# TinPal: An Enhanced Interface for Pattern Locks

Harshal Tupsamudre, Sukanya Vaddepalli, Vijayanand Banahatti, Sachin Lodha
TCS Research, Pune, India
Email: firstname.lastname@tcs.com

***Abstract*—Pattern lock scheme in which users connect 4-9 dots in a $3 \times 3$ grid is one of the most popular authentication methods on mobile devices. However, numerous research studies show that users choose patterns from a small space which makes them vulnerable to a variety of attacks such as guessing attacks, shoulder-surfing attacks and smudge attacks.**

**In this work, we enhance the existing $3 \times 3$ interface with a visual indicator mechanism and demonstrate how this slight modification can influence users' pattern choices, thereby improving the security of the pattern lock scheme. We refer to this enhanced interface as *TinPal*. As users draw their pattern, TinPal highlights the next set of unconnected dots that can be reached from the currently connected dot. We gauge the impact of this highlighting mechanism on users' pattern choices by performing a comparative study of two groups, where one group creates pattern using the existing interface while the other group creates pattern using TinPal. The study results show that participants who used TinPal created more secure patterns than participants who used the existing interface.**

## I. Introduction

Today smartphones have become an integral part of people's daily lives. According to Ericsson mobility report [2], the number of worldwide smartphone subscriptions has already crossed 3 billion mark and it is expected to reach 6.8 billion by year 2022. People use smartphones for a wide range of applications which include sending emails, browsing web, chatting, making audio/video calls, taking pictures and doing financial transactions. Consequently, these portable devices are a gateway to a plethora of personal and sensitive information. Therefore, smartphones come equipped with several authentication mechanisms such as number passwords (PINs), textual passwords, graphical passwords and biometrics (e.g., fingerprint, face and iris).

Graphical passwords are considered as a usable alternative to textual passwords since many studies [19], [17], [27] show that humans have remarkable ability to remember visual information than textual information. The most prominent example of graphical passwords is Google's recall-based *Android Pattern Lock Scheme* in which users connect a series of dots in $3 \times 3$ grid to unlock their phone. This pattern lock scheme is perceived to be more usable than PINs [29]. Further, biometric alternatives are available only in high-end devices and are nevertheless considered to be less secure than $3 \times 3$ patterns [5]. Moreover, the use of biometric security poses huge privacy risks [1].

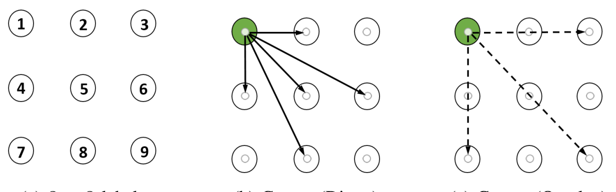
Workshop on Usable Security (USEC) 2018
18 February 2018, San Diego, CA, USA
ISBN 1-891562-53-3
https://dx.doi.org/10.14722/usec.2018.23021
www.ndss-symposium.org

Although $3 \times 3$ patterns are considered to be usable, they are prone to a wide range of attacks including *guessing attacks*, *shoulder-surfing attacks* and *smudge attacks*. Many research studies [25], [8], [24] show that users' pattern choices are highly biased and drawn from a small space. For instance, patterns resembling English letters such as 'Z', 'S', 'M', 'N', 'L', and 'G' are very prevalent among users. Further, a large fraction of patterns are composed using simple strokes which could be easily memorized by an observer [28]. The characteristics such as knight moves, overlaps, direction changes, intersections (crosses) which enhance the visual complexity of patterns are almost never used. The pattern lock scheme is also shown to be susceptible to smudge attack [9], a type of side-channel attack in which the attacker infers user's pattern using physical traces left (by fingers) on the screen. Further related work [7] even suggests that it is possible to recover the entire pattern from the partial traces by exploiting users' biased choices.

**Notations**. To make it easier when referring to a particular pattern, we label all dots arranged in the $3 \times 3$ grid in row-major order, where the upper-left dot is labelled as 1 and the lower-right dot is labelled as 9 as shown in Figure 1a. A pattern is therefore represented as an ordered sequence of dots, e.g., 38519647. When referring to a line segment (connection) between two consecutive dots $d_1$ and $d_2$ in a pattern, we use the notation $d_1 \rightarrow d_2$. For instance, the line segment between consecutive dots 3 and 8 in the pattern 9538127 is represented as $3 \rightarrow 8$.

The rules [25] for creating patterns are given below. Due to these restrictions, the total number of $3 \times 3$ patterns is 389,112.
(i) At least 4 dots must be chosen.
(ii) No dot can be used twice.
(iii) Only straight lines are allowed.
(iv) One cannot jump over dots not visited before.

We observed that the first three rules (i), (ii) and (iii) are enforced by the existing pattern lock interface [3], however *there is no mechanism in the existing interface to make users aware of the rule (iv)*. The fourth rule implies that dot $d_i$ can be connected to dot $d_j$ directly, if all dots along the (straight line) path are already connected. For instance, one can connect dot 1 to dot 3 ($1 \rightarrow 3$) if dot 2 is already connected or connect dot 2 directly to dot 8 ($2 \rightarrow 8$) if dot 5 is already connected (Figure 1a). The existing pattern lock interface never informs users about such connection options which might result in constrained pattern choices. If these kind of connections are never used in the patterns, then the theoretical search space is reduced to just 139,880, *i.e.*, about 1/3rd of 389,112. We conjecture that many users are simply not aware of all possible connection choices during pattern creation due to which they resort to insecure behaviour.



(a) 3 × 3 labels (b) Corner (Direct) (c) Corner (Overlap)

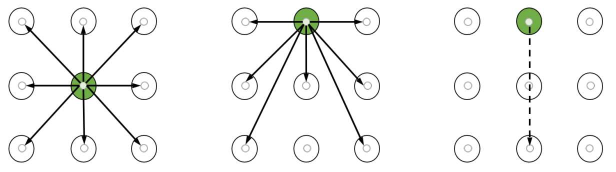

(d) Center (e) Side (Direct) (f) Side (Overlap)

Fig. 1: Connectivity rules for corner, center and side dots. There are two types of connectivity rules, *direct* and *overlap*.

The connectivity rules in the pattern lock scheme are not easy to comprehend. For instance, consider the connectivity rules pertaining to dot 1.

1) From dot 1, one can connect any one of the five segments, $1 \rightarrow 2$, $1 \rightarrow 4$, $1 \rightarrow 5$, $1 \rightarrow 6$ or $1 \rightarrow 8$ as depicted in Figure 1b. We refer to such connectivity rules as *direct rules*.
2) From dot 1, one can also join $1 \rightarrow 3$ if dot 2 is already connected. This connection can happen in two ways.
    a) *Overlapping segment* : It occurs when the connection to dot 2 is immediately followed by dots 1 and 3. In this case, the line segment $2 \rightarrow 1$ is covered completely by the line segment $1 \rightarrow 3$. An example of such pattern is 5213847 (Figure 2a).
    b) *Overlapping dot* : It occurs when the connection to dot 2 is followed by some other dot(s) followed by dots 1 and 3. An example of such pattern is 62413589 (Figure 2b).

   Similarly, one can also join $1 \rightarrow 7$ if dot 4 is already connected or join $1 \rightarrow 9$ if dot 5 is already connected (Figure 1c). We refer to such conditional connectivity rules as *overlap rules*.

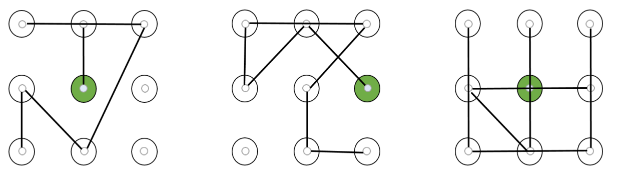

(a) Pattern 5213847 (b) Pattern 62413589 (c) Pattern 528463971

Fig. 2: Illustration of direct and overlap rules.

There is no mechanism available in the existing pattern lock interface to inform users about the different connectivity rules (direct and overlap). Spelling out these connectivity rules in text-form is a tedious task. This problem is further aggravated since the connectivity depends on whether the dot is located at the corner, center or to the side of a corner dot in the $3 \times 3$ grid as shown in Figure 1. There are four corner dots $\{1, 3, 7, 9\}$, one center dot 5 and four side dots $\{2, 4, 6, 8\}$. A corner dot can be connected directly to any of the 5 non-corner dots (Figure 1b), center dot can be connected to any of the remaining 8 dots (Figure 1d) while a side dot can be connected directly to any of the 7 dots (Figure 1e). Due to overlap rule, a side dot can be connected to the remaining eighth dot if dot 5 is already connected (Figure 1f).

Table I shows the theoretical distribution of overlaps in $3 \times 3$ patterns. If overlaps are never used in the patterns then the search space diminishes to just 139,880. The maximum number of overlaps that can occur in a pattern is 5, and it is observed for instance in patterns that begin with the center dot, followed by all side dots followed by all corner dots. One such pattern (528463971) is shown in Figure 2c.

| *#Overlaps* | *Count* | *Percentage* |
|---|---|---|
| 0 | 139,880 | 35.95% |
| 1 | 159,480 | 40.98% |
| 2 | 69,896 | 17.96% |
| 3 | 16,912 | 4.35% |
| 4 | 2,688 | 0.69% |
| 5 | 256 | 0.07% |
| #Patterns | 389,112 | 100% |

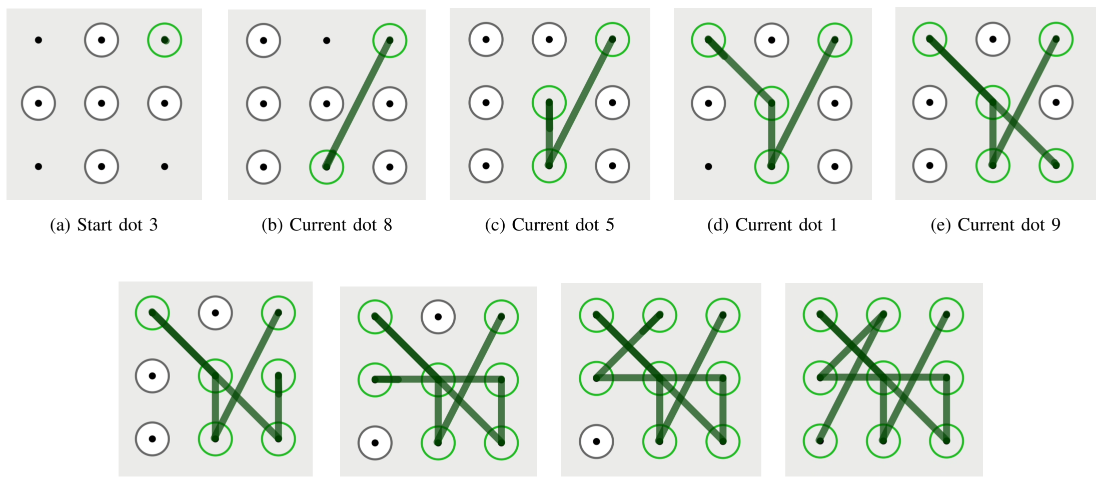
TABLE I: Distribution of overlaps in patterns.

## A. *Contribution*

In this work, we alter the existing $3 \times 3$ interface with a visual indicator mechanism to make users aware of different connectivity rules. Specifically, as users draw their pattern, the new $3 \times 3$ interface highlights the next set of unconnected dots that can be reached from the currently connected dot, thus making users aware of different connection options at each step during pattern creation as well as during recall. We refer to this highlighting interface as *TinPal*. We note that the proposed interface does not force or persuade users to connect any particular dot, instead it simply informs users about the set of choices available for consideration from the currently connected dot.

The working of TinPal is illustrated in Figure 3. The figure shows a step-by-step snapshot of TinPal while the pattern is being drawn by the user.

1) Suppose that the user starts her pattern with dot 3. From dot 3, the user can visit any non-corner dot $\{2, 4, 5, 6, 8\}$ (direct rule). However, the user cannot visit dot 1 as dot 2 is still unconnected or dot 7 as dot 5 is unconnected or dot 9 as dot 6 is unconnected (overlap rule). Hence, only non-corner dots $\{2, 4, 5, 6, 8\}$ are highlighted by TinPal as shown in Figure 3a.
2) Next, the user visits dot 8. From dot 8, the user can visit any dot except dot 2. This is because dot 5 is still not connected (overlap rule). Therefore, all unconnected dots $\{1, 4, 5, 6, 7, 9\}$ except dot 2 are highlighted as shown in Figure 3b.
3) Subsequently, the user connects dot 5. From dot 5, the user can visit any unconnected dot $\{1, 2, 4, 6, 7, 9\}$ as highlighted in Figure 3c.



(a) Start dot 3 (b) Current dot 8 (c) Current dot 5 (d) Current dot 1 (e) Current dot 9

(f) Current dot 6 (g) Current dot 4 (h) Current dot 2 (i) Current dot 7

Fig. 3: A step-by-step illustration of pattern creation on TinPal. This interface highlights the next set of unconnected dots that can be visited from the currently connected dot. The connectivity rules are conveyed to users in real-time while the pattern is being drawn. The pattern created in the above example is 385196427.

4) The user chooses dot 1. From there, the user can visit any unconnected non-corner dot $\{2, 4, 6\}$ (direct rule). Since dot 5 is already connected, the user can also visit dot 9 (overlap rule). However, the user cannot visit dot 7 as dot 4 is not yet connected. Hence, the set $\{2, 4, 6, 9\}$ is highlighted by TinPal as depicted in Figure 3d.
5) Next choice is dot 9. From there, the user can visit any unconnected non-corner dot $\{2, 4, 6\}$ (direct rule). Further, since dot 8 is already connected, the user can now visit dot 7 (overlap rule). Hence, the set $\{2, 4, 6, 7\}$ is highlighted as indicated in Figure 3e.
6) Next, the user visits dot 6. From dot 6, in addition to unconnected dots 2 and 7 (direct rule), the user can also visit dot 4 since dot 5 is already connected (overlap rule). Therefore, the set $\{2, 4, 7\}$ is highlighted in Figure 3f.
7) Next choice is dot 4. From there, the user can go to either dot 2 or dot 7 (direct rule) as highlighted in Figure 3g.
8) The user connects dot 2. Now, the only choice available is dot 7 (direct rule) which is highlighted in Figure 3h.
9) Finally, the user connects dot 7 and the pattern 385196427 is recorded as shown in Figure 3i.

The contributions of this paper are as follows:

- We enhance the existing $3 \times 3$ interface with a visual indicator mechanism to help users select more diverse patterns. Specifically, the new interface (referred to as TinPal) highlights the set of reachable dots from the currently connected dot, thus making connectivity rules more salient to users. The highlighting mechanism works in real-time while the pattern is being drawn.

- We also give an algorithm that takes the currently connected dot as an input and outputs the next set of unconnected dots that can be visited from the current dot.

- We evaluate the impact of our visual indicator mechanism with a comparative user study involving 99 participants. We found that the participants who used TinPal (experimental group), created patterns using significantly large number of dots and complex features such overlaps, knight moves and direction changes, than those who used the existing $3 \times 3$ interface (control group).

- We also estimate the guessing resistance of patterns created in the control group and the experimental group using a Markov model based guessing algorithm. Within first 20 attempts, the algorithm cracked 12% patterns in the control group, but none (0%) in the experimental group. We also had access to 69,797 $3 \times 3$ patterns (ASIACCS'17 dataset) reported recently in [24]. Within 20 attempts, the algorithm (trained on 69,797 patterns) cracked 12.24% patterns in the control group, but only 4% patterns in the experimental group.

The organization of this paper is as follows. First, we provide a brief overview of graphical passwords and review the work related to the security of the pattern lock scheme. Subsequently, we describe the design choices we made along with the working mechanics of the proposed interface. Next, we describe the user study and present both security and usability results. Finally, we discuss the future work.

## II. Related Work

Graphical passwords are considered as promising alternative to textual passwords since many research studies [19], [17], [27] suggest that graphical information is easier to remember than textual information. Based on the difficulty of retrieving graphical information from the visual memory, graphical schemes are broadly divided into three categories [11]: *recognition-based*, *cued recall-based* and *recall-based*.

A typical example of a recognition-based scheme is *PassFaces* [12], where the user selects an image face from a set of 9 image faces and correctly recognizes it among decoy images during login. To attain sufficient security, there are multiple iterations, with each iteration employing a different set of 9 images. An example of cued recall-based scheme is *Pass-points* [26] in which the user selects a sequence of points on a system-assigned image. During login, this image acts as a memory cue which helps the user to recall the selected points in the correct order. An example of recall-based scheme is *Pass-Go* [23], in which the user draws one or more strokes on a $n \times n$ grid. The $3 \times 3$ pattern lock scheme is a special case of Pass-Go, where $n$ is set to 3 so that the grid could fit on smartphones.

### A. Android Pattern Lock

Security of the pattern lock scheme has been well studied in the literature. In 2010, Aviv *et al.* [9] demonstrated that it is possible to reconstruct the user's entire pattern from the oily traces left on the screen. This attack is popularly known as *smudge attack*. Later in 2013, Andriotis *et al.* [7] showed that it is possible to recover the entire pattern even from the partial traces by exploiting users' biased choices. They surveyed 144 participants and found that more than half of them started their pattern from the upper-left dot. Further analysis revealed that 18.75% of the participants used the path $1 \rightarrow 2 \rightarrow 3$ in their pattern. These observations along with the partial physical traces reduced the search space drastically.

In 2013, Uellenbeck *et al.* [25] found that users' pattern choices are biased and prone to guessing attacks. They collected approximately 2900 patterns from 584 participants on 5 different layouts. Their analysis revealed that most users create $3\times3$ patterns with horizontal (e.g., $1 \rightarrow 2$) and vertical strokes (e.g., $1 \rightarrow 4$) which reduces the security of $3\times3$ patterns to just 3-digit random PINs. They also found that patterns created on the circular interface (8 dots on the circumference and 1 dot in the center) were more secure than $3\times3$ patterns. Similar results were reported in a recent study by Tupsamudre *et al.* [24]. They proposed a different circular interface, called Pass-O, with all 9 dots on the circumference and evaluated it with a large-scale study (21,053 users, 123,190 patterns). However, both [25], [24] focused on security and lacked rigorous usability evaluation. In 2016, Aviv *et al.* [8] studied the security of $4 \times 4$ patterns and found that the majority of them are just extended versions of $3 \times 3$ patterns.

Most graphical password schemes are susceptible to shoulder-surfing attacks [11]. In 2015, Zezschwitz *et al.* [28] performed a systematic evaluation of the shoulder-surfing susceptibility of the pattern lock scheme. They found that line visibility, pattern length, number of knight moves, number of overlaps and number of intersections (refer to section V-A for definitions) play an important role in thwarting shoulder-surfing attacks. Recently, Davin *et al.* showed that different viewing angles, hand positions and phone sizes can also affect the efficacy of shoulder-surfing attacks. Various strength meters have been proposed [6], [22], [21] to encourage users to create visually complex patterns. These meters nudge users to draw longer patterns with overlaps, knight moves, direction changes and intersections.

Other related works are by Siadati *et al.* [20] and Cho *et al.* [13]. Siadati *et al.* proposed a persuasive interface that suggests a random starting point to the user while Cho *et al.* proposed three SysPal policies that mandate users to include 1 to 3 random dots at any position in their pattern. However, we note that mandating the use of random dots in the pattern does not ensure that the resulting patterns will exhibit secure characteristics such as knight moves and overlaps simply because users may not be aware about the feasibility of such connections. In fact, Cho *et al.* found that in all three SysPal policies, the most frequently used segments were $1 \rightarrow 2$ and $2 \rightarrow 3$, and $i \rightarrow i+1$ was more frequently used than $i+1 \rightarrow i$ for all $i = 1, 2, 4, 5, 7$ and 8, implying that most patterns were drawn from left to right.

On the other hand, the highlighting interface TinPal proposed in this paper does not mandate users to connect any specific dot, it just informs them about the set of reachable dots in each step during pattern creation as well as during recall. Our study results show that patterns drawn on TinPal used significantly longer strokes and large number of knight moves, overlaps and direction changes as compared to those drawn on the existing interface. Moreover, the proportion of SysPal patterns that were cracked in 20 attempts are: 9.97% (1-dot), 9.36% (2-dot) and 14.11% (3-dot). These numbers suggest (also stated in [13]) that *mandating too many points could potentially reduce the overall password space*. Further, for patterns created using TinPal, the proportion of patterns cracked in 20 attempts is 0% (and 4% using ASIACCS'17 dataset [24]). To the best of our knowledge, our study is the first attempt that gauges the impact of informing users about the next valid points in the $3 \times 3$ pattern lock scheme.

## III. Interface Design and Mechanics

We employ two design principles, namely, *visibility* and *consistency* [18], to enhance the existing $3 \times 3$ interface. According to the visibility principle, the system should have proper mechanisms to convey to users what actions are possible. The visual indicator mechanism in TinPal makes the set of available choices visible to users. Specifically, it highlights the next set of unconnected dots that can be visited from the currently connected dot to help users choose diverse $3 \times 3$ patterns. This highlighting of dots happens in real-time while the pattern is being drawn.

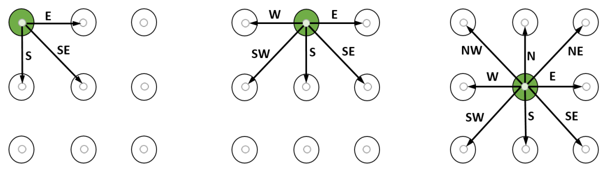
The *consistency* principle on the other hand makes the interface intuitive to use. The highlighting of dots in TinPal happens not only during pattern creation, but also during recall. This eliminates the confusion since the behaviour of the interface is consistent during creation and recall. Further, the highlighting of dots could also serve as cue when the user is trying to retrieve the pattern from her memory.

TinPal also retains the *feedback* mechanism of the existing interface, thereby enforcing the first three requirements for creating patterns [25]. For instance, if the user connects less than 4 dots, it displays the *feedback* message, "Connect at least 4 dots. Try again." and it enforces users to connect two dots using straight lines only.

### A. Mechanics

Now, we present an algorithm that takes the currently connected dot $d$ as an input and outputs the next set $S_d$ of unconnected dots that can be reached from the current dot $d$.

This algorithm is invoked whenever the user connects a new dot on TinPal. The set of dots returned by this algorithm are highlighted on the interface. To simplify the algorithm, we define a neighbourhood relation on the $3 \times 3$ grid. We say that a dot $d_1$ is a neighbour of a dot $d_2$ on the $3 \times 3$ grid if it lies in any of the eight directions (north, north-east, east, south-east, south, south-west, west and north-west) from dot $d_2$. Figure 4 depicts the neighbourhood of dots 1, 2 and 5 in all possible directions. Dot 1 being a corner dot has 3 neighbours, *i.e.*, dots 2 (east), 4 (south) and 5 (south-east). Dot 2 being a side dot has 5 neighbours, *i.e.*, dots 1, 3, 4, 5, and 6, whereas the center dot 5 has all the other dots as its neighbours.



(a) Neighbours of 1 (b) Neighbours of 2 (c) Neighbours of 5

Fig. 4: Neighbours of dot 1 (corner), dot 2 (side) and dot 5 (center).

Algorithm 1 starts by adding all the unconnected dots to the set $S_d$. Then it eliminates those dots from the set $S_d$ that are not reachable from the current dot $d$. For elimination, the algorithm employs Table II which describes the neighbours of all 9 dots on the $3 \times 3$ grid. A row in the table represents a dot and a column represents a direction. If a dot $d$ does not have any neighbour in direction $j$, then the corresponding entry in the neighbourhood table is marked with symbol $\perp$ (indicating empty). To eliminate the unreachable dots from the set $S_d$, the algorithm scans the entire $d^{th}$ row, *i.e.*, all 8 neighbours of dot $d$ in the neighbourhood table. If the entry (in the $j^{th}$ direction) is marked with $\perp$ then the algorithm does not take any action. Otherwise, the algorithm checks if the neighbouring dot $e$ (in the $j^{th}$ direction) is already connected. If it is not, then the algorithm eliminates the neighbour of the neighbouring dot $e$ in the $j^{th}$ direction from the set $S_d$. We illustrate the working of this elimination algorithm through an example.

Suppose that the user starts her pattern by connecting dot 1. Subsequently, the elimination algorithm is invoked to determine the set $S_d$ of unconnected dots that can be reached from the current dot 1. The algorithm begins by adding all unconnected dots $\{2, 3, 4, 5, 6, 7, 8, 9\}$ to the set $S_d$. The algorithm then scans the first row of the neighbourhood table to identify and remove the unreachable dots from dot 1 in the set $S_d$. As its neighbour in the east dot 2 is still unconnected, the algorithm removes dot 3 (which lies further to the east of dot 2) from the set $S_d$. This elimination occurs due to the overlap rule which says that dot 1 cannot be directly connected to dot 3 as dot 2 is still unconnected. Similarly, as the south neighbour dot 4 is yet to be connected, the algorithm removes dot 7 (which lies further to the south of dot 4) from the set $S_d$. Also, as the south-east neighbour dot 5 is unconnected, the algorithm removes dot 9 (which lies further to the south-east of dot 5) from the set $S_d$. Therefore, the set $S_d = \{2, 4, 5, 6, 8\}$ is returned for the highlighting purpose.

| *Dot* | *N* ↑ | *NE* ↗ | *E* → | *SE* ↘ | *S* ↓ | *SW* ↙ | *W* ← | *NW* ↖ |
|---|---|---|---|---|---|---|---|---|
| 1 | ⊥ | ⊥ | 2 | 5 | 4 | ⊥ | ⊥ | ⊥ |
| 2 | ⊥ | ⊥ | 3 | 6 | 5 | 4 | 1 | ⊥ |
| 3 | ⊥ | ⊥ | ⊥ | ⊥ | 6 | 5 | 2 | ⊥ |
| 4 | 1 | 2 | 5 | 8 | 7 | ⊥ | ⊥ | ⊥ |
| 5 | 2 | 3 | 6 | 9 | 8 | 7 | 4 | 1 |
| 6 | 3 | ⊥ | ⊥ | ⊥ | 9 | 8 | 5 | 2 |
| 7 | 4 | 5 | 8 | ⊥ | ⊥ | ⊥ | ⊥ | ⊥ |
| 8 | 5 | 6 | 9 | ⊥ | ⊥ | ⊥ | 7 | 4 |
| 9 | 6 | ⊥ | ⊥ | ⊥ | ⊥ | ⊥ | 8 | 5 |

TABLE II: Neighbourhood table for the $3 \times 3$ grid.

**Algorithm 1** Elimination Algorithm

```
1:  procedure Elimination Algorithm
2:  Input: Current dot $d$
3:  Output: Set $S_d$ of unconnected dots that can be visited
    from the current dot $d$
4:      $S_d \leftarrow$ all unconnected dots
5:      $m \leftarrow$ number of columns in the neighbourhood table
6:      $j \leftarrow 1$
7:      while $j \leq m$ do
8:          $e \leftarrow$ table[d][j]
9:          if $e \neq \perp$ && !isConnected(e) then
10:             $S_d \leftarrow S_d \setminus$ {table[e][j]}
11:         end if
12:         $j \leftarrow j + 1$
13:     end while
14: end procedure
```

## IV. User Study

We evaluate the impact of our visual indicator mechanism on users' pattern choices with a comparative user study. The study was conducted in lab during August 2017. Participants were recruited using internal mailing lists within our organization. A total of 99 users responded to our e-mail and completed the study. Participants were randomly split into two groups: a control group containing 49 participants and the experimental group containing 50 participants.

- **Control Group**. Participants in this group created their pattern on the existing $3 \times 3$ interface.
- **Experimental Group**. Participants in this group created their pattern on TinPal which highlights the next set of available dots that can be visited from the currently connected dot.

A recent study [10] found statistically significant differences between patterns collected on the mobile device of participants and patterns collected using other collection methods. Therefore, we opted for pattern collection on the participant's own mobile device. To make the study available across all mobile devices, we created an HTML/Javascript web application using Java J2EE platform. Thus, participants could easily access the study using any standard web-browser on their mobile devices without having to install any additional software. We tried to simulate the look and feel of Android Lock Pattern as closely as possible. Participants were given a pen and a chocolate worth \$2 for completing the study. There is no IRB in our organization for approving studies involving human subjects. However, we took all the necessary steps in order to be compliant with privacy regulations. The data was collected anonymously and used for research purposes only.

## A. Participants

The demographics of our participants are summarized in Table III. Majority of participants were between 20 and 30 years of age, and right handed. The proportion of male participants in the experiment was slightly higher than that of the female participants. Also the proportion of participants with a background in CS/IT/security exceeded those with no such background. Further, all participants had at least an undergraduate degree and belonged to the same nationality. We found no statistically significant difference in gender, handedness, age or background of participants across two groups (Fischer's Exact Test, $p > 0.01$).

During experiment, we also asked participants questions related to the mobile device they own and the screen lock they use (if any) to protect it. These device and lock statistics are also presented in Table III. A large fraction of participants used Android phones. Further, majority of them were familiar with the Android pattern lock scheme. We found no statistically significant difference in the mobile OS experience and pattern lock familiarity across two groups (Fischer's Exact Test, $p > 0.01$).

| | *Control* | *Experimental* |
|---|---|---|
| **Gender** | | |
| Male | 51.02% | 56.00% |
| Female | 48.98% | 44.00% |
| **Handedness** | | |
| Right | 95.92% | 98.00% |
| Left | 4.08% | 2.00% |
| **Age Group** | | |
| 20-25 | 67.35% | 80.00% |
| 26-30 | 26.53% | 16.00% |
| 31-35 | 6.12% | 4.00% |
| **Background** | | |
| CS/IT/Security | 59.18% | 54.00% |
| Others | 40.82% | 46.00% |
| **Mobile OS** | | |
| Android | 95.92% | 92.00% |
| iOS | 2.04% | 6.00% |
| Windows | 2.04% | 2.00% |
| **Screen-lock** | | |
| Android Pattern | 59.18% | 58.00% |
| PIN | 30.61% | 38.00% |
| Password | 24.29% | 26.00% |
| Fingerprint | 38.78% | 42.00% |
| Slide-to-lock | 24.29% | 26.00% |
| None | 10.20% | 4.00% |
| #Participants | 49 | 50 |

TABLE III: Demographics of participants in control and experimental groups.

## B. Procedure

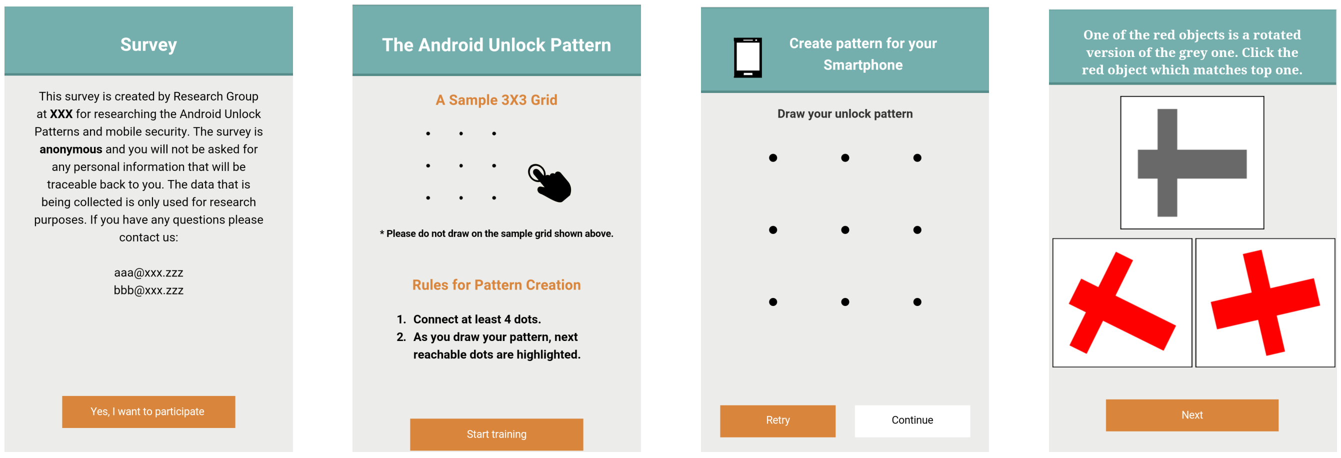
The structure of our study is similar to the one described in [15]. The study was conducted in lab in 6 stages: 1) Information, 2) Brief introduction to Android Lock Patterns, 3) Pattern creation, 4) Distraction task, 5) Questions on demographics, device and screen-lock, and 6) Pattern recall. The details of each stage are given below.

1) *Information.* Participants were requested to open the study link on their mobile device. On visiting the link, they were shown the information page (Figure 5a) which contained a brief information about the study, its purpose and the data usage policy. This page was common for both the groups.
2) *Introduction to Pattern Lock.* After agreeing to participate in the study, participants were shown a sample $3 \times 3$ grid along with the pattern creation rules. The control group participants were informed about the four pattern creation rules [25] while the experimental group participants were informed about the highlighting feature as depicted in Figure 5b. To ensure that participants in the control group become familiar with the existing $3 \times 3$ interface and participants in the experimental group become familiar with TinPal, we also had a training page where the participants could explore the assigned interface by creating as many patterns as they like.
3) *Pattern Creation.* After completing the training, participants were given the smartphone scenario [15], in which they were asked to create a new pattern to protect their mobile device. Participants in the control group created pattern on the existing $3 \times 3$ interface while participants in the experimental group created pattern on our highlighting interface, TinPal. After submitting a valid pattern, participants were asked to re-confirm their pattern. If the confirmation failed, participants could try creating the pattern again.
4) *Distraction Task.* After creating the pattern, participants in both groups were asked to solve mental rotation puzzles. Specifically, participants were shown a target object at the top and they had to identify which one of the bottom two objects matches the target as illustrated in Figure 5d. All participants had to attempt three such puzzles. The purpose of these puzzles was to clear their visual working memory. The dataset required for building the puzzles was used from [4].
5) *Questions.* To further prolong the recall stage, we asked participants few questions pertaining to demographics, their mobile device and the screen-lock they use (if any).
6) *Pattern Recall.* Finally, participants were asked to recall their pattern within five attempts. Participants in the control group used the existing pattern lock interface while participants in the experimental group used TinPal.

## C. Limitations

The sample used in our comparative study is younger and more tech-savvy, and therefore, may have better memory than average which could influence the results. Further, as the study was conducted in lab, the sample is small (99 participants) and with a large sample we could observe further patterns. However, the objective of our study was to determine whether informing users about connectivity rules through the highlighting mechanism had any impact on their pattern choices which we found to be statistically significant with this small sample. We note that our sample size (99 participants) is larger than that of Cho *et al.* lab study (46 participants) [13].

Further, it is not easy to demonstrate if patterns created in the study are realistic. Fahl *et al.* [14] attempted to address the


Survey

This survey is created by Research Group at **XXX** for researching the Android Unlock Patterns and mobile security. The survey is **anonymous** and you will not be asked for any personal information that will be traceable back to you. The data that is being collected is only used for research purposes. If you have any questions please contact us:

aaa@xxx.zzz
bbb@xxx.zzz

Yes, I want to participate

The Android Unlock Pattern

A Sample 3X3 Grid

* Please do not draw on the sample grid shown above.

Rules for Pattern Creation

1. Connect at least 4 dots.
2. As you draw your pattern, next reachable dots are highlighted.

Start training

Create pattern for your Smartphone

Draw your unlock pattern

Retry
Continue

One of the red objects is a rotated version of the grey one. Click the red object which matches top one.

Next

(a) Information Screen (b) Introduction to Pattern Lock (TinPal) (c) Create Pattern (d) Solve Puzzle

Fig. 5: Selected screens of the user study.

questions related to ecological validity in password studies and found that passwords created by participants asked to role-play a scenario resemble their real passwords. They also found that passwords created in the lab study were more representative of actual passwords than those in the online study and priming subjects did not make any difference. Therefore, we conducted our study in lab and asked participants to create a pattern for the smartphone scenario.

## V. Results

To evaluate the impact of TinPal on users' pattern choices, we compare certain characteristics of patterns created in the control group (existing interface) and the experimental group (TinPal). Specifically, we look at pattern characteristics such as pattern length, stroke length, start points, end points, direction changes, knight moves, overlaps, intersections and trigrams. These characteristics are considered to be effective in preventing guessing attacks [7], [25], [6], [8], [24] and shoulder-surfing attacks [22], [21], [28]. We also measure the guessability of patterns created in both groups using an attack technique based on $n$-gram Markov model [25], [8], [20], [24], [13]. Finally, we investigate the usability aspects like memorability (recall attempts) and efficiency (creation and recall time) of the two groups.

### A. Pattern Characteristics

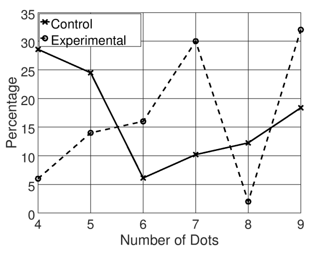
Table IV compares the characteristics of patterns created in two groups. The mean, the median and the standard deviation of each pattern characteristic along with the result of statistical analysis is shown in the table. As the data is not normally distributed (Shapiro-Wilk test rejects the null hypothesis that data is normal with a $p < 0.01$), we use Wilcoxon-Mann-Whitney test [16] to determine whether pattern characteristics on the two interfaces differed significantly. We use a significance level of $\alpha = 0.01$. Since we perform statistical tests on six different pattern features (Table IV) with $\alpha = 0.01$, we apply Bonferroni correction and alter its value to $0.01/6 = 0.0016$ and claim statistical significance if $p < 0.0016$.

**Pattern Length.** It is the most basic feature and represents the number of dots connected in the pattern [25], [6], [22], [21], [8], [28], [24]. Figure 6a depicts the distributions of pattern length in both groups. The number of control group participants who used more than six dots to connect their pattern is only 40.82% while the number of such experimental group participants is 64%. However, there does not appear to be a very significant difference in the number of dots used for connecting patterns across both groups ($p = 0.006870 > 0.0016$).

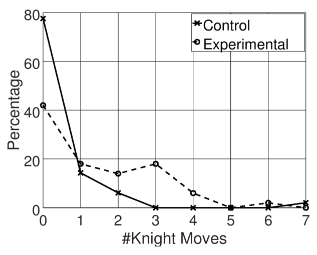
| *Characteristic* | *Control* | *Experimental* | *p-value* | *significant* |
|---|---|---|---|---|
| **Pattern Length** | | | | |
| Mean | 6.08 | 7.04 | 0.006870 | no |
| Standard deviation | 1.90 | 1.60 | | |
| Median | 5.00 | 7.00 | | |
| **Stroke Length** | | | | |
| Mean | 6.05 | 8.39 | 0.0000074 | **yes** |
| Standard deviation | 2.63 | 2.69 | | |
| Median | 5.41 | 8.27 | | |
| **Knight Moves** | | | | |
| Mean | 0.41 | 1.36 | 0.0000590 | **yes** |
| Standard deviation | 0.94 | 0.92 | | |
| Median | 0.00 | 1.00 | | |
| **Overlaps** | | | | |
| Mean | 0.12 | 0.72 | 0.000585 | **yes** |
| Standard deviation | 0.33 | 1.02 | | |
| Median | 0.00 | 0.00 | | |
| **Direction Changes** | | | | |
| Mean | 1.57 | 2.9 | 0.000408 | **yes** |
| Standard deviation | 1.40 | 1.91 | | |
| Median | 2.00 | 3.00 | | |
| **Intersections** | | | | |
| Mean | 0.43 | 0.64 | 0.041358 | no |
| Standard deviation | 1.99 | 1.25 | | |
| Median | 0.00 | 0.00 | | |
| **#Trigrams** | 105 | 159 | | |
| **#Patterns** | 49 | 50 | | |

TABLE IV: Comparison of pattern characteristics across two groups.

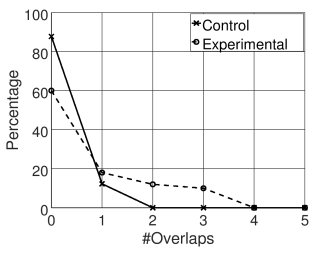
**Stroke Length.** The line segments used for creating $3 \times 3$ patterns are not all similar. For instance, the line segments $1 \rightarrow 2$, $1 \rightarrow 3$, $1 \rightarrow 5$, $1 \rightarrow 6$ and $1 \rightarrow 9$ all have different physical lengths. To capture this distance notion, we use the concept of stroke length [8], [24] which is defined as the sum of Euclidean distances of all line segments within the pattern. In order to compute the Euclidean distance of a line segment,



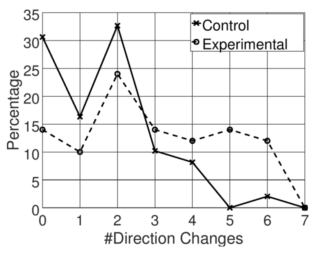
(a) Pattern length



(b) Knight moves



(c) Overlaps



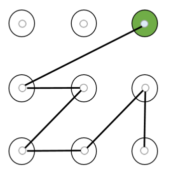
(d) Direction changes

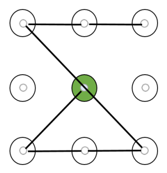
Fig. 6: Distribution of length, knight moves, overlaps and direction changes across two groups.

we label the upper-left dot in the $3 \times 3$ grid as (0,0) and the lower-right dot as (2,2). Thus, the Euclidean distance of the segment $1 \rightarrow 2$ is 1, that of $1 \rightarrow 5$ is $\sqrt{2}$, that of $1 \rightarrow 3$ is 2, that of $1 \rightarrow 6$ is $\sqrt{5}$ and that of $1 \rightarrow 9$ is $2\sqrt{2}$.

Stroke length of a pattern is considered as an important feature in countering shoulder-surfing attacks [22] as well as guessing attacks [8], [24]. The stroke length of patterns in the experimental group (8.39) is significantly higher than the stroke length of patterns in the control group (6.05). Normalizing the stroke length with respect to the pattern length reveals the mean length of the line segment used for creating patterns in both groups. The normalized stroke length of patterns[1] in the control group is $\frac{6.05}{6.08} \approx 1$ while in the experimental group it is $\frac{8.39}{7.04} \approx 1.19$ (19% increase).

**Knight Moves.** The concept of a knight move is similar to the one defined in the game of chess. A knight move occurs when a dot is connected to another dot, that is two units away in the horizontal (vertical) direction and one unit away in the vertical (horizontal) direction. In other words, line segments with the Euclidean distance of $\sqrt{5}$ are referred to as knight moves. For instance, the segment $3 \rightarrow 4$ in the pattern 3457869 is a knight move (Figure 7a). The number of knight moves is considered to be an important feature in thwarting both shoulder-surfing attacks [28] and guessing attacks [6]. The distribution of knight moves in patterns across both groups is shown in Figure 6b. Only 22.45% of the patterns in the control group had at least one knight move, while the proportion of such patterns in the experimental group is 58%. Overall we found that patterns on TinPal were created using significantly more knight moves than those on the existing interface ($p = 0.0000590$).


(a) Pattern with knight move


(b) With overlap move

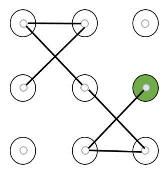

(c) With intersections

Fig. 7: Patterns with knight move, overlap move and intersections.

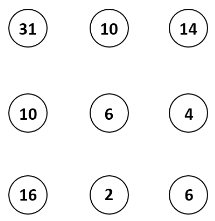
**Overlap.** A connection between two dots in a pattern is referred to as an overlap, if the dot between them is already connected. In other words, the line segments with the Euclidean distance of either 2 or $2\sqrt{2}$ are referred to as overlaps. For instance, the line segment $9 \rightarrow 1$ in the pattern 5789123 constitutes an overlap (Figure 7b). Its length is $2\sqrt{2}$. Another instance of an overlap is the connection $7 \rightarrow 9$ in the pattern 8792615. Its length is 2. The number of overlap moves is considered to be an important feature in resisting both shoulder-surfing [28], [22], [21] and guessing attacks [6]. The distribution of overlaps in patterns across both groups is shown in Figure 6c. Only 12.24% of the participants in the control group created their pattern with at least one overlap whereas the number of such participants in the experimental group is 40%. Overall results indicate that participants in the experimental group created patterns with significantly more overlaps than participants in the control group ($p = 0.000585$).

**Direction Changes.** A direction change occurs when two consecutive line segments in a given pattern have different Euclidean distances [24]. For instance, two consecutive line segments $3 \rightarrow 4$ and $4 \rightarrow 5$ in the pattern 3457869 constitute a direction change since these segments have different Euclidean distances ($\sqrt{5}$ and 1 respectively). Simple patterns such as 321456987 ('S' shape) composed of unit distance line segments do not comprise any direction change [24]. The distribution of direction changes used while creating patterns across two groups is shown in Figure 6d. Only 20.40% of the participants in the control group used more than two direction changes, while the proportion of such patterns in the experimental group is 52%. Overall results show that participants in the experimental group changed direction more often in their pattern than participants in the control group ($p = 0.000408$).

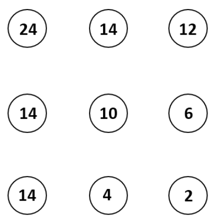
**Intersections.** An intersection occurs when two non-consecutive line segments in a pattern cross each other. For instance, the line segments $6 \rightarrow 8$ and $9 \rightarrow 5$ in the pattern 6895124 intersect each other (Figure 7c). Here another intersection occurs between the line segments $5 \rightarrow 1$ and $2 \rightarrow 4$. The number of intersections in a pattern is considered as an important feature in countering shoulder-surfing attacks [22], [21], [28]. After analysing the data sets, we found that the highlighting mechanism had no effect on the number of intersections in the patterns across two groups ($p > 0.0016$).

**Start and End Points.** As reported in the previous studies [7], [25], [6], [8], [24], the upper-left dot is the most popular starting choice and the lower-right dot is the most popular ending choice for creating patterns in both groups (Figure 8). About 31% of the participants in the control group and 24% in

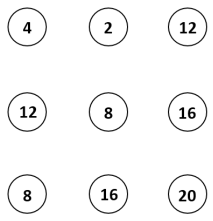
[1]We used the ratio of means since it is always less than or equal to the means of ratio by Jensens' inequality.



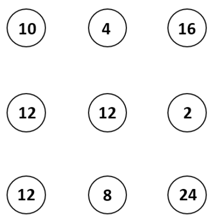
(a) Start Point (Control)



(b) Start Point (Experimental)

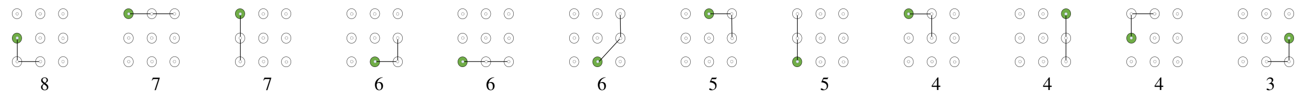


(c) End Point (Control)

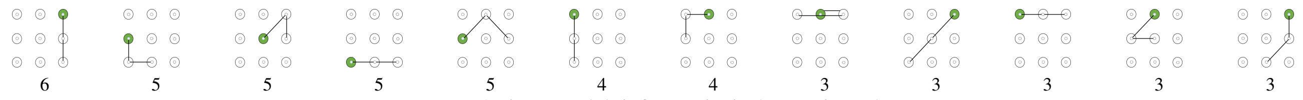


(d) End Point (Experimental)

Fig. 8: Start and end point distributions across two groups (percentages are rounded to the nearest integer).



Top 12 trigrams and their frequencies in the control group.



Top 12 trigrams and their frequencies in the experimental group.

Fig. 9: Top 12 trigrams in the datasets along with their frequencies.

the experimental group began their patterns from the upper-left dot whereas about 20% of the participants in the control group and 24% in the experimental group ended their patterns on the lower-right dot. We estimate the uncertainty in the start and end point choices of both groups using the *Entropy* measure:

$$Entropy\ H = \sum_{i=1}^{9} p_i \cdot log_2(1/p_i) \tag{1}$$

where $p_i$ is the probability of choosing a dot $i$ as a start (or end) point. The uncertainty in the start point choices of the control group and experimental group is similar, 2.82 bits and 2.93 bits respectively. The uncertainty in the end point choices of the control group and experimental group is also similar, 2.96 bits and 2.94 bits respectively. Thus, the highlighting mechanism did not seem to have any effect on the start and end point choices of the participants in the experimental group.

**Trigrams.** Any continuous sequence of 3 dots in a pattern constitutes a trigram, *e.g.*, the trigrams in the pattern 34578 are 345, 457 and 578. Patterns in the control group were composed using total 200 (105 distinct) trigrams whereas patterns in the experimental group were composed using total 252 (159 distinct) trigrams. Top 12 trigrams for each group along with their frequencies are shown in Figure 9. These 12 trigrams constitute about $1/3^{rd}$ of the total trigrams in the control group and $1/5^{th}$ of the total trigrams in the experimental group. Thus, the trigrams used for creating patterns in the experimental group were more diverse than the control group.

## B. Pattern Guessability

We measure the guessability of patterns created in the control group (existing interface) and the experimental group (TinPal) using Markov model based attack technique described in [25]. Markov models exploit the fact that the subsequent choices in a human-generated sequence are based on the previous choices. For instance, in English language letter *h* is most likely to follow letter *t* than letter *p*. In case of $3 \times 3$ patterns, dot 2 is more likely to follow dot 1 than dot 9. The $n$-gram Markov model predicts the next letter in a sequence based on the past $n-1$ choices. The probability of a $l$ length sequence $s_1 s_2 \dots s_l$ is modelled as:

$$P(s_1 \dots s_l) = P(s_1 \dots s_{n-1}) \cdot \prod_{i=n}^{l} P(s_i | s_{i-n+1} \dots s_{i-1}) \tag{2}$$

In addition to training Markov model on our dataset, we also train another Markov model on 69,797 $3 \times 3$ patterns collected by Tupsamudre *et al.* [24] (ASIACCS'17 dataset) and report the guessability results using both models. Since Android enforces a policy that allows a maximum of 20 failed attempts [13], we focus on pattern cracking results for the first 20 guesses only.

**Our dataset.** We need $9 \cdot 8 = 72$ values to learn 2-grams and $9 \cdot 8 \cdot 7 = 504$ values to learn 3-grams. Since the control group dataset has 49 patterns and corresponding 298 datapoints that are insufficient to learn 3-gram probabilities, we resort to 2-grams and use Laplace smoothing to account for unseen 2-grams. We perform 10-fold cross-validation on each pattern set, *i.e.*, we divide each pattern set into 10 approximately equal-sized subsets, one of the subset is used as test set and the remaining 9 subsets are combined into training set. We estimate the probabilities of all possible patterns (389,112) using training set, sort them in descending order of probabilities and simulate guessing attack on the test set. We repeat this validation 10 times where every subset is used as a test set. We perform this entire process 10 times and report the average results. Within first 20 attempts, the guessing algorithm could crack 12% patterns in the control group and none in the experimental group.

**ASIACCS'17 dataset.** We also had access to 69,797 $3 \times 3$ patterns collected by Tupsamudre et al. [24]. Since their dataset is huge, we train a 3-gram Markov model to estimate the probabilities of all possible patterns (389,112), and simulate guessing attacks on our datasets. Within first 20 attempts, the guessing algorithm could crack 12.24% patterns in the control group and 4% patterns in the experimental group.

## C. Usability Results

Now, we present the results pertaining to usability. Specifically, we compare memorability and efficiency of patterns created on the existing interface with those created on TinPal. As the usability data, *i.e.*, login attempts, pattern creation time and pattern recall time are not normally distributed, we use Wilcoxon-Mann-Whitney test [16] and claim the results to be statistically significant if $p < 0.01$.

**Memorability.** We measure memorability using the following two metrics:

- number of users who successfully recalled their pattern.
- number of login attempts required to recall the pattern.

Before advancing to the pattern recall stage, participants in both groups spent 3.72 minutes on an average solving the distraction task (Figure 5d) and answering questions related to demographics, their devices and screen locks. The results pertaining to pattern recall are shown in Table V. 97.96% (48/49) of the participants in the control group and 98% (49/50) of the participants in the experimental group successfully recalled their pattern within three attempts. We found no significant difference in the login attempts of both groups (FET, $p = 1$).

| | *Control* | *Experimental* |
|---|---|---|
| Attempt 1 | 35 (71.43%) | 39 (78.00%) |
| Attempt 2 | 11 (22.45%) | 8 (16.00%) |
| Attempt 3 | 2 (4.08%) | 2 (4.00%) |
| Successful | 48 (97.96%) | 49 (98.00%) |
| Recall time | 1.71s | 2.24s |

TABLE V: Login attempts of participants across two groups.

**Efficiency.** We measure efficiency using two metrics:

- time required to create the pattern during creation stage.
- time required to recall the pattern during recall stage.

Before advancing to the pattern creation stage, participants in both groups spent some time in the training stage. This stage was exploratory and provided participants with an opportunity to become familiar with the assigned interface. Participants in the control group explored 1.65 patterns on an average on the existing interface whereas participants in the experimental group were more curious about TinPal and explored 5.6 patterns on an average. The average time required to create a pattern in the control group is 2.10s while the time required in the experimental group is 3.50s. We found significant difference in the creation time between two groups ($p = 0.000078$) which suggests that participants in the experimental group spent relatively longer time creating their pattern and paid attention to the highlighted dots.

The average time required to recall a pattern in the control group (1.71s) is also less than that required in the experimental group (2.24s). As a consequence, there is a significant difference in the recall time between two groups ($p = 0.00215$). However, we note that the average stroke length of patterns created in the control group (6.05) is less than those created in the experimental group (8.39). After normalizing the recall time with respect to the stroke length, the recall time of patterns in the control group ($\frac{1.71}{6.05} \approx 0.28s$) is similar to the recall time of patterns in the experimental group ($\frac{2.24}{8.39} \approx 0.27s$).

**Acceptability.** We asked participants in the experimental group an open-ended question: "*Which pattern lock interface do you prefer, the existing one that you have used before (on your phone) or the new one that you saw in our experiment?*" Of the 29 (58%) experimental group participants who reported using Android pattern screen-lock before (Table III), 93.10% said that they would prefer the new one used in the experiment while 6.90% said that they would prefer the existing one. Some of the remarks made by participants are as follows.
*'The new interface is better because of feedback feature.'*
*'The new one gives more idea about the variety of patterns we can make.'*
*'I prefer the existing interface, need simple patterns only.'*

## VI. Discussion and Future Work

In this work, we propose a new $3 \times 3$ interface, TinPal, which informs users about different available connection options during both pattern creation and its recall. The results of our comparative study indicate that TinPal influenced users' pattern choices without much affecting the usability. Participants who used TinPal created patterns with longer strokes and visually complex features such as knight moves, overlaps and direction changes compared to those who used the existing interface. Guessability results show that patterns created on TinPal are more guessing-resistant than those created on the existing interface. These results are encouraging as TinPal just informed users about all available options and did not force or nudge them towards a particular option.

Our data indicates that TinPal had no effect on the start point choices of the users, and they remain biased. For example, the upper-left dot is still the most popular choice. One way to reduce this bias is to suggest a random starting point to the user as done in [20]. After the user connects the suggested point, TinPal will come into action and highlight the next set of reachable dots from the connected dot.

SysPal policies [13] mandate users to use 1 to 3 randomly chosen dots in their pattern. However, these policies do not ensure that the resulting patterns will be composed using knight moves ($1 \rightarrow 6$) or overlaps ($1 \rightarrow 3$) as users may not be aware of such connection options. On the other hand, our study results show that patterns drawn on TinPal use significantly longer strokes and large number of knight moves, overlaps and direction changes. SysPal policies can be combined with TinPal so that along with mandating users to use randomly chosen dots in their pattern they also inform users about all possible connection options which could further strengthen the security of the pattern lock scheme.

It would be interesting to study the combined effect of TinPal and different pattern strength meters proposed in the literature [6], [22], [21]. With the existing interface, users might not be aware of all potential choices for creating their pattern which can reduce the impact of pattern strength meters. The combination of TinPal and a pattern strength meter could be a very effective one. As the user draws her pattern, TinPal will make her aware of all available choices that could be reached from the current dot and at the same time the strength meter will indicate which of the available choices are secure.

## References

[1] "Biometric security poses huge privacy risks," https://www.scientificamerican.com/article/biometric-security-poses-huge-privacy-risks/, accessed on 7 February 2018.

[2] "Ericsson mobility report," https://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-june-2017.pdf, accessed on 7 February 2018.

[3] "How to set a pattern lock in android," https://www.wikihow.tech/Set-a-Pattern-Lock-in-Android, accessed on 7 February 2018.

[4] "Mental rotation task," http://www.psytoolkit.org/experiment-library/mentalrotation.html, accessed on 7 February 2018.

[5] "Unlock with your fingerprint," https://support.google.com/pixelphone/answer/6285273, accessed on 7 February 2018.

[6] P. Andriotis, T. Tryfonas, and G. Oikonomou, "Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method," in *Proceedings of the Second International Conference on Human Aspects of Information Security, Privacy, and Trust - Volume 8533*. New York, NY, USA: Springer-Verlag New York, Inc., 2014, pp. 115–126. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-07620-1

[7] 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. [Online]. Available: http://doi.acm.org/10.1145/2462096.2462098

[8] A. J. Aviv, D. Budzitowski, and R. Kuber, "Is bigger better? comparing user-generated passwords on 3x3 vs. 4x4 grid sizes for android's pattern unlock," in *Proceedings of the 31st Annual Computer Security Applications Conference*, ser. ACSAC '15. New York, NY, USA: ACM, 2015, pp. 301–310. [Online]. Available: http://doi.acm.org/10.1145/2818000.2818014

[9] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies*, ser. WOOT'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=1925004.1925009

[10] A. J. Aviv, J. Maguire, and J. L. Prak, "Analyzing the impact of collection methods and demographics for android's pattern unlock," in *Workshop on Usable Security*, ser. USEC 2016. Internet Society.

[11] R. Biddle, S. Chiasson, and P. Van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 19:1–19:41, Sep. 2012. [Online]. Available: http://doi.acm.org/10.1145/2333112.2333114

[12] S. Brostoff and M. A. Sasse, "Are passfaces more usable than passwords? a field trial investigation," in *People and Computers XIV-Usability or Else!*. Springer, 2000, pp. 405–424.

[13] G. Cho, J. H. Huh, J. Cho, S. Oh, Y. Song, and H. Kim, "Syspal: System-guided pattern locks for android," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 338–356.

[14] S. Fahl, M. Harbach, Y. Acar, and M. Smith, "On the ecological validity of a password study," in *Proceedings of the Ninth Symposium on Usable Privacy and Security*, ser. SOUPS '13. New York, NY, USA: ACM, 2013, pp. 13:1–13:13. [Online]. Available: http://doi.acm.org/10.1145/2501604.2501617

[15] M. Loge, M. Duermuth, and L. Rostad, "On user choice for android unlock patterns," in *European Workshop on Usable Security*, ser. EuroUSEC 2016. Internet Society.

[16] A. Marx, C. Backes, E. Meese, H.-P. Lenhof, and A. Keller, "Edison-wmw: Exact dynamic programing solution of the wilcoxon–mann–whitney test," *Genomics, proteomics & bioinformatics*, vol. 14, no. 1, pp. 55–61, 2016.

[17] A. Paivio, T. B. Rogers, and P. C. Smythe, "Why are pictures easier to recall than words?" *Psychonomic Science*, vol. 11, no. 4, pp. 137–138, Apr 1968. [Online]. Available: https://doi.org/10.3758/BF03331011

[18] Y. Rogers, H. Sharp, and J. Preece, "Interaction design: beyond human-computer interaction." John Wiley & Sons, 2011, p. 21.

[19] R. N. Shepard, "Recognition memory for words, sentences, and pictures," *Journal of verbal Learning and verbal Behavior*, vol. 6, no. 1, pp. 156–163, 1967.

[20] H. Siadati, P. Gupta, S. Smith, N. Memon, and M. Ahamad, "Fortifying android patterns using persuasive security framework," *UBICOMM 2015*, p. 81, 2015.

[21] Y. Song, G. Cho, S. Oh, H. Kim, and J. H. Huh, "On the effectiveness of pattern lock strength meters: Measuring the strength of real world pattern locks," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 2343–2352. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702365

[22] C. Sun, Y. Wang, and J. Zheng, "Dissecting pattern unlock," *J. Inf. Secur. Appl.*, vol. 19, no. 4, pp. 308–320, Nov. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jisa.2014.10.009

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

[24] H. Tupsamudre, V. Banahatti, S. Lodha, and K. Vyas, "Pass-o: A proposal to improve the security of pattern unlock scheme," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS '17. New York, NY, USA: ACM, 2017, pp. 400–407. [Online]. Available: http://doi.acm.org/10.1145/3052973.3053041

[25] 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. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516700

[26] 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, pp. 102–127, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.ijhcs.2005.04.010

[27] J. C. Yuille, *Imagery, memory, and cognition: Essays in honor of Allan Paivio*. Lawrence Erlbaum Assoc Incorporated, 1983.

[28] E. v. Zezschwitz, A. De Luca, P. Janssen, and H. Hussmann, "Easy to draw, but hard to trace?: On the observability of grid-based (un)lock patterns," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 2339–2342. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702202

[29] E. v. Zezschwitz, P. Dunphy, and A. De Luca, "Patterns in the wild: A field study of the usability of pattern and pin-based authentication on mobile devices," in *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, ser. MobileHCI '13. New York, NY, USA: ACM, 2013, pp. 261–270. [Online]. Available: http://doi.acm.org/10.1145/2493190.2493231