Fortifying Android Patterns using Persuasive Security Framework

Hossein Siadati
New York University
New York, Brooklyn
email: hossein@nyu.edu

Payas Gupta
New York University Abu Dhabi
Abu Dhabi
email: payasgupta@nyu.edu

Sarah Smith, Nasir Memon
New York University
New York, Brooklyn
email: sesmith325@gmail.com
nasir@nyu.edu

Mustaque Ahamad
Georgia Institute of Technology
New York University Abu Dhabi
Atlanta, Georgia
email: mustaq@cc.gatech.edu

***Abstract*—Android Pattern, form of graphical passwords used on Android smartphones, is widely adopted by users. In theory, Android Pattern is more secure than a 5-digit PIN scheme. Users' graphical passwords, however, are known to be very skewed. They often include predictable shapes (e.g., Z and N), biases in selection of starting point, and predictable sequences of the points that make them easy to guess. In practice, this *decreases* the security of Android Pattern to that of a 3-digit PIN scheme for at least *half* of the users. In this paper, we effectively *increase* the strength of Android Patterns by using a persuasive security framework, a set of principles to get users to behave more securely. Using these principles, we have designed two user interfaces that *persuade* users to choose *stronger* patterns. One of the user interfaces is called BLINK, where the starting point of the pattern is suggested to user, effectively *nudging* her to create a pattern with a significantly *less predictable* starting point. The other user interface is called EPSM, where the system gives *continuous* feedback to user while she is creating a new pattern, effectively persuading her to create a complex pattern. Security and usability of our proposed designs evaluated by conducting a user study on 270 participants recruited from Amazon MTurk demonstrated that while only 49% of subjects choose *strong* patterns in Android Pattern user interface, our suggested designs increase it to 60% in BLINK and 77% in EPSM version.**

Keywords–Android; nudging; persuasive security; blinking.

I

I. INTRODUCTION

The rising trend of smartphones in our daily lives and the amount of personal information being carried on these devices call for stronger authentication measures than ever. Smartphones are used to perform sensitive personal and financial tasks including online banking, messaging, and used as a two-factor authentication. PINs have been the traditional way of locking a phone and securing critical data on it [1]. However, *Android Pattern*, a graphical password scheme, has seen a tremendous increase in adoption due to its perceived user-friendliness [2]. According to a recent study, 40% of Android users are using Android Pattern to unlock their devices instead of a PIN [3].

A pattern can be denoted by a sequence of numbers indicating the position of points on the screen (see Figure 1). In the Android Pattern scheme, enrollment and verification works as in a typical password-based user authentication, where a user chooses a secret (i.e., a pattern) in an enrollment phase and recalls it at the time of verification. Whereas the *theoretical password-space* of Android Pattern is larger than that of a 5-digit PIN scheme, Uellenbeck et al. [4] demonstrated biases in starting points (i.e., some points are more frequently chosen than others) and n-grams (i.e., frequent subsequences of patterns) that make user patterns guessable. Based on these findings, they were able to guess about 50% of the patterns with only 1000 guesses. In other words, the *effective password-space* of Android Pattern is equivalent to that of just a 3-digit PIN scheme for 50% of Android users! The problem is that users do not appear to effectively use the large password-space of Android Pattern.

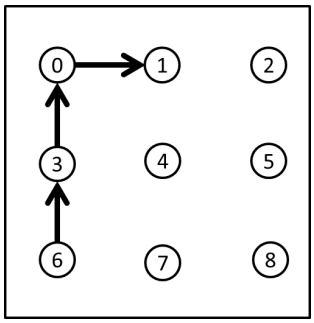


Figure 1. Path from point o_6 to point o_1. $o_6 - o_3 - o_0 - o_1$ indicates that the pattern is started from o_6, then moved to o_3, then to o_0 and then finally ended at o_1.

Several intuitive solutions appear promising but unfortunately fail to address the problem. For example, with *blacklisting*, the authentication system forbids frequently-chosen patterns, but this only shifts the distribution to a new set of frequently-chosen patterns, and does not hinder a resourceful attacker. With *random assignment*, the authentication system chooses a random pattern for the user, but this comes with a significant cost on usability and memorability. With *rearrangement*, Uellenbeck et al. [4] removed frequently-chosen starting points and rearranged all points, but found that this approach by itself does not expand the *effective* password-space. With *user education*, users are taught the differences between weak and strong passwords so that they may prefer the latter over the former, but we found in a survey (described later in this paper) that users are already aware of these differences, and yet this awareness does not translate into choosing stronger passwords.

In this paper, rather than overwhelming users with instructions and cumbersome security measures, or forcing them to choose certain patterns, we use a *persuasive security authentication framework* to *nudge* or *persuade* users to behave more securely [5].

We present two persuasive mechanisms that nudge users to choose *strong* patterns, thereby expanding the *effective password-space*, and reducing the advantage the adversary may have from a priori knowledge of pattern distribution.

 ISBN: 978-1-61208-418-3

Specifically, these user interfaces are:

1) Embedded Pattern Strength Meter (EPSM): A mechanism that provides realtime visual feedback based on the pattern's strength while it is being drawn.
2) BLINK: A mechanism that provides recommendations and nudges users at appropriate points — by blinking — to eliminate the problem of starting point bias, and to persuade users to create stronger patterns.

In summary, this paper makes the following research contributions:

- We show that EPSM helps users to create stronger, more complex patterns compared to Android Pattern. EPSM dramatically reduces the success of a guessing attack: a hypothetical attacker is able to guess 50% of Android patterns by only 1000 guesses, whereas the same attacker is able to guess only 22.6% of the patterns in EPSM.
- With BLINK, we show that we can eliminate the starting point bias. Consequently, the probability that a point is chosen as a starting point is diffused across all points in the unlock pattern grid. This makes patterns stronger against guessing attacks (40% of the patterns can be guess by 1000 guesses, 10% less than the Android patterns).
- To derive these results, we tested BLINK and EPSM with 270 participants recruited from the Amazon Mechanical Turk. We show that BLINK and EPSM improve the security of Android Pattern with only a negligible usability and memorability impact.

The rest of the paper is organized as follows. In Section II, we discuss background details on state-of-the-art approaches for graphical passwords and persuasive authentication. We also provide details from the past work which may seem plausible options at first, however, may not help to improve the security of Android Pattern. In Section III, we provide our persuasive security mechanism design choices. In Section IV, we present experimental details of our user study followed by results in Section V. We conclude in Section VI.

II. BACKGROUND

The first type of grid-based graphical passwords called "Draw a Secret" (DAS) was proposed by Jermyn et al. [6]. In DAS, user creates a password by drawing a pattern that connects cells of a grid on a screen. Followup works have proposed variations of DAS to improve on its security and usability. Most notably, Tao et al. introduced Pass-Go [7] that uses intersections of the cell in a grid (instead of the cells) and improved its usability. Android Pattern is a type of Pass-Go system and is widely adopted by Android users [3].

In this section, we briefly describe the problem of *bias* in users-choices of patterns, and its effect on the security of Android users (see Section II-A). We also list the previous efforts taken to fortify the security of Android Pattern and describe why they have insignificant effect (see Section II-B). Thereafter, we describe how the *persuasive security framework* can help to address the problem (see Section II-C). Finally, we show how to calculate the strength of patterns using Markov model (see Section II-D).

A. Biases of user-chosen patterns

Thorpe et al. [8][9] demonstrated the limitations of user selected patterns and effective password space of DAS by analyzing the memorable space of graphical passwords where patterns are partially or completely symmetric. They conclude that the *effective password-space* of DAS is much smaller than *theoretical password-space*. Andriotis et al. [10] have studied the biases of patterns chosen by users. They have found that 50% of the users choose the top-left point as the starting point of their patterns. Uellenbeck et al. [4] analyzed the bias of choosing the sequence of the points in their patterns as well. Exploiting these biases, they have estimated that 50% of users choose a pattern weaker than that of a 3-digit PIN.

B. Efforts to fortify the patterns

Previous efforts to increase the security of grid-based graphical passwords can be categorized into two different classes. The first class is focused on increasing the theoretical password-space (and implicitly increasing the effective password space used by users), either by increasing the size of the grid or introducing new degrees of freedom such as rotation and layering to the user interface. The second class is focused on developing approaches that explicitly expand the effective password space used by user. In this section, we enumerate significant works in both classes aimed to improve the security of free-form graphical passwords.

1) Background: Dunphy et al. [11] suggested "Background Draw-a-Secret" (BDAS) to improve the security of patterns by adding a background to the grid of the DAS scheme. Using a usability test, authors showed an increase in the length of patterns [12]. However, Gao et al. [13] and Zhao et al. [14] demonstrated the ease of guessing patterns based on the detectable hot-spots in the background images in the *Window 8 graphical password*, an approach similar to BDAS.

2) Rotation: Chakrabarti et al. [15] proposed a scheme called R-DAS adding rotation as a degree of freedom to DAS. This intuitively increases the theoretical password-space and may increase the effective password space. By drawing the same pattern but using rotation between several strokes, users hypothetically can achieve a stronger pattern. However, authors did not study the usability and effect of rotation on what users generate as their patterns. Applying rotation on Android Pattern is not practical because it is a single-stroke scheme. In addition, Android Pattern is used for frequent authentication and rotation possibly hampers its usability to a great extent.

3) Layering: Chiang et al. [16] proposed an extension of DAS called Touch-screen Multi-layered Drawing (TMD) where they add "wrap cells" that allow users to continuously draw their passwords across multiple layers. This improves the theoretical password-space. However, the usability study shows that biases of starting point and shape of the patterns remain pertinent.

4) Blacklisting: An intuitive approach to strengthen the security of patterns is to blacklist certain patterns that are used frequently (e.g., a pattern like "Z" or "N", or any pattern starting from the top-left point) and do not allow users to choose them as their pattern. Uellenbeck et al. [4] have experimented such an approach by removing the most frequently used starting point, o_0, from the Android unlock screen (i.e., blacklisting all patterns that start from that point).

 ISBN: 978-1-61208-418-3

They noticed that this resulted in a *new* frequently used starting point (o_1). Indeed, the blacklisting approach only shifts the distribution of patterns, and does not transform the skewed distribution to a uniform one.

5) Random assignment: Another option to strengthen the security of patterns is to assign a random pattern to the user. This resolves the problem of skewed distribution of patterns. Nonetheless, random assignment will suffer from practical weaknesses including usability [17] and memorability.

6) Rearrangement: Some believe that the shape of the grid and the arrangement of cells create some inherent biases on what users choose as their patterns (e.g., choosing straight vertical or horizontal lines, instead of a cross line, because of the visual effects of the grid). Therefore, rearranging of the points of a grid in a shape other than square (e.g., circle or random) is studied as a potential technique to remove such biases. However, it has been observed that the biases only shift to a new set of points and results in a new set of frequently used sub-sequences [4]. These modifications do not help to remove the biases of the patterns, and do not increase the security of scheme.

C. Persuasive password security as an alternative approach

Persuasive Technology is a psychological framework which can be defined as "interactive computing systems designed to change people's attitudes and behaviours" [18]. Built on top of persuasive technology, there has been prior work on persuasive password security which persuades users to choose strong passwords by creating suitable user-interfaces [19][20]. A persuasive user-interface guides user to choose options that are desirable from the perspective of the designer of the system. In the same way, a persuasive password user-interface guides user to choose passwords that are strong. Chiasson et al. [21] have proposed and studied the "cued click point" a variant of PassPoints [22] which employs persuasive password security techniques to reduce the biases and reduces the predictable hotspots from 40% to 8%.

Forget et al. have proposed a persuasive authentication framework [5] that enumerates possible techniques for persuasion. These include simplification, personalization, monitoring, conditioning, and social interaction, as applied to a user-interface. For example, personalization includes suggestions of secure options to the user, and monitoring includes feedback to users about the security of their choices.

D. Pattern strength

In this section, we discuss guessing attacks on the Android Pattern system. Assuming the attacker has a perfect knowledge of the system and the distribution of all Android pattern used by users, an attacker can build a probabilistic model for computing the probability $P(X)$ of every possible pattern X. A pattern X_1 is considered stronger than pattern X_2 if $P(X_2) > P(X_1)$; resulting in an attacker tries X_2 before trying X_1 to guess someone else's pattern. In summary, more likely patterns are guessed before less likely ones. Therefore, to evaluate the strength of a pattern, we develop a score function $f(X)$ based on the probability of a given pattern X, in which, a more likely pattern gets a low score, and a less likely one gets a higher score.

Uellenbeck et al. [4] demonstrated that a Markov probabilistic model can effectively estimate the probability of patterns as:

$$P(X = o_1 o_2 \dots o_m) = P(o_1 o_2 \dots o_{n-1}) * \prod_{i=n}^{m} P(o_i | o_{i-n+1} o_{i-n+2} \dots o_{i-1}) \quad (1)$$

To compute this probability, we need an appropriate training dataset to compute the conditional probability

$$P(o_i | o_{i-n+1} o_{i-n+2} \dots o_{i-1})$$

For a 3-gram Markov model, we should compute the probability $P(o_i|o_{i-2}o_{i-1})$ for all different combinations of the nodes. We use an estimation of this probability instead, by collecting enough sample of patterns, using an appropriately designed experiment. If a sequence does not occur in our dataset, the probability of *zero* is assigned to that n-gram, leading to estimation of zero as the probability of a rare pattern. To fix this issue, we use Kneser-Ney smoothing (an advanced form of absolute-discounting interpolation) [23]. It is considered as the most effective method of smoothing.

We use a simple score function based on the Markov probabilistic model. MM-score(X) = $-log(p(X))$, where probability of X, $P(X)$ is computed by (1). We refer to this as MM-score function in the rest of the paper. A pattern X_1 is stronger than pattern X_2 iff:

$$MM - score(X_1) > MM - score(X_2)$$

We categorize all patterns of Android Pattern into 3 different levels of security: *weak*, *medium*, and *strong*. We compute the strength of all possible Android Pattern based on MM-score defined above. Defining an interval for the score of the patterns in each of these levels is subjective. A minimum security of a 4-digit PIN system is considered appropriate for authentication in ATMs [24][25] and smartphones. Therefore, we classify the patterns which offers the security of a 2-digit PIN as *weak* patterns accounting to a total of 100 patterns. The next 900 patterns are labeled as *medium* security, as they provide a maximum security of a 3-digit PIN, and all other patterns are labeled as *strong* patterns, since they provide the minimum security of a 4 to 5 digits PIN.

III. Proposed Approach

In this section, we first present our findings on what users are aware and where they lack understanding of strength of patterns (see Section III-A). Instead of assigning random pattern to a user, imposing unreasonable restrictions or enforcing them to not use certain blacklisted patterns/points, we propose persuasive security mechanisms to nudge/persuade users to choose secure patterns, without potentially hampering the usability or security of the system. In this paper, we propose a) EPSM (using self-monitoring) and b) BLINK (using personalization) to help users create stronger patterns by infusing knowledge of the global pattern distribution to the system (see Section III-B and Section III-C, respectively).

 ISBN: 978-1-61208-418-3

A. Plausible awareness and obliviousness

To provide better security suggestions or instructions, it is first important to understand what users are aware of and where they are lacking. To understand this, we conducted a short online survey using Amazon Mechanical Turk (MTurk). From 336 total participants, we analyzed the responses from only 266 participants. We eliminated a) anyone who provided contradicting answers to the same question asked multiple times with different wording. b) who completed the survey in less than 30 seconds or took more than 5 minutes. All participants, even those we eliminated, were paid $0.50. Participation for this survey was restricted to only Android users who have either used or are using Android unlock pattern as authentication mechanism on their phones. This survey and all the experiments reported in this paper were approved by the Institutional Review Board of the New York University (IRB approval reference IRB-13-9674) and Institutional Review Board of the New York University Abu Dhabi. Data collected from the participants was anonymized and protected according to the procedures described in the corresponding IRB submission documents.

We used a within-subjects design and asked ("*How strong is the following pattern?*") the participants to rate the strength of six patterns as shown in Figure 2 on a 5-point Likert scale. We chose patterns from three different security levels based on their strength, *weak*, *medium*, and *strong*, as is defined in section II-D. Patterns 2(a) and 2(b) are weak patterns, patterns 2(c) and 2(d) are of medium security level, and patterns 2(e) and 2(f) are strong patterns (refer to Figure 2). To avoid biases, we randomized the order that the patterns were shown to the participants.

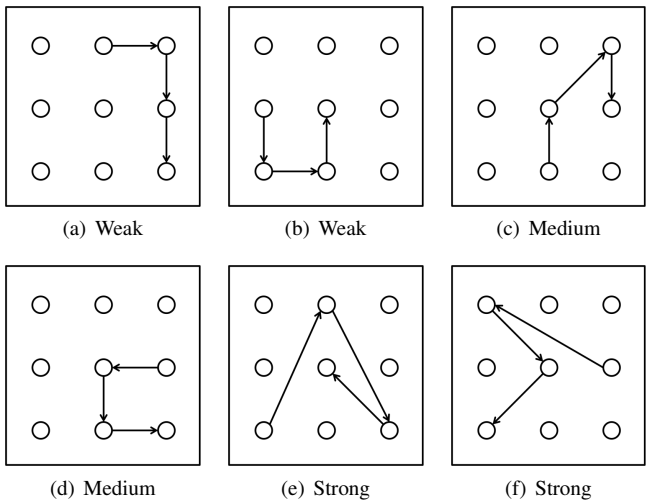


Figure 2. Choice of different proposed patterns (based on pattern strengths)

As it can be observed from result of the survey in Figure 3, users *are aware* of the relative strength of patterns and can distinguish between complex (Figures 2(e) and 2(f)) and easy to guess (Figures 2(a) and 2(b)) patterns.

However, this knowledge is not translated into selection of strong patterns by a large number of users and many still choose weak patterns. However, we exploit this *awareness* and design a simple but effective feedback mechanism called EPSM which provides feedback to users about the security

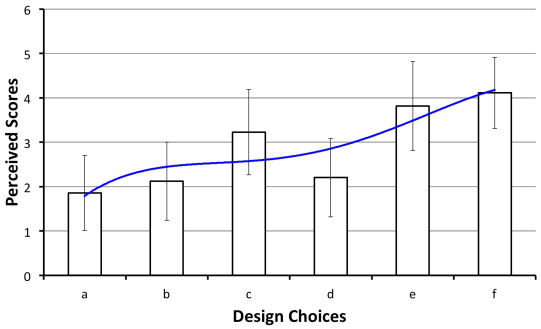


Figure 3. Perceived strength of patterns by users. It is mostly consistent with computed strength class for patterns.

of their pattern (by updating the color of the pattern). EPSM does not provide any hint on how to create a better pattern because users are already *aware* of which patterns are more secure.Therefor, users will be able to update their patterns to a stronger one.

Moreover, based on this survey, we observed that the perceived security strength of patterns 2(b) and 2(d) is almost same (as shown in Figure 2), where their strength is not the same in reality (because of the different starting points and n-grams they use). We believe that this could be because of the similar shapes of the two patterns. This suggests that even if user chooses a pattern of a shape similar to a weaker pattern but with a different starting point, it can increase the strength of the pattern. For this purpose we propose BLINK in which the system suggests a different starting point to the users.

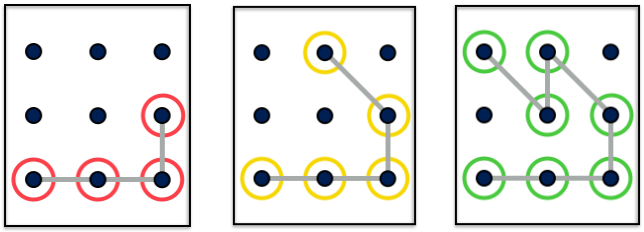
B. EPSM: Embedded pattern strength meter

Self-monitoring is one of the persuasive security principles helping users to adjust their security behavior [20] and was used to design EPSM. Andriotis et al. [26] showed the promise of this approach by providing a text-based feedback to users about the strength of their patterns after they complete drawing it. In their experiment, one out of five subjects changed their patterns after knowing their patterns are not strong.

In EPSM, instead of giving a delayed feedback after users generate their patterns and using a separate user interface element (e.g., text-based feedback), we provide a *continuous* and *embedded* feedback while the user is drawing a pattern. This design choice was made because a) it is more effective to provide continuos feedback influences the user's decision of what to choose as her pattern. b) smartphones have relatively small screen size which demands a compact representation of information and feedback. This is helpful for users to adjust the strength of their patterns as they create the patterns.

EPSM provides a fine-grained continuous embedded feedback by coloring the user's pattern according to pattern's strength level. The *red* color alarms a *weak* pattern, *yellow* indicates *moderate*, and *green* represents a *strong* pattern (see Figure 4). The system also pops-up a message describing the meaning of each color "*As you draw your pattern, the color of your pattern changes from red to green. Red one is bad (others can guess your pattern), yellow is good, and green is perfect.*" Regardless of the strength of the pattern the color of the pattern remains *red*, until the pattern satisfies the minimum required length (i.e., four). Thereafter, based on the

strength of the pattern, the color of the pattern gets updated (see Section II-D for details on pattern strength). Note that change in the color usually goes from red to yellow to green, however in certain cases it can sometimes goes the other way around, i.e., from green to red. For example, a half-drawn Z ($o_0 - o_1 - o_2 - o_4 - o_6 - o_7$) is a more secure pattern than a full drawn Z ($o_0 - o_1 - o_2 - o_4 - o_6 - o_7 - o_8$).



(a) Red (b) Yellow (c) Green

Figure 4. Sequence showing the change in colors for a single pattern, as user draw it

C. BLINK- Nudging

Nudging, a concept in behavioral science, argues that positive reinforcement and suggestions can influence the motives, incentives and decision making of groups and individuals [27]. Nudging can be used to suggest stronger patterns to users. However, suggesting a random pattern hampers usability and memorability of the Android Pattern. A more practical option is to provide partial suggestions. For example, suggesting users where to start their patterns is helpful to remove the bias of starting points (e.g., more than 40% of the users use upper most left point to start their patterns [4][10]) that can be used by attackers to guess the patterns easily.

In a pilot study, we examined a number of techniques to suggest a starting point to the users without hampering the usability of Android Patterns. Based on our observations and users suggestions, we concluded that a) suggestion by blinking a point is very effective, and b) users need to be told what is expected be done with the suggested point without hampering the usability of the system. Our final design uses a blinking point (see Figure 5) and similar to EPSM it pops-up a recommendation message stating "*It is recommended to start your pattern with the blinking point but NOT MANDATORY*", as it helps user to understand what to do and how to proceed with the blinking point.

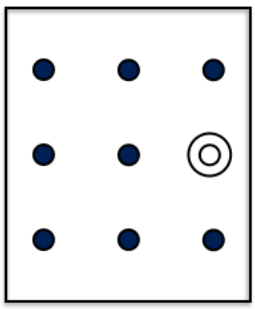


Figure 5. Suggesting a start point by blinking

During the registration phase, the system randomly recommends one of the 9 points in the screen to the user by creating an additional circle around the recommended point and blinking indefinitely. The circle stops blinking when the user starts drawing a pattern.

IV. Experimental Setup

To measure the efficacy of the designed persuasive security schemes, we conducted a between-subjects usability study on Amazon Mechanical Turk (MTurk). Subjects were assigned randomly to three different user interfaces: a) the control group, where users were assigned to work with the normal Android Patterns user interface (NORMAL), b) the BLINK group, where users were assigned to the BLINK user interface, and c) the EPSM group, where users were assigned to the EPSM user interface. We compare the security, memorability and usability of our proposed user interfaces (EPSM and BLINK) with that of the normal Android Patterns (NORMAL).

Because of Amazon's MTurk policy, we could not ask our participants to install an app on their smartphone to participate in our user study, therefore, we implemented the NORMAL, EPSM and BLINK using web technologies (HTML, CSS, and Javascript) accessible by visiting a link on participants' smartphones. In order to avoid the possibility of doing the experiment on a desktop or any other device beside an Android phone, we checked the "Browser Agent" field of the HTTP requests and only permitted those requests issued by an Android phone. Web pages are rendered differently on different devices and browsers, and there is a possibility that users do not see the user interface as we expected. To detect any such distortions in users' experience, we asked the participants about the quality of the user interface, at the end of the user study in the post user study survey. Any data from those who reported a distortion in the main page is excluded from our analysis.

The user study procedure involved two main steps a) *Registration* - Participants were assigned randomly to one of the groups, i.e., NORMAL, EPSM or BLINK. After that, they were asked to choose a pattern and then to verify it immediately. All participants were instructed to imagine that they have received a new phone and would like to set an Android Pattern on it. They were asked to choose a pattern of a minimum length four. b) *Survey* - After creating a pattern, participants were asked to complete a survey. We paid $0.40 to each participant upon the completion of our user study.

TABLE I. GROUP DEMOGRAPHICS.

Group	Total Participants
NORMAL	92 M (28); F (64)
EPSM	72 M(25); F(47)
BLINK	106 M(31); F(75)

We recruited a total of 270 US-based workers to participate in our experiment. Note, that these are different participants than the participants described in Section III-A. We also confined our user study to those who are familiar with the Android Patterns. Demographics of the participants and number of participants are given in Table I.

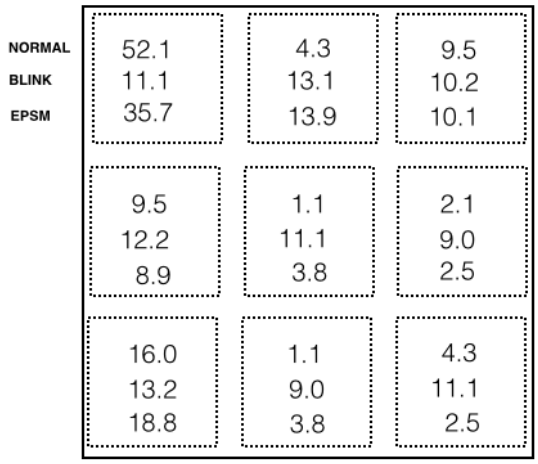
V. Results

In this section, we analyze the patterns obtained during our user study, and compare the security, usability and memorability of each design schemes proposed.

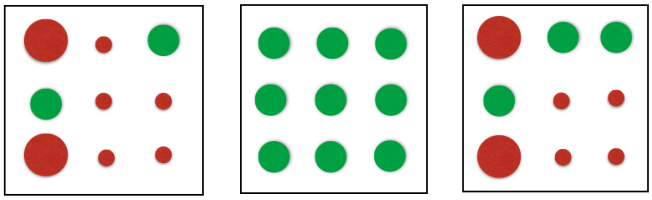
 ISBN: 978-1-61208-418-3

A. Starting point distribution

Figure 6(a) shows the percentage of patterns starting from each of the nine points in all three schemes. As expected in NORMAL and EPSM, we can observe that the starting point probabilities of the top two corner points (52.1% and 35.7%) are much higher than the other points. This is because there was no recommendation provided for eliminating the starting point bias in NORMAL and EPSM.



(a) Start point distribution(%)



(b) NORMAL (c) BLINK (d) EPSM

Figure 6. Distribution of starting point of the patterns in the NORMAL, BLINK and EPSM. BLINK removes the bias of starting points.

On the other hand, the bias of starting point probability in BLINK is eliminated and it is distributed almost uniformly. This happens because BLINK suggests the starting points randomly, and 85% of users use the suggested point. For some others, this has apparently nudged them to choose the start point of their patterns wisely. Table II shows the percentage of the suggestions used by users for each of the nine points in the BLINK.

TABLE II. PERCENTAGE OF USED SUGGESTION FOR EACH POINT IN THE BLINK USER INTERFACE. POINTS ARE SUGGESTED UNIFORMLY AND RANDOMLY.

Point	0	1	2	3	4	5	6	7	8
Used	80%	90%	77%	100%	84%	73%	100%	81%	85%

B. Pattern strength

Theoretical password-space of the normal Android Pattern is higher than that of a 5-digit PIN, but a large number of users (51%) use patterns with strength level (see Section II-D) of a 3-digit PIN (*weak* and *medium* strength patterns). We designed EPSM and BLINK to persuade users to use patterns

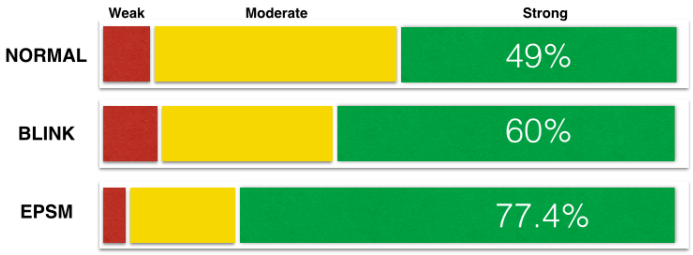


Figure 7. Strength level of the patterns created by users of NORMAL, BLINK and EPSM. Red, yellow, and green bars indicate weak, moderate, and strong patterns respectively.

of greater strength. Figure 7 shows the strength level of the patterns generated in our user study. The percentage of users that create strong patterns is increased to 60% in the BLINK, and 77.4% in the EPSM. This shows that BLINK and EPSM are able to persuade users to choose stronger patterns.

C. Security against partial guessing attack

In this section, we study the resilient of patterns generated by each user interface against guessing attacks. In a guessing attack, an attacker has access to an oracle (a blackbox) that gets a password and answers yes if it is correct. An oracle may answer an unlimited number of queries or apply some limitations on the amount or speed of returning the results. For example, on Android phones, a maximum number of 20 guesses are granted before the attacker get locked out completely. An oracle may also appear in the form of a secure hardware module that is rate-limited for the purpose of deterrence of attackers, and answer the queries very slowly (e.g., iPhone [28]).

An optimal attacker tries the weak patterns before the strong ones because they are more likely to be used by users. Indeed, such an adversary builds a dictionary

$$D_{pattern} = \{pt_1, pt_2, ..., pt_n\}$$

as a set of all possible patterns. Then, he computes the probability of occurrence of each pattern based on a probabilistic model (see equation 1), and then sorts the patterns based on their probability to compute the ordered list $G = (g_1, g_2, ..., g_n)$ where

- $g_i \in D_{pattern} \quad \forall \quad 1 \leq i \leq n$
- $P(g_i) \leq P(g_j) \quad \forall \quad i \leq j$

The attacker tries patterns in the order in which they appear in the ordered list G. Since guessing very strong patterns is time consuming, and is not cost-effective in many cases, an attacker usually guesses a portion of passwords with a reasonably small effort. This is called *partial guessing attack* and is used to evaluate the security of text-based passwords [29] and Android Patterns [4]. Accordingly, we evaluate the security of our proposed user interfaces against *partial guessing attack*.

Figure 8 shows the success rate of guessing attack against NORMAL, EPSM and BLINK. As we can observe, it is evident that patterns in EPSM and BLINK are stronger than NORMAL, and an attacker needs more effort (i.e., number of guesses) to guess a portion of them in comparison with patterns in NORMAL. Specifically, an attacker needs only 886 guesses to guess 50% of the patterns in NORMAL, whereas he

 ISBN: 978-1-61208-418-3

needs 3344 and 1918 guesses to guess 50% of the patterns in EPSM and BLINK, respectively. In terms of *partial entropy* of patterns, this translates to 9.79 bits of entropy for the patterns used by 50% of NORMAL, and 11.7 and 10.9 bits of entropy for the same proportion of users in EPSM and BLINK.

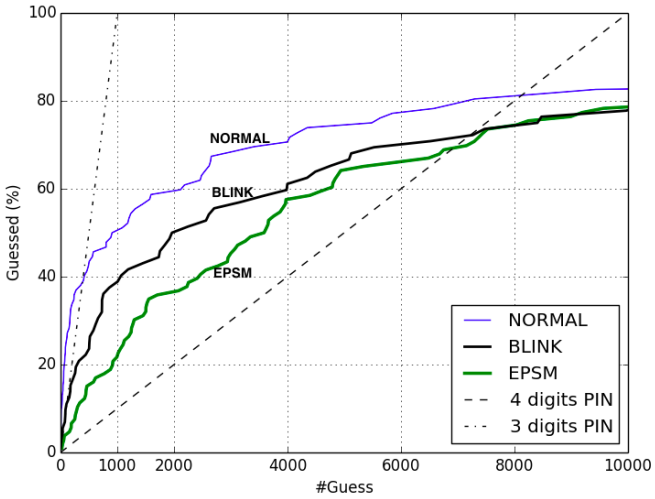


Figure 8. Guessing attack against NORMAL, BLINK and EPSM. Guessing model is built over the NORMAL version.

D. Pattern Length

Table III shows statistics about the length of patterns for each design schemes. Even though there are minor differences between the length of patterns in EPSM as compared to NORMAL and BLINK, Kruskal Wallis test (This is a non-parametric equivalent of the one-way analysis of variance (ANOVA)) does not show any significant difference between length of the patterns (χ^2=2.5, p>0.28). This emphasizes that the increased security offered by EPSM is not resulted by longer patterns, but is because of using more complex patterns with higher strength levels.

TABLE III. LENGTH OF PATTERNS.

Group	Average length	Std. Dev
NORMAL	6.13	1.61
BLINK	6.12	1.78
EPSM	6.5	1.72

E. Short-term recall rate

One of the design considerations for our new variations is to create a strong yet easy to use pattern scheme without hampering the users' recall rate. In the frequent authentication schemes (e.g. unlocking phone that is done several times a day), the repetition of password entry helps users to recall the pattern over long intervals. Consequently, it is reasonable to measure the short-term memorability of the patterns [30].

Since the maximum idle timeout before the Android phone gets locked down in normal Android Patterns is 30 minutes, we tested the memorability of the patterns after 20 minutes. It was not possible to guaranty that the subjects we recruited from the Amazon Mechanical Turk will take the recall task within 20 minutes. Therefore, we ran a between-subjects in-lab study and recruited 60 students to evaluate the recall rate of

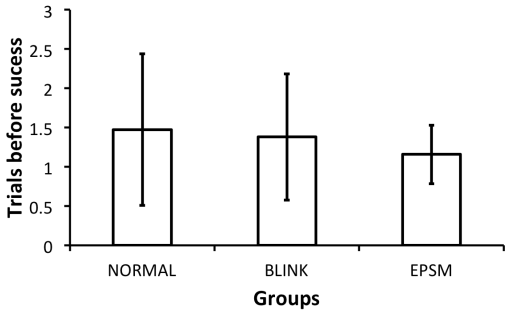


Figure 9. Recall accuracy (in terms of trials before success)

the proposed user interfaces. When they created their patterns using NORMAL, BLINK, and EPSM, we asked them to return back after 20 minutes to do a follow-up test where we asked them to re-enter their pattern. Figure 9 shows the recall rate of the patterns for each version after 20 minutes. We compute the recall rate accuracy in terms of how many trials participants took on average to verify themselves against the system. Based on the pairwise *t-test* conducted we found that there is no statistically significant difference between the the recall rate of NORMAL and BLINK; and NORMAL and EPSM.

VI. Conclusion

In this paper, we proposed two Android Patterns schemes with the goal of improving the security of patterns chosen by users. We used the principles from *persuasive security framework* to nudge users to choose starting points uniformly and to use more complex sequence of points in their patterns. We recruited 270 participants from Amazon Mechanical Turk and conducted a usability user study to measure the effect of our proposed schemes on security and usability of the system.

While only 49% of subjects choose *strong* patterns in standard Android Patterns, our suggested schemes increase it to 60% in BLINK and 77.4% in EPSM version. Accordingly, the partial entropy of the patterns is increased from 9.79 in NORMAL to 10.9 in BLINK and 11.7 in EPSM. These improvements are achieved without hampering the usability in term of the length of the pattern and short-term recall rate.

VII. Acknowledgement

The first, third, and fourth authors were supported by NSF grant 1228842. We would like to thanks Markus Jakobsson for suggestions of the design for the experiment, and all the anonymous reviewers for their comments and feedbacks towards this work.

References

[1] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner, "Are you ready to lock?" in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 750–761.

[2] D. Van Bruggen, S. Liu, M. Kajzer, A. Striegel, C. R. Crowell, and J. D'Arcy, "Modifying smartphone user locking behavior," in Proceedings of the Ninth Symposium on Usable Privacy and Security. ACM, 2013, p. 10.

[3] D. C. Van Bruggen, "Studying the impact of security awareness efforts on user behavior." Ph.D. Thesis, University of Notre Dame, 2014.

 ISBN: 978-1-61208-418-3

[4] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the security of graphical passwords: The case of android unlock patterns," in Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 161–172.

[5] A. Forget, S. Chiasson, and R. Biddle, "Persuasion as education for computer security," in World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, vol. 2007, no. 1, 2007, pp. 822–829.

[6] I. Jermyn et al., "The design and analysis of graphical passwords," in Proceedings of the 8th USENIX Security Symposium, vol. 8. Washington DC, 1999, pp. 1–1.

[7] H. Tao and C. Adams, "Pass-go: A proposal to improve the usability of graphical passwords." IJ Network Security, vol. 7, no. 2, 2008, pp. 273–292.

[8] J. Thorpe and P. Van Oorschot, "Towards secure design choices for implementing graphical passwords," in Computer Security Applications Conference, 2004. 20th Annual. IEEE, 2004, pp. 50–60.

[9] P. C. van Oorschot and J. Thorpe, "On predictive models and user-drawn graphical passwords," ACM Transactions on Information and system Security (TISSEC), vol. 10, no. 4, 2008, p. 5.

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

[11] P. Dunphy and J. Yan, "Do background images improve draw a secret graphical passwords?" in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 36–47.

[12] Z. Zhao, G.-J. Ahn, J.-J. Seo, and H. Hu, "On the security of picture gesture authentication," in Proceedings of the 22Nd USENIX Conference on Security, ser. SEC'13. Berkeley, CA, USA: USENIX Association, 2013, pp. 383–398.

[13] H. Gao, W. Jia, N. Liu, and K. Li, "The hot-spots problem in windows 8 graphical password scheme," in Cyberspace Safety and Security. Springer, 2013, pp. 349–362.

[14] Z. Zhao, G.-J. Ahn, J.-J. Seo, and H. Hu, "On the security of picture gesture authentication." in USENIX Security, 2013, pp. 383–398.

[15] S. Chakrabarti, G. V. Landon, and M. Singhal, "Graphical passwords: drawing a secret with rotation as a new degree of freedom," in Proceedings of the Fourth IASTED Asian Conference on Communication Systems and Networks. ACTA Press, 2007, pp. 561–173.

[16] H.-Y. Chiang and S. Chiasson, "Improving user authentication on mobile devices: A touchscreen graphical password," in Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services. ACM, 2013, pp. 251–260.

[17] E. A. Stobert, "Memorability of assigned random graphical passwords," Ph.D. dissertation, Carleton University, 2011.

[18] B. J. Fogg, "Persuasive technology: using computers to change what we think and do," in Ubiquity, vol. 2002, no. December. ACM, 2002, p. 5.

[19] D. Weirich and M. A. Sasse, "Persuasive password security," in CHI'01 Extended Abstracts on Human Factors in Computing Systems. ACM, 2001, pp. 139–140.

[20] A. Forget, S. Chiasson, P. C. van Oorschot, and R. Biddle, "Persuasion for stronger passwords: Motivation and pilot study," in Persuasive Technology. Springer, 2008, pp. 140–150.

[21] S. Chiasson, P. C. van Oorschot, and R. Biddle, "Graphical password authentication using cued click points," in Computer Security–ESORICS 2007. Springer, 2007, pp. 359–374.

[22] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "Passpoints: Design and longitudinal evaluation of a graphical password system," Int. J. Hum.-Comput. Stud., vol. 63, no. 1-2, Jul. 2005, pp. 102–127.

[23] J. M. Gawron, "Discounting," http://www-rohan.sdsu.edu/~gawron/compling/course_core/lectures/kneser_ney.pdf, accessed: 2015-03-19.

[24] I. O. for Standardization, "Iso 9564," http://en.wikipedia.org/wiki/ISO_9564, accessed: 2015-03-19.

[25] B. Milligan, "The man who invented the cash machine," http://news.bbc.co.uk/2/hi/business/6230194.stm, accessed: 2015-03-19.

[26] P. Andriotis, T. Tryfonas, G. Oikonomou, and C. Yildiz, "A pilot study on the security of pattern screen-lock methods and soft side channel attacks," in Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, ser. WiSec '13. New York, NY, USA: ACM, 2013, pp. 1–6.

[27] R. Thaler and C. Sunstein, Nudge. Yale University Press, 2008.

[28] S. Garfinkel, "The iphone has passed a key security threshold," http://www.technologyreview.com/news/428477/the-iphone-has-passed-a-key-security-threshold/, accessed: 2015-03-19.

[29] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012, pp. 538–552.

[30] A. Beautement and A. Sasse, "Gathering realistic authentication performance data through field trials," in SOUPS USER Workshop, 2010.

 ISBN: 978-1-61208-418-3