Towards Constructing Consistent Pattern Strength Meters with User's Visual Perception

Leo Hyun Park[1], Eunbi Hwang[1], Donggun Lee[2],
and Taekyoung Kwon[1](✉)

[1] Yonsei University, Seoul, South Korea
{dofi,ebhwang95,c15336,taekyoung}@yonsei.ac.kr
[2] Ministry of National Defense, Seoul, South Korea

Abstract. Pattern lock strength meters designed for securing Android devices are inconsistent in their metering, e.g., assigning higher scores to weaker patterns. In this paper, we raise this inconsistency problem by analyzing five existing pattern strength meters. We reveal that they commonly miss some important visual features and even assign erroneous weights to features. As a preliminary study toward a consistent pattern strength meter in the future, we design a rigorous user study to identify the visual features of a pattern that correspond to real-world users' criteria to score the strength of the pattern. We conducted an online survey for 3,851 users to collect reliable labels for 625 patterns. The statistical result of the user study sheds light on a pattern strength meter that reflects the user's visual perception with various visual features.

Keywords: Pattern lock · Pattern strength meter · Shoulder surfing attack

1 Introduction

Android pattern lock, which is one of the authentication methods used to protect a smartphone, originates from the earlier recall-based systems such as Draw-A-Secret (DAS) [18] and Pass-Go [30]. A pattern lock user draws a pattern shape on 3 × 3 grid in a touchscreen and enrolls it. When unlocking the smartphone, the user only needs to draw the enrolled pattern. As a graphical password, pattern lock utilizes the fact that graphical information is easier to be remembered by humans than text information [5,28]. It is also preferred by users because of its good error recovery [37]. Although the recent trend is using a biometric authentication that has been developed newly, in this case, users should adopt the pattern lock or PIN as a secondary authentication method.

Android pattern lock is one of the most common authentication methods for smartphone [16,21]. It is reported in a previous study that about 40% of Android users are using the pattern lock [35]. Furthermore, in our user study, we found that 35.91% of Android smartphone users are currently using the pattern lock,

S.-H. Seo and H. Seo (Eds.): ICISC 2022, LNCS 13849, pp. 81–99, 2023.
https://doi.org/10.1007/978-3-031-29371-9_5

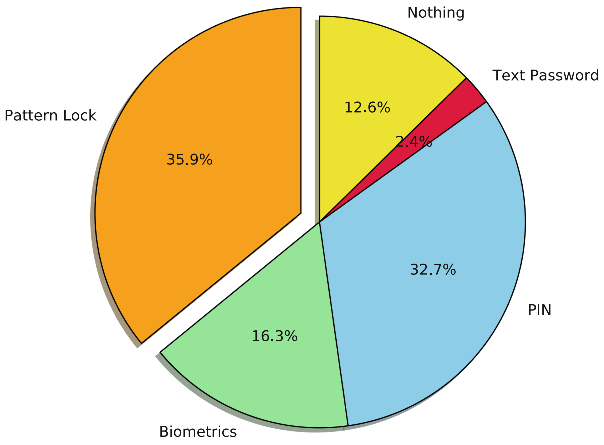


Fig. 1. Proportion of current authentication method usage of survey participants

which accounts for the most among authentication schemes (see Fig. 1). Users who have experience using the pattern lock comprised 93.35% of all participants. This indicates that Android pattern lock still influences heavily protecting users' smartphones.

Despite a number of smartphone users using Android pattern lock, a variety of security issues with the pattern lock have been raised. Theoretically, the possible number of unique patterns in Android pattern lock is 389,112. It is a tremendous amount but actual pattern usage differs from the theory. Users commonly use simple and usable but insecure patterns. This decreases the number of patterns that an attacker should consider and makes the pattern lock vulnerable to the guessing attack [27,32]. Moreover, as a simple pattern is easier to be remembered by both user and attacker, it is easily exposed to the shoulder-surfing attack [36]. Because the shoulder-surfing attack does not need any prior knowledge about the pattern lock, it is more dangerous considering that anyone near the user can perform the attack. Therefore, there needs equipment that leads users to choose more complex and secure patterns to prevent two types of attacks on the pattern lock.

From decades ago, there have been a lot of studies about password strength meters as equipment to increase the security of user's text password [13]. Against the brute-force attack [13,14], the dictionary attack [8,9,13,19,33,34,39], and the guessing attack [9], those existing works applied features such as Markov model and entropy to their meters. The text password strength meters are deployed on websites, encouraging users to choose more secure passwords [34]. Inspired by the case of the text password, there have been several studies about a pattern strength meter to prevent shoulder-surfing attack and guessing attack [2,6,27,29,32]. They extracted various features based on their own criteria, designed a metric to measure the strength of a pattern, and performed user studies to confirm the validity of their meters. They commonly concluded that pattern strength meters can help choose more secure patterns. However, all of the existing pattern strength meters have an inconsistency problem. In other words, they have the possibilities that they estimate a simple pattern that is vulnerable to attacks complex. Likewise, they might estimate a complex pattern that is robust to attacks simple. Their inconsistencies can cause a fatal defect that they can recommend a vulnerable pattern to users. Due to this reason, the existing pattern strength meters are premature to be applied to public users.

In this paper, we conduct a preliminary study toward consistent pattern strength meters. We first summarize five existing pattern strength meters [2,6,27,29,32] and identify the reason for their inconsistency problem. They commonly miss some important features and assign improper weight values to used features. Furthermore, they designed their meters from the subjective perspective of authors, not the real-world users' perspective. We claim that features relevant to users' visual perception should be applied as much as possible to the strength metering to solve the problem. In this respect, we raise a fundamental question: visual features of patterns correspond with the perception of real-world users? We perform a large-scale online survey subjected to 3,851 android users to answer the question. In this process, various feature values of patterns were measured and a clustering algorithm was applied to select 1,000 survey patterns. Through the statistical analysis of the survey result, we obtained reliable strength scores of 625 patterns among 1,000 patterns. Our study result implies that a future pattern strength meter based on abundant features and their proper weights can clearly explain how human recognizes a pattern and scores the strength of the pattern. In summary, this paper makes the following contributions:

- We raise the inconsistency problem of the existing five pattern strength meters through several pattern examples that are misestimated. We identify that the reason for their problem is the lack of used features and improper feature weights due to the subjective perspective.
- We perform a large-scale online survey for 3,851 android users (Sect. 3). Unlike previous studies, 100 of our survey participants who responded to one pattern can give the ground truth of the strength of the pattern. We also obtain reliable strength scores of 625 patterns through statistical analysis.
- Further, we discuss solutions to resolve the problems and to measure an accurate pattern strength (Sect. 4). We define requirements for a consistent pattern strength meter. We also discuss how the survey result can be utilized to construct the strength meter.

2 Pattern Strength Meters

Despite of efforts of the previous works, the existing pattern strength meters have an inconsistency in measuring the strength of a pattern. In other words, they are possible to judge a weak pattern to be strong and a strong pattern to be weak. This can lead to a serious problem in that those meters guide users to choose weak patterns. In this section, we introduce the existing five pattern strength meters [2,6,27,29,32], analyzing their inconsistency.

We first analyze an error caused by a single pattern meter. Figure 2 shows error patterns that we found the existing meters measure erroneously. In Figs. 2(a), (b), (d), and (e), the right pattern generally looks simpler than the left one for human, but each meter concludes that the right pattern is much more complex than the left one. In Figs. 2(c), the right pattern generally looks more complex than the right one, but the Sun meter estimates similar complexities for both patterns. We repeatedly sorted 389,112 patterns in ascending orders of the five meters. We then extracted the Nth strong pattern in each existing meter to identify erroneous patterns.

2.1 Existing Pattern Strength Meters

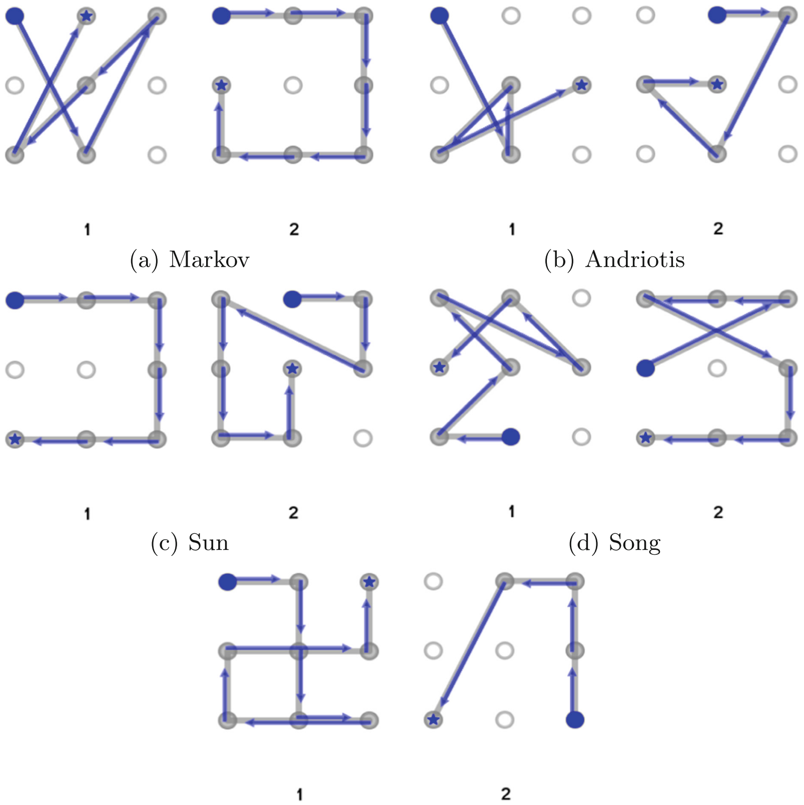
Uellenbeck Meter. Uellenbeck et al. [32] measured the security of a pattern against the guessing attack, based on the hidden Markov model. Their pattern strength metric based on an n-gram Markov model can be defined as follows.

$$P(c_1, ..., c_m) = P(c_1, ..., c_{n-1}) \cdot \prod_{i=n}^{m} P(c_i | c_{i-n+1}, ..., c_{i-1}) \tag{1}$$

In the above equation, c_n indicates a 3-gram pattern sequence token, $P(c_1, ..., c_{n-1})$ indicates an initial probability, and $P(c_n | c_1, ..., c_{n-1})$ indicates a transition probability. They collected user patterns of hundreds of participants in the user study to collect the probability of each token.

It is more real to utilize probabilities from the usage distribution of real-world users, but it is impossible to investigate the usage distribution of all 389,112 patterns and all users. We, therefore, implemented their meter, defining the probability that the current dot moves to another dot as the transition probability. From now on, we call Uellenbeck's meter *the Markov meter* in this paper.

The major error we can find from the Markov model is that the security measurement relies heavily on the number of dots in a pattern. This metric can reflect the security against the guessing attack but cannot reflect the security against the shoulder-surfing attack. In Fig. 2(a), the complexity ranking of the right pattern in the Markov meter is higher than the left one. The right pattern seems less secure than the left one because of the lack of features such as cross point, the direction of segments, and the angle of segments. However, the Markov meter does not consider those features and over-estimated the right pattern.



(a) Markov

(b) Andriotis

(c) Sun

(d) Song

(e) Bier

Fig. 2. Error patterns caused by existing single pattern strength meters. Each meter estimated the much larger strength score of the right pattern than the left one except for the Sun meter. Sun meter estimated similar strength scores for both patterns. The circle in the pattern denotes the starting point, and the asterisk in the pattern denotes the endpoint.

***Andriotis Meter*:** Andriotis et al. [2] utilized five pattern features and defined conditions for each feature to increase a security score. The score increase condition x_i is as follows. 1) x_1 is 1 if the starting point is not upper left, otherwise, it is 0. 2) x_2 is $|P| - 5$ where $|P|$ is length of the pattern if $|P| >= 6$, otherwise, it is 0. 3) x_3 is 1 if the number of turns is more than or equal to 2, otherwise, it is 0. 4) x_4 is the number of knight moves. 5) x_5 is the number of overlaps. The final pattern score θ is defined with

$$\theta = \sum_{i=1}^{5} x_i. \tag{2}$$

Figure 2(b) illustrates an example of the error of the Andriotis meter. In this figure, the ranking of the left pattern and that of the right pattern is identical in the Andriotis meter, even though the left one seems visually more secure than the right one for humans. From the perspective of the Andriotis meter, the left pattern could not get the strength score from the starting point. The left pattern also has several directions but got the additional score only by 1 because of their policy. Furthermore, they missed noticeable features of the pattern such as narrow angles and cross points. As a result, the Andriotis meter underestimated the left pattern.

***Sun Meter*:** Sun et al. [29] tried to apply a similar strength metric as for text password to pattern lock. Using several pattern characteristics, they transformed the traditional entropy equation to

$$PS_p = S_p \times log_2(L_p + I_p + O_p) \tag{3}$$

In this metric, S_p, L_p, I_p, and O_p indicate the number of points, the sum of segments' euclidean distance, the number of cross points, and the number of overlaps, respectively.

There is concern that the Sun meter does not consider the direction and angle of segments so patterns with those features can get low strength scores. In addition, the number of points has a great influence since it is applied to true value while other features are reduced with a log scale. The same weights for the other three features can make long patterns get a high strength score easily. In Fig. 2(c), the ranking of the left pattern and the right pattern is similar in the Sun meter. The left pattern does not have noticeable features and seems simple. Sun meter, however, overestimated the number of points and the length of the pattern.

***Song's Meter*:** Song et al. [27] designed a function for pattern strength meter, which is combined from three pattern features considering both guessing attack and shoulder-surfing attack.

$$M_P = 0.81 \times \frac{L_P}{15} + 0.04 \times N_P + 0.15 \times \frac{min(I_P, 5)}{5} \tag{4}$$

They extracted a feature that had not been extracted by other existing approaches before. L_P is sum of segments' vertical and horizontal length, N_P is the ratio of non-repeated sub-patterns, and I_P is the number of intersections. The repetition of the same segments makes a pattern seem simple to users and increases guessability. The weights of the three features were initialized to 0.33 in common. They updated their weights as the above equation through a user study.

They assigned a too-large weight to the pattern length but a small weight to the sub-pattern feature. In Fig. 2(d), the ranking of the left pattern is lower than the right pattern in Song meter, while the left pattern seems more complex for humans. Song meter over-estimated the right pattern since they assigned a large weight to euclidean distance. They also underestimated the left pattern and reduced its ranking improperly since they missed narrow angles of segments of the pattern.

***Bier's Meter*:** Bier et al. [6] concentrated on the directional feature of segments. Their pattern strength metric is as below

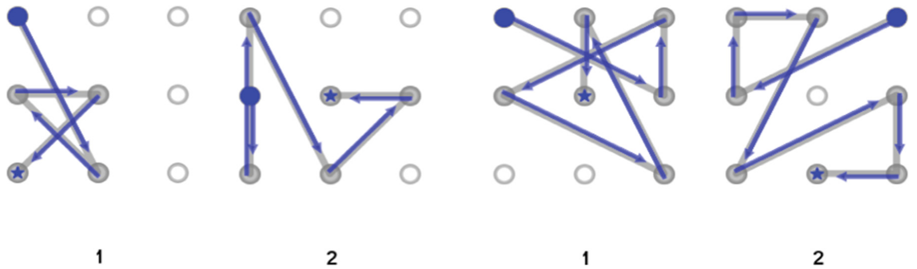
$$m(P_0, d_1 d_2 ... d_k) = (1 - p(P_0))(1 - \alpha^k)\frac{1}{3k}\sum_{i=1}^{s(k)} w(d_i). \tag{5}$$

Given k segments, d_i indicates ith segment. $p(P_0)$, α, $w(d_i)$ indicate weights for starting point, the sensitivity of the number of points, and ith segment, respectively. They assigned larger weights for diagonal segments than vertical and horizontal segments.

Bier meter is missing important features such as euclidean distance, cross points, overlap, and angle of segments. In Fig. 2(e), the ranking of the left pattern is lower than the right pattern in the Bier meter. The left pattern was underestimated even though its many turns and overlap increase the complexity. Bier meter also over-estimated knight move of the right pattern, increasing its ranking unnecessarily.

2.2 Common Problem of Pattern Strength Meters

We additionally found the common error cases from the five existing meters. We could identify their common problem from those cases. Figure 3 depicts two patterns that the five meters commonly under-estimated or over-estimated. In Fig. 3(a), compared with the right pattern, the left pattern seems much more complex. This example represents that the existing meters commonly overlook the angle and density of the left pattern. Meanwhile, in Fig. 3(b) illustrates an opposite example. In this figure, compared with the left pattern, the right pattern seems much simpler. We can conclude that the existing meters overly concentrated on pattern features such as the length, and the number of points.



(a) Under estimated pattern (left)

(b) Over estimated pattern (right)

Fig. 3. Common error patterns caused by five existing pattern strength meters. All of the existing meters estimated the smaller strength score of the left pattern of (a) than the right one. In addition, they estimated the larger strength score of the right pattern of (b) than the left one.

3 Our Study

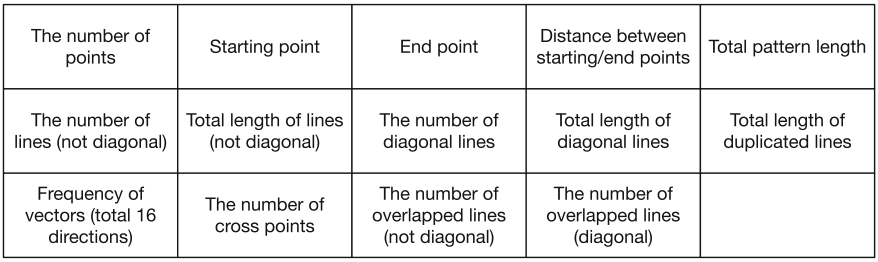
In this section, we perform a user study based on a survey to identify whether the strength scores of patterns measured by visual features correspond with those of real-world users. In the following subsections, we explain how we designed, performed, and analyzed the survey and also explain how we created the pattern strength scores.

3.1 Survey Pattern Selection

The ideal approach is collecting labels of patterns as many as possible from real-world users. However, it is impossible to ask for all of 389,112 patterns to users. We need to choose a part of those patterns to be included in the training dataset. The chosen patterns should be able to represent other patterns and create objective data. If patterns have more points and become more complex, people may not be able to answer their accurate strengths. For this reason, we use patterns whose number of points does not exceed six for this user study. We found that 34,792 patterns satisfy this criterion. We still have too many patterns to be considered so we grouped similar patterns among them into clusters.

We utilized scikit-learn [24], the Python-based open source machine learning library, for pattern clustering. We used kmeans++ among the available algorithms. We used 29 visual features, which are extractable from a pattern itself, in Fig. 4 for clustering. Each feature has a different scale so feature values are normalized from 0 to 1 by the min/max scaler. Intersections make lines of a pattern more densely such that the pattern gets more complex. Therefore, we increased the weight of the intersection ten times because we thought that intersections have significant importance to pattern strength.

We chose representative patterns (i.e. centroids), that will be displayed to respondents in the survey, from 1,000 clusters. It is difficult for a respondent to answer all 1,000 patterns, so we need to make them answer for the proper num-



The number of points	Starting point	End point	Distance between starting/end points	Total pattern length
The number of lines (not diagonal)	Total length of lines (not diagonal)	The number of diagonal lines	Total length of diagonal lines	Total length of duplicated lines
Frequency of vectors (total 16 directions)	The number of cross points	The number of overlapped lines (not diagonal)	The number of overlapped lines (diagonal)	

Fig. 4. A total of 29 visual features used for pattern clustering. For the frequency of vectors (left bottom in the figure), each direction of vectors is an independent feature.

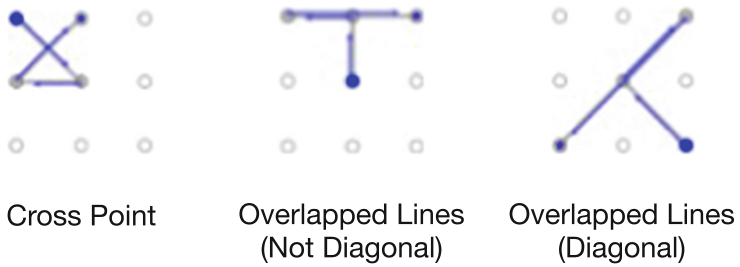


Fig. 5. Example patterns that describe an intersection (i.e., cross point) and overlapped lines

ber of patterns to obtain an objective answer. Therefore, we created 40 survey groups, limiting the number of patterns a respondent can answer to 25. There are simple (i.e. weak) patterns and complex (i.e. secure) patterns among the chosen patterns. For respondents to answer from a weak pattern to a secure pattern, we created five temporary pattern complexity groups and let the respondents answer for all complexity groups. The respondents are asked to answer five patterns for each complexity group.

We sorted 1,000 patterns based on our own criteria to determine which complexity group they belong to. We considered that any features that have a large value significantly affect the pattern strength, so we sorted the patterns in ascending order of feature values. The order of the priority of features is the number of unique directions, the sum of intersections and overlaps, the number of points, intersections, overlaps, and the total euclidean length. Patterns are firstly sorted in ascending order of the number of directions. When two patterns have the same value, the pattern with the smaller sum of intersections and overlaps, which is the next priority feature, is considered simpler than the other. We divided the sorted patterns into five groups of the same size, then assigned the first 200 patterns into complexity group 1 and the last 200 patterns into complexity group 5.

3.2 Survey Design

We have 40 survey groups through the result of clustering. 25 patterns are assigned to each survey group. We set 100 respondents for each survey group and we planned to recruit a total of 4,000 respondents. We expected that 100 samples of a pattern are enough to derive the ground-truth of the strength score. The respondents were limited to Android smartphone users located in the United States. We used Amazon Mechanical Turk where we can accommodate a lot of participants to request our surveys. The questionnaire design for each survey group is identical to each other but the only difference is in the shape of the patterns. To display the pattern shape, we printed a point sequence of a pattern as an image. One questionnaire contains two main survey sections. Both survey

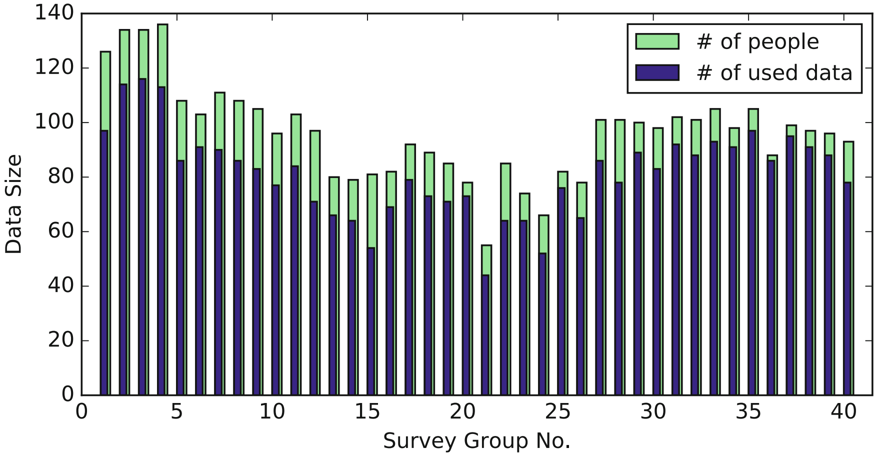


Fig. 6. The number of participants and used data size for each survey group of user study

sections are used to calculate the label of a pattern. The organization of a survey is as follows.

Survey Section 1 consists of five questions. Each question is related to one complexity group and shows five patterns in the group. Respondents should watch the patterns and answer the complexity ranking of the patterns. The ranking of patterns in one question must be different from each other. In this survey section, we want to identify the detailed differences in scores among patterns in the same complexity group.

Survey Section 2 consists of 25 questions. Each question shows a pattern and respondents should answer the objective complexity score, ranging from one to five, of the pattern. Score one means the pattern is the weakest, and score five means the pattern is the most secure. In this section, five questions are assigned for each complexity group. We want to identify the objective score in this section, that is not related to complexity groups. It is possible for the same respondent to make bias by answering the same pattern in both sections. Therefore, we deployed Section 1 patterns of the even survey group in Section 2 of the odd survey group. In the same way, we deployed Section 1 patterns of the odd survey group in Section 2 of the even survey group.

A total of 3,851 respondents were recruited as the result of 40 survey groups. A lot of respondents participated in the initial phase of survey groups, but their participation became slow such that some groups could not recruit over 100 respondents. There were a variety of the age of respondents, ranging from teenagers to 60s, and their education level. The survey group that recruited the most respondents had 136 respondents, and the group that recruited the least respondents had 55 respondents. On average, each survey group recruited 96.275 respondents.

We did not use all of the respondent's data. As the label of training patterns must be measured by the reliable labeling method, we used only the reliable ones among all data. We regarded the data of respondents who gave an answer that makes no sense as noise and rejected them. For instance, some respondents answered the most complex pattern as the simplest, and vice versa. We also rejected the data of randomly answered respondents. The number of respondents and used samples are illustrated in Fig. 6. The number of total used samples for labeling was 3,257 over 40 survey groups. The survey group with the most samples had 116 samples, and the group with the least samples had 44 samples. On average, each survey group had 81.425 samples.

3.3 Strength Score Measurement

In this step, we measure the strength score of the patterns used in the survey. Although we utilize both survey sections to determine the strength of a pattern, they have different purposes and structures. In this respect, we obtain scores of a pattern using different methods for each survey section and then combine two scores. Survey Sect. 1 results in the relative score of a pattern compared to the other four patterns in the same complexity group. For the conversion from a relative score to the absolute value, a score range of five complexity groups should be defined. Therefore, we first analyze the result of survey Sect. 2 to define their range. Fortunately, we identified that there is a statistical difference among pattern complexity groups so we can consider those groups are separated. However, we cannot assure that the grouping result is definitely objective. In the real world, the most complex pattern in the Nth group may be more complex than the simplest one in the N+1th group. For this reason, we permit overlap of the range of two complexity groups to some degree.

We calculated the objective score of Sect. 2 by averaging the responses of all respondents who answered in the same pattern. The score range of each complexity group is determined by the minimum/maximum Sect. 2 scores in the group. Same as Sect. 2, we calculated the Sect. 1 score by averaging all responses of a pattern. The scale of the Sect. 1 score changes when the relative Sect. 1 score is converted to the objective score. Given a pattern P, its complexity group G, its Sect. 2 score S^2, and its relative Sect. 1 score S^1, the equation to obtain the objective Sect. 1 score $S_P^{1'}$ is defined as

$$S_P^{1'} = \frac{(max(S_G^2) - min(S_G^2))}{4} \times (S_p^1 - 1) + min(S_G^2). \tag{6}$$

$mim(S_G^2)$ and $max(S_G^2)$ means the minimum and maximum score of G. $\frac{(max(S_G^2)-min(S_G^2))}{4}$ means the interval between two adjacent objective scores. The relative score 1 of Sect. 1 is converted to the objective score $min(S_G^2)$. The relative score 5 of Sect. 1 is converted to $max(S_G^2)$. The relative score 2, 3, and 4 is converted to objective scores based on the score interval of G. In conclusion, the equation of combining Sect. 1 score $S_P^{1'}$ and Sect. 2 score S_P^2 to measure the

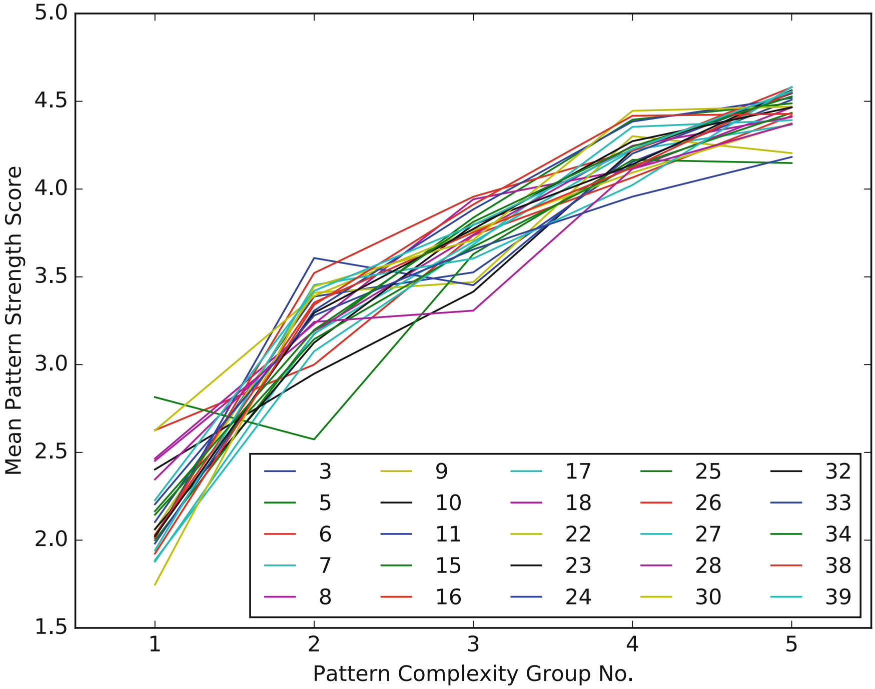


Fig. 7. Mean pattern strength score of pattern complexity groups of user study. Each color of line describes a survey group number.

pattern strength label L_p of a pattern P is defined as

$$L_P = \sqrt{(S_P^{1'})^2 + (S_P^2)^2}. \tag{7}$$

3.4 Survey Results

We make sure that the five complexity groups over 40 independent survey groups match people's perspectives and that the difference in scores between the complexity groups is significant. If our grouping contains an error, it leads to an error in the design of the questionnaire and the survey results become difficult to analyze. We confirmed this by conducting a statistical analysis based on Mixed Factorial ANOVA. We eliminated 15 survey groups during the statistical test, we only labeled the strength scores of patterns in the remaining 25 survey groups.

The normality which is the basic assumption of ANOVA analysis was established with more than 30 survey groups. The homogeneity which is another basic assumption of ANOVA analysis wasn't established since the sample sizes of some surveys were too small or too large to satisfy homogeneity of variance. The 1st, 2nd, 12th, 13th, 14th, 21st, 29th, 31st, and 35th survey groups made this problem, so we excluded those survey groups to satisfy homogeneity of variance.

Table 1. Within-Subjects effect test

Source	Type3 sum of square	Degree of freedom	Mean square	F value	P value
Pattern group	6348.427	3.681	1724.597	4661.218	.000
Pattern*Survey group	276.939	88.347	3.129	8.456	.000
Error (Pattern group)	2699.420	7295.954	.370		

Table 2. Within-Subjects contrast test

Source	Type3 sum of square	Degree of freedom	Mean square	F value	P value
Pattern group	5917.302	1	5917.302	12680.546	.000
Pattern *Survey group	71.927	24	2.997	6.422	.000
Error (Pattern group)	924.889	2982	.467		

Levene's test of equality of error variance showed no difference in all pattern groups based on median (p-value > 0.05). Meanwhile, as there was a significant mean difference among the remaining 31 survey groups (p-value < 0.025), we identified that the 4th, 19th, 20th, 36th, 37th, and 40th survey groups had a large difference in mean among groups by conducting LSD post-analysis that is sensitive to the average difference. We removed the results of these survey groups and made the remaining survey groups have no mean difference (p-value > 0.05).

Mixed Factorial ANOVA analysis suggested that the pattern complexity group did not satisfy the sphericity which is the basic assumption of Mixed Factorial ANOVA (p-value < 0.05). Nevertheless, we assumed that the sphericity was ensured because the Greenhouse-Geissser value was close to 1. Table 1 showed that the mean difference among pattern complexity groups was significant (p-value > 0.05). Also, from Table 2, which is the result of the contrast test, we found that the difference was significant in linear models (p-value < 0.05). As shown in Fig. 7, the complexity of the pattern group is upward. Therefore, through the survey result, we identified survey groups that have no difference from other groups and confirmed that there is a linear upward difference among pattern complexity groups. In addition, we also found that the pattern complexity of those patterns measured by our approach follows the visual perception of real-world users.

4 Discussion

Through the analysis of error cases of the existing pattern strength meters identified in Sect. 2, we define their two main problems. First, each of them is missing at least one visual feature which affects the safety of a pattern. The strength of text passwords can be represented by simple features. On the other hand, the strength of patterns must be represented with more complex visual features. We showed that, in Sect. 2, incorporating used and missing features of existing meters can reenact the criteria of real-world users to evaluate patterns. For more accurate metering, we can consider further features such as Markov model [32], repeating sub-patterns [27], or the angle of two lines.

The second problem of existing pattern strength meters is that they assigned wrong weights to their features due to the intervention of the author's subjective perspective. Song et al. [27] adopted a machine learning model, but its weights of features were initialized by the author. Even though the strength of our survey patterns was accurately derived, we cannot manually measure the strength of all existing patterns. Therefore, we suggest the strength of a pattern should be measured by the machine learning model alone rather than by applying someone's opinion to assign accurate weights for various features. Deep learning is a promising solution to extract latent features. DNN consists of layers with neurons. Each neuron of different layers are connected by weights and biases (i.e., parameters). The topology of DNN can be designed freely. A sophisticated DNN can solve a difficult problem such as a non-linear problem. We believe that DNN can extract the latent features from the perspective of a human.

Meanwhile, the machine learning model requires a ground-truth for training. The strength scores of 625 survey patterns are a reliable ground-truth because they were measured by multiple users and evaluated by the statistical test. If we deploy a regression model, the model learns the appropriate weights of features from the label of survey patterns. The model then calculates the strength of the remaining patterns with feature values of the patterns and weight parameters of the model.

5 Related Work

5.1 Security of Android Pattern Lock

Android pattern lock has a security issue in that users prefer to use only a few pattern spaces to draw actual patterns within the theoretical limits of pattern space [2,32]. There is a trade-off between security and usability according to the complexity of lock patterns [29]. However, users tend to select the simple pattern which is easily stolen and replicated for usability rather than security. Various types of attacks targeted to android pattern lock have been proposed, such as guessing attacks [3,10,27], shoulder surfing attacks [22,31], a smudge attacks [4], a video-based attack [41,42], and a thermal attack [1]. Such attacks have a common ground that they are performed via a leakage of pattern shapes [25], where simple patterns are more vulnerable to those attacks. In this study, we focused on guessing attack and shoulder-surfing attack which are more feasible in real-world.

Some previous works proposed some modifications of existing schemes including pattern lock to prevent the leakage of graphical passwords [11,17,38,43]. They focused on an increment of resistance against only one specific attack. However, they could not deal with other attacks they do not consider while more than two attacks that target the Android pattern lock can coexist in the real world. Moreover, in general, they could not guarantee a significant improvement in security or they reduced the usability of their schemes.

As the essential motivation of attacks on graphical authentication is to crack the shape of private passwords, the behavior-based authentication leveraged not only private passwords but also user behavior collected by embedded sensors in a smartphone to prevent those attacks [7,12,15,23,26,40]. Especially, Ku et al. [20] applied the behavioral approach to android pattern lock. They turned a private pattern into a public one by displaying the pattern to multiple users. They used only user's behavior information to distinguish users. As a consequence, they could remove existing threats on the traditional android pattern lock. However, this system is still hard to be accepted by public users who are firmly using android pattern lock which.

5.2 Password Strength Meter

One of the methods to offer a secure password authentication system for users is maintaining the current scheme and recommending for them to use secure passwords [19]. Text password policies had been studied to create passwords that are robust against guessing attack. Policies were created based on LUDS formulation that counts lower and uppercase letters, digits, and symbols while the policies depend on different websites using passwords as an authentication method [19,39]. However, the LUDS formulation had problems of usability and ineffectiveness against guessing attack [39].

To resolve this problem, studies about password meters have begun [34]. Ur et al. [33] implemented a meter that scores a password by combining various heuristics related to a neural network and created data-based text feedback. Castelluccia et al. [9] implemented an adaptive password strength meter (APSM) that estimates password strength using the Markov model. It was accurate on the guessability of a password and robustness against other attack models. Some studies proved that password meters are helpful for password creation [14,33]. Users who utilize a password meter create longer passwords than those who do not utilize the meter, and passwords created with help of the meter displaying a visual bar are slower to be cracked than those without the meter [14,34].

5.3 Pattern Strength Meter

There are five previous studies that are most relevant to our work [2,6,27,29,32]. They developed pattern strength meters that improve the security of android pattern lock by assessing the strength score of a pattern and encouraging users to use secure patterns. Uellenbeck et al. [32] utilized the Markov model to measure the guessability of a pattern. Although they did not consider shoulder-surfing attack, we included Markov probability in our feature set because we consider guessing attack as well as shoulder-surfing attack. The other four studies focused on security against shoulder-surfing attack. They established their metrics to calculate the visual complexity of a pattern. Various numerical features such as starting point, length, directions, cross points, and overlaps were included in their metrics. They have two main problems that cause inconsistency in pattern metering. First, they included only few features in their metrics so they could not fully reflect the user's visual perception. As a solution for the first problem, we combined most of their features into our feature set and also included new features (i.e., angles). Second, except Song et al. [27], they assigned the wrong weights to their features because of their subjectivity. Song et al. [27] initialized feature weights and updated their weights by regression. However, they collected a label of a pattern from a limited number of users. As a result, their labels could not represent the ground-truth and they assigned wrong weights to features as well. We collected reliable strength scores of patterns, from large-scale user survey, that can also be used for training a further strength meter as the labels.

6 Conclusion

As smartphone contains users' private data more than before and android pattern lock becomes a target of various attacks, the need for novel equipment that protects the smartphone from those attacks is continuously increasing. We proposed a novel pattern strength meter that reflects the user's visual perception, overcomes the inconsistency problem of existing pattern strength meters and eventually encourages users to create more secure patterns. Based on various visual features of a pattern, the proposed pattern strength meter can score the accurate robustness of a pattern against a guessing attack and a shoulder-surfing attack. We performed a large-scale online survey of android users. From the survey, we could obtain the ground-truth of the user's perspective about a pattern and identified that complexities of patterns measured by our features follow the visual perception of real-world users. We are considering future work on the pattern strength meter with some improvements toward the ground-truth of the strengths of all patterns.

Acknowledgements. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1A2C1088802).

References

1. Abdelrahman, Y., Khamis, M., Schneegass, S., Alt, F.: Stay Cool! Understanding thermal attacks on mobile-based user authentication. In: Conference on Human Factors in Computing Systems, pp. 3751–3763. CHI 2017, ACM, USA (2017). https://doi.org/10.1145/3025453.3025461. http://doi.acm.org/10.1145/3025453.3025461
2. Andriotis, P., Tryfonas, T., Oikonomou, G.: Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method. In: Tryfonas, T., Askoxylakis, I. (eds.) HAS 2014. LNCS, vol. 8533, pp. 115–126. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07620-1_11
3. Aviv, A.J., Budzitowski, D., Kuber, R.: Is Bigger better? comparing user-generated passwords on 3x3 vs. 4x4 grid sizes for android's pattern unlock. In: Annual Computer Security Applications Conference, pp. 301–310. ACSAC 2015. ACM (2015)
4. Aviv, A.J., Gibson, K., Mossop, E., Blaze, M., Smith, J.M.: Smudge attacks on smartphone touch screens. In: Workshop on Offensive Technologies, pp. 1–7. WOOT 2010. USENIX (2010)
5. Biddle, R., Chiasson, S., Van Oorschot, P.C.: Graphical passwords: learning from the first twelve years. ACM Comput. Surv. (CSUR) **44**(4), 19 (2012)
6. Bier, A., Kapczyński, A., Sroczyński, Z.: Pattern lock evaluation framework for mobile devices: human perception of the pattern strength measure. In: Gruca, A., Czachórski, T., Harezlak, K., Kozielski, S., Piotrowska, A. (eds.) ICMMI 2017. AISC, vol. 659, pp. 33–42. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67792-7_4
7. Buriro, A., Crispo, B., DelFrari, F., Wrona, K.: Hold and sign: a novel behavioral biometrics for smartphone user authentication. In: the Security and Privacy Workshops, pp. 276–285. SPW 2016. IEEE (2016)
8. Burr, W., Dodson, D., Polk, W.: Electronic authentication guideline. Tech. rep, National Institute of Standards and Technology (2004)
9. Castelluccia, C., Dürmuth, M., Perito, D.: Adaptive password-strength meters from Markov models. In: NDSS (2012)
10. Cha, S., Kwag, S., Kim, H., Huh, J.H.: Boosting the guessing attack performance on android lock patterns with smudge attacks. In: Asia Conference on Computer and Communications Security, pp. 313–326. AsiaCCS 2017. ACM (2017)
11. Cho, G., Huh, J.H., Cho, J., Oh, S., Song, Y., Kim, H.: SysPal: system-guided pattern locks for android. In: Symposium on Security and Privacy, pp. 338–356. S & P 2017. IEEE (2017)
12. Crawford, H., Ahmadzadeh, E.: Authentication on the go: assessing the effect of movement on mobile device keystroke dynamics. In: Symposium on Usable Privacy and Security, pp. 163–173. SOUPS 2017. USENIX (2017)
13. De Carnavalet, X.D.C., Mannan, M., et al.: From very weak to very strong: analyzing password-strength meters. In: NDSS, vol.14, pp. 23–26 (2014)
14. Egelman, S., Sotirakopoulos, A., Muslukhov, I., Beznosov, K., Herley, C.: Does my password go up to eleven?: the impact of password meters on password selection. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2379–2388. ACM (2013)
15. Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. IEEE Trans. Inf. Forensics Secur. **8**(1), 136–148 (2013)
16. Harbach, M., De Luca, A., Egelman, S.: The anatomy of smartphone unlocking: a field study of android lock screens. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 4806–4817. ACM (2016)

17. Higashikawa, S., Kosugi, T., Kitajima, S., Mambo, M.: Shoulder-surfing resistant authentication using pass pattern of pattern lock. IEICE Trans. Inf. Syst. **101**(1), 45–52 (2018)
18. Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K., Rubin, A.D.: The design and analysis of graphical passwords. In: 8th Usenix Security Symposium. USENIX (1999)
19. Komanduri, S., Shay, R., Cranor, L.F., Herley, C., Schechter, S.: Telepathwords: preventing weak passwords by reading users' minds. In: 23rd {USENIX} Security Symposium ({USENIX} Security 14), pp. 591–606 (2014)
20. Ku, Y., Park, L.H., Shin, S., Kwon, T.: Draw it as shown: behavioral pattern lock for mobile user authentication. IEEE Access **7**, 69363–69378 (2019)
21. Kunda, D., Chishimba, M.: A survey of android mobile phone authentication schemes. Mobile Netw. Appl. **26**, 2558–2566 (2018)
22. Lashkari, A.H., Farmand, S., Zakaria, D., Bin, O., Saleh, D., et al.: shoulder surfing attack in graphical password authentication. arXiv preprint arXiv:0912.0951 (2009)
23. Li, L., Zhao, X., Xue, G.: Unobservable re-authentication for smartphones. In: Network and Distributed System Security Symposium, NDSS 2013 (2013)
24. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
25. Rao, V.V., Chakravarthy, A.: Analysis and bypassing of pattern lock in android smartphone. In: 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–3. IEEE (2016)
26. Sitová, Z., et al.: HMOG: new behavioral biometric features for continuous authentication of smartphone users. IEEE Trans. Inf. Forensics Secur. **11**(5), 877–892 (2016)
27. Song, Y., Cho, G., Oh, S., Kim, H., Huh, J.H.: 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, pp. 2343–2352. ACM (2015)
28. Standing, L., Conezio, J., Haber, R.N.: Perception and memory for pictures: single-trial learning of 2500 visual stimuli. Psychonomic Sci. **19**(2), 73–74 (1970)
29. Sun, C., Wang, Y., Zheng, J.: Dissecting pattern unlock: the effect of pattern strength meter on pattern selection. J. Inf. Secur. Appl. **19**(4–5), 308–320 (2014)
30. Tao, H., Adams, C.: Pass-go: a proposal to improve the usability of graphical passwords. IJ Netw. Secur. **7**(2), 273–292 (2008)
31. Tari, F., Ozok, A., Holden, S.H.: A Comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In: Symposium on Usable Privacy and Security, pp. 56–66. SOUPS 2006. ACM (2006)
32. Uellenbeck, S., Dürmuth, M., Wolf, C., Holz, T.: Quantifying the security of graphical passwords: the case of android unlock patterns. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 161–172. ACM (2013)
33. Ur, B., et al.: Design and evaluation of a data-driven password meter. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 3775–3786. ACM (2017)
34. Ur, B., et al.: How does your password measure up? the effect of strength meters on password creation. In: Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12), pp. 65–80 (2012)
35. Van Bruggen, D.: Studying the impact of security awareness efforts on user behavior, Ph. D. thesis, University of Notre Dame (2014)

36. Von Zezschwitz, E., De Luca, A., Janssen, P., Hussmann, H.: Easy to draw, but hard to trace?: on the observability of grid-based (un) lock patterns. In: Conference on Human Factors in Computing Systems, pp. 2339–2342. CHI 2015. ACM (2015)
37. Von Zezschwitz, E., Dunphy, P., De Luca, A.: 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, pp. 261–270. ACM (2013)
38. Von Zezschwitz, E., Koslow, A., De Luca, A., Hussmann, H.: Making Graphic-based Authentication Secure against Smudge Attacks. In: Proceedings International Conference on Intelligent User Interfaces, pp. 277–286. IUI 2013. ACM (2013)
39. Wheeler, D.L.: zxcvbn: Low-budget password strength estimation. In: 25th {USENIX} Security Symposium ({USENIX} Security 16), pp. 157–173 (2016)
40. Xu, H., Zhou, Y., Lyu, M.R.: Towards continuous and passive authentication via touch biometrics: an experimental study on smartphones. In: Symposium on Usable Privacy and Security. SOUPS 2014, vol. 14, pp. 187–198 (2014)
41. Ye, G., et al.: Cracking android pattern lock in five attempts. In: Network and Distributed System Security Symposium, NDSS 2017 (2017)
42. Ye, G., et al.: A video-based attack for android pattern lock. ACM Trans. Privacy Secur. (TOPS) **21**(4), 19 (2018)
43. Zakaria, N.H., Griffiths, D., Brostoff, S., Yan, J.: Shoulder surfing defence for recall-based graphical passwords. In: Symposium on Usable Privacy and Security, p. 6. SOUPS 2011. ACM (2011)