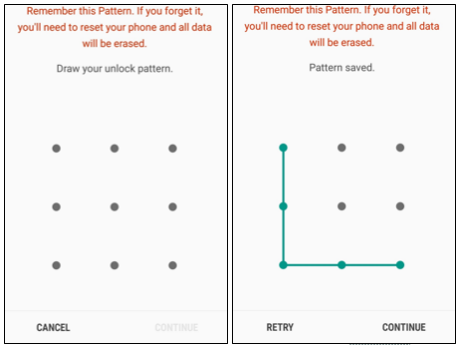

Figure 1: Android Pattern Lock interface on a Galaxy S8.

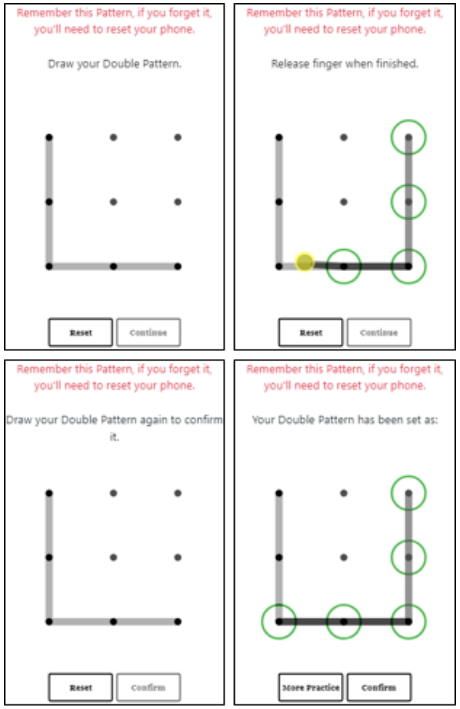

Figure 2: An example of our DPat interface during the selection phase.

Is Two Better Than One? Extending Android Unlock Patterns to Utilize Multiple Patterns

Tim Forman
United States Naval Academy
m201902@usna.edu

Adam J. Aviv
The George Washington University
aaviv@gwu.edu

ABSTRACT

Android unlock patterns are a knowledge-based graphical authentication that have a simple design, whereby users create and recall a single-stroke pattern drawn on a 3x3 grid. Unlock patterns are known to suffer from security issues due to human factors in the selection process that can make a user's pattern easy to guess. As a solution to increase the complexity of user selected patterns without complicating the well-worn UI, we propose *Double Pattern* whereby a user selects and enters two separate pattern-strokes on the same 3x3 grid. Here, we present our preliminary work on a prototype design and user-study with the long term goals of a full evaluation for both security and usability.

CCS CONCEPTS

• **Security and privacy → Mobile platform security**.

KEYWORDS

Mobile Security; Android Unlock Patterns; Android Interface; Usable Security

ACM Reference Format:
Tim Forman and Adam J. Aviv. 2019. Is Two Better Than One? Extending Android Unlock Patterns to Utilize Multiple Patterns. In *Proceedings of Fifteenth Symposium on Usable Privacy and Security (SOUPS'19)*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.475/123_4

(1) Informed Consent
(2) Introduction to Double Patterns
(3) Device Usage and Biometric Utilization Questions
(4) Question Regarding Currently Utilized Authentication Method
(5) Introduction of DPat interface
(6) Practice Use of DPat Interface
(7) Scenario Selection: Unlock and either Shopping or Banking (random order and selection)
(8) DPat Selection for Each Scenario and Followup Questions (x2)
(9) Simple Usability Scale Questions
(10) Recall DPat for Each Scenario (x2)
(11) Comparison to Other Authentication Methods
(12) Open Response Questions Regarding User Selected Double Patterns
(13) Demographic Information Questions
(14) Integrity Question and Final Submission
(15) Survey Feedback Page Containing Finish Code for Mechanical Turk

Figure 3: Survey Procedures: survey questions omitted due to space.

INTRODUCTION

Android unlock patterns are perhaps the mostly widely used graphical authentication interface. This is a testament to its simple and intuitive design whereby a user draws, in a continuous stroke, a "pattern" on a 3x3 grid of contact points. Despite known susceptibility to smudge attacks [1, 3] and bias towards contact point utilization [9], the utilization of Android unlock patterns remains quite popular [5].

One notable shortcoming of Android unlock patterns, is that as a knowledge-based authenticator, there are significant human-induced flaws in the pattern selection process. Although there are 389,112 possible patterns, users tend to select from a much smaller set of patterns in predictable ways [1, 9]. While there have been a number of proposals to improve the current state, such as providing user guided selection [4] or password-meters [7], these interfaces require add-on design principles that significantly change the user-interface (UI) of Android unlock patterns during the selection process.

Here, we propose a new approach to this problem that is twofold – our approach captures the intuitive UI of Android unlock patterns and finds transparent, natural steps to improve upon the current state, by increasing the security complexity of the authenticator without complicating the UI and requiring more in-depth comprehension on the part of the user. As a first step, we report on a prototype design and evaluation of a *Double Pattern* (*DPat*) system. DPat maintains the same 3x3 grid UI of traditional patterns, with the simple addition that a user selects and recalls *two patterns* during authentication. Each pattern is entered on the same grid using two different strokes.

Herein, we present some preliminary analysis of ($n = 24$) participants in a prototype user-study of DPat, where we implemented a web-based survey that recreated a number of security scenarios (similar to prior work by Loge et al. [6]) and asked participants to create and recall patterns, as well as report on the usability and security perceptions of the system.

We are now in the process of conducting our full study, with ($n = 236$) participants having taken our survey thus far, exclusively on mobile devices. We look to continue our collection, and our future work will consist of processing this data using standard statistical techniques to inform us of the human-factors involved with Double Patterns.

SOUPS'19, August 2019, Santa Clara, CA, USA

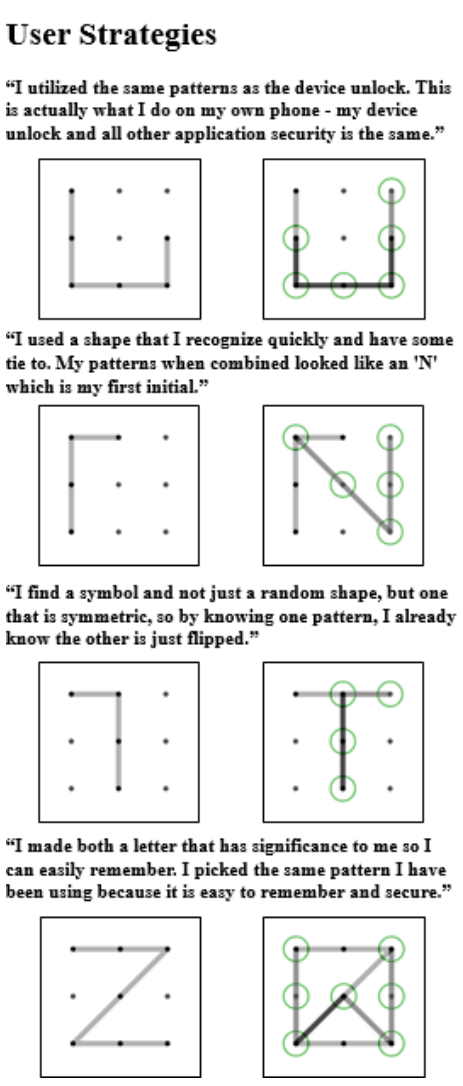

Figure 4: Various user methodologies used during the DPat selection phase in our prototype.

BACKGROUND AND RELATED WORK

Android Unlock Patterns originated from a simple graphical password scheme named Pass-Go [8], however Android patterns are more restricting than the original interface. An Android pattern consists of a 3x3 grid, and a user "draws" a pattern in a single stroke by connecting nodes (or contact points) in the 3x3 grid. Not all patterns are valid, and there exists a basic rule-set. The rule-set is as follows:

(1) No less than four contact points must be selected.
(2) No repetition of contact points.
(3) Only straight lines are allowed.
(4) All contact point along a path must be connected, unless previously selected.

For an example of the pattern interface typically by seen by users, refer to Figure 1.

Previous studies into the security of Android unlock patterns have proposed changes in the form of additions to the existing interface. However, a potential pitfall of these alterations is the increased user-comprehension complexity of the interface. For example, Choi et al. proposed system guided assistance during selection, SysPal [4], that would require users to use certain contact points to select their pattern. While this increases the diversity and security of patterns, it also creates added burden during the selection process and significant interface changes. Password meters may have a similar effect [7], which does not require drastic interface changes, but also requires increased user comprehension and the meter advice may be ignored. In proposing double pattern it is our hope that this simple and natural interface update can have a significant increase on the security of the system without increasing the complexity of the pattern selection interface.

DOUBLE PATTERN DESIGN

Our interface is simple in that the only additional step required of users is to draw a second pattern during the selection phase. The rules of pattern selection as well as the visual features of the interface itself remain the same as using the original 3x3 grid. Figure 2 illustrates an example of how a user creates a Double Pattern. Similar to the structure of the original interface, the user is prompted to create their Double Pattern, and then required to verify it for the new security policy to take effect. The only notable difference is that the user is drawing two patterns in sequence, vice one pattern. During recall/unlock, the user would simply enter both patterns, in the same order, on a single 3x3 grid.

DESIGN AND METHODOLOGY

We developed a prototype online survey using Django and a JavaScript pattern entry tool (patternLock [10]). Our current survey must be completed on a mobile device using Amazon Mechanical Turk, but for the purposes of prototyping, we used desktop computers. Our sample consisted primarily

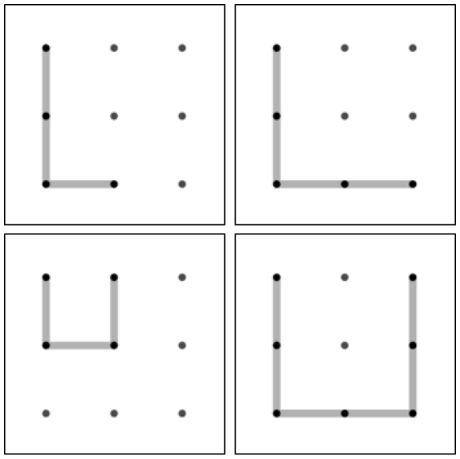

Figure 5: Common patterns selected include the 'L' and 'U' shapes, found to be common in other related pattern studies [2].

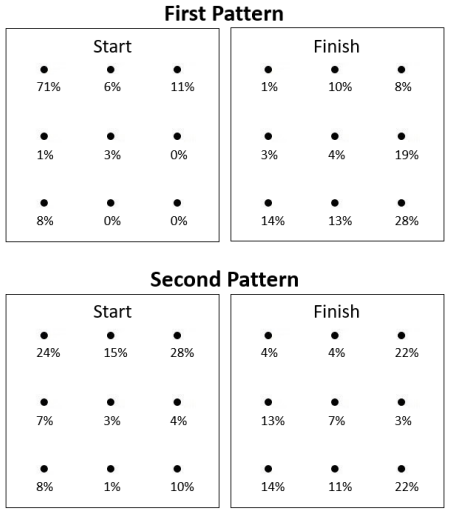

Figure 6: Usage of contact points during the DPat selection phase.

of students from the Computer Science Department at our institution who volunteered to provide feedback on the survey and preliminary data.

We designed the procedure of the survey to allow for both selection and recall of Double Patterns from two distinct scenarios. The scenario presented to all participants is that of unlocking/securing your mobile device, the other scenario is either securing a shopping application on a mobile device or securing a banking application on a mobile device. These scenarios were used by Loge et al. [6], and in a cross-data set comparison by Aviv et al. [2]. It has been shown that having different security settings can help diversify the data set. The order of the two scenarios was randomized within the survey. The overall structure of the survey is presented in Figure 3. In addition to our collection of data during the selection and recall phases, we also asked demographic, usability, and open response questions.

PRELIMINARY ANALYSIS

For the prototype run of our study, we recruited $n = 24$ participants. For the prototype, we mostly focused on refining the survey rather than the direct question responses from the participants. However, the Double Patterns selected are informative of the potential in this space and where we focus our analysis. Among our sample, we collected data pertaining to the users' level of confidence using the Double Pattern interface, their methodologies related to the choice of two (rather than one) patterns 4, and statistical data relating to the actual Double Patterns selected. Figure 5 illustrates the four most common patterns we found in our prototype. We also examined contact point usage in Double Pattern selection, illustrated in Figure 6. The primary concern expressed by users using the DPat interface was memorability. As we collect more data, we will be able to explore these areas in depth, as well as tactile input speed as compared to traditional patterns and 6+ digit PINs.

CONCLUSION AND FUTURE WORK

We propose Double Patterns, a natural design progression for Android unlock patterns that maintains the well-worn UI with the potential to increase the security complexity while maintaining usability. Reflecting on our preliminary analysis and examining similar work in this field, we believe that Double Pattern warrants further investigation. It is clear that the intuitive nature of the interface would allow easy adoption and has a lot of potential. Within our prototype, there was substantial interest focused on the further development of our survey as well as the results of our study.

We are currently in the process of collecting data from Amazon Mechanical Turk users. Using this data, we will look to better understand the security of this system using the standard statistical procedures, such as guessability [1, 9], as well as do a more substantive analysis of the usability response and qualitative feedback. We believe that based on our prototype, our DPat interface will reflect a statistically lower level of guessability, while not sacrificing usability or memorability with users.

REFERENCES

[1] Adam J. Aviv, Devon Budzitowski, and Ravi Kuber. 2015. Is Bigger Better? Comparing User-Generated Passwords on 3x3 vs. 4x4 Grid Sizes for Android's Pattern Unlock. In *Annual Computer Security Applications Conference (ACSAC '15)*. ACM, Los Angeles, California, USA, 301–310.

[2] Adam J. Aviv and Markus Dürmuth. 2018. A Survey of Collection Methods and Cross-Data Set Comparison of Android Unlock Patterns. *CoRR* abs/1811.10548, 1–20.

[3] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. 2010. Smudge Attacks on Smartphone Touch Screens. In *USENIX Workshop on Offensive Technologies (WOOT '10)*. USENIX, Washington, District of Columbia, USA, 1–7.

[4] Geumhwan Cho, Jun Ho Huh, Junsung Cho, Seongyeol Oh, Youngbae Song, and Hyoungshick Kim. 2017. SysPal: System-Guided Pattern Locks for Android. In *IEEE Symposium on Security and Privacy (SP '17)*. IEEE, San Jose, California, USA, 338–356.

[5] Marian Harbach, Emanuel von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. 2014. It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception. In *Symposium on Usable Privacy and Security (SOUPS '14)*. USENIX, Menlo Park, California, USA, 213–230.

[6] Marte Løge, Markus Dürmuth, and Lillian Røstad. 2016. On User Choice for Android Unlock Patterns. In *European Workshop on Usable Security (EuroUSEC '16)*. ISOC, Darmstadt, Germany.

[7] Youngbae Song, Geumhwan Cho, Seongyeol Oh, Hyoungshick Kim, and Jun Ho Huh. 2015. On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks. In *ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, Seoul, Republic of Korea, 2343–2352.

[8] Hai Tao and Carlisle Adams. 2008. Pass-Go: A Proposal to Improve the Usability of Graphical Passwords. *International Journal of Network Security* 7, 2, 273–292.

[9] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns. In *ACM Conference on Computer and Communications Security (CCS '13)*. ACM, Berlin, Germany, 161–172.

[10] Sudhanshu Yadav. 2018. patternLock. https://github.com/s-yadav/patternLock. *GitHub repository*.