On the Effectiveness of Pattern Lock Strength Meters – Measuring the Strength of Real World Pattern Locks

Youngbae Song\textsuperscript{1}, Geumhwan Cho\textsuperscript{1}, Seongyeol Oh\textsuperscript{1}, Hyoungshick Kim\textsuperscript{1} and Jun Ho Huh\textsuperscript{2}

\textsuperscript{1}Computer Engineering, Sungkyunkwan University
\textsuperscript{2}Honeywell ACS Labs, USA

\{youngbae, geumhwan, seongyeol, hyoung\}@skku.edu
junho.huh@honeywell.com

ABSTRACT
We propose an effective pattern lock strength meter to help users choose stronger pattern locks on Android devices. To evaluate the effectiveness of the proposed meter with a real world dataset (i.e., with complete ecological validity), we created an Android application called EnCloud that allows users to encrypt their Dropbox files. 101 pattern locks generated by real EnCloud users were collected and analyzed, where some portion of the users were provided with the meter support. Our statistical analysis indicates that about 10\% of the pattern locks that were generated without the meter support could be compromised through just 16 guessing attempts. As for the pattern locks that were generated with the meter support, that number goes up to 48 guessing attempts, showing significant improvement in security. Our recommendation is to implement a strength meter in the next version of Android.

Author Keywords
Security; Password; Pattern Lock; Password Strength Meter

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
People often create weak passwords that are easy to remember without paying much attention to security. As a result, the real password space is much smaller than the theoretical space [5, 17], making brute-force and dictionary attacks possible and effective. One way to help users choose stronger passwords is to use a “password strength meter” and provide immediate feedback on the security level of passwords that users are about to use. The effectiveness of such meters have already been evaluated [9, 13, 26].

Although password strength meters have been evaluated to be effective in improving security of traditional passwords, it is not really known how such meters can be designed for “pattern lock-based passwords” that are used on mobile phones, and whether they can be effective on improving strength of graphical passwords such as pattern locks. This paper focuses on the popular Android pattern lock graphical passwords, and proposes a strength meter that measures the strength of users’ pattern locks based on how strong a pattern lock will be against shoulder-surfing or password guessing attacks. In designing the meter, we carefully considered different factors that can indicate pattern lock strength, including the length of a pattern lock, the number of points touched on the 9-point grid, and the number of point-to-point lines (referred to as “segments”) that intersect each other in a given pattern lock. To evaluate the correctness and accuracy of the designed meter, we conducted a user study involving 101 participants, asking participants to perform shoulder-surfing attacks on pattern locks that were categorized as weak, medium, or strong through the meter. Our study confirmed that the pattern locks that are categorized by the meter as strong are more difficult to compromise through shoulder-surfing attacks than those that are categorized as medium or weak.

After evaluating the accuracy of the designed meter, we conducted a second experiment to study the effectiveness of the meter in helping users select stronger pattern locks. To achieve ecological validity, that a field study was conducted in the wild without telling users what the experiment was about. Users downloaded a free (fully functional) Android application from the Android Play store called “EnCloud” that was designed to encrypt Dropbox content – no information about the pattern lock study was provided to users. To use EnCloud, users were required to generate a pattern lock for authentication, where some users were provided with the meter to help them choose stronger pattern locks. Because EnCloud’s pattern lock interface is very similar to the original Android pattern lock interface and serves a similar purpose, we claim that the collected results closely correspond to what the real-world pattern locks look like. We compared the characteristics of pattern locks that were generated normally (without meter assistance) and those that were generated with meter assistance, showing that the majority of the users benefited from meter assistance and generated more secure pattern locks. For the pattern locks that were generated with meter assistance, the partial guessing entropy [6] is 8.96 bits of information (when the goal of the attack is to compromise 10\% of the pattern
locks) compared to 7.38 bits for those that were generated without meter assistance. The key contributions of the paper can be summarized as follows:

- design of an effective pattern lock strength meter that considers characteristics like the pattern lock length and the number of intersecting segments;
- evaluation of the effectiveness of the designed strength meter with complete ecological validity, showing that users find the meter very useful and the meter is significantly effective in strengthening security pattern locks;
- empirical analysis on the pattern locks that are used in the wild, showing that the actual space of the pattern locks used in the Android platform might be much smaller than the theoretical space.

In the following section, we explain pattern lock characteristics that we considered while designing the proposed meter. Section “The First Study: Performing Shoulder-Surfing Attacks on Pre-Categorized Pattern Locks” discusses the results from our first user study on performing shoulder-surfing attacks on pattern locks. In Sections “The Second Study: Gauging the Effectiveness of the Meter in the Wild” and “Statistical Analysis of the Pattern Lock Strength,” we discuss the results from the second study and the effectiveness of the meter. Implications of the study results are discussed in Section “Discussion.” In Section “Ethical Considerations,” we explain how ethical issues were considered in the user studies. Our conclusions are at Section “Conclusions.”

DESIGNING A PATTERN LOCK STRENGTH METER

We first design a reasonable pattern lock strength meter by analyzing different pattern lock characteristics that affect their security against common threats like shoulder-surfing attacks or password guessing attacks.

Pattern Lock Strength Meter Design

Fig. 1 shows a prototype implementation of the pattern lock strength meter that we designed. The visual slider on the top of the screen indicates the strength of a pattern lock being created by a user, which gets updated in real time as the user selects more dots to be included in a pattern. Fig. 1 shows three groups of two screenshot examples where the first screenshot from each group represents a weak pattern lock and the second screenshot represents a strong pattern lock.

The goal is to design a meter that is effective against the two most commonly considered threats in graphical passwords, which are shoulder-surfing attacks and password guessing attacks. To measure and visualize the strength in a scale of weak, medium, or strong, we created a scoring function that uses the heuristics related to a pattern lock’s total length, repeated sub-patterns, and intersecting segments (point-to-point lines). Initially, we equally divided strength categories and scores into weak (scores between 0.00~0.33), medium (0.34~0.67) and strong (0.68~1.00), where the maximum meter strength score is 1. Those heuristics were adapted from [26]. We now explain the pattern lock characteristics (parameters) that are used in the scoring function:

- Pattern length \((L_p)\) is the sum of the lengths of all the segments used in an input pattern \(p\); the length of a segment is measured as the distance between the two points that are connected.
  
  To calculate the distance between two points \((x_1,y_1)\) and \((x_2,y_2)\), we took the largest value from \(|x_1-x_2|\) and \(|y_1-y_2|\), rather than using another complex distance metric like the Euclidean distance. The maximum length possible is 15. Our intuition is that the security of a pattern lock increases as its length increases (just as in textual passwords [26]).

- Ratio of non-repeated segments \((N_p)\) is the ratio of “the number of times a segment does not appear in the longest repeated sub-pattern” to “the total number of segments in an input pattern \(p\)”.
  
  Symbols \(R_p\) and \(S_p\) are used to represent “the number of times that a segment appears in the longest repeated sub-pattern in \(p\)” and “the total number of segments in \(p\),” respectively. A repeated sub-pattern is a sub-pattern that occurs two or more times consecutively, i.e., \(N_p = (S_p - R_p)/S_p\). Our intuition is that the higher the probability of predicting the next point in a pattern, the weaker the pattern lock (similar to the effects of repeating characters in a textual password).

- Number of intersecting points \((I_p)\) is the number of times a segment in a given pattern intersects another segment. The maximum number of intersecting points possible in the Android system is 14. Our intuition is that the more intersecting points there are in a given pattern, the more complex and secure against shoulder-surfers. However, through a pilot study, we observed that the pattern lock security is not greatly affected after reaching \(I_p = 5\). Therefore, we used \(\min(I_p,5)\) rather than \(I_p\) in the scoring function.

Those three parameters are used in the meter strength scoring function and are referenced throughout the paper. Here is the function for the measuring the meter strength score \(M_p\), where \(w_L, w_N,\) and \(w_I\) are the relative weights for \(L_p, N_p,\) and \(I_p\), respectively:

\[
M_p = w_L \cdot \frac{L_p}{15} + w_N \cdot \frac{N_p}{5} + w_I \cdot \frac{\min(I_p,5)}{5}
\]

In the initial design of the meter (i.e., before optimizing the use of the parameters in the scoring function), those three parameters were given equal weightings (i.e., \(w_L = w_N = w_I = 1/3\)) as we had no data on the relative impact each parameter can have on the strength of a pattern lock. The next section describes a shoulder-surfing attack experiment, which was conducted to work out the correct weightings that should be placed on those parameters to improve the accuracy of the meter.

THE FIRST STUDY: PERFORMING SHOULDER-SURFING ATTACKS ON PRE-CATEGORIZED PATTERN LOCKS

To validate the correctness of the strength meter, our first experiment was designed to demonstrate correlations between the strength scores of pattern locks and their security against...
shoulder-surfing attack, and help us optimize the scoring function described above.

Study Design
The aim of the study is to (1) validate the correctness of the initial meter design, and (2) determine the relative importance (weightings) of the three parameters to optimize the scoring function. The study was designed similar to [29], examining how strong pattern locks are against shoulder-surfing attack. In a typical shoulder-surfing attack, an attacker discovers a password by taking a peek over a user’s shoulder during an authentication process. Graphical passwords, including pattern locks, are prone to such an attack [29]. To that end, if the design of the meter is correct, high strength scores should indicate more robustness against shoulder-surfing attacks and vice versa. The study was conducted in a controlled laboratory environment to avoid any distractions.

We recruited 101 participants in total by posting fliers about our study on bulletin boards in a university. The mean age was 33 (19–57), where 83 of the participants were male and 18 were female. The participants were randomly assigned to a set of six pattern locks containing 2 weak, 2 medium, and 2 strong patterns. One individual was recruited to play the role of “victim” and was given the task of entering each of the six assigned patterns on a Samsung Galaxy Note 2 phone in a random order. The rest of the participants were asked to play the role of “shoulder surfer” and peek and remember the pattern locks entered by the victim. Before the victim entered a pattern lock, participants (shoulder surfers) were asked to move toward his or her best viewing position (e.g., to the left or to the right side of the victim). While the victim entered each of the six pattern locks, a shoulder surfer observed the victim’s login process. When the victim completed entering a pattern lock (i.e., when the login process was complete), the shoulder surfer was asked to remember and enter the victim’s pattern lock without any limit in the number of attempts. The attack was considered as failure when the shoulder surfer gave up on guessing the pattern. Since it is unlikely in reality that a shoulder surfer would be able to access the victim’s phone immediately after observing the login process, the participants were asked to take a short break (about 10 seconds) before entering the observed pattern lock.

The participants were strongly encouraged to put in their best efforts to remember the pattern locks entered by the victim, being rewarded with a $1 honorarium for successfully remembering a pattern lock.

Study Results
The total number of tested pattern locks was 606 (through 101 participants), among which 432 (71.29%) were successfully attacked and 174 (28.71%) were not successfully reproduced by the participants.

To validate the correctness of the proposed scoring function, we first analyze the relationship between the strength score of a pattern lock $p$ and the likelihood of shoulder-surfing attack being successful on $p$. The likelihood of shoulder-surfing attack being successful is the proportion of the matched sub-patterns between the victim’s pattern $p$ and the pattern $p_u$ guessed by the shoulder surfer $u$. The proposed metric for attack success rate is as follows:

\[ A_p(u) = \frac{\max\{i \mid p[1..i] = p_u[1..i]\}}{\max(|p|, |p_u|)} \]

where $p[1..i]$ and $p_u[1..i]$ are a prefix of $p$ and $p_u$, respectively.

The results showed that there is a significant correlation between $A_p$ and its strength score (correlation coefficient = 0.415, $p < 0.01$).

We categorized the tested patterns into “Compromised Patterns” (pattern locks successfully reproduced by a participant) and “Robust Patterns” (pattern lock unsuccessfully reproduced by a participant) according to the attack results, and computed the mean values of strength scores ($M_p$), pattern lengths ($L_p$), numbers of the segments used for the longest repeated sub-pattern ($R_p$), numbers of intersecting points ($I_p$), and attack success rates ($A_p$) for those two groups. Table 1 shows these results. This comparison shows that pattern locks categorized as strong with a high strength score are indeed harder to compromise through shoulder-surfing attack than those categorized as weak or medium that have relatively lower strength scores. We study the characteristics of those two groups in the following subsections.

(Images of weak and strong pattern locks, with labels: (a) Length, (b) Non-repeated segments, (c) Intersecting points. Figure 1. Examples of weak (left) and strong (right) pattern locks. The slider indicates the strengths of the chosen pattern locks. Each column shows the strength difference with a feature: (a) Length $L_p$, (b) Non-repeated segments $N_p$, or (c) Intersecting points $I_p$.)

Table 1. Mean values of strength scores ($M_p$), pattern lengths ($L_p$), numbers of the segments used for the longest repeated sub-pattern ($R_p$), numbers of intersecting points ($I_p$), and attack success rates ($A_p$) for two groups.
Length of a Pattern Lock
Compromised Patterns (Compro.) have a mean length of 6.05 with a standard deviation of 2.13. Robust Patterns (Robust) have a mean length of 8.68 with a standard deviation of 1.73. There is a significant difference in the pattern length between the two groups ($p < 0.01$, one-tailed unpaired t-test).

Ratio of Non-repeated Segments
We also look at the ratio of “the number of times a segment does not appear in the longest repeated sub-pattern” to “the total number of segments in an input pattern $p$.” For this metric, Compromised Patterns have a mean of 0.59 with a standard deviation of 0.33, while Robust Patterns have a mean of 0.65 with a standard deviation of 0.23. There is a significant difference in the pattern length between the two groups ($p < 0.01$, one-tailed unpaired t-test).

Number of Intersecting Points
As for the analysis on the number of intersecting points (see Section “Designing a Pattern Lock Strength Meter”), Compromised Patterns have a mean number of 1.25 intersecting points with a standard deviation of 1.53. Robust Patterns have a mean number of 3.17 intersecting points with a standard deviation of 2.15. We can also see that they are significantly different in the number of the intersecting used in points ($p < 0.01$, one-tailed unpaired t-test).

Parameter Optimization
To improve the accuracy of the meter, we optimized the weightings of the three parameters (see Section “Designing a Pattern Lock Strength Meter”) based on the study results. The inter-relationships between those parameters are examined by linear regression to optimize their relative weightings. With the regression coefficients, we adjusted weightings of the pattern lock length, the number of the segments used for the longest repeated sub-pattern, and the number of intersecting points from 1/3 to 0.81 for $w_L$, 0.04 for $w_N$ and 0.15 for $w_I$, respectively (i.e., the pattern length is relatively much more important in measuring the pattern lock strength than the other two). To demonstrate improvement, we measured the correlation between a pattern $p$’s strength score $M_p$ and its attack success rate $A_p$ again with the optimized weightings (correlation coefficient $= 0.462$, $p < 0.01$). The correlation coefficient increased from 0.415 to 0.462, implying that the new weightings improve the meter accuracy.

As we mentioned in Section “Designing a Pattern Lock Strength Meter,” initially, the pattern strength categories were equally divided into weak (0.00–0.33), medium (0.34–0.67) and strong (0.68–1.00), where the maximum strength score is 1. Based on the results, we also adjusted those three ranges to more accurately reflect on the pattern lock strength. To do that, we used the confidence intervals for the strength scores (updated with new parameter weightings) of Compromised Patterns and Robust Patterns groups. For Compromised Patterns, the 95% confidence interval of strength scores is from 0.3709 to 0.4000, while the 95% confidence interval of strength scores for Robust Patterns is from 0.5610 to 0.6026. Hence, we redefined the categories into weak (0.00–0.40), medium (0.41–0.56), and strong (0.57–1.00). For better and more intuitive visualization though, those three ranges were normalized to show up as the same size (i.e., 1/3 each) on the slider. The mean of the finally adjusted strength scores ($\hat{M}_p$) is presented in Table 1, showing that the score gap between Compromised Patterns and Robust Patterns increases immensely compared to the original scores. This implies that the adjusted strength scores are more accurate.

### Table 1. Basic statistics for the First Study

<table>
<thead>
<tr>
<th></th>
<th>Freq.</th>
<th>$M_p$</th>
<th>$M_p$</th>
<th>$L_p$</th>
<th>$N_p$</th>
<th>$I_p$</th>
<th>$A_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compro.</td>
<td>432</td>
<td>0.410</td>
<td>0.382</td>
<td>6.058</td>
<td>0.587</td>
<td>1.245</td>
<td>1.000</td>
</tr>
<tr>
<td>Robust.</td>
<td>174</td>
<td>0.603</td>
<td>0.638</td>
<td>8.684</td>
<td>0.651</td>
<td>3.167</td>
<td>0.371</td>
</tr>
</tbody>
</table>

THE SECOND STUDY: GAUGING THE EFFECTIVENESS OF THE METER IN THE WILD
The aim of the second study is to gauge how effective the proposed pattern lock strength meter is in the real-world.

Study Design
To achieve complete ecological validity, we developed an independent security application called EnCloud (see Fig. 2) that is equipped with our meter and made it available on Google Play.

When the EnCloud application is installed and launched for the first time, it asks a user’s consent to anonymously disclose information about his or her behavior for research purposes. If the user agrees, the user is asked to choose a pattern lock for authentication purposes (to prevent unauthorized access to the user’s files stored in Dropbox). After choosing a pattern lock, the user is asked to enter the same pattern again for confirmation; if the reentered pattern matches the original pattern, that pattern is saved; otherwise, the user is required to repeat the same procedure until the patterns match successfully. Because the purpose of EnCloud is to protect Dropbox data, our intuition is that the users’ pattern locks are created with similar level of security and caution as the pattern locks that the users use to lock their Android phones.
Half of the users were asked to create a pattern lock with the presence and support of the strength meter, and the remaining half were asked to create a pattern lock without the meter support. The first group is referred to as the ‘With Meter’ group and the second group is referred to as the ‘Without Meter’ group. We compare the security of pattern locks generated between those two groups and study the differences in their pattern generation behaviors (see Fig. 3). The users were never informed about the study intentions.

To ensure that there is no bias in the group selection process, we rely on the back-end server to create a random number based on the unique registration identifier submitted by EnCloud, and send that number back to EnCloud. The registration identifier is derived from the Universally Unique Identifier (UUID) of users’ phones. If that number is an even number, EnCloud enables the meter support (With Meter) and also allows users to try different pattern locks by clicking on the “initialization” button. If it is an odd number, users generate pattern locks as per normal without the meter support (Without Meter). The back-end server collects and checks the user’s registration identifier to prevent double registration. Only the MD5 hashes of the UUIDs are ever sent to the server though. We collected 101 users’ pattern locks in total during a two months period.

**Study Results**

A total of 101 users installed EnCloud and actively used it. 52 (50.5%) users created their pattern locks with the meter support (With Meter) and 49 (49.5%) created their pattern locks without the meter support (‘Without Meter’). Table 2 shows the characteristics of the pattern locks generated between the two user groups. As expected, the strength scores of user selected pattern locks (in Table 2) are lower than the scores of system generated pattern locks (in Table 1); this is true even for With Meter pattern locks.

**Table 2. Meter parameter changes: ‘Without Meter’ vs ‘With Meter’**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Users</td>
<td>(M_p)</td>
<td>(L_p)</td>
<td>(N_p)</td>
</tr>
<tr>
<td>Without Meter</td>
<td>49</td>
<td>0.251</td>
<td>4.755</td>
<td>0.408</td>
</tr>
<tr>
<td>With Meter</td>
<td>52</td>
<td>0.365</td>
<td>6.423</td>
<td>0.414</td>
</tr>
</tbody>
</table>

**Strength Score**

With Meter pattern locks have a mean strength score of 0.251 with a standard deviation of 0.164 (see Table 2). In comparison, Without Meter pattern locks have a mean score of 0.365 with standard deviation of 0.242, which is about 0.114 higher on average. Such a difference in the strength scores is statistically significant (\(p < 0.01\), one-tailed unpaired t-test).

**Pattern Lock Length**

The pattern lock lengths are significantly different between the two groups. Without Meter pattern locks have an average length of 4.755 with a standard deviation of 2.376. The largest length is 12 but just one pattern lock has that length. With Meter pattern locks have an average length of 6.423 with a standard deviation of 3.268. The largest length is 15 and three pattern locks have that length. That difference between the two groups is statistically significant (\(p < 0.01\), one-tailed unpaired t-test).

**Ratio of Non-repeated Segments**

We compare the ratio of non-repeated segments between the two groups. Without Meter pattern locks have a mean ratio of 0.408 with a standard deviation of 0.262, while With Meter pattern locks have a mean ratio of 0.414 with a standard deviation of 0.281, which is slightly higher. We failed to show statistical significance in the difference between the two groups though (\(p = 0.458\), one-tailed unpaired t-test).

**Number of Intersecting Points**

The number of intersecting points is compared next. With Meter pattern locks have a mean number of intersecting points of 1.096 with a standard deviation of 3.303. In comparison, Without Meter pattern locks have a mean number of intersecting points of 0.184 with a standard deviation of 0.601. That difference is statistically significant (\(p < 0.05\), one-tailed unpaired t-test).

**Frequency of the 9 Points Used in Pattern Locks**

We also analyze the usage frequency of each of the 9 points in the \(3 \times 3\) grid. Those 9 points are numbered from 1, starting with the point in the top left corner, to 9, which is the point in the bottom right corner of the grid. The results between the two groups are compared in percentages (see Fig. 4). In
In this section, we measure security of the pattern locks from a statistical viewpoint. We treat pattern locks as events: since each point in a pattern lock can be represented as a sequence of numbers, the lock can be represented as a sequence of numbers. The efficiency of performing guessing attacks on passwords can be estimated by analyzing the usability of these sequences. In general, passwords that are more predictable are less secure. Therefore, we compare the security of pattern locks by analyzing the usage frequency of segments used in pattern locks. This information can be used to infer the likelihood of a password being guessed.

### STATISTICAL ANALYSIS OF THE PATTERN LOCK STRENGTH

In this section, we use the $N$-gram Markov model to estimate the probability of a pattern lock being used. A $N$-gram model is a probabilistic model that uses the history of $N$ previous symbols to predict the next symbol. This model is useful for estimating the likelihood of a pattern lock being used, as it takes into account the frequency of segment usage.

#### $N$-gram Markov Model

The $N$-gram Markov model is used to study probability distributions over sequences of observations. Using the Markov model, we can estimate the probability of an event occurring. For example, in English text, the letter that comes after $t$ is more likely to be $h$ than $q$. If guessing an English word that starts with $t$ is the goal (such as “the”), one should start with words that start with $th$ first, and not with those that start with $tq$. We treat pattern locks as events: since each point in a pattern lock represents a number between 1 and 9, a pattern lock can be represented as a sequence of numbers. The $N$-gram Markov model is used to estimate the probability of the number (point) sequences $x_1, \ldots, x_m$ as

$$P(x_1, \ldots, x_m) = P(x_1, \ldots, x_{n-1}) \cdot \prod_{i=n}^m P(x_i|x_{i-n+1}, \ldots, x_{i-1})$$

### Table 3. Usage frequency of segments in pattern locks.

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>H1</th>
<th>D1</th>
<th>V2</th>
<th>H2</th>
<th>D2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unused</td>
</tr>
<tr>
<td>Without Meter</td>
<td>12</td>
<td>12</td>
<td>16</td>
<td>6</td>
<td>6</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>With Meter</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>%</td>
</tr>
<tr>
<td>Without Meter</td>
<td>32.0%</td>
<td>41.0%</td>
<td>22.1%</td>
<td>0.5%</td>
<td>2.2%</td>
<td>2.2%</td>
<td>100%</td>
</tr>
<tr>
<td>With Meter</td>
<td>30.8%</td>
<td>33.9%</td>
<td>19.8%</td>
<td>1.7%</td>
<td>1.7%</td>
<td>12.1%</td>
<td>100%</td>
</tr>
</tbody>
</table>

### Figure 5. Frequencies of the segments used in EnCloud users’ pattern locks: 'Without Meter' vs 'With Meter'.

Without Meter pattern locks, the most frequently used points are 2 and 8, which were used 37 times (13.6%). The least frequently used point is 4, which was only used 23 times (8.5%) as shown in (b) of Fig. 4. With Meter pattern locks, the most frequently used points are 1, 5, 6, and 9, which were used 41 times (12.0%), and the least frequently used points are 4 and 7, which were used 31 times (9.1%) as shown in (c) of Fig. 4. Overall, the usage frequency looks more evenly distributed in the With Meter pattern locks.

#### Segments Used

A segment in a pattern lock is defined as a line that connects two points together. We counted the usage frequency of all the segments used in pattern locks for the two groups (With Meter and Without Meter) and calculated the total number of segments used in pattern locks. Fig. 5 shows the proportion of the usage frequency for each segment: darker the color, higher the number of segments used.

The total number of segments used in Without Meter pattern locks is 222. But there are only 48 distinct segments in that 222. The average number of segments used in pattern locks is 4.625 with a standard deviation of 4.639. The most frequently used segment connects points 8 and 9, which was used 17 times (5.88%). The usage frequency of segments for With Meter pattern locks appears more evenly distributed.

Computing Shannon entropy [20] for those distributions shows that usage frequency distribution for segments used in With Meter pattern locks has a higher entropy of 1.641 compared to an entropy value of 1.510 for Without Meter pattern locks. Shannon entropy is 1.908 when all the segments are equally used.

Finally, we analyze the usage frequency of segments used in the pattern locks. Segments are categorized into six groups based on their type (V: vertical, H: horizontal, or D: diagonal) and length (1 or 2), and are counted (see Table 3). For example, D2 counts diagonal segments that have length 2. The With Meter pattern locks have a significantly higher number of D2 (12.1%) than the Without Meter pattern locks (2.2%).

### Table 3. Usage frequency of segments in pattern locks.

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>H1</th>
<th>D1</th>
<th>V2</th>
<th>H2</th>
<th>D2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unused</td>
</tr>
<tr>
<td>Without Meter</td>
<td>12</td>
<td>12</td>
<td>16</td>
<td>6</td>
<td>6</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>With Meter</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>%</td>
</tr>
<tr>
<td>Without Meter</td>
<td>32.0%</td>
<td>41.0%</td>
<td>22.1%</td>
<td>0.5%</td>
<td>2.2%</td>
<td>2.2%</td>
<td>100%</td>
</tr>
<tr>
<td>With Meter</td>
<td>30.8%</td>
<td>33.9%</td>
<td>19.8%</td>
<td>1.7%</td>
<td>1.7%</td>
<td>12.1%</td>
<td>100%</td>
</tr>
</tbody>
</table>
To use the Markov model we have to determine the initial probabilities $P(x_1, \ldots, x_{n-1})$ and the transition probabilities $P(x_n|x_1, \ldots, x_{n-1})$, which means the probability of a number sequence occurring $x_n$ associated with the state of $x_1, \ldots, x_{n-1}$. For example, if we calculate the probability of a pattern (1, 2, 3, 4) using the 3-gram Markov model, we can write the equation as

$$P(1, 2, 3, 4) = P(1, 2) \cdot P(3|1, 2) \cdot P(4|2, 3)$$

**Partial Guessing Entropy**

Guessing entropy [18] is a useful metric to evaluate the average number of successive guesses that an attack needs to make to find the correct answer. It is often used for measuring the strength of passwords against guessing attacks. However, since the guessing entropy metric cannot be used to measure the guessing difficulty just for a desired portion of passwords (i.e., the smaller subset of real world passwords that are actually targeted), an alternative metric called Partial guessing entropy [6] (or $\alpha$-guessing entropy) have been recently proposed and is popularly used. It measures the average number of trials to correctly guess a fraction $\alpha$ of the entire password set.

For $0 \leq \alpha \leq 1$, let $\mu_n = \min \{j|\sum_{i=1}^{j} p_i \geq \alpha\}$ where $p_i$ is the probability of $i^{th}$ element occurring in non-increasing order, and let $\lambda_{\mu_n} = \sum_{i=1}^{\mu_n} p_i$, which is the actual fraction covered. With those notations, partial guessing entropy is defined as follows:

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

We note that the traditional guessing entropy is a special case of partial guessing entropy with $\alpha = 1$.

**How the Two Statistical Techniques are Used**

We compare the pattern lock distributions of the two groups. To calculate the partial guessing entropy of a pattern lock set, we need to know the probability distribution of all the possible pattern locks. However, our collection comprises of 32 With Meter pattern locks and 49 Without Meter pattern locks, which are much smaller than the theoretical space of 389,112 possible patterns. Therefore, the $N$-gram Markov model is used to show approximate probability distributions.

**Measuring Entropy**

We used the 3-gram Markov model with a simple Laplace smoothing algorithm – each count is incremented by 1 – to cover rare $N$-gram cases; this is the same guessing model used in [25]. The estimated probabilities are sorted in an non-increasing order and the probabilities of the top 200 pattern locks are plotted in (a) of Fig. 6. Although the probability curve for With Meter seems more evenly distributed than the curve for Without Meter, the probabilities for patterns in both curves decrease dramatically below the 20th patterns, indicating that their pattern distributions are skewed in favor of a small number of commonly used patterns.

Next, we measure the partial guessing entropy to demonstrate the security of pattern locks against guessing attacks. Our results are shown in (b) of Fig. 6. In this figure, when $\alpha$ is low ($\leq 0.1$), the guessing entropy estimates do not seem too different between the two groups. But as $\alpha$ increases, the difference between the guessing entropy estimates increases significantly, clearly demonstrating the effectiveness of the meter in strengthening pattern lock security. 10% of Without Meter pattern locks can be compromised with just 16 trials (i.e., $\alpha = 0.1$), while at least 48 trials are needed to compromise the same portion of With Meter pattern locks. To compromise 50% of Without Meter pattern locks 2,354 trials are needed (i.e., $\alpha = 0.5$), while at least 2,911 trials are needed to compromise the same portion of With Meter pattern locks. This shows that the With Meter pattern locks are much more difficult to guess.

We express those results in “bits of information” for more intuitive comparison with other measurements. This conversion is done as follows:

$$\tilde{G}_\alpha(\chi) = \log \left( \frac{2 \cdot G_\alpha(\chi)}{\lambda_{\mu_n}} - 1 \right) + \log \frac{1}{2 - \lambda_{\mu_n}}$$

Our estimate results are shown in Table 4. Partial entropy estimates are calculated with various $\alpha$ levels, ranging from 0.1 to 0.5. The guessing entropy estimates for the With Meter pattern locks are higher than the Without Meter pattern locks at all levels, the 3-gram models. For comparison, we calculated the partial entropy estimates for the set of pattern locks with the uniform distribution ($U_{389112}$), the set of 4-digit PINs with the uniform distribution ($U_{10000}$), the set of 5-digit PINs with the uniform distribution ($U_{100000}$), and the real world 4-digit PINs [2]. Intriguingly, for all $\alpha$ values shown in Table 4, the guessing entropy estimates for the Without Meter pattern locks are higher than those of the real world 4-digit PINs with $\alpha$.

**DISCUSSION**

**On the Effectiveness of the Pattern Lock Meter**

Our user study results confirm (with ecological validity) that a well-designed pattern lock strength meter is indeed effective in helping users choose more secure pattern locks. Statistical analysis (see Section “The Second Study: Gauging the Effectiveness of the Meter in the Wild”) shows that the pattern
locks generated with the meter support have higher guessing entropy estimates than those that are generated without the meter support. Based on these key findings, is to consider implementing the meter in the next version of Android to encourage users to move to stronger pattern locks.

On the Security of Pattern Locks
By quantifying the security of pattern lock authentication through entropy estimates (based on real EntCloud users’ pattern lock data), we show that real-world pattern locks (i.e., Without Meter pattern locks) are, as expected (based on theoretical password space), more secure than 4-digit PINs (2.19 entropy bits higher for α = 0.1 and 2.53 bits higher for α = 0.5). Users who are more concerned about security should consider using pattern locks over 4-digit PINs.

Unlike a previous study that was conducted in a lab setting [25], we collected and observed real EntCloud users’ pattern locks and behaviors without ever revealing the study intentions. Because of such environmental differences in the studies, there are some differences in the results: our entropy estimates were substantially lower than their entropy estimates (1.34 bits lower) when α = 0.1, but our estimates were inversely higher than theirs when α > 0.2.

**Most Likely Used Pattern Locks**
We use the Markov model to identify pattern locks that are most likely to be used. Fig. 7 shows the top 10 most likely used pattern locks for Android. Here are some interesting observations on them:

- **Short length:** The lengths of all of those pattern locks are less than or equal to 5. Except for the 1st, 3rd and 8th pattern locks, the rest of the pattern locks have length of 4, which is the minimum length required in.

- **Small number of turns:** Except for the 7th pattern lock, all others have just one single turn in them. This implies that many users prefer using simple patterns that can be drawn quickly and easily.

- **Popular directions:** All of those patterns start from the left and move to the right, and from the top and move to the bottom. We believe that those characteristics are strongly related to the directions in which many written languages are interpreted – we looked at the EntCloud users’ country information on Google Play to confirm this. Those trends indicate that information about users’ locations and languages can provide hints for adversaries to be more effective in guessing pattern locks.

**Unused Segments**
From looking at the usage frequency of individual segments in the collected pattern locks (see Fig. 5), we identify segments that are rarely or never used. In the pattern locks that are generated without the meter support, diagonal segments with length 2 (D2) are rarely used or never used as they are more difficult to draw than vertically or horizontally straight segments. Fig. 8 shows some representative examples of such unused segments. As shown in Table 3, one possible solution to that segment selection bias is to use a pattern lock meter. Another possible way is to use a different underlying layout (e.g., a circular layout [25]) instead of using the current 3 × 3 grid layout.

### Table 4. Comparing bits of information of several distributions and different values for the target fractions.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>α = 0.1</th>
<th>α = 0.2</th>
<th>α = 0.3</th>
<th>α = 0.4</th>
<th>α = 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-gram</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>With Meter</td>
<td>8.96</td>
<td>10.33</td>
<td>11.32</td>
<td>12.17</td>
<td>12.92</td>
</tr>
<tr>
<td>Without Meter</td>
<td>7.38</td>
<td>9.56</td>
<td>10.83</td>
<td>11.79</td>
<td>12.61</td>
</tr>
<tr>
<td>Random Patterns (U_{389112})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real Users’ 4-digit PINs [17]</td>
<td>18.57</td>
<td>18.57</td>
<td>18.57</td>
<td>18.57</td>
<td>18.57</td>
</tr>
<tr>
<td>Random 4-digit PINs (U_{10000})</td>
<td>5.19</td>
<td>7.04</td>
<td>8.37</td>
<td>9.38</td>
<td>10.08</td>
</tr>
<tr>
<td>Random 5-digit PINs (U_{100000})</td>
<td>13.29</td>
<td>13.29</td>
<td>13.29</td>
<td>13.29</td>
<td>13.29</td>
</tr>
<tr>
<td>1st 2nd 3rd 4th 5th 6th 7th 8th 9th 10th</td>
<td>16.61</td>
<td>16.61</td>
<td>16.61</td>
<td>16.61</td>
<td>16.61</td>
</tr>
</tbody>
</table>

**Figure 7. The top 10 most likely used patterns in the Markov model.**
User Behaviors in the Presence of the Meter

To understand user behaviors in the presence of the pattern lock meter, we conducted a separate follow-up study with 14 EnCloud users. From the study, we found that 71.43% of the participants prefer using pattern locks over PINs to authenticate themselves. About 57.14% (i.e., more than half) of the participants who were given the meter upon choosing a pattern lock decided to select a stronger pattern because of the low meter scores. Those participants felt that their original patterns were not sufficiently strong.

ETHICAL CONSIDERATIONS

It was not our intention to collect personal information or use collected data for commercial or illegal purposes. Before installation, all EnCloud users were informed that their data may be used in a scientific research and we asked for their consent. Only the anonymized data was used for statistical analysis, and we only collected MD5 hashes of the UUIDs of users phones.

RELATED WORK

Previous studies [28] have shown that pattern lock graphical passwords are more usable than text-based passwords such as PINs, while providing a sufficiently large password space of 389,112 possible patterns. Graphical passwords are typically classified as recall-based, recognition-based, cued recall-based schemes [4]. Jermyn et al. proposed Draw-A-Secret (DAS) as the first recall-based scheme in 1999 [16]. BDSA [12], YAGP [14] and Passdoodles [27] improved security and usability from the first one. Android pattern locks are a special case of the Pass-Go scheme [23], which uses the intersections on a 2D grid and covers a $2^{109}$ password space with an average password length of 17.

Graphical passwords tend to have better memorability than text-based passwords [1, 10] because human brains are better at remembering graphical information [22]. Zeszchitz et al. [28] showed that graphical pattern locks are more favorable and outperform PINs during error recovery.

Only a few studies have analyzed the security of graphical passwords though [8, 11, 15, 24]. Uellenbeck et al. [25] shared their skeptical views on the actual, small graphical password space that is used in reality. They conducted a large-scale user study to observe the actual pattern locks that participants choose, and interviewed the participants about the strategies they used in choosing patterns. The participants played a game where they were asked to choose a pattern lock (from a pre-generated pool of pattern locks) and to guess pattern locks chosen by other participants. Based on those user study results, they quantified the strength of pattern locks using the Markov model which are typically used to measure the security of passwords [7, 19]. Their results, however, could have been affected by the incentives that were given to participants to choose memorable yet sufficiently secure pattern locks; their participants were also aware of the study intentions. Another limitation of their studies is the fact that pre-generated patterns were used. In comparison, our analysis was performed on a real world pattern lock dataset that was collected through the users of EnCloud. Their results also demonstrated that users’ strategies for generating pattern locks on the current grid layout are highly biased. They tried to minimize that bias by modifying and rearranging the grid layout and found that a circular layout helped improve the entropy estimates of pattern locks.

Passwords strength meters have been designed to help users choose stronger passwords. It is usually presented in the form of a visual bar that colors itself differently depending on the strength of a password. To design an effective password strength meter, Ur et al. [26] investigated a variety of real world password meters, showing that a well-designed meter does help users choose longer passwords. Egelman et al. [13] also demonstrated the usefulness of password meters. However, their results indicate that even with the meter support, users would still create weak passwords on their low-risk accounts.

Although password meters have been proven to be effective, Carnavalet et al. [9] demonstrated that a badly designed meter can provide misleading strength information to users. They found many meters being used in popular websites that classify weak passwords as strong passwords. Some meters were inconsistent in measuring the strength scores.

As for pattern locks, Androitis et al. [3] proposed a pattern lock strength meter, and conducted a user survey to observe their pattern lock selection behaviors. Based on the study results, they discuss the following three heuristics: (1) a pattern lock that has the top leftmost node as the starting point is considered weak because more than 50% of users start drawing their patterns from the top leftmost node; (2) a pattern lock that consists of less than 6 points is considered weak; and (3) a pattern lock that has more than two directional changes is considered strong. Their results also show that pattern lock strength meters can be effective in helping users choose stronger pattern locks. Their study results, however, may not be sufficient to show the real impacts of pattern lock meters because the participants were aware of the study intentions, and could have been encouraged to create stronger pattern locks. In contrast, we achieved complete ecological validity by creating and distributing EnCloud, which had 101 real users generating pattern locks to protect their data [21]. Because EnCloud users were never told about the study intentions and were never asked to change pattern locks, we were able to gather and analyze unbiased, real world pattern locks and user behaviors.
CONCLUSIONS
We designed an effective pattern lock strength meter to help users choose stronger pattern locks in Android devices. We evaluated the effectiveness of the meter by collecting real world pattern locks generated by users of an Android app called EnCloud, which allows users to encrypt their Dropbox files. Our statistical analysis showed that (1) the meter is indeed effective in improving security of pattern locks, and that (2) pattern locks are more secure than real world 4-digit PINs.

As part of the future work, we plan to redesign the meter by using the Markov model we constructed for pattern locks to further improve the meter accuracy. That kind of meter design could be particularly useful in establishing an accurate and sound methodology for measuring pattern lock guessability.

It is also our plan to explore different visualization effects that can be added on to the meter to make the meter more effective.

ACKNOWLEDGEMENTS
This work is partly supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1010) supervised by the NIPA (National IT Industry Promotion Agency) and is also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2014R1A1A003707).

REFERENCES