Is Bigger Better? Comparing User-Generated Passwords on 3x3 vs. 4x4 Grid Sizes for Android's Pattern Unlock

Adam J. Aviv
United States Naval Academy
aviv@usna.edu

Devon Budzitowski
United States Naval Academy
dev@comcast.net

Ravi Kuber
University of Maryland, Baltimore County
rkuber@umbc.edu

ABSTRACT

Android's graphical authentication mechanism requires users to unlock their devices by "drawing" a pattern that connects a sequence of contact points arranged in a 3x3 grid. Prior studies demonstrated that human-generated 3x3 patterns are weak (CCS'13); large portions can be trivially guessed with sufficient training. An obvious solution would be to increase the grid size to increase the complexity of chosen patterns. In this paper we ask the question: *Does increasing the grid size increase the security of human-generated patterns?* We conducted two large studies to answer this question, and our analysis shows that for both 3x3 and 4x4 patterns, there is a high incidence of repeated patterns and symmetric pairs (patterns that derive from others based on a sequence of flips and rotations), and many 4x4 patterns are expanded versions of 3x3 patterns. Leveraging this information, we developed an advanced guessing algorithm and used it to quantified the strength of the patterns using the *partial guessing entropy*. We find that guessing the first 20% ($\tilde{G}_{0.2}$) of patterns for *both* 3x3 and 4x4 can be done as efficiently as guessing a random *2-digit* PIN. While guessing larger portions of 4x4 patterns ($\tilde{G}_{0.5}$) requires 2-bits more entropy than guessing the same ratio of 3x3 patterns, it remains on the order of cracking random 3-digit PINs. Of the patterns tested, our guessing algorithm successful cracks 15% of 3x3 patterns within 20 guesses (a typical phone lockout) and 19% of 4x4 patterns within 20 guesses; however, after 50,000 guesses, we correctly guess 95.9% of 3x3 patterns but only 66.7% of 4x4 patterns. While there may be some benefit to expanding the grid size to 4x4, we argue the majority of patterns chosen by users will remain trivially guessable and insecure against broad guessing attacks.

1. INTRODUCTION

As an alternative to text-based passwords, graphical passwords [7] enable users to authenticate through a process related to image selection or sketch/gesture matching. The motivation for graphical passwords is part psychological [19, 26] — humans are bad at committing sequences of alphanumeric characters to memory and precisely recalling that information, but better at remembering and recalling graphical stimuli — and part a desire for increased complexity as compared to user choice of easily guessed text-based passwords [13, 16, 17].

ACSAC '15, December 07 - 11, 2015, Los Angeles, CA, USA

http://dx.doi.org/10.1145/2818000.2818014 .

While many graphical password systems have been proposed (see [7] for a comprehensive survey), with the advent of mobile- and touchscreen-computing, it is not until recently that graphical passwords have become widespread. In particular, Android's graphical authentication mechanism, the password pattern or pattern unlock scheme, is perhaps the most widely used graphical password system to date. This is attributed in part to the fact that the graphical password system comes standard on all Android devices, and that Android is the most widely used mobile Operating System.

Based on earlier graphical systems (e.g., Pass-Go [22]), in order to authenticate, Android users are required to "draw" a pattern that connects a sub-set of four or more contact points arranged in a 3x3 grid. If the pre-selected pattern is entered accurately, entry to the device is granted. The Android password pattern system has been studied in many contexts, including attacks on patterns [5, 6], security perceptions [4, 11], prevalence of use [25], and user choice [1, 2, 18, 21, 23, 14]. Through these analyses, it has been shown that, despite there being 389,112 possible patterns, users select patterns from a much smaller set, and that the majority of these user-selected patterns can be easily guessed with roughly the same difficulty as guessing random 3-digit PINs [23]. The addition of password meters [18, 21] and strength scores [1] can increase the complexity of human choice; however, the guessability is still higher than desired [18] thereby impacting levels of security.

One intuitive and somewhat obvious strategy to encourage users to select stronger password patterns is to increase the grid size. In custom modifications to Android, such as CyanogenMod [10], users are allowed to select from grid sizes ranging from 3x3 up to 12x12. An obvious question is then: *Does increasing the grid size increase the security of human-generated patterns?* Increasing the grid size significantly increases the total available patterns by many orders of magnitude — a 4x4 grid has 4,350,069,823,024 possible patterns — so one would expect that the complexity of human generated 4x4 patterns to be substantially greater than that of 3x3 patterns.

To address this question, we conducted two large, Institution Review Board (IRB) approved studies to collect human-generated 3x3 and 4x4 patterns. First, we designed and administered an in-lab, pen-and-paper study that follows the methodology of prior work [23] requiring users to generate 3x3 and 4x4 patterns in an adversarial situation. Participants are asked to first select patterns of their own (so called "defensive patterns"), and they are then asked to generate more patterns (so called "offensive patterns") in an attempt to guess the defensive patterns of others. Second, we developed an online survey that works in the browser where users can optionally *self-report* their own personal patterns using their own mobile devices. The online survey was administered on Amazon Mechanical Turk, and attracted 750 respondents, of which 440 self-reported their 3x3 patterns. The in-lab survey attracted 80 participants who provided 494 valid 3x3 patterns and 504 valid 4x4 patterns.

Analyzing the data sets, we find that there is a general consistency, such as in start/end points and pattern length, between the 3x3 patterns collected online and in-lab, which suggests that the in-lab study of 4x4 patterns provides a reasonable proxy for patterns which users may actually choose in the wild. Further analysis shows that across all the data sets, users are inclined to choose either the same patterns as others or a symmetric pair of other users' patterns (that is, a pattern that can be derived through some combination of flips or rotations). In total, 76.4%, 80.5%, and 40.48% of self-reported 3x3, pen-and-paper 3x3, and pen-and-paper 4x4 patterns, respectively, are either repeated or have a symmetric pair in the data set. Further, we find that 32.9% of 4x4 patterns are simple embeddings of 3x3 patterns that are mapped into the 4x4 grid space.

Using these observations, we developed, to the best of our knowledge, the most accurate pattern guessing algorithm to date and use it to estimate the *partial guessing entropy* [8] of the collected patterns which considers an attacker that wishes to crack some fraction of the patterns (also referred to as α-*guesswork* where α is the target fraction to guess). To verify the efficacy of the guesser, we reserved the self-reported 3x3 patterns as a test set, and trained the guesser using a cross-fold validation of the 3x3 and 4x4 pen-and-paper patterns.

During the training phase, we find that the guessability of 3x3 pen-and-paper patterns is similar to that in prior work [23] at $\alpha = 0.5$, but, using our guesser, cracking the first 20% ($\tilde{G}_{0.2}$) of patterns requires many fewer guesses than prior work and shows that it requires only slightly more work than guessing random *2-digit* PINs. Surprisingly, the guess work required to crack the first 20% of pen-and-paper 4x4 patterns is less than cracking the same portion of 3x3 patterns, on the order of guessing random 2-digit PINs. However, guessing the first 50% ($\tilde{G}_{0.5}$) of the 4x4 patterns is 2-bits of entropy higher than that of cracking a 3x3 pattern.

When applying the guesser to the reserved test set of self-reported patterns and training on the pen-paper 3x3 patterns, we find that the guessing rate is consistent with the cross-fold validations, and requires only slightly more work (0.5 bits) to guess the first 50% of the self-reported data. In total, after 50,000 guesses, our guesser can crack 95.9%, 97.2%, and 66.7% of the self-reported 3x3, pen-and-paper 3x3, and pen-and-paper 4x4 patterns, respectively. Interestingly, if we consider the online attack scenario where an attacker only has a limited number of attempts before a lockout, typically 20, 15% of the self reported 3x3, 16.7% of pen-and-paper 3x3, and 19.9% of the pen-and-paper 4x4 patterns would be successfully guessed.

To summarize, the contributions of this paper include:

- The first analysis of newly collected of Android password patterns using grid sizes larger than 3x3 (i.e. using a 4x4 grid);
- The first analysis of self-reported 3x3 patterns compared to in-lab studies;
- Confirming the analysis and efficacy of the data collection methodology of prior work;
- New observations on the prevalence of repetition, symmetric-pairing, and embedding of human-generated 3x3 and 4x4 patterns;

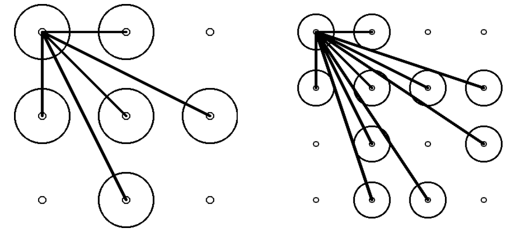


Figure 1: The available next contact points from the contact point in the upper left for 3x3 and 4x4 grids

- Advancement in guessing strategies for cracking both 3x3 and 4x4 patterns;
- Analysis and comparison of the guessability of 3x3 and 4x4 patterns.

We find that there may be some benefits to increasing the grid-size from 3x3 to 4x4. For example, increasing the grid size does not strongly affect memorability and does decrease the likelihood of naive guessing. However, the low complexity and general guessability of human-generated patterns remain. We conjecture that this will likely be true for even larger grid sizes beyond 4x4. As the grid space becomes more dense with more contact points, the ease of entry for more complex, less guessable, patterns is reduced due to the increased likelihood of accidentally interacting with the contact points. As such, we should expect that the guessability rates for human-generated patterns on larger grid sizes will suffer from the same problems as 4x4 patterns, and have similar, easier than desired, guessability rates.

2. RELATED WORK AND BACKGROUND

As a response to the tradeoff between security and memorability of alphanumeric paswords and PINs, the community has developed a wide variety of graphical password schemes (described in more detail by Biddle et. al [7]) in an attempt to provide increased complexity and increased memorability from graphical stimuli. The Android graphical password pattern is an example of such a scheme and is based on draw-metric graphical passwords system and is related to systems like Draw-A-Secret [12], PassPoints [27], PassGo [22] and PassShapes [26], to name a few.

The Android graphical password pattern, as a variant of these draw-metric schemes, presents the user with a grid of 3x3 contact points on which the password pattern is "drawn." If the pattern is successfully recreated, entry to the device is granted. The drawing of a pattern is constrained such that (1) a pattern must contain at least 4 contact points, (2) a contact point may only be used once, (3) a pattern must be entered without lifting, and (4) a user may not avoid a previously un-selected contact point. The same rules apply for both the 3x3 and 4x4 grid spaces. In Figure 1, an example of the allowable strokes originating from the upper left corner are shown for both grid spaces. Note that once a contact point is selected, a user may trace over it to get to another contact point, but the point is only considered selected once.

It is well documented through brute force enumeration [5] that there are 389,112 possible 3x3 patterns. We have performed a similar calculation for 4x4 patterns (requiring a number of optimization techniques) to determine that there are 4,350,069,823,024 possible 4x4 patterns, which is approximately 2^{42} possible patterns as compared to approximately 2^{18} 3x3 patterns. As another point of reference, the number of possible 4x4 patterns is similar in magnitude as randomly selected 7-character password that could use all upper-case characters (26), lower-case characters (26), numerics (10), special symbols (32), and spaces ($95^6 \sim 2^{46}$), while the number of possible 3x3 patterns is more similar in magnitude of a 3-character password ($95^4 \sim 2^{19}$).

The Android pattern has been well studied by researchers. Foremost, von Zezchwitz et. al conducted a study of locking strategies on Android devices [25] and found that PINs outperformed pattern lock when comparing input speed and error rates despite the fact that patterns remain very popular among Android users. The number of patterns available to users on grids has been calculated on numerous occasions [5, 23, 14]. Attacks on the patterns have also been proposed. These include, for example, smudge attacks [5] which use the oily-residues of prior entries to determine the pattern as well as attacks that leverage the on-board motion sensors to determine pattern input [6]. The ease of shoulder surfing attacks on pattern entry has also been studied [24]. Furthermore, studies have been undertaken examining unlocking behavior [11] and perceptions of security [4]. Findings suggest that participants considered secure lock screens (e.g. Android unlock patterns) unnecessary in 24.1% of situations [11].

The earliest analysis of user-generated patterns was conducted by Andriotis et. al [2]. Users were asked to provide one "easy" and one "hard" pattern, in an in-lab setting. Follow-up work has been conducted by Andriotis et. al [1], Song et. al [18], and Sun et. al [21]. The researchers have all collected user-generated patterns for the purpose of developing complexity measures and/or password meters to increase security of user choices. While these schemes do generally increase the complexity of user choice (by requiring the user to select patterns with longer strokes or using more contact points), the human-selected patterns are still relatively easily guessed as compared to random 4-digit PINs [21].

This paper is most similar to is a study by Uellenbeck et. al [23]. These researchers collected and analyzed the partial guessing entropy [8] of a large corpus of user generated Android patterns. We adopt much of the methodology of that study here. In particular, we employ the same metrics for guessability, and we also adopt (with minor modification) the primary data collection methodology where users are asked to select patterns of their own and guess the patterns that other users selected. Our pattern guesser is also modeled after the guesser described by Uellenbeck et. al with added focus to the likelihood measures (i.e., the Markov model construction) and pattern repetitions/symmetries. We confirm the primary findings of their study that in many situations the guessability of Android patterns is as challenging as guessing random 3-digit PINs. We advance upon these findings by showing that in many situations, and for 4x4 patterns, that the guessing challenge is more similar to guessing random 2-digit PINs.

Finally, to the best of our knowledge, this is the first work to study human-generated patterns for grid sizes beyond 3x3. While Uellenbeck et. al. [23] did consider different layouts of the contact point, e.g., in a circle, with the top left point removed, and randomly aligned, the number of contacts were always equal to or less than nine. In the research described in this paper, we consider the 4x4 grid space, with a total of 16 contact points.

3. METHODOLOGY

In this section, we describe the data collection methodology. Note that we use two primary data collection methods: an in-lab/pen-and-paper study (termed *pen-and-paper*) and an online/self-reporting study (termed *self-report*). For the pen-and-paper study, we model the methodology of Uellenbeck et. al [23], and we use this method to collect both 3x3 and 4x4 patterns. For the self-report study, we developed a survey that functions in the browser and administered it on Amazon Mechanical Turk. The survey is designed to model the pattern entry of Android within the browser so that participants can enter patterns on their own mobile device without having to install any specialized applications. We use the self-reported data for two purposes: (1) to compare against the pen-and-paper data to establish the efficacy of the pen-and-paper methodology; and (2) as a reserved test set to measure the performance of the guessing algorithm.

All the protocols used herein were reviewed and approved by our IRB, meet appropriate ethical standards, and incur minimal risk to the participants. The limitations of our methodology are discussed following their description.

3.1 Study 1: Pen-and-Paper

To encourage participants to generate realistic patterns during the study, we employed the *adversarial methodology* described by Uellenbeck et. al [23]. This method encourages users to generate patterns which they believe others will use (and thus would probably use themselves). The crux of the adversarial method requires users to first select patterns as their own (so called *defensive patterns*) and are rewarded for generating additional patterns that others selected (so called *offensive patterns*). Our protocol differs from Uellenbeck et. al's, as we ask participants to generate 3 defensive patterns and 10 offensive patterns rather than one in each category. We also ask participants to attempt to recall their defensive patterns at the end of the survey to gauge the memorability of 4x4 patterns compared to 3x3 patterns.

Conducted using pen-and-paper, participants are asked to draw patterns using marker/pen on grids printed on paper handouts. While drawing their pattern, participants follow the same rules as they would for drawing patterns on a mobile device by placing the marker/pen at the starting contact point and drawing the pattern without lifting. To differentiate the start point of the pattern from the end point, participants are asked to circle the start point on the grid. The specifics of the protocol and recruitment are described below.

Protocol. Participants are divided into focus groups where, within each group, a group leader (a researcher) would direct participants through the procedures. Rewards are provided to the participants, in the form of edible treats (e.g., chocolate), for the ability to guess others' patterns and recall their own patterns. The fact that participants would be asked to recall their own passwords, is withheld until the end of the focus group; however, participants are informed up front about rewards for guessing others' patterns. The protocol proceeds in five phases for both 3x3 and 4x4 pattern study groups: *instructions*, *selection*, *guessing*, *survey*, and *recall*.

The purpose of the instructional phase is to inform the participants of the procedures of the study. For consistency, group leaders are provided with an oral script and participants received handouts with instructions. The handouts differ between groups in regard to the grid sizes only (i.e., 3x3 vs 4x4), all other instructions remained the same. The handout also contained information about what makes a pattern valid and what makes a pattern invalid, which is also described aloud to participants.

During the selection phase, participants are instructed to select *three* patterns to be their own defensive patterns. There are explicit oral and written instructions (in bold and colored red) regarding the criteria which selected patterns should follow:

> Choose passwords that are easy for you to recall but hard for others to guess.

Once the three passwords are selected, the paper-sheet on which they are drawn is collected and reserved.

Next, in the guessing phase, participants are asked to make *ten* guesses (offensive patterns) of what they think other participants may have selected as their patterns. Participants may guess their own pattern, but rewards are only provided if the participants correctly guesses other participants' patterns. Again, once completed, the paper-sheets are collected with all the offensive patterns.

The survey phase that follows serves two purposes. First, it is used to collect standard demographic information about the participants, such as age and gender, but it also used to keep the participants occupied while the group leaders matches the guesses (offensive patterns) to the selected patterns (defensive patterns). Once the matching and survey is completed, the results are revealed and rewards distributed.

Finally, the group leader reveals to the participants that they can earn additional rewards if they are able to recall their own initial three patterns (occurring approximately 15-20 minutes after initial selection). A final handout is provided to the participants on which they attempt to draw their originally selected patterns. A reward is provided for each properly and accurately recalled pattern. Once the survey is complete, the group leader allows the participants to view all the patterns selected and converse with each other. During the survey, conversing was not allowed.

Recruitment and Collection. We conducted the study using 10 focus groups during a period of 6 weeks. In total 80 individuals participated, and the group sizes varied between 8 and 20. Two-thirds (48) of the participants were male and one-third (24) were female. Recruitment was conducted at the institutions of the authors, and the ages of the participants varied between 18 and 40 years. We collected 494 3x3 patterns (380 offensive and 114 defensive) and 504 4x4 patterns (385 offensive and 119 defensive). There were a small number of patterns that were rejected from analysis as they did not follow the pattern generation rule (see Section 2), and some participants failed to provide all three defensive or all ten offensive patterns.

3.2 Study 2: Self-Reported Patterns

The self-reporting study is used to augment the data collected during the pen-and-paper survey. Prior work examining text-based passwords has shown that self-reported statistics model actual user behavior [20, 28]. As such, we designed an on-line/in-browser survey written in HTML5 and Javascript that is able to mimic the pattern entry system for Android. The survey was administered on Amazon Mechanical Turk. Participants were compensated \$0.75 for completing the study. To encourage participants to provide their real patterns, we allowed participants to opt-out of reporting their pattern during the survey and instead report statistics of their pattern (such as start point and common tri-grams in their pattern). In total, 750 individuals participated, of those, 440 self reported their 3x3 patterns. Of the 440, 251 were male and 189 were female, ranging in age between from 18 to 55+.

To ensure honesty among the participants, we added two attention tests. First, participants must enter their pattern (or statistics about their pattern) twice, once at the start of the survey and once at the end. If these varied, the participant is excluded (but not rejected in Amazon Mechanical Turk). Second, we required participants to answer a truth-based question at the end of the survey along the lines of asking "Did you provide honest answers to this survey?" If the participant failed to check "yes" then they are excluded (but not rejected). Finally, we also incorporated a Captcha into the survey to ensure the process could not be easily automated.

3.3 Limitations

The foremost limitation of the methodology is the use of pen-and-paper to collect patterns. Clearly, patterns are not typically entered using pen-and-paper but instead on mobile devices using touchscreens. We compensate for this limitation by using the self-reported data set to verify the properties of the pen-and-paper data. As we will show, the basic statistics of the two data sets are very similar, and we argue that pen-and-paper model offers a reasonable substitute for real, in-the-wild data, especially for 4x4 patterns which are not commonly used.

There are also reasonable limitations regarding the veracity of the self-reported data set. We argue that this data probably better represents patterns as they are actually seen in the wild than other reported data sets [2, 21, 23], and there are a number of factors to suggest this.

First, participants provided patterns on their own mobile devices which are likely locked with the pattern provided. We argue that there is likely a reflex in this setting to just enter the real pattern much like there is a reflex to type your password in a password box if one comes up on the screen [9]. Second, there were very low rates of failure among the attention tests — two participants were excluded for not-matching their patterns and three were excluded for not indicating that they told the truth — suggesting that participants likely provided attentive and honest responses. Third, participants could optionally *not* enter their password, so those that did were more likely to reveal their real password. This, however, may suggest that the revealed passwords may be simpler/easier/less-secure than those that were not revealed. An analysis of the self-reported statistics from the self-reported patterns are consistent with respect to length, start/end point, and common sub-sections, which suggests that participants provided honest answers.

Another limitation is that the pen-and-paper 4x4 patterns cannot be cross-referenced to another study, such as a self-reporting survey like the 3x3 patterns were, due to the simple fact that 4x4 patterns are not typically used except on specially modified Android devices (e.g., CyanogenMod). As such, we argue that just as the data from the pen-and-paper 3x3 patterns is a good proximate for real 3x3 patterns it is likely that the pen-and-paper 4x4 patterns are a good proximate for how users would choose real 4x4 patterns if they were used widely.

Finally, there is further limitations in the pen-and-paper data collection methodology with regarding to participant priming and entry exhaustion. As for potential issues with priming, indicating to participants that they must first choose patterns that others will guess may prime them towards choosing harder passwords than they would realistically use; however, the fact that participants then in turn assume others choose weaker passwords than themselves, as evidenced in the results (see Section 5), provides some balance with respect to password strength in the data when looking at a combination of both defensive and offensive patterns.

Regarding entry exhaustion, a potential limitation in the data may occur when participants are asked to generate many patterns in one sitting (ten in a row for offensive patterns, and thirteen total considering the defensive as well) may have led participants to over simplify (or over complicate) their guesses, which would lead to weaker patterns overall and for offensive patterns in particular. The results suggest that this impact was small. When comparing the guessability of offensive patterns to those self-reported, where a participant provided just a single pattern, we find that self-reported patterns are actually more easily guessed as compared to the offensive patterns where ten patterns must be provided in one burst. This suggests that overall, participants from pen-and-paper surveys likely provide realistic patterns with respect to guessability features as would be used in the wild.

4. DATA CHARACTERIZATION

Before proceeding to the guessability measures, we first wish to provide some characterization of the data that will inform our

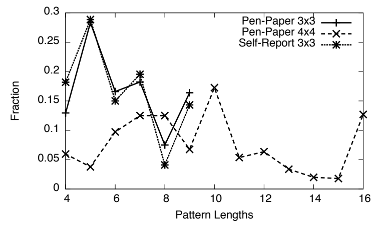


Figure 2: The distribution of length in the data sets.

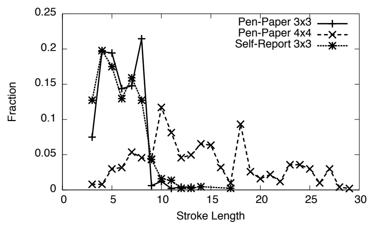


Figure 3: The distribution of stroke-lengths in the data set

	Length	Norm. Length	Stroke Length	Norm. Stroke Length
Self-Report 3x3	6.0 [5:7]	0.7 [0.6:0.8]	5.8 [4.0:7.0]	2.9 [2:3.5]
Pen-Paper 3x3 (All)	6.3 [5:7]	0.7 [0.6:0.8]	5.9 [4.1:7.4]	2.9 [2.2:3.7]
Pen-Paper 3x3 (Off.)	6.3 [5:8]	0.7 [0.6:0.9]	5.9 [4.3:7.7]	3.0 [2.2:3.8]
Pen-Paper 3x3 (Def.)	6.0 [5:7]	0.7 [0.6:0.8]	6.0 [4.8:7.0]	3.0 [2.4:3.5]
Pen-Paper 4x4 (All)	9.6 [7:12]	0.6 [0.4:0.8]	9.5 [6.8:11.4]	3.2 [2.3:3.8]
Pen-Paper 4x4 (Off.)	9.8 [7:12]	0.6 [0.4:0.8]	9.6 [7.0:11.5]	3.2 [2.3:3.8]
Pen-Paper 4x4 (Def.)	8.8 [6:11]	0.6 [0.4:0.7]	9.0 [6.0:11.0]	3.0 [2.0:3.7]

Table 1: Statistics of the length measures (mean $[q_1{:}q_3]$): Norm. Length calculated by dividing by total available points, Norm. Stroke Length calculated by mapping the 3x3 and 4x4 grid on a Cartesian plane 1.0x1.0

guessing algorithm. Foremost, we look at basic statistics of the patterns, such as their length, start/end points, common sub-sections, and the most common patterns. We will also present analysis of the frequency of patterns repetitions and symmetries, and the embedding of 3x3 patterns into 4x4 patterns. Finally, we provide analysis of the memorability and human-powered guessability of patterns, such as how common it was for 3x3 and 4x4 patterns to be both recalled and compromised by other participants.

4.1 Basic Features

Length. We first consider the most basic feature of the patterns, the length of the pattern. Figure 2 presents the results of the length analysis — the measure of the total number of contact points used — and for the pen-and-paper 3x3 patterns and self-report 3x3 patterns collected, the distributions of pattern lengths are very similar. As 4x4 patterns have more contact points, the lengths are longer overall.

Additionally, we were interested in the stroke-length (or distance [3]) which has been shown to correlate with perception of security [4]. The stroke length is calculated using the Cartesian distance of each line segment of the pattern where the contact points are labeled with (x, y) values. For patterns drawn on 3x3 and 4x4 grids, we labeled the contact point in the upper left as $(0, 0)$ and the one in the lower right $(2, 2)$ or $(3, 3)$ depending on the grid size. Calculating the stroke distance of the patterns resulted in the distribution presented in Figure 3. Again, for pen-and-paper 3x3 and self-reported 3x3 patterns, the distributions are very similar, and for 4x4 patterns, with more points, we expect the stroke length to be longer, covering the entirety of the grid.

To draw a better comparison between 3x3 and 4x4 patterns, we normalized the two length measures. To normalize the length, we divided the number of used contact points by the total available contact points. To normalize the stroke-length, we mapped the 3x3 and 4x4 grids into a Cartesian space of size 1x1, where the upper left contact remained $(0, 0)$ but the lower right contact point was $(1, 1)$ for both 3x3 and 4x4 patterns. These results are presented in Table 1; interestingly, when normalized, the length of 4x4 patterns are shorter overall and do not have substantially longer stroke lengths. This furthers the argument that the selected 4x4 patterns are very similar to embedded 3x3 patterns in the 4x4 space, that is 4x4 patterns have the same shape/structure as 3x3 patterns.

Start and End Conditions. Results relating to start and end points are presented in Figure 4. Unsurprisingly, in all the data sets, the most common start point is the contact point in the upper left corner, which has been reported in prior studies [1, 2, 14, 18, 21, 23]. We identified that this trend also continues for 4x4 patterns, and perhaps is even more prevalent (37.5%) considering the increase in the total number of start points. Again, as reported in prior studies, patterns typically end in the bottom right contact point, and this continues to be true for all the data sets.

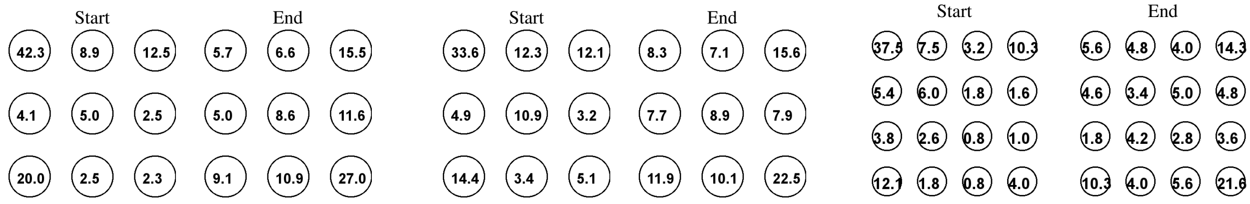
4.2 Pattern Repetitions and Symmetries

Common Sub-Sequences. We wish to look at common sub-sequences of patterns, namely tri-grams as these have been shown to perform best when generating likely patterns for guessing [23]. Figure 5 displays the most common tri-grams (sequences of 3 or more connected points) for all the data sets, and additionally the top quad-grams for pen-paper 4x4 patterns. As shown, the most common grams appear frequently, nearly twice as many as the 12-th most common. Another interesting property of the common sub-sequences is the prevalence of sub-sequences along the exterior of the grid space, a property that persists in the 4x4 data. We will use the distributions of tri-gram as a likelihood measure for patterns, both 3x3 and 4x4, when developing the Markov model for the guesser.

Repetitions. An important observation leveraged in our guesser is that the prevalence of repeated patterns in the data set is quite high. Figure 6 shows the top 5 most frequently occurring patterns in each of the data sets. Note that the most frequent selected pattern appears a lot; roughly, 3% of the data set is the most frequently occurring pattern. Figure 7 shows the distribution of repeated patterns as a cumulative fraction graph. All the data sets have a high occurrence of patterns that repeat, and, more strikingly, roughly 20% of the patterns in all data sets repeat at least 4 times. Table 2 provides summary statistics of this feature and those described below.

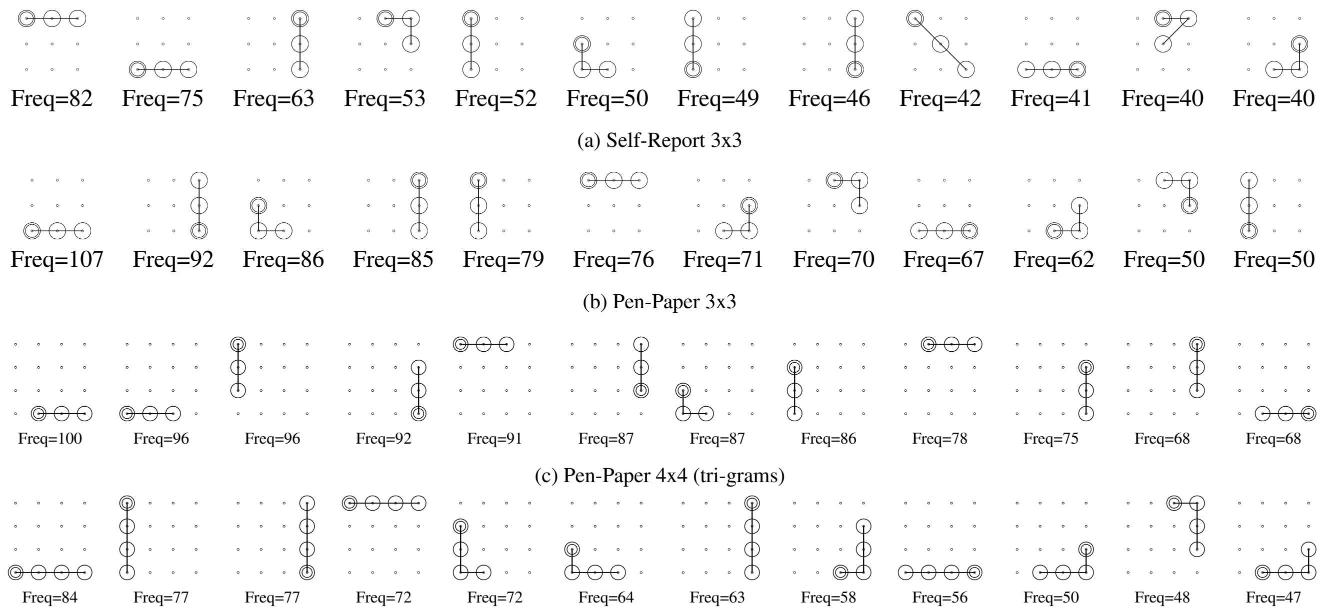
Symmetries. Further analysis of the data shows that while a large portion of each of the data sets is repetitive, there also exist many symmetric pairs. We define a symmetric pair as two patterns that can be transformed into the other through some sequence of flips, rotations, or reversals. A repeated pair, then, is also a symmetric pair, and when including all symmetric pairs, we find that 80% of the pen-and-paper 3x3 patterns and 76% of the self-reported 3x3 patterns have symmetric pairs. Only 40% of the 4x4 pen-and-paper patterns have symmetric pairs, and, while still a large fraction, there are other properties of 4x4 patterns that can be leveraged in the guesser. Figure 8 shows the distribution of symmetric pairs in the data set as a cumulative fraction graph.

Embeds. As can be observed in Figure 6 for the most frequent patterns, some of the 4x4 patterns are just enlarged 3x3 patterns, such as the 'Z' or 'L' shaped patterns. We describe this as a *embedding* of a 3x3 pattern into a 4x4 grid space. There exist 16 possible embedding for every 3x3 pattern corresponding to the 16



(a) Self-Report 3x3 (b) Pen-Paper 3x3 (c) Pen-Paper 4x4

Figure 4: Frequency of Pattern Start and End Points (in percent)



(d) Pen-Paper 4x4 (quad-grams)

Figure 5: Top 12 Occuring Tri-grams for 3x3/4x4 and Quad-grams for 4x4

ways of removing one row and one column from a 4x4 grid to form a 3x3 sub-grid. For each sub-grid, we consider a mapping of the 3x3 pattern adding intermediate points as necessary to form a valid 4x4 pattern. Only unique embeddings are considered. We analyzed the pen-and-paper 3x3 data embedded into the 4x4 grid space and found that nearly a third (32.9%) of 4x4 patterns are just mappings of 3x3 patterns. We apply this fact in our guessing algorithm to train the guesser on likely 4x4 patterns.

4.3 Memorability and Naive Compromises

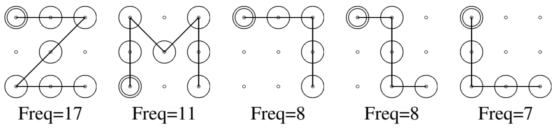
We have an opportunity to measure the memorability and "human-powered guessability" (so called naive compromises) of 3x3 and 4x4 patterns collected during the pen-and-paper survey. Recall from the methodology that participants were asked to select three patterns of their own (defensive patterns) and also guess ten patterns of others (offensive patterns). Additionally, participants were asked to recall their three defensive patterns at the end of the survey, roughly 15-20 minutes past when those patterns were selected.

Naive Compromises. While rewards were only provided to participants who guess patterns within their study group, we can look across study groups and get a sense of the naive, human-powered guessability of patterns. For 3x3 patterns, 39/114 (or 34%) of the defensive patterns appear as offensive patterns. For 4x4 patterns, 16/119 (or 13%) of the defensive patterns appear as offensive patterns. Note these numbers are slightly inflated due to repetition of patterns (a single guess can compromise multiple patterns). Still, the rate of compromise for 3x3 patterns is more than twice as high as that of 4x4, which suggests for human-powered, naive guessing of patterns, 4x4 patterns are likely more secure against a typical human adversary.

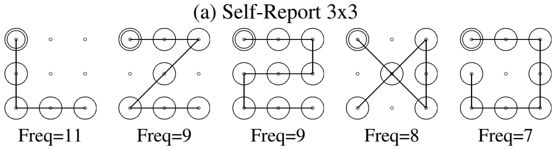
	Size	Repetitions	Symmetries	Embedding
Self-Report 3x3	440	203 (46.1%)	336 (76.36%)	n/a
Pen-Paper 3x3 (All)	491	245 (49.9%)	398 (81.1%)	n/a
Pen-Paper 3x3 (Off.)	378	187 (48.3%)	309 (79.8%)	n/a
Pen-Paper 3x3 (Def.)	113	16 (14%)	54 (47%)	n/a
Pen-Paper 4x4 (All)	501	179 (35.7%)	204 (40.7%)	166 (33.1%)
Pen-Paper 4x4 (Off.)	382	156 (40.8%)	177 (46.3%)	142 (37.1%)
Pen-Paper 4x4 (Def.)	119	10 (8.4%)	10 (8.4%)	24 (20.1%)

Table 2: The Fraction of Repetitions, Symmetries, and Embedding of 3x3 patterns in 4x4 patterns

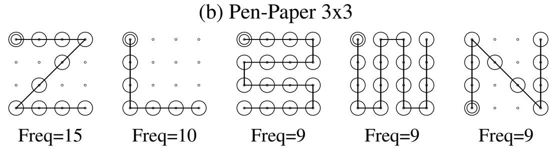
Recall Rates. The memorability of the patterns is estimated in the recall rate of the defensive patterns at the end of the study. Both 3x3 and 4x4 patterns had similar recall rates: 54/114 (47%) of the defensive 3x3 patterns and 50/120 (42%) defensive 4x4 patterns could be recalled at the end of the survey. Combined with the lower naive compromise rate of 4x4 patterns, this would suggest that there would be some benefit to increasing the grid size: low impact on memorability and an increased resilience to naive guessing. As we will show in the next section, however, the true guessability of 4x4 patterns compared to 3x3 patterns is actually much closer when using advanced techniques.



(a) Self-Report 3x3



(b) Pen-Paper 3x3



(c) Pen-Paper 4x4

Figure 6: Top 5 Most Frequently Occurring Patterns

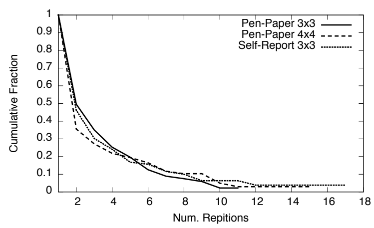


Figure 7: Cumulative fraction of patterns that repeat

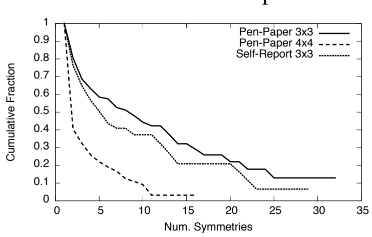


Figure 8: Cumulative fraction of patterns that have symmetries

5. PATTERN GUESSABILITY

Our primary mechanism for quantifying the relative strength of 3x3 and 4x4 patterns is to measure the guessability by constructing a guessing routine and measuring the rate of correctly guessed patterns. We model our guessing algorithm on prior work [23] with some advances in the likelihood measures and pattern generation techniques. We train the guesser to the pen-and-paper 3x3 and 4x4 patterns using a cross-fold validation and tune the performance based on the partial guessing entropy [8], which measures ability of the attacker to guess some fraction of the patterns. This is a common technique employed in prior work [18, 23] and related work on password strength [13, 16]. We verify the efficacy of the guesser by finally applying it to the reserved test set (self-reported 3x3 patterns) with training from the pen-and-paper 3x3 patterns.

In the rest of this section, we first describe the likelihood measures and the guessing algorithm, followed by a description of the partial guessing entropy metric. The section concludes with a presentation of the results.

5.1 Guessing Algorithm

We based our guessing routine on the one described by Uellenbeck et. al [23] with limited modifications. The primary difference between our and Uellenbeck et. al's routine, is that we leverage more of the properties of human-generated patterns during the initial guessing routine. As noted in the prior section, we observe that human-generated patterns have high repetition rates as well as have many symmetries, and we wish to guess those patterns earlier and ranked by some likelihood score. Our likelihood measure is also based on a tri-gram Markov model; however, we have advanced this technique to better draw from the start and end conditions of the patterns as well as the distribution of pattern lengths. Similar to Uellenbeck et. al, we find that tri-grams work best for likelihood measures.

Markov Model Likelihood Estimates. To compute the likelihood of a given pattern, we employ a standard Markov model with probabilities estimated from the training set. Pattern transitions, i.e., the connection of two contact points to form a segment in the pattern, is modeled based on the likelihood of a transition between two tri-grams. For example, the probability (0,1,2) transitions to (1,2,3) is estimated based on occurrences of that transition in the training set, but the probability (0,1,2) transitioning to (1,5,3) would always be 0 because it is an impossible transition. Of course, not all valid tri-grams transitions will occur in the training set, so we apply the standard Laplance smoothing (or constant smoothing with $k = 1$). As noted in prior work [23], the smoothing technique has little to no effect on the results.

Additionally, we wish to take advantage of the start and end points, e.g. that patterns begin in the upper left and end in the lower right, beyond just the most common initial tri-grams (differing from [23]). To do that, we consider additional start and end states in the transition matrix where (-1,-1,0) is the tri-gram for a pattern starting at the 0th contact point, and (8,-1,-1) is the tri-gram ending in the 8th contact point.

Finally, we want the likelihood measure to also account for the likely length of the patterns. While Markov models are sufficient for measuring transition likelihoods, they do not model the expected length. As such, we also multiply the probability of a transition sequence by the probability a pattern is that length, again, estimated from the training set.

Formally, we define a pattern x of length n as the sequence

$$x = \{x_{-2}, x_{-1}, x_0, \ldots, x_{n-1}, x_n, x_{n+1}\}$$

where, for $i < 0$ and $i \geq n$, $x_i = -1$ which represents a start/end state so that we can properly capture the beginning and end probabilities of a pattern. The probability of a given pattern $P(x)$ is then defined as

$$P(x) = P(l(x) = n) \cdot \prod_{i=0}^{n+1} P(x_i | x_{i-1}, x_{i-2}) \tag{1}$$

where $l(x)$ is the length of the pattern not considering start and end nodes. The formula encapsulates both the start and end state probability, as well as interior transitions, and the probability a pattern is a given length. We use this likelihood estimate to rank patterns during guessing, as well as generating likely patterns that have not been seen previously in the training set.

Generating Likely Patterns. Additionally, it is necessary to use the Markov model to generate patterns that were not encountered in the training set. We do this by sampling from the conditional (or transition) probability distributions associated with $P(x_i|x_{i-1}, x_{i-2})$ while also considering the possibility of a transition to an end state. To do so, we need to calculate the probability of the next contact point $P(x_i)$ using the following equation:

$$P(x_i) = \begin{cases} P(x_i|x_{i-1}, x_{i-2}) \cdot P(l(x) \geq i) & \text{if} \quad x_i \geq 0 \\ P(x_i|x_{i-1}, x_{i-2}) \cdot P(l(x) = i - 1) & \text{if} \quad x_i < 0 \end{cases}$$

The probability $P(x_i|x_{i-1}, x_{i-2})$ can be estimated from the training set using the same transition information obtained from the likelihood measures. The probability $P(l(x) > i)$ is defined as $1 - \sum_{l=0}^{s^2} P(l(x) = l)$ where s is the grid dimension (i.e, either 3 or 4 for 3x3 or 4x4 grids).

Another way to describe the use of the length probabilities in this formula is that when generating a pattern, you must consider the impact of a transition to a non-end state ($x_i \geq 0$) and a transition to an end-state ($x_i < 0$) with respect to the distribution of pattern lengths. In the case where the transition is to a non-end state, we must consider the probability that the pattern is longer than its current length ($P(l(x) \geq i)$). Conversely, if the transition is an end state, then we must consider the probability that the pattern is this length ($P(l(x) = i - i)$. Recall, that $l(x)$ considers the length of the pattern without the start and end states. With a transition to an end state, $l(x)$ would equal $i - 1$ as $x_i < 0$.

The pattern generation algorithm proceeds by first selecting x_0 by sampling from the distribution $P(x_0)$ and continues to sample for each next contact point using $P(x_i)$ until an end state is selected. We further limited the generation routine to consider only valid transitions when calculating the conditional probabilities; for example, when considering transition to a contact point x_i that was previously selected but outside the scope of the conditional of x_{i-1} and x_{i-2}, the probability of that transition should be zero.

Algorithm Description. The goal of the guessing algorithm is to not only guess as many patterns as possible, but to also do so as fast as possible. The order of the guesses directly affects the partial guessing entropy. To ensure the best possible order of guesses, the algorithm will train a Markov model from the training input and generate a sequence of ranked guesses based on the likelihood measure with added preferences for patterns appearing in the training set and symmetries/reversals of patterns from the training set.

The Markov model must first be trained to compute the likelihood measures. The training of the model is based on using all the training data to estimate the transition probabilities; additionally, we include all unique symmetries and reversals not found in the training data as part of the probability estimates for the Markov model. The weighting of training input to unique symmetries is two to one, that is, a pattern appearing in the input training data is treated as occurring twice for each instance (including repetitions) and the unique symmetries are considered occurring only once. We do this to capture transitions that occur in symmetries, as symmetries are highly prevalent for human-generated patterns and should be accounted for. Finally, Laplance smoothing is used to ensure no zero probability transitions exist for valid transitions that do not appear in either the training or the unique symmetries of the training.

Once the training of the Markov model is complete, the first stage of guessing commences. Motivated by the high occurrence of repeated patterns in the data, it makes sense for the guessing algorithm to first guess *all* unique patterns provided in the training data, ranked in order of the number of repetitions in the training data with ties broken by the likelihood measure from the Markov model. The next stage of guessing attempts to leverage the high rate of symmetries in the data. As such, we next guess all unique symmetries of patterns that appear in the training set ranked in order of the likelihood measure.

For a training set of 400 samples (as is the case for a five-fold cross validation with 500 items), the initial guesses constitute the first $\sim$ 1800 guesses for 3x3 patterns and the first $\sim$ 2900 guesses 4x4 patterns. The remaining guesses are generated by sampling from the Markov model as described in the previous section. To ensure that more likely patterns are guessed first from the generated patterns — the sampling routine does not guarantee that the most likely patterns are generated in order — we first generate enough patterns to ensure that sufficiently likely patterns are considered. All the generated patterns are then sorted based on the likelihood measure. In our experiments, we generated enough patterns to make 50,000 total guesses. Written in Python, the guessing routine takes about two minutes to generate 50,000 guesses for 4x4 patterns.

5.2 Partial Guessing Entropy

To measure the performance of the guesser with respect to the guessability of the data set, we use the *partial guessing entropy* [8] which considers an attacker who is satisfied with guessing some fraction of the passwords in the set. In this scenario, we are attempting to quantify the strength of patterns people choose by considering an adversary that can perform guessing across all individuals' devices. We are concerned with how long it takes, as measured in number of guesses, to correctly guess some fraction of the users devices where there does not exist any lockouts (e.g., after 20 guesses). The scenario of guessing without lockouts does not exactly model the reality of guessing users' unlock pattern in online manner on physical devices; however, we can observe the fraction of patterns guessed within 20 attempts, the lockout limit. More so, allowing guesses beyond the first 20, enables us to quantify the strength of patterns more generally using partial guessing entropy and compare the strength of user generated 3x3 and 4x4 patterns as well as to other reported results.

Partial guessing entropy is formalized by first letting α be the fraction of passwords the attacker wishes to guess, then μ_α is the minimum number of guesses required to guess $N \cdot \alpha$ of the passwords where N is the number of passwords in the corpus. Let,

$$\mu_\alpha = \min \left\{ i_0 \;\middle|\; \sum_{i=1}^{i_0} p_i \geq \alpha \right\}$$

where p_i is the probability of guessing the i^{th} password. Let $\lambda_{\mu_\alpha} = \sum_{i=1}^{\mu_\alpha} p_i$ be the fraction of passwords cracked after μ_α guesses. The actually number of passwords cracks may be greater than $\alpha \cdot N$ because passwords may repeat in the data set, e.g., two users have the same password. The partial guessing entropy is then defined

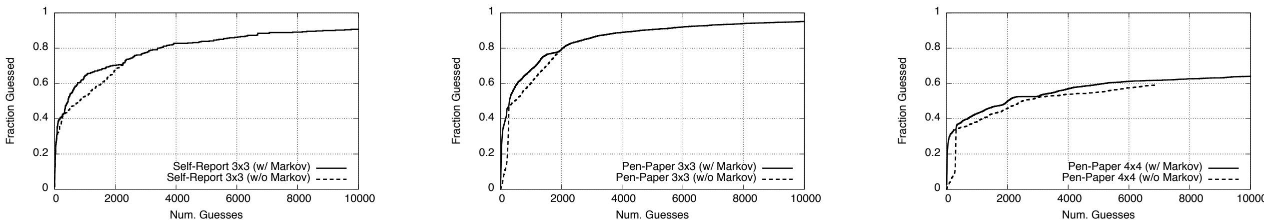
$$G_\alpha(X) = (1 - \lambda_{\mu_\alpha}) \cdot \mu_\alpha + \sum_{i=1}^{\mu_\alpha} i \cdot p_i$$

This can be expressed as bits of entropy using the following formulation:

$$\tilde{G}_\alpha = \lg\left(\frac{2 \cdot G_\alpha(X)}{\lambda_{\mu_\alpha}} - 1\right) + \lg \frac{1}{2 - \lambda_{\mu_\alpha}} \tag{2}$$

As described in [8], the term $\lg \frac{1}{2-\lambda_{\mu_\alpha}}$ is provided to ensure that the metric is constant for uniform distributions as would be the case if the guessing routine is using no information and guessing randomly. The partial guessing entropy (in bits) for a uniform distribution is simply $\tilde{G}_\alpha = \lg(N)$.

In the context of measuring the partial guessing entropy for the data sets we collected, we vary from this approach in the same two ways as in [23]. First, the size of the sample set required to properly estimate X to compute p_i is beyond what we are able to collect. The second reason is that we are concerned with the metrics associated with a specific attack methodology that attempts to model the optimal order for guessing patterns. To compensate, like prior work [23], we define p_i as the fraction of passwords cracked with the i^{th} guess of running the guessing algorithm.

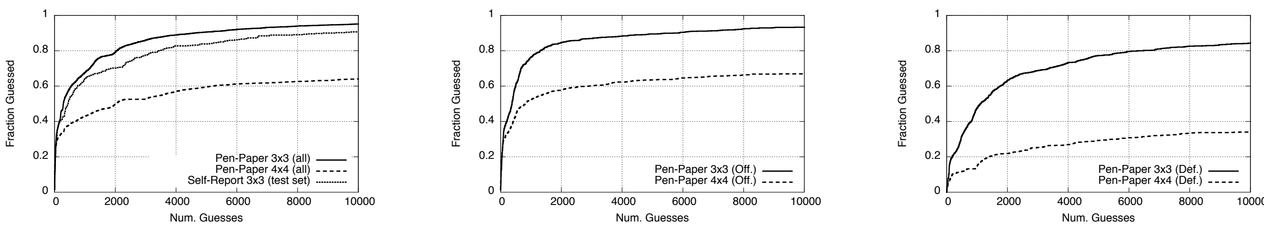


(a) Self-Report 3x3 Patterns

(b) Pen-and-Paper 3x3 Patterns

(c) Pen-and-Paper 4x4 Patterns

Figure 9: Guessing Entropy Estimates *with* and *without* Markov Likelihood Rankings



(a) *All* patterns using the average of 10 runs of a 5-fold cross-validation with 500 randomly selected patterns and self-reported 3x3 patterns

(b) *Offensive* patterns using the average of 10 runs of a 5-fold cross-validation with 100 randomly selected patterns

(c) *Defensive* patterns using the average of 10 runs of a 5-fold cross-validation with 100 patterns

Figure 10: Guessing Entropy Estimates

5.3 Guessability Results

Training the Guesser. The first task is to tune the guesser using the training data found in the pen-and-paper 3x3 and 4x4 data. To do that, we conducted 10 runs of a randomized five-fold cross validation using 500 samples per run. Note that there are only 494 total samples for pen-and-paper 3x3 patterns, so one fold was slightly smaller; however, since we are concerned with the average across 10 runs with 5 validations per run, this effect is negligible. The guesser was set to make 50,000 guessing attempts.

Through the training of the guesses, we attempted a number of different orderings of guesses — some examples include, conducting a total ordering of all likely patterns using the likelihood measure and only using the pattern generator and then ordering — and we found that the most effective guessing strategy is to always first guess patterns found in the training set ordered by repetition and likelihood, followed by the symmetries/reversals ordered by likelihood, followed by the generated patterns, again, ordered by likelihood.

For 4x4 pattern guessing, we additionally wanted to leverage the embedding of 3x3 patterns into the 4x4 grid. To do so, we treated 3x3 embedded patterns as additional training for the Markov model much like the symmetries were, and they were also treated as pre-generated patterns to guess during the generation phase. Including these covers the first $\sim$ 7000 guesses for 4x4 and correctly guesses 60% of the 4x4 patterns, on average during the cross-validation.

We also found that the likelihood measure from the Markov model is crucial to achieving efficient pattern guessing. Figure 9 presents the fraction of patterns correctly guessed for each guess for each of the data sets with and without the use of the Markov model to order the guesses. Ordering the guesses using the Markov model vastly improves the performance of the guesser.

In total, after 50,000 guesses, our guesser can crack 96.2% and 67.4% of pen-and-paper 3x3 and pen-and-paper 4x4 patterns, respectively, and 16.7% and 19.9% of 3x3 and 4x4 patterns, respectively, after 20 guessing attempts,i.e., the phone lockout point. When considering only the defensive and offensive patterns, we ran cross-fold validations with 100 samples due to the smaller size of the data sets, and the guesser was still able to guess 94.5% and 88.5% of the offensive and defensive 3x3 patterns respectively (12.5% and 4.0% after 20 guesses). The guesser correctly guessed 61% of the 4x4 offensive patterns and only 37% of the 4x4 defensive patterns (16.7% and 3.2% after 20 guesses). Visuals of the guessing rate are found in Figure 10. These results are further summarized in Table 3.

Testing the guesser. With the guesser well trained, it can be applied to the reserved testing set which was withheld from the prior analysis. The self-reported 3x3 patterns, as shown previously, has many of the same properties as the pen-and-paper 3x3 patterns. We wish to determine how easily these patterns can be guessed when only training on the pen-and-paper 3x3 patterns.

Without entering the pattern generation phase, the guesser can crack 70% of the self reported patterns with the first 2,256 guesses using only the training input of the pen-and-paper 3x3 patterns. Using the Markov model to reach 50,000 guesses total, the guesser can guess 96.3% of the self-reported 3x3 patterns. Within 20 guesses, 15% of the patterns are guessed. These results are presented in Figure 10a and Table 3.

Partial Guessing Entropy. The partial guessing entropy for each of the data sets, as well as references to related results [23, 18, 15], is presented in Table 3 for $\alpha = 0.1, 0.2$ and 0.5.

The partial guessing entropy for all the test data sets for guessing the first 10% of the patterns ($\tilde{G}_{0.1}$) is less than prior reported entropy rates [23, 18]. Most interesting, the guessability of the first 20% of the 4x4 patterns actually requires fewer guesses than guessing the similar proportion of 3x3 patterns for both 3x3 data sets. We believe this has to do with the fact that the most common 4x4 patterns are even more common than the most common 3x3 patterns (see Figure 6 for the most common 4x4 patterns); however the least common 4x4 patterns are less common than the least common 3x3 patterns. For $\alpha = 0.5$, the entropy is > 2 bits higher for 4x4 patterns compared to either of the 3x3 data sets.

When comparing the guessability of patterns to random 2-, 3-, and 4-digit PINs, it becomes apparent that in some cases, such as $\alpha = 0.1, 0.2$, guessing patterns is as easy as guessing a random

	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.5$	Perc. Guessed Total	Perc. Guessed with 20 attempts
Self-Reported 3x3	**6.62**	**6.95**	**9.49**	**95.9%**	**15.0%**
Pen-Paper 3x3 (all)	**6.59**	**6.99**	**8.93**	**97.2%**	**16.7%**
Pen-Paper 3x3 (Off.)	6.98	7.69	9.31	95.3%	12.5%
Pen-Paper 3x3 (Def.)	9.43	9.79	10.98	90.2%	4.0%
Pen-Paper 4x4 (all)	**6.23**	**6.64**	**11.61**	**66.7%**	**19.9%**
Pen-Paper 4x4 (Off.)	6.46	7.57	10.40	67.7%	16.7%
Pen-Paper 4x4 (Def.)	6.23	6.64	11.61	37.4%	3.2%
Uellenbeck et. al 3x3 (Off.) [23]	7.56	7.74	8.19		
Uellenbeck et. al 3x3 (Def.) [23]	8.72	9.10	10.90		
Song et. al 3x3 (w/ Meter) [18]	8.96	10.33	12.29		
Song et. al 3x3 (w/o Meter) [18]	7.38	9.56	10.83		
Random 3x3 Pattern ($U_{389,112}$)	18.57	18.57	18.57		
Random 4x4 Pattern ($U_{4,350,069,823,024}$)	41.98	41.98	41.98		
Random 6-dit PIN ($U_{1,000,000}$)	19.93	19.93	19.93		
Random 5-dit PIN ($U_{100,000}$)	16.60	16.60	16.60		
Random 4-dit PIN ($U_{10,000}$)	13.29	13.29	13.29		
Random 3-dit PIN ($U_{1,000}$)	9.97	9.97	9.97		
Random 2-dit PIN (U_{100})	6.64	6.64	6.64		
Real Users' 4-Digit PINs [18, 15]	5.19	7.04	10.08		

Table 3: Partial Guessing Entropy Comparisons

selection of 2-digit PINs. Even in the hardest setting, guessing half the data set ($\alpha = 0.5$), guessing 3x3 patterns is easier than guessing a random selection of 3-digit PINs and guessing 4x4 patterns is < 2 bits harder but much easier than guessing a random selection of 4-digit PINs. Interestingly, the guessability of patterns seems to be more in line with the difficulty of guessing real users 4-digit PINs (last line of table) which is another common unlock mechanism on mobile devices.

6. CONCLUSION

Like text-based passwords [13, 16, 17], we know that humans choose non-complex/insecure/easily-guessable Android graphical-unlock passwords [1, 2, 14, 18, 21, 23]. One obvious, and easy, solution that could encourage users to increase the complexity of chosen patterns is to increase the grid size from 3x3 to 4x4. To test the veracity of this solution, we conducted two large user studies, and found that the rate of repeated patterns for both 3x3 and 4x4 patterns is very high, as well as the rate of symmetric pairs. Further, we found that many 4x4 patterns are simple embeddings of 3x3 patterns. We then developed an advanced guessing algorithm to measure the guessability of 3x3 and 4x4 patterns, finding that 4x4 patterns are just as easily guessed as 3x3 patterns in many situations. As such, we believe that increasing the grid size will have minimal impact overall on the security of human-generated patterns. At the very least, for 4x4 patterns, we showed that while the overall number of guessed patterns is lower than 3x3, the guessability of the common 4x4 patterns requires less guesswork. We conjecture that increasing the grid size beyond 4x4 will not affect much change. As the grid size is increased, the ease of entering more complex patterns will be reduce as the number of contact points becomes more dense. The probability of mis-entering a pattern will be too high, which will likely encourage users to continue to choose easily guessable patterns, perhaps even more guessable than 4x4 patterns.

Acknowledgments

This work was support in part by the Office of Naval Research and the National Security Agency. At the Naval Academy, high school intern Jeanne Luning-Prak contributed in developing the online survey, and Midshipman Justin Maguire aided in data entry for the paper surveys. Flynn Wolf at the UMBC also assisted in administering paper surveys. Finally, we thank Rida Bazzi for shepherding this paper, and the anonymous reviewers for their helpful feedback in improving this paper.

References

[1] P. Andriotis, T. Tryfonas, and G. Oikonomou. Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method. In *Human Aspects of Information Security, Privacy, and Trust*, pages 115–126. Springer, 2014.

[2] P. Andriotis, T. Tryfonas, G. Oikonomou, and C. Yildiz. A pilot study on the security of pattern screen-lock methods and soft side channel attacks. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, WiSec'13, pages 1–6, 2013.

[3] M. Arianezhad, D. Stebila, and B. Mozaffari. Usability and security of gaze-based graphical grid passwords. In *Financial Cryptography and Data Security Workshop on Usable Security*, USEC'13, pages 17–33. Springer, 2013.

[4] A. J. Aviv and D. Fichter. Understanding visual perceptions of usability and security of Android's graphical password pattern. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC'14, pages 286–295, 2014.

[5] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 2010 Workshop on Offensive Technology*, WOOT'10, 2010.

[6] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith. Practicality of accelerometer side channels on smartphones. In *Proceedings of the 28th Annual Computer Security Applications Conference*, ACSAC'12, pages 41–50, 2012.

[7] R. Biddle, S. Chiasson, and P. C. Van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys (CSUR)*, 44(4):19, 2012.

[8] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP'12, pages 538–552, 2012.

[9] C. Bravo-Lillo, L. Cranor, J. Downs, S. Komanduri, S. Schechter, and M. Sleeper. Operating system framed in case of mistaken identity: measuring the success of web-based spoofing attacks on os password-entry dialogs. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 365–377, 2012.

[10] Cyanogenmod. Cyanogenmod|Android community operating system, 2015.

[11] M. Harbach, E. von Zezschwitz, A. Fichtner, A. De Luca, and M. Smith. ItâĂŹs a hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *Symposium on Usable Privacy and Security (SOUPS)*, 2014.

[12] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin. The design and analysis of graphical passwords. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, SSYM'99, 1999.

[13] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Proceedigns of the IEE Symposium on Security and Privacy*, SP'12, pages 523–537, 2012.

[14] G. Kessler. Technology corner: Calculating the number of Android lock patterns: An unfinished study in number theory. *Journal of Digital Forensics, Security and Law*, 8(4):57–64, 2013.

[15] H. Kim and J. H. Huh. PIN selection policies: Are they really effective? *computers & security*, 31(4):484–496, 2012.

[16] M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur. Measuring password guessability for an entire university. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & communications security*, CCS'13, pages 173–186, 2013.

[17] R. Morris and K. Thompson. Password security: A case history. *Communications of the ACM*, 22(11):594–597, 1979.

[18] Y. Song, G. Cho, S. Oh, H. Kim, and J. H. Huh. On the effectiveness of pattern lock strength meters: Measuring the strength of real world pattern locks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, 2015.

[19] L. Standing, J. Conezio, and R. N. Haber. Perception and memory for pictures: Single-trial learning of 2500 visual stimuli. *Psychonomic Science*, 19(2):73–74, 1970.

[20] J. M. Stanton, K. R. Stam, P. Mastrangelo, and J. Jolton. Analysis of end user security behaviors. *Computers & Security*, 24(2):124–133, 2005.

[21] C. Sun, Y. Wang, and J. Zheng. Dissecting pattern unlock: The effect of pattern strength meter on pattern selection. *Journal of Information Security and Applications*, 19(4):308–320, 2014.

[22] H. Tao and C. Adams. Pass-go: A proposal to improve the usability of graphical passwords. *International journal of Network Security*, 7 (2):273–292, 2008.

[23] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz. Quantifying the security of graphical passwords: The case of Android unlock patterns. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, CCS'13, 2013.

[24] E. von Zezschwitz, A. De Luca, P. Janssen, and H. Hussmann. Easy to draw, but hard to trace?: On the observability of grid-based (un)lock patterns. 2015.

[25] E. von Zezschwitz, P. Dunphy, and A. De Luca. Patterns in the wild: A field study of the usability of pattern and PIN-based authentication on mobile devices. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 261–270, 2013.

[26] R. Weiss and A. De Luca. Passshapes: Utilizing stroke based authentication to increase password memorability. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges*, NordiCHI '08, pages 383–392, 2008.

[27] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon. Passpoints: Design and longitudinal evaluation of a graphical password system. *Int. J. Hum.-Comput. Stud.*, 63(1-2):102–127, July 2005.

[28] M. Zviran and W. J. Haga. Password security: an empirical study. *Journal of Management Information Systems*, pages 161–185, 1999.