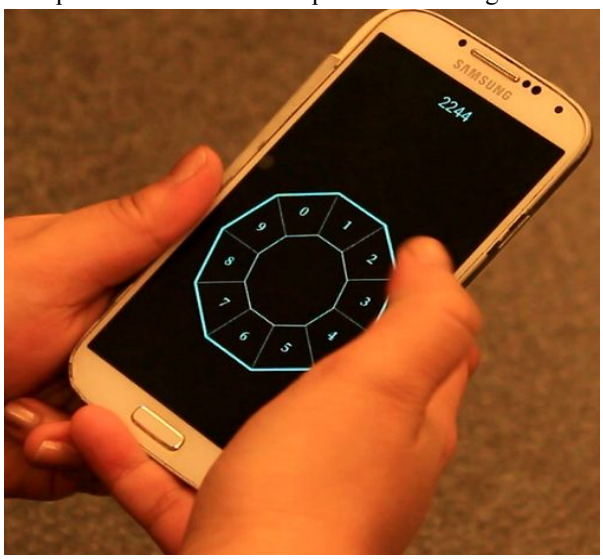

Figure 2. Swipin Lock Screen after the user has just entered the password and right before unlock occurs.

Alternatively, the user can also move their finger over the center of the dial to avoid touching a number that might be in the line of motion to reach the intended number.

The user can also move to the center from the number to allow re-entering the same number again.

This design enables user to be able have the least hindrance in swiping from one number to another number. This concept is an effective one because user can remember their passwords a s gesture as opposed to a number. This technique exploits the best of pattern recognition along with the security and memorability of a 4-Digit PIN.

RELATED WORK

The world is moving into two broad classifications for smartphone authentication: biometric and non-biometric authentication.

In the biometric authentication mechanisms of today, the system tries to authenticate the user on the basis of facial recognition, fingerprint recognition, voice/speech recognition or a combination of those. There are many real problems with these techniques[1]. The algorithms for implementing can be computationally exhaustive and might require special hardware neither of which are accessible to users with older or non-cutting edge devices. Fingerprint recognition is an idea that has been around for a long time[3] but has only been implemented in the most recent phones making it an inaccessible technology for a huge number of smartphones. On the other hand voice/speech recognition algorithms have their drawbacks for phone unlocking. Facial recognition algorithms require heavy computation for running the computer vision algorithms in the back end[4,5]. From usability perspective, the situation is even worse for facial/voice recognition such as the obvious hindrance of dictating your password out loud[6]. For all these techniques, another major drawback is the amount of time it takes for user to teach the machine their face/voice/fingerprint initially. Also, all these methods still employ non-biometric fallback authentication rendering their security only as good as the non-biometric authentication.

As far as non-biometric authentication methods are concerned, the work has been innovative and a lot of them have a cool factor. Arif, et al explore a unique tap+gesture authentication mechanism[9]. But among other things, they have a bit of a learning curve which makes them difficult to translate into everyday application.

For these purposes we decided to try to improve existing and widely-used non-biometric authentication interactions.

We devised a design space (Fig. 3) to situate our technique in relation to previous non-biometric authentication interactions, suggest connections between techniques, and direct attention to relatively under-explored combinations.
The rows indicate the authentication categories. The columns delineate the input strategy: Keyboard-based Input vs Swipe Action.

	Keyboard Based Input	**Swipe Action**
Alphanumeric	✓	✓
4-Digit Pin	✓	*Proposed*
Gesture/Pattern	✗	✓

Figure 3. Design space of swipe interaction, with rows for authentication categories: Alphanumeric Password, 4-Digit PIN and Gesture/Pattern Password—and columns for our two interaction strategies: Keyboard-based Input and Swipe Action.

The alphanumeric keyboard is the most widely used form of input mechanism.[14,15] So using it for passwords allows users to access the familiar keyboard and employ text passwords. One of main drawbacks of using this for smartphone authentication is that users prefer to spend as little time as possible unlocking their cellphones and alpha-numeric password entry is pretty time consuming[1,2]. This will not be surprising since studies show that people unlock their phone as many as 200 times a day[8]. Swipe entry for text-entry keyboards is not new either. Speak-as-you-swipe[11] by Sim, et al. even explores Swipe and Speech as a multi-modal input for mobile authentication. But that only explores alpha-numeric password entry. Neither the technology nor Swipe has yet been applied to entering 4-Digit PINs for quick authentication.

While many gesture mechanisms have been explored, they all have drawbacks.[13] Some of these are in the form of heavy computational overload because the resolution of the input area is too high because of the movement towards high pixel density in smartphones in the last decade. The pattern input combines the pros of gestures with the low resolution grid which makes it less computationally expensive. However, people end up using the most simplistic patterns for authentications thereby yielding "passwords with entropy far below the theoretical optimum".[7]

The 4 Digit PIN has been in use ever since it was chosen as the authentication mechanism for ATM machines by their inventor in 1967. And yet it is astounding that the input interface has not evolved in all these years despite the vast improvement in the input device resolution from being clunky buttons to seamless capacitive touch screens. A lot of work has gone into testing and improving security of the 4-Digit PIN as an authentication mechanism.

More recently, Zezschwitz, et al came up with the similarly named SwiPIN[10]. However, in this paper they used a very different input mechanism which involved swiping across two differently colored rectangles at the bottom of the screen for abstracting the PIN entry process for protection against over the shoulder surfers. This paper still had the digits in the standard 0-9 dial-pad grid and did not involve swiping over the digits in any way.

The idea of having a curved dial is also not revolutionary since the original telephones used to have a round dials because a rotation mechanism was involved for dialing. Once button press came to the scene the more familiar grid layout became commonplace. CurveDial[12] talks about "Vernier effect for speed parameter entry".

Therefore, in this paper, we propose Swypin which will satisfy the design space of being a non-biometric authentication mechanism which combines 4-Digit PIN authentication with a swipe-entry input interface. We propose that this will have a two-fold benefit of 1) producing an interaction interface that is more usable and 2) aiding memorability of the password itself.

IMPLEMENTATION

The application developed is designed for Android devices ranging from API 10 to API 23 which cover 100% of all existing android devices. For the purposes of this implementation Samsung Galaxy S4 was used as the main development and testing device.

For this purpose a lock screen application has been developed which will automatically launch when the phone is woken up using any of the buttons. The lock screen design includes two concentric decagons with their being 10 slices between the two decagons with digits from 0-9 on them as shown in Figure 1. For entering a 4-digit pin with all distinct numbers, the user can use one finger or thumb to draw a gesture on the number dial. For entering a number twice in succession the user must just drag their finger/thumb to the inner circle once to register first numbers selection and then

can drag finger/thumb back to that number and the application will recognize it as two distinct entries.

This was achieved by listening for touch on the unlock screen. As soon as touchdown is detected the classification algorithm kicks in to decide which number is being touched.

The algorithm monitors ACTION_MOVE and fetches the x-y coordinates to get current location of touch. It then checks to see if the current location is still on the same number. If not then it registers a new number touched. This is done by defining the area in terms of pixels for each of the numbers. The central area is classified as a neutral zone on touching which the current number detected is set to -1 and so you can either go back to the number you came from or move towards a new number.

The open source project by Joisar[23] was found to be adequate for the algorithm to set a simplistic custom lock screen on the phone. So for the purposes of this project this project was modified to include the touch recognition and classification algorithm that was discussed above.

SYSTEM ANALYSIS

The 4-Digit PIN password offers an entropy of 10^4 which is decent on its own but actually much higher when compared to the pattern lock. While implementing this application we had a chance to examine the Pattern Based Password mechanism. The way it works is that it fetches a string of 9 bits which correspond to the 9 zones on the screen classifies by relative pixel position much like we did in Swipin. This 10 bit starts off as 0s and the bit is flipped to 1 for each of the dots on the grid triggered. This gives us an entropy of $\sum_{r=4}^{9} {}^{9}_{r}C$ because the algorithm enforces at least 4 dot pattern, which brings us to a total entropy of 382 which is very little compared to the 10^4 of 4-Digit PIN.

While Alphanumeric passwords fare better in terms of entropy, they are also difficult to remember and input which is a problem to be solved by designers of the keyboard layout itself or the input mechanism for Alphanumeric input, and does not compete with our design.

So we compare Swipin instead with the traditional 4-Digit PIN input mechanism that employs buttons and button-click listeners, which we will now be calling the Tap mechanism.

Participants & Procedure

For the purposes of this study we designed an application that would employ Swipin as well as an implementation of the traditional 4-Digit PIN button-based input mechanism.

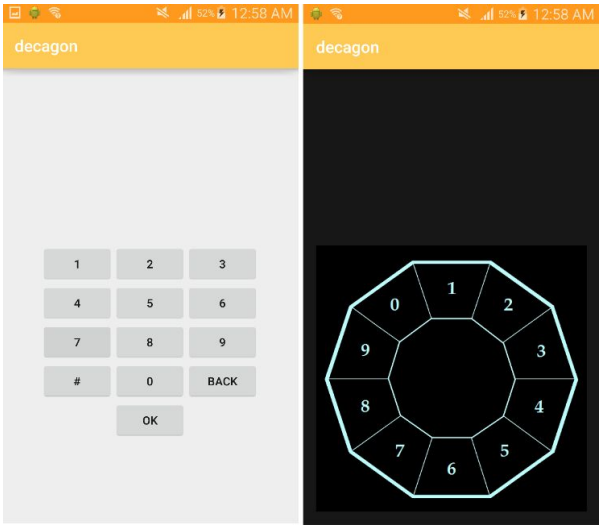

Figure 4: On the left we have the traditional button-based PIN input mechanism and on the right we have Swipin implemented as a stand alone application.

The application was designed to capture the time taken, in nanoseconds, to begin and complete one PIN input on any of the methods.

For the purposes of the study we used 10 people in the age range 20-37 (Median: 21, 3F/7M).

We asked them to perform the following tasks: a)Enter the PIN 1234 on Swipin a total of 10 times b) Enter the PIN 6348 on Swipin a total of 10 times c) Enter the PIN 1234 on Tap a total of 10 times d) Enter the PIN 6348 on Tap a total of 10 times.

We employed these 2 PIN in particular as they differ in complexity of input with '6348' being relatively difficult to input on both the methods and '1234' being the most common of all 4-Digit PIN passwords.

Design

The experiment consisted of 2 phases. In the first phase participants worked with Swipin and entered 2 passwords 10 times each. In the second phase participants worked with Tap and entered 2 passwords 10 times each. This was performed for 10 participants. This gave us a factorial design of 400 (10 Participants x 2 PINs x 10 Attempts x 2 Systems).

The participants were explained what Swipin was with one quick demonstration on how one might use it. However, the first time they tried it themselves was during the testing, unlike the Tap method which people are largely familiar with already. The participants were encouraged to perform the tasks as quickly as possible and in quick successions.

The application logged the time taken to perform the input of 1 PIN on each of the systems in nanoseconds.

Results and Discussion

For each of the tasks we analyzed the data in terms of Attempt Number on X-Axis and the time taken in nanoseconds on the Y-Axis.

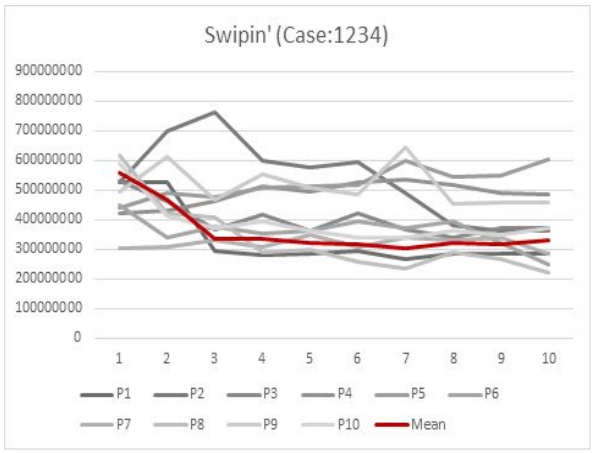

Figure 5: Swipin Case:1234 shows that the mean time taken for entering the password declines with number of attempts.

For Swipin Case:1234, the average time taken by participants dropped by a factor of 2 by the third attempt and stayed there in a more or less linear way through the rest of the study.

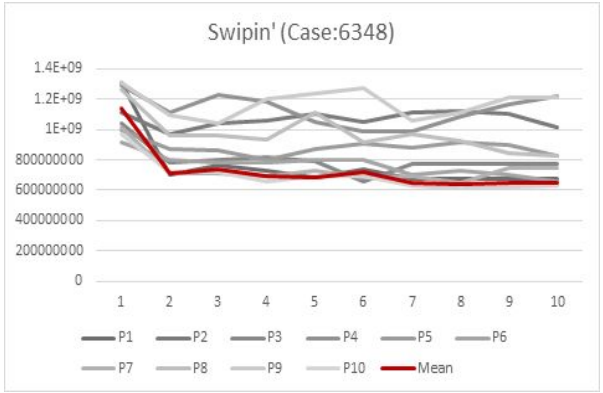

Figure 6: Swipin Case:6348 shows that the mean time taken for entering the password declines with number of attempts.

For Swipin Case:6348 the average time taken dropped by a factor of ~1.5 as early as the second attempt and then maintained that time for the rest of the study.

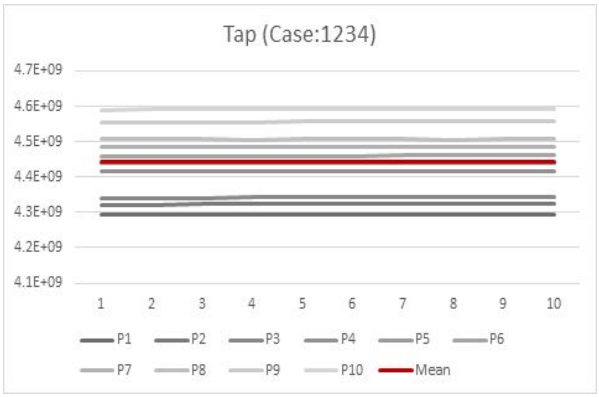

Figure 7: Tap Case:1234 shows that the mean time taken for entering the password declines with number of attempts.

For Tap Case:1234 the average time taken remained more or less constant at around 4443 milliseconds.

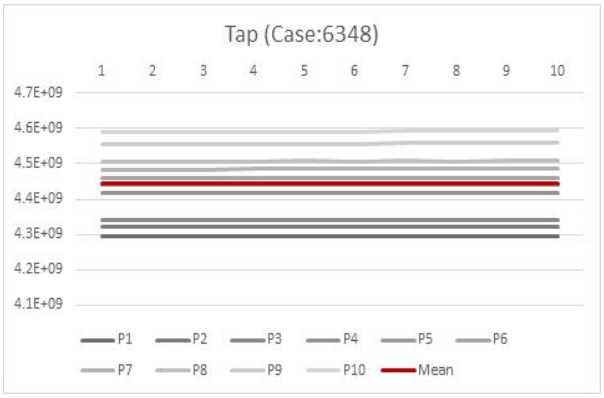

Figure 8: Tap Case:6348 shows that the mean time taken for entering the password declines with number of attempts.

For Tap Case: 6348 the average time remained more or less constant throughout the study at around 4445 milliseconds which was less than the average time for Tap Case:1234 by 3 milliseconds.

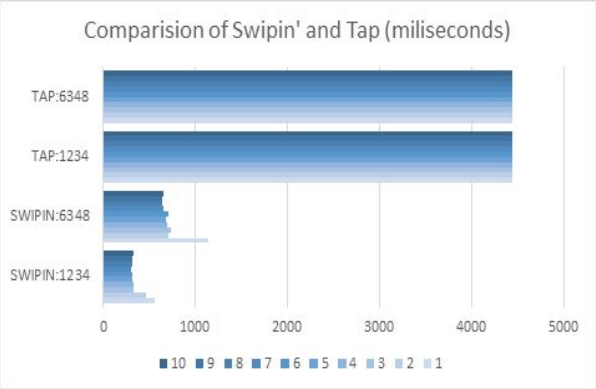

Figure 9: Comparison between Swipin and Tap input times in milliseconds clearly demonstrates that Swipin is faster than Tap by a factor of 9 for contiguous input like 1234 and a factor of 5 for a more complicated input like 6348.

On evaluating and comparing the two methods, the following is clear:

a) Swipin has the potential to be an acquired skill where the number of attempts for getting better input time can be as low as 2 or 3, following which one might expect to more or less reach the best speed they will acquire for a particular input.
b)The Tap method does not improve on input time with number of attempts.
c)Swipin is clearly fast and easy to pick up on.
d)Swipin is definitely faster than Tap method.

USER STUDY

Participants & Procedure

The study took 10 participants from the previous study after they had performed the tasks for a quick survey. The participants were given as much time as they desired to fill out the survey.

The survey asked the participants to grade the following on a scale of 1 to 5 (1 being least favorable):

Ease of Use, Design, Would You Use this?, Do you prefer this over traditional unlock screen?

The survey also asked users the subjective questions: What do you like the most about Swipin? and What do you like the least about Swipin?

Results and Discussion

The users gave very positive responses to the rating questions with at least a mean of 4 for each and a mean of 5 for the question "Would you use this?" This response encourages us to believe that Swipin is a successful design for 4-Digit PIN based smartphone unlocking.

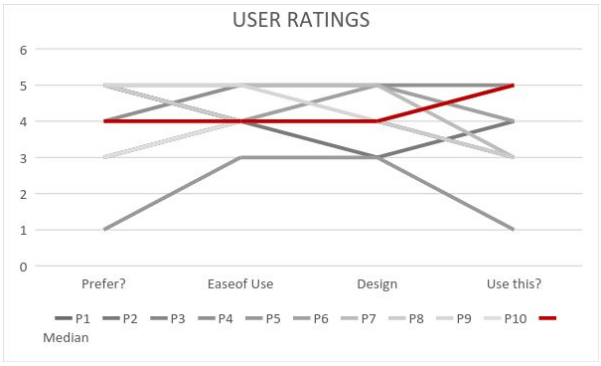

Figure 10: User Ratings for Swipin show very favorable responses with at least a median of 4/5 for all parameters and 5/5 for the question "Would you use this?"

The users mentioned the words "fast", "easy", "efficient" and "fun" to answer the question "What do you like the most about Swipin?". Users mentioned that they "liked to swipe" and that they "did not have to lift their finger" while answering the above question. One user said that "it is easy if your code numbers are in a row (1,2,3,4) but hard otherwise". Another user said of the system that "It becomes easier to memorize the password". Another user said that they liked the design of the Swipin UI.

While answering the question "What do you like the least about Swipin?" the user mentioned that it was harder to enter number which were "all over the place". Two users expressed doubts about whether it was better than the 3x3 grid and said that they would prefer Fingerprint Recognition over anything else. One user expressed concern saying that "the traditional keyboard layout is more intuitive".

DISCUSSION

Swipin is a successful design in term of speed, efficiency and user reviews. The System Analysis demonstrates that the speed of input increases as the user makes more attempts at the input with significant improvement as early as the second or third attempt. On the other hand, Tap method does not hold a lot of scope for input speed improvement.

The User study demonstrated that the user opinions of Swipin are highly favorable and very encouraging for future work to be performed to improve the design aspects of Swipin.

The users also collaborated with our initial hypotheses that swipe input is favorable to tap input. It also

collaborates with the hypotheses that Swipin would help with memorability of the password.

LIMITATIONS AND FUTURE WORK

Users explicitly mentioned that while they liked the overall idea of Swipin, they wondered if the design itself could be improved upon. Further work needs to be put in towards conducting analysis of number arrangements and the look and feel of the dial.

From implementation perspective, the current system classifies coordinates in absolute pixel coordinates. For the system to be made portable to all devices, the algorithm would have to consider the relative pixel coordinates instead.

One of the users mentioned that the centre area of the design seemed to be wasted, whereas the center is currently being used to classify touch input for in-between number movement and for re-entering the same number multiple times. But this question does open up the possibilities of perhaps detecting taps, etc. in the centre space as an additional interaction for the purpose of adding another layer to the password or as a gesture to perform a task.

CONCLUSION

In this paper we have presented the Swipin application for unlocking your smartphone with swipe-select based 4-Digit PIN input. We developed the Swipin design over many iterations to get a design that work on many levels. We performed a system analysis which demonstrates that Swipin is faster than the traditional Tap method of 4-Digit PIN input. It is also a better user experience in terms of speed, efficiency and fun. We also demonstrated that Swipin has a lot of potential with almost all the users saying that they would personally like to use it and a lot of them going so far as to say that they prefer it over existing methods.

ACKNOWLEDGEMENTS

I would like to extend my gratitude to all the participants of the user studies for their cooperation and valuable feedback.

REFERENCES

1. Shari Trewin , Cal Swart , Larry Koved , Jacquelyn Martino , Kapil Singh , Shay Ben-David, Biometric authentication on a mobile device: a study of user effort, error and task disruption, *Proceedings of the 28th Annual Computer Security Applications Conference, December 03-07, 2012, Orlando, Florida, USA*
2. P. Bao, J. Pierce, S. Whittaker, and S. Zhai. Smart phone use by non-mobile business users. *In MobileHCI, Stockholm, Sweden, 2011.*
3. T. Y. Tang , Y. S. Moon , K. C. Chan, Efficient implementation of fingerprint verification for mobile embedded systems using fixed-point arithmetic, *Proceedings of the 2004 ACM symposium on Applied computing, March 14-17, 2004, Nicosia, Cyprus*
4. M. Kirby , L. Sirovich, Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, v.12 n.1, p.103-108, January 1990*
5. Andrew Senior , Rein-Lien Hsu , Mohamed Abdel Mottaleb , Anil K. Jain, Face Detection in Color Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence, v.24 n.5, p.696-706, May 2002*
6. Shaojian Zhu , Yao Ma , Jinjuan Feng , Andrew Sears, Don't listen! I am dictating my password!, *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility, October 25-28, 2009, Pittsburgh, Pennsylvania, USA*
7. Darren Davis , Fabian Monrose , Michael K. Reiter, On user choice in graphical password schemes, Proceedings of the 13th conference on USENIX Security Symposium, p.11-11, August 09-13, 2004, San Diego, CA
8. Tao, Q., & Veldhuis, R. N. (2006). Biometric Authentication for a Mobile Personal Device. *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*. doi:10.1109/mobiq.2006.340409
9. Ahmed Arif, Michel Pahud, Ken Hinckley, William Buxton (2013). A tap and gesture hybrid method for authenticating smartphone users. *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services -MobileHCI*
10. Emanuel von Zezschwitz , Alexander De Luca , Bruno Brunkow , Heinrich Hussmann, SwiPIN: Fast and Secure PIN-Entry on Smartphones, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, April 18-23, 2015, Seoul, Republic of Korea*
11. Khe Chai Sim, Speak-as-you-swipe (SAYS): a multimodal interface combining speech and gesture keyboard synchronously for continuous mobile text entry, *Proceedings of the 14th ACM international*

conference on Multimodal interaction, October 22-26, 2012, Santa Monica, California, USA

12. Grham Smith , m. c. schraefel , Patrick Baudisch, Curve dial: eyes-free parameter entry for GUIs, *CHI '05 Extended Abstracts on Human Factors in Computing Systems, April 02-07, 2005, Portland, OR, USA*
13. Mauricio Cirelli , Ricardo Nakamura, A Survey on Multi-touch Gesture Recognition and Multi-touch Frameworks, *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces, November 16-19, 2014, Dresden, Germany*
14. Shumin Zhai , Per Ola Kristensson, Interlaced QWERTY: accommodating ease of visual search and input flexibility in shape writing, *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, April 05-10, 2008, Florence, Italy*
15. Sunyu Hwang , Geehyuk Lee, Qwerty-like 3x4 keypad layouts for mobile phone, *CHI '05 extended abstracts on Human factors in computing systems, April 02-07, 2005, Portland, OR, USA*
16. Caroline Appert and Shumin Zhai. 2009. Using strokes as command shortcuts.*Proceedings of the 27th international conference on Human factors in computing systems - CHI 09* (2009). DOI:http://dx.doi.org/10.1145/1518701.1519052
17. Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. 2013. Memorability of pre-designed and user-defined gesture sets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*(2013). DOI:http://dx.doi.org/10.1145/2470654.2466142
18. T. J. Hazen, E. Weinstein, B. Heisele, A. Park, and J. Ming. Multimodal face and speaker identification for mobile devices. In R. I. Hammoud, B. R. Abidi, and M. A. Abidi, editors, *Face Biometrics for Personal Identification: Multi-Sensory Multi-Modal Systems*. Springer, 2007.
19. Guo, Y., Yang, L., Ding, X., Han, J., and Liu, Y. *Opensesame: Unlocking smart phone through handshaking biometrics. In INFOCOM* (2013).
20. Matsuo, K., Okumura, F., Hashimoto, M., Sakazawa, S., and Hatori, Y. *Arm swing identification method with template update for long term stability.* In ICB (2007).
21. Kim, I. Keypad against brute force attacks on smartphones. *IET Information Security 6, 2* (2012), 71-76.
22. Jakobsson, M., Shi, E, Golle, P., and Chow, R. Implicit authentication for mobile devices. *In Proc. HotSec '09. USENIX* (2009), 9-9
23. Source: https://github.com/Joisar/LockScreenApp