Easy to Draw, but Hard to Trace? On the Observability of Grid-based (Un)lock Patterns

Emanuel von Zezschwitz[1], **Alexander De Luca**[1,2], **Philipp Janssen**[1], **Heinrich Hussmann**[1]
[1]Media Informatics Group, University of Munich (LMU), Munich, Germany
[2]DFKI GmbH, Saarbrücken, Germany
{emanuel.von.zezschwitz, alexander.de.luca, hussmann}@ifi.lmu.de, janssen@cip.ifi.lmu.de

ABSTRACT
We performed a systematic evaluation of the shoulder surfing susceptibility of the Android pattern (un)lock. The results of an online study ($n = 298$) enabled us to quantify the influence of pattern length, line visibility, number of knight moves, number of overlaps and number of intersections on observation resistance. The results show that all parameters have a highly significant influence, with line visibility and pattern length being most important. We discuss implications for real-world patterns and present a linear regression model that can predict the observability of a given pattern. The model can be used to provide proactive security measurements for (un)lock patterns, in analogy to password meters.

Author Keywords
Pattern; Authentication; Observability; Security

ACM Classification Keywords
D.4.6. Security and Protection: Authentication

INTRODUCTION
Mobile devices (e.g. smartphones) provide access to potentially sensitive data. As a consequence, several security mechanisms were introduced to hamper unauthorized use. Besides personal identification numbers (PIN) and biometric approaches (e.g. fingerprint), graphical approaches have widely been adopted. The most prominent example is Google's pattern (un)lock which is similar to a usability optimized version of the Draw-a-Secret concept [5], the first recall-based graphical password system. Gesture-based approaches have already been shown to be usable alternatives to PIN [8]. At the same time, they are highly prone to observation attacks [9]. This can be a serious drawback as grid-based authentication is mainly deployed on mobile devices and interaction with such devices often takes place in the public [4].

With the introduction of the Android pattern (un)lock, researchers started investigating usability and security features of such approaches on mobile devices. Aviv et al. [2] showed that grid-based authentication is prone to smudge attacks, an attack which exploits oily residues on the screen to deduce a formerly entered pattern. Uellenbeck et al. [6] analyzed the guessability of patterns and showed that users are biased in their pattern choice and therefore only a small fraction of the theoretical password space is actually used. Andriotis et al. [1] proposed to utilize proactive security meters to support users in selecting more secure patterns. They define pattern strength based on features like length, overlapping nodes and knight moves (see Figure 1). However, the importance and the relative weight of these features was not analyzed.

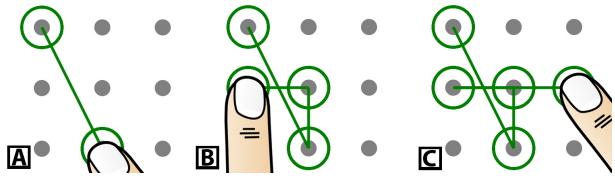

Figure 1. The input of a pattern (length = 5) with visible lines comprising the analyzed features "knight move" (A), "intersection" (B) and "overlap" (C). During the user study, the finger was always fully visible and the angle was fixed at 45° (adjusted for presentation).

Although shoulder surfing problems are often suggested and alternative approaches are proposed (e.g. [3, 7]), the observability of the pattern (un)lock has not yet been systematically investigated. In contrast to the evaluation of smudge attacks and guessing attacks, pattern (un)lock solely serves as the baseline condition for novel shoulder surfing resistance approaches (e.g. [7]). To our knowledge, the most relevant work was done by Zakaria et al. [9]. They analyzed the observability of the Draw-a-Secret scheme and proposed decoy strokes and disappearing strokes as a potential solution.

In this paper, we present the first systematic evaluation of the observability of grid-based (un)lock patterns. We conducted an online study with 298 participants who attacked 5960 patterns of various length and complexity. Our approach allowed us to weigh the impact of single pattern characteristics like length, knight moves, overlaps and visual appearance. We present a prediction model to assess the shoulder surfing risk for a given pattern and discuss the implications for user-selected patterns. This work contributes to the field of usable security by providing ground truth for the shoulder surfing vulnerability of such (un)lock patterns and can be the basis for novel types of proactive security measurement systems which could help users to choose patterns which are easy to enter, but harder to observe.

CHI'15, April 18 - 23 2015, Seoul, Republic of Korea.

ACM 978-1-4503-3145-6/15/04...$15.00.
http://dx.doi.org/10.1145/2702123.2702202

THREAT MODEL

In our threat model, a user draws the pattern in a (semi-)public setting. The attacker, who has no previous knowledge about the characteristics (e.g. length) of the drawn pattern, has perfect sight on the display. There are no occlusions and no distracting reflections. The attacker sees the whole authentication exactly once as there is no technical equipment involved (e.g. video recording). Immediately after the attack, the observer gets in possession of the device (e.g. by theft) and redraws the observed pattern to authenticate.

OBSERVABILITY STUDY

Since we wanted to measure how easily an attacker can reproduce an observed pattern, the patterns were generated by machine, simulating the unlock behavior of a human, and the observation and reproduction could be carried out through a Web application. Please note that there is no IRB at our institution for this kind of studies. Nevertheless, we made sure that the study complied with strict privacy regulations.

Design and Implementation

The study followed a repeated measures design. We specified the following independent variables: *a*) *line visibility* [false/true], *b*) *length* [4-9], *c*) *knight move* [0-4], *d*) *overlap* [0-3] and *e*) *intersection* [0-7].

Line visibility specifies if drawn patterns are visualized or not. The *length* of a given pattern is described by the number of activated cells. A *knight move* describes the connection of two cells which are not directly neighboured (Figure 1, A). *Intersections* (Figure 1, B) occur whenever an already drawn line is crossed by another line. *Overlaps* (Figure 1, C) result from crossing over already activated cells. Patterns were randomly generated based on the standard rules for Android devices: a valid pattern must comprise a minimum length of four, cells can only be activated once (and stay activated thereafter) and a pattern must consist of straight lines only (cells on a straight line cannot be skipped). Pattern *length* and *intersections* were randomly assigned. A *knight move* or an *overlap* was used with a probability of 20%, whenever such a move was possible. *Line visibility* was alternated and therefore assigned to exactly 50% of the tested patterns. Therefore, we tested similar patterns (with similar complexity) with both, visible and invisible lines.

Shoulder surfing *success* was measured in two ways: binary (true/false) and as percentaged success rate. Success rate represents the sum of correctly activated cells divided by the entirety of correct cells (length of the expected pattern). A cell was specified correct when its position in the matrix and its position within the pattern matched the expected cell.

The study software was developed using JavaScript. User interaction was logged using PhP/ MySQL. The size of the $3 * 3$ matrix was 500px per edge. Each observation started with a three seconds countdown. Afterwards, the (animated) pattern was drawn. The input was simulated by a finger as seen in Figure 1. Single strokes took about 500 ms, depending on pattern complexity (input speed was derived from [8]). Participants started drawing by pressing the left mouse button and finished by releasing it.

Procedure and Participants

Each session consisted of 23 shoulder surfing attacks. First, an introduction page explained the procedure and all other important aspects of the user study. Whenever the participants felt ready, they pressed start and the training task began. Each user was trained the same three differently complex patterns. After the training was finished, 20 more patterns were tested. Each shoulder surfing attack comprised the following steps: a) three seconds countdown, b) pattern observation, c) pattern input and d) feedback.

Each pattern was observed exactly once *(b)*. The guessed pattern *(c)* was submitted using a confirm button and could be cleared using a reset button. A maximum of three attempts was given to submit the correct pattern. After a correct pattern or three failed attempts, users rated the attack *(d)* using two Likert scale questions. After all 20 patterns were tested, the participants answered a short questionnaire collecting demographic data and task specific information (e.g. shoulder surfing experience). Furthermore, we asked if any additional equipment (e.g. pen and paper) was used for the attacks. We did not mention this aspect in the introduction as this could have influenced the participants' behavior. If additional equipment was reported, we excluded the sample from the analysis. On average, the session was completed within 13 minutes (SD=5). All participants had the chance to win one of two eBook readers. The required email addresses were stored separately and could not be joined with the study data. To keep the participants motivated, the chance to win one of the devices increased with the number of successfully attacked patterns. Multiple participation was forbidden.

Two users were removed due to using additional equipment, leaving 298 correct data sets, 151 (51%) were male. Participants were invited using university mailing lists and social networks. The average age was 32 (14-73, SD=13). The majority used Android devices (59%) on a daily base, 29% were using other smartphones (e.g. iPhone), the rest (12%) was not using a smartphone. 31% did not use a lock screen, 30% used PIN, 28% used the pattern (un)lock and 11% used other methods. 15 participants had already been victims of shoulder surfing attacks, 44 had actively observed an authentication.

RESULTS

We removed 61 outliers from the original 5960 patterns. The removed samples had extreme values which exceeded the specified range of the independent variables (e.g. knight moves > 4). Table 1 (left) shows the main statistics of the final pattern set. The algorithm generated patterns of various complexity. We tested simple patterns which are likely to be used by humans (e.g. 12% of the tested patterns comprised visible lines without any special move) and more complex ones which are unlikely to be used in the wild. Only testing the full range of complexity allowed us to assess the impact of each single pattern feature. Finally, we restricted the analysis to the user's first guess as preliminary analyses showed that the chance of correctly drawing a pattern after the first failed attempt was only 6%.

	Descriptive Statistics			Binary Logistic Model		Linear Regression Model		
	Mean (SD)	Median	Range	B (SE)	Odds Ratio (95% CI)	B (SE)	β	VIF
Line	-	-	false [0], true [1]	-1.12 (.07)	0.33* (.29, .37)	14.42 (.69)	.23*	1.00
Length	6.36 (1.72)	6,00	4-9	-0.60 (.026)	0.55* (.52, .58)	-5.27 (.27)	-.29*	1.78
Knight move	0.92 (0.93)	1,00	0-4	-0.38 (.05)	0.68* (.62, .75)	-3.99 (.53)	-.12*	2.02
Overlap	0.38 (0.65)	0,00	0-3	-0.22 (.05)	0.80* (.72, .89)	-3.20 (.60)	-.07*	1.27
Intersection	1.06 (1.39)	1,00	0-7	-0.13 (.04)	0.88* (.82, .95)	-2.05 (.39)	-.09*	2.52
Constant	-	-	-	5.52 (.17)	-	97.59 (1.85)	-	-

Table 1. Left: Mean values, median and ranges of the tested pattern features; Center: B-values and odds ratio of the logistic regression model predicting binary success; Right: B-values, standardized betas and variance inflation factor of the linear regression model predicting the success rate. Line visibility was coded: 0=false, 1=true. All tested features have a significant individual influence on shoulder surfing success (*$p < .001$).

Feature Weights

Firstly, we define *success* as *false* (coded as 0) and *true* (coded as 1). With this measure, 3565 (51.7%) patterns were successfully shoulder surfed, 57.9% of them had visible lines. In the group of unexposed patterns (48.3%), 37.8% had visible lines. Individual independent t-tests for each pattern feature and *success* reveal that successfully attacked patterns are significantly shorter (M=5.7, SD=1.5) than unexposed ones (M=7.4, SD=1.4; $t_{5897} = 44.5$) and comprise significantly less *knight moves*, *overlaps* and *intersections* (all $p < .001$).

In order to determine the impact of specific feature values on observation success, we conducted a binary logistic regression analysis. The results are depicted in Table 1 (center). All tested features have a significant individual influence on shoulder surfing success (all $p < .001$). The resulting prediction model is able to correctly estimate 75.8% of the binary outcome of an observation attack ($\chi^2(5) = 20089.9$, $p < .001$, $R^2_{Nagelkerke} = 0.404$).

The odds ratio (Table 1) reveals that switching off line visibility reduces the observation risk by 67%. Furthermore, increasing the pattern length by one reduces the chance for observers by 45%. Adding a knight move reduces the risk by 32%. Additional overlaps (20%) and intersections (12%) have a smaller, but still significant relative weight. Figure 2 illustrates the relationship of the two most important factors (*length* and *line visibility*) and shoulder surfing success.

Success Rate Prediction

To allow a fine-grained prediction, we additionally specified *success* as the portion of correctly observed cells. The computed value ranges from 0 (no cell) to 100 (all cells). The average success rate for all observed patterns was 78.8% (SD=30.9). Analyzing the success rate of unexposed patterns reveals that participants were able to observe 46.4% (SD=28.1) of the input, even if the binary outcome of a guess was wrong. When lines were visible, participants observed 86.2% (SD=24.9) on average, without visualizing the input this amount dropped to 71.4% (SD=34.4).

To predict the portion of success of an attack on a given pattern, we performed a simple multiple regression analysis. The data met the assumption of independent errors (Durbin-Watson value = 2.035). Furthermore, preliminary analyses indicated no multicollinearity and the histogram as well as the P-P plot of standardized residuals indicated that errors were approximately normally distributed.

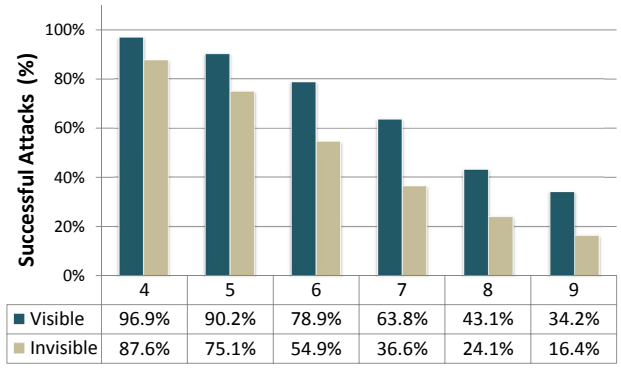

Figure 2. The number of successfully observed patterns with respect to their length and line visibility.

A highly significant regression equation was found ($R^2 = 0.263, R^2_{Adjusted} = 0.263, F_{(5,5893)} = 421.32, p < .001$). Table 1 (right) illustrates the details of the model. The standardized β-values reveal that length has the biggest impact ($\beta = .29$), followed by line visibility ($\beta = .23$), knight moves ($\beta = .12$), intersections ($\beta = .09$) and overlaps ($\beta = .07$). All features are significant individual predictors ($p < .001$).

Further analyses revealed that personal attributes like gender and experience with Android patterns did not significantly influence the shoulder surfing success.

User Perception

After each attack, users rated the ease of observation and the ease of input. The questions were based on two Likert scales ranging from 1 (“very easy”) to 5 (“very hard”).

A Spearman’s rank-order correlation indicates a strong positive correlation of ease of input and ease of observation, $r_s(5897) = 0.96, p < .001$. Overall, participants indicated medium difficulty for both tasks. Observation and input of correctly observed patterns were rated “easy” (Mdn=2), while both tasks were rated “very hard” (Mdn=5) for unexposed patterns. Both tasks become more difficult with increasing feature values. For example, patterns with the length of four were rated “very easy” (Mdn=1) to draw and “very easy” (Mdn=1) to observe while patterns with the length of six were rated “medium” (Mdn=3) and patterns with the length of nine were rated “very hard” (Mdn=5) in observation and input. The same is true for knight moves, intersection and overlaps. The average observation difficulty was rated “medium” (Mdn=3) with visible lines and “hard” (Mdn=4) when the pattern was not visualized.

DISCUSSION

Overall, 51.7% of all tested patterns were successfully attacked within one observation. Even if this means that almost half of the patterns were not exposed, it has to be noted that this does not indicate shoulder surfing resistance. The analysis revealed that users were able to partly recognize most of the patterns even if the correct pattern was not found. Therefore, we conclude that Android patterns are easy to attack, even in one-time observations.

However, the analysis of the different feature weights indicates that every feature can significantly increase shoulder surfing resistance. Pattern length and line visibility are the most important factors. Switching to invisible lines has the potential to reduce the chance of a successful observation attack by 67%, increasing the length by one cell can reduce the risk by 45%. Furthermore, every additional "special move" has a significant impact. While these results seem to indicate that long, complex and invisible patterns provide a straightforward solution to shoulder surfing, we are not advocating complex patterns. Of course, usability aspects and user behavior in the wild need do be considered.

Previous analyses showed that users often select patterns which are short and biased towards simple strokes [1, 6]. In addition, our perception analysis indicates that more complex patterns are perceived hard to enter. Therefore, we assume that "special moves" like overlaps and knight moves are hardly used in the wild. Even more critical, Harbach et al. [4] found that most users have line visibility activated. When we assume that users have visible lines and their patterns do not comprise any "special move" we find that 93% of such patterns were exposed in our experiment.

This indicates that most user-selected Android patterns are highly prone to shoulder surfing attacks. Our prediction model can be used for a new type of proactive security-level checkers which visualizes the estimated shoulder surfing risk for a given pattern. Such systems could help users to avoid high risk patterns and support them in finding patterns which are harder to trace, but still easy to enter.

LIMITATIONS

The study was thoroughly designed and the data was carefully analyzed. However, there are inherent limitations in our approach which we would like to discuss.

To be able to perform the required number of observation attacks, we randomized pattern generation and simulated user input. Therefore, some real-world factors were not considered. For example, we did not vary input speed and did not change the angle of view. In addition, we only simulated right-handed input. Consequentially, it is possible that the weight of single features differs in a real-world setting depending on the current conditions and the performance of the device owner. However, we assume that the feature relation will stay the same. In addition, our model is only applicable to one-time observations. Based on our results, we assume that none of the tested pattern features provides significant protection from multiple observations.

CONCLUSION AND FUTURE WORK

In this paper, we presented a systematic evaluation of the shoulder surfing vulnerability of (un)lock patterns. Our results indicate that line visibility and length are the most important security factors, but pattern complexity plays a significant role, too. We presented a regression model which can significantly predict the observability of a given pattern.

Since this work provides ground truth for the real world vulnerability of current user-defined Android patterns, we plan to implement a proactive pattern checking system based on our prediction model. We assume that such a system can help users to select patterns which are less prone to shoulder surfing attacks. In addition, it will help us to further analyze the interplay of pattern security and usability aspects. Finally, real-world aspects like occlusions, input speed and different angles of view have to be evaluated.

REFERENCES

1. Andriotis, P., Tryfonas, T., and Oikonomou, G. Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method. In *Col. HAS'14*, vol. 8533 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, 115–126.
2. Aviv, A. J., Gibson, K., Mossop, E., Blaze, M., and Smith, J. M. Smudge attacks on smartphone touch screens. *WOOT 10* (2010), 1–7.
3. De Luca, A., Hang, A., Brudy, F., Lindner, C., and Hussmann, H. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *Proc. CHI '12*, ACM (New York, NY, USA, 2012), 987–996.
4. Harbach, M., von Zezschwitz, E., Fichtner, A., De Luca, A., and Smith, M. It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception. In *Proc. SOUPS '14*, USENIX Association (Menlo Park, CA, July 2014), 213–230.
5. Jermyn, I., Mayer, A., Monrose, F., Reiter, M. K., and Rubin, A. D. The design and analysis of graphical passwords. In *Proc. SSYM'99*, USENIX Association (Berkeley, CA, USA, 1999), 1–1.
6. Uellenbeck, S., Dürmuth, M., Wolf, C., and Holz, T. Quantifying the security of graphical passwords: The case of android unlock patterns. In *Proc. CCS '13*, ACM (New York, NY, USA, 2013), 161–172.
7. van Eekelen, W., van den Elst, J., and Khan, V.-J. Dynamic layering graphical elements for graphical password schemes. *Creating the Difference* (2014), 65.
8. von Zezschwitz, E., Dunphy, P., and De Luca, A. Patterns in the wild: a field study of the usability of pattern and pin-based authentication on mobile devices. In *Proc. MobileHCI '13*, ACM (New York, NY, USA, 2013), 261–270.
9. Zakaria, N. H., Griffiths, D., Brostoff, S., and Yan, J. Shoulder surfing defence for recall-based graphical passwords. In *Proc. SOUPS '11*, ACM (New York, NY, USA, 2011), 6:1–6:12.