

Augmented Unlocking Techniques for Smartphones Using Pre-Touch Information

Matthew Lakier
mlakier@uwaterloo.ca
University of Waterloo

Yixin Wang
y3244wan@uwaterloo.ca
University of Waterloo

Dimcho Karakashev
dzkaraka@uwaterloo.ca
University of Waterloo

Ian Goldberg
iang@uwaterloo.ca
University of Waterloo

ABSTRACT

Smartphones store a significant amount of personal and private information, and are playing an increasingly important role in people's lives. It is important for authentication techniques to be more resistant against two known attacks called shoulder surfing and smudge attacks. In this work, we propose a new technique called 3D Pattern. Our 3D Pattern technique takes advantage of a new input paradigm called pre-touch, which could soon allow smartphones to sense a user's finger position at some distance from the screen. We implement the technique and evaluate it in a pilot study ($n=6$) by comparing it to PIN and pattern locks. Our results show that although our prototype takes about 8 seconds to authenticate, it is immune to smudge attacks and promises to be more resistant to shoulder surfing.

1 INTRODUCTION

Smartphones are increasingly used to store private information such as personal photos, contacts, and financial information. However, smartphones are also frequently used in public spaces or in social gatherings, necessitating the protection of this private information via user authentication. Authentication or "unlocking" techniques include the common manual (e.g., PINs and gesture-based pattern locks) and biometric (e.g., fingerprint reading, iris scanning, and face recognition) techniques, and less commonly *continuous authentication* techniques, which continuously monitor the user's behaviour, such as touch or swipe patterns, locking the device if it believes a different person has started to use it.

In this work, we focus on manual authentication techniques, because they are among the most common techniques used on smartphones as a way to protect private information. Even users utilizing fingerprint readers often are required to enter a PIN for added security, for example, when rebooting or authorizing payments.

Harbach et al. [10] showed that bystanders looking at other people's phones as they type their PIN ("shoulder surfers") are able to reliably deduce the PINs. This unfortunately hinders the effectiveness of PINs. Many different PIN entry techniques have been proposed to improve shoulder-surfing resistance, such as scrambled keyboards [23], haptic and sound-based PINs [2], gestures (including swiping) [25], graphical PINs [5], and techniques based on remembered user behaviour [3]. In general, techniques have a tradeoff between performance and resistance to shoulder surfing [10].

Recently, there has been a move towards creating systems that support *pre-touch sensing*, that is, using information about user's fingers just before the screen is actually touched [12, 15, 26]. Similarly to how pre-touch information has been used for expanding target selection [27], we identify an opportunity to apply pre-touch sensing to improve the effectiveness of PIN entry techniques. We create a novel version of the Android pattern lock that expands the traditional 3×3 grid out of the screen into a $3 \times 3 \times 3$ cube. Points are connected by moving a finger in 3D space above the surface of the phone. Because pre-touch information is not available on current smartphones, we simulate pre-touch using a motion capture system. This enables us to implement a prototype version of the 3D Pattern lock.

A pilot study with six participants shows that the 3D Pattern technique is slower and more error-prone than the PIN and pattern techniques. These results could be partially attributed to the novelty of pre-touch input and the motion-capture approach used by the prototype, both of which would be ameliorated when pre-touch input becomes available in commonly used devices. We also find that the 3D Pattern technique has increased shoulder-surfing resistance compared to PINs. Further, the empirical CDF indicates that a larger study could reveal an improvement in shoulder-surfing resistance over the pattern technique as well. Finally, the 3D Pattern technique is naturally immune to smudge attacks [1].

The main contributions of this paper are (1) the design of a novel smartphone authentication technique called 3D Pattern using pre-touch, and (2) an implementation and evaluation of the 3D Pattern technique in comparison to conventional PIN and pattern locks.

2 BACKGROUND AND RELATED WORK

In this section, we discuss different types of attacks against smartphone authentication techniques, as well as past techniques designed to defend against these attacks.

2.1 Shoulder Surfing

Shoulder surfing is a widely known attack in which the adversary tries to infer the victim's authentication secret by looking over his or her shoulder. There is a significant body of research into mitigating the impact of shoulder-surfing attacks. An in-depth survey conducted by Eiband et al. [7] considered the threat not only in the context of authentication, but also in the context of routine smartphone usage. The survey showed that 130 out of 174 participants indicated that shoulder-surfing attacks occurred on public transportation. Victims most commonly defended against

such an attack by modifying their posture or cancelling the authentication. Furthermore, a study conducted by Harbach et al. [11] found the perceived risk of shoulder surfing to be high in only 11 of 3410 situations. This demonstrates that people are not actively defending themselves against shoulder surfing, and more work is needed to improve the shoulder-surfing resistance of authentication techniques.

There is significant amount of previous work that has attempted to address shoulder surfing, but the proposed solutions either do not adequately defend against shoulder surfing or result in other problems such as longer authentication times or increased error rates.

2.2 Smudge Attacks

PIN keypads and pattern locks are commonly used methods for phone authentication. Unfortunately, these techniques are vulnerable to smudge attacks, because the user leaves oily residues on the screen. Previous work has demonstrated that smudge attacks are especially effective on pattern locks as users drag their fingers over the screen. Smudge attacks can also be used to limit the input space for PIN locks. Aviv et al. [1] found that as long as the line of sight is not perpendicular, it is easy to observe entered patterns based on smudges. Under ideal conditions, 92% of the patterns entered were partially identifiable and 68% of the entered patterns were fully recoverable. Under less ideal conditions, 37% of the patterns were partially recoverable and 14% were fully recoverable.

These results demonstrate that even if the adversary is not able to actively observe the process of authentication, he or she can still recover the password with considerable success. In our work, we leverage pre-touch information to limit the number of touches the user makes on the screen, mitigating the effect of smudge attacks.

2.3 PIN and Password Locks

Before the advent of smartphones, Tan et al. [24] devised a password entry technique designed for a computer mouse; however, a similar method could be applied for touchscreens. In this technique, the user presses left and right to subtly highlight a letter on a scrambled keyboard. The user then uses the mouse to drag a tile on top of this letter. While dragging, the scrambled keyboard letters disappear so an onlooker cannot tell which letter is being selected.

Now that smartphones are commonplace, traditional authentication techniques have been adapted to work on the small touchscreens of smartphones. Kovelamudi et al. [16] compared speed and shoulder-surfing resistance of a scrambled PIN entry keypad and a normal PIN entry keypad. They found that the scrambled keypad was slower but more resistant to shoulder surfing.

Several works have examined the possibility of augmenting PIN keypads with gestures. SwiPIN, by von Zezschwitz et al. [25], divided the PIN keypad into two sections. Each number in each section corresponded to a different swipe gesture direction. Performing a swipe gesture on the correct section of the screen would insert the corresponding number. Their study demonstrated that this technique improved resistance against smudge attacks. Khan et al. introduced “ForcePINs” [14], with which each PIN digit could be entered with different levels of finger pressure on the screen, to add an additional layer of challenge for shoulder surfers. However,

results showed that there was no statistically significant difference in shoulder-surfing resistance between regular PINs and ForcePINs, because when users pressed harder, they also pressed for a noticeably longer time.

Other works have looked beyond purely visual representations of PINs by incorporating haptic and audio feedback. Bianchi et al. [2] created an observation-resistant authentication technique by providing no visual clues to the user. The technique renders a wheel on the screen with identical sections. However, when users drag their fingers over the sections of the wheel, tactile feedback is presented with varying lengths and strengths. To select a section, users drag their fingers to the middle of the wheel. After each entry, the sections are shuffled to provide resistance against smudge attacks. Similarly, VibraInput [17] used an on-screen, rotary wheel with two levels. The outer level contained the letters A through D, each corresponding to a fixed vibration pattern (that has to be remembered by user). The inner level corresponded to the PIN numbers 0 through 9. Upon starting PIN entry, the phone would vibrate the pattern of a letter. The user would then rotate the outer wheel to align the letter with the number to select on the inner wheel. By repeating this process, the technique could use process of elimination to ascertain the PIN number. The overall technique would repeat until the entire PIN was entered.

Two-Thumbs-Up (TTU) [20] prevents shoulder-surfing attacks by requiring the user to cover the screen with their hands. This forms a “handshield” and enters a challenge mode. If users move their hands away from the screen, the authentication technique disappears. TTU randomly associates five “response” letters with two digits each, presenting the digits and letters on either side of the screen. The user then has to tap on the letter corresponding to the next PIN digit. After a certain number (dependent on PIN length) of correctly selected letters, the authentication process is complete.

2.4 Pattern Locks

Harbach et al. [10] focused on comparing PIN locks and pattern locks. They were able to observe the behaviour of 134 smartphone users over one month, revealing differences between the two techniques. Results showed that although pattern locks are faster, users are six times as likely to make mistakes compared to PIN locks. When including failed attempts, there were no differences in authentication time between the two techniques. When a user made a mistake entering a PIN or pattern, subsequent successful attempts took more time, presumably because the user took more care when repeating the authentication. Visual feedback did not influence the error rate nor the entry time. Similarly, our 3D Pattern technique improves shoulder-surfing resistance by reducing visual feedback during authentication.

De Luca et al. suggested using a stroke-based visual authentication scheme [6], expecting visual patterns to be easier to remember in comparison to traditional numeric PINs or alphanumeric passwords. A similar technique, the pattern lock, was ultimately incorporated into the Android operating system. Unfortunately, as outlined above, pattern locks have been shown to be weak against shoulder surfing and smudge attacks. In contrast, DRAW-A-PIN, by Nguyen et al. [19], has the user draw each PIN digit on the screen

using their finger. Results indicated that this approach was capable of mitigating shoulder surfing attacks.

2.5 Graphical

Another category of PIN entry techniques uses pictures or other graphics. In SemanticLock [21], users arrange icons on the screen in a memorable way. The user is authenticated based on correct placement of the icons. In a similar work, Awase-E [22], Takada and Koike leverage photos taken on a user’s smartphone. The lock screen breaks a user-chosen photograph up into smaller chunks, and shows nine chunks of various photographs all at once. The user then has to select the tile from the correct photograph four times in a row to unlock the phone.

2.6 Small-Scale Interaction

Several works have attempted to mitigate the impact of smudge attacks by limiting the touch interaction to a small area on the screen. TinyLock [18] resists smudge attacks without trading off usability. Users draw their pattern in a tiny grid, making it harder to observe finger motions due to the small interaction area. To finish authentication, the user rotates a virtual wheel on top of the grid, distorting the smudges from the pattern. Similarly, ClickPattern [9] shows a keypad in a randomly shuffled order in a small area at the bottom of the screen. The user presses the keys to enter numbers corresponding to a pattern. This technique has the same lack of shoulder surfing resistance as the Android pattern lock because the pattern is visualized on the screen. However, it improves smudge attack resistance because of the small input area on the screen. Our 3D Pattern technique avoids the need to limit the interaction area by not requiring the user’s finger to make contact with the screen.

2.7 Behavioural Authentication

There is considerable research exploring whether or not lock screens are even necessary at all, by applying *continuous authentication*, also known as implicit authentication. Continuous authentication systems analyze an individual’s regular patterns of touches on the screen, and build a model. A different user would have different patterns, and could be denied access by the system. With the Touchalytics project [8], Frank et al. were able to use continuous authentication to identify the user with an error rate below 4%, even after a week of elapsed time between training and testing. Unfortunately, Khan et al. [13] showed that an attacker, merely watching a video of the target using their phone, could bypass swipe-based continuous authentication at least 75% of the time.

Some work has explored applying the principles of continuous authentication to augment traditional lock screen techniques. Buschek et al. [3] use spatial touch features in addition to previously used temporal touch features on keyboards to verify users based on their individual text entry behaviours. Examples of spatial touch features include touch offsets, angles, and pressures. By incorporating such spatial features, user recognition accuracy was improved.

2.8 Pre-Touch

Many recent works on touchscreen interactions have started exploring pre-touch information; that is, positional information about

the user’s hands or fingers before making contact with the screen. For example, with TouchCuts and TouchZoom, Yang et al. [27] used pre-touch finger distance to expand nearby targets on screen, facilitating easier target selection. This general approach has not yet been explored in the context of authentication techniques resistant to shoulder surfing.

Another common application of pre-touch information is for reducing the perceived latency of touchscreen interactions. Xia et al. employed this approach for tabletop displays [26], achieving a touch location prediction error of about 1 cm. The approach was implemented by tracking the user’s index finger location using motion capture with fiducial markers, which are small retro-reflective spheres that can be precisely tracked by IR cameras. In the prototype of our 3D Pattern technique, we also use a motion capture system for finger position tracking.

We anticipate pre-touch sensing to become available on commodity smartphones in the near future. In 2016, Hinckley et al. [12] explored how a smartphone with a self-capacitance touchscreen could enable pre-touch information to be sensed, and applied this information in various smartphone applications. We envision that our pre-touch PIN entry techniques will be able to be used on smartphones without additional motion tracking hardware.

3 GOALS AND THREAT MODEL

We assume that the attacker is in close vicinity of the victim during authentication or has a video (e.g., from security camera footage) of the victim authenticating, enabling shoulder-surfing attacks. In regards to smudge attacks, the attacker may have a brief period of physical access to the phone after the victim has authenticated, allowing the attacker time to observe smudges on the screen and potentially deduce the credentials. We consider specialized attacks such as acoustic side-channel attacks [4] to be outside the scope of this work.

With this threat model in mind, we propose the following goals for a pre-touch augmented PIN entry technique:

- protect against shoulder-surfing and smudge attacks;
- leverage pre-touch information of the user’s finger;
- not involve significant extra cognitive load or memorization on the part of the user (e.g., unlike previous acoustic PIN entry techniques, which rely on the user remembering sound associations); and
- be not significantly slower or more error-prone than other conventional techniques (e.g., PIN, Android pattern lock).

4 PIN ENTRY TECHNIQUES

In this section, we describe in detail the three techniques we implemented for PIN entry: PIN, pattern, and 3D Pattern.

Our implementations of the PIN and pattern locks are designed to be as similar as possible to Android’s built-in PIN and pattern locks, respectively. For the PIN keypad, we present the numbers zero through nine in a grid along with the masked-out PIN on the screen. For the pattern lock, we present a 3×3 grid of points on the screen, which users connect together by dragging their finger from point to point. A line is drawn between the points that the user connects.

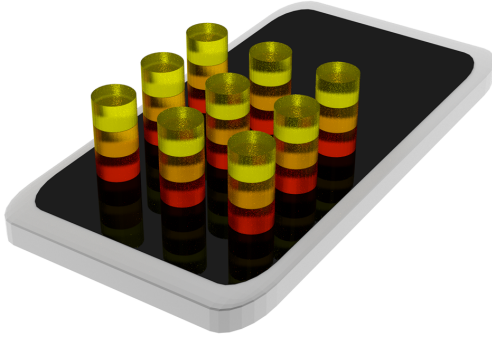
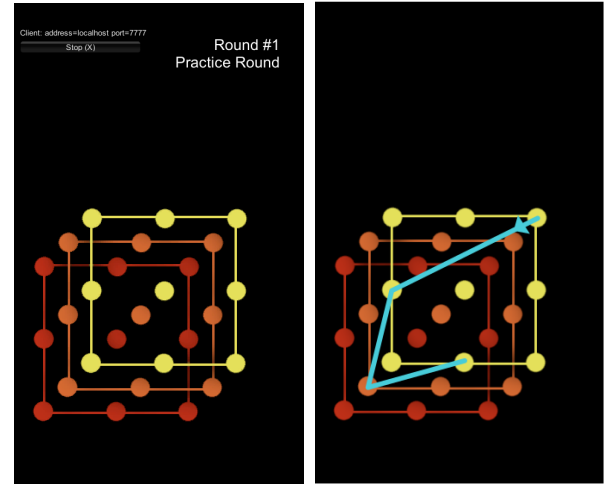


Figure 1: Cylinder representation of our implementation of the 3D Pattern technique. The cylinders are coloured depending on the layer they represent. Users must only hold their finger within the layer of the cylinder to select the corresponding point during authentication.

4.1 3D Pattern Lock

The 3D Pattern lock is inspired by Android’s conventional pattern lock. A simple adaptation of the pattern lock to a pre-touch environment would be to duplicate the normal 3×3 grid of points, but have users enter the pattern with their finger hovering over the screen rather than touching it, thus eliminating the smudge attack vector. This simple “hover pattern” use of pre-touch should maintain most of the security and usability properties of the normal pattern lock, save for being immune to smudge attacks, and so we do not analyze this technique in detail. Instead, our novel idea is to use pre-touch information to extend the pattern lock concept into a full third “z” dimension. Rather than a 3×3 grid of circles, our 3D Pattern lock is a $3 \times 3 \times 3$ cube of cylinders (see Figure 1). By using cylinders, users do not have to think about where to place their fingers within a layer, only about which layer to place their finger within. This simplifies the rendering of depth cues. The smartphone renders an orthogonal projection of the cube (see Figure 2a). Each depth, or “layer”, of the cube is represented in a different colour. The user authenticates by connecting the points in a chosen sequence. Our technique does not require users to slide their fingers on the screen. This inherently protects against smudge attacks since users will not leave oily residues on the screen.

Assuming that the user is allowed to connect any four points such that no point is reused, a theoretical password space upper bound is $27 \times 26 \times 24 \times 23 = 387504$ patterns. However, as a further improvement to usability, we limit the space of valid 3D patterns to include only those that start on the topmost layer, do not bypass the middle layer, do not bypass a point within a layer or use a point more than once (as with the traditional Android pattern lock), and do not connect points across a layer with a distance of more than $\sqrt{3}$ units (to avoid difficult-to-input diagonal lines). Even with these assumptions, the password space for the 3D Pattern technique is still larger than previous conventional techniques. Using a recursive algorithm in Python, we found 19192 possible 3D Patterns. This means that the worst-case password space for our 3D Pattern technique is better than that of both PIN (10000) and



(a) A screenshot of the smartphone screen as the user sees it before authentication using the 3D Pattern technique. (b) An example reference image shown to a participant when authenticating using the 3D Pattern technique.

Figure 2: The on-screen representation of the 3D Pattern technique. The 3D cube is orthogonally projected on the screen, with each layer represented using a different colour.

pattern (1400, computed using a similar Python script as above) locks in the case of a four-digit PIN or pattern.

As with the conventional Android pattern lock, our 3D Pattern lock has two modes: (1) *with feedback* and (2) *without feedback*. In *with feedback* mode, as the user moves his or her finger between the different points, a line is rendered between each of the connected points. In *no feedback* mode, these lines are not rendered.

All three of our implemented techniques have additional haptic feedback. This helps the users perceive whether their input has been detected. In a very quiet environment, an attacker may be able to hear the haptic feedback, but in the ambient noise of the experiment room, the experimenters could not hear or otherwise detect the haptic feedback as participants were authenticating. Similarly, an attacker should not be able to hear the haptic feedback in a public environment.

Our 3D Pattern lock additionally always renders a “cursor” on the screen. The cursor changes colour depending on the finger’s distance from the screen. Yellow represents that the finger is in the closest layer to the user, orange represents the middle layer, and red represents the layer closest to the screen. Figure 3 demonstrates the authentication process.

5 IMPLEMENTATION

Pre-touch information is not yet available on current commercial smartphones. As a result, we simulated pre-touch with fiducial-based motion capture. Our system was set up in a $2\text{ m} \times 2\text{ m} \times 3\text{ m}$ room instrumented with six Vicon motion-capture cameras. The cameras work together to triangulate the position of fiducial markers on the user’s finger. The small grey spheres in Figure 3 are examples of these markers. The motion capture system tracks both



(a) Starting authentication



(b) Connecting second point



(c) Connecting third point



(d) Finished authentication

Figure 3: Study participant authenticating using our 3D Pattern technique. (a) The participant’s finger starts in the yellow layer, the closest layer to the participant. (b) The participant moves his finger diagonally to a point in the middle layer. (c) The participant moves his finger to the point below in the bottom layer. (d) The participant moves his finger diagonally up to a point in the middle layer, and the complete pattern has been entered.

the smartphone and the finger positions. The absolute positions of each of these objects in 3D space is transformed, resulting in finger coordinates relative to the phone screen.

Our prototype implementation includes a main PC, which instruments the remaining parts of the system. The responsibilities of the main PC include determining what to draw on the phone screen, controlling the experiment, verifying PINs, and logging useful information. The three authentication techniques were implemented using the Unity game engine.¹ The source code is available at <https://github.com/spamalot/3D-Pattern-Lock>. All six Vicon cameras are connected to a Vicon server through a network switch. This server calculates the absolute 3D positions of the user’s finger and smartphone and forwards this information to the main PC. The connection between the smartphone and the main PC is implemented as a client-server architecture over a Wi-Fi connection. The smartphone renders the authentication technique to the user and accepts touch and dragging input. The system architecture is depicted in Figure 4. If our system were to be implemented in practice, of course, all computation and sensing would be performed on the smartphone itself.

To evaluate our technique, we compare it to two other popular authentication techniques: PIN and pattern locks. We replicate Android’s implementation of the techniques as closely as possible to make a fair evaluation of our proposed technique. We do this by rendering Android’s layout as a background, and overlaying buttons or swipe zones on top of this image. This preserves the spacing between UI elements present in the Android implementations. For the Pattern lock, we empirically matched the width of each point’s hit box with that of the Android implementation.

6 EXPERIMENT

We conducted an experiment to understand how quickly and accurately users can authenticate using the 3D Pattern technique, as well as the resistance of our technique against shoulder surfing. Our hypotheses are that the 3D Pattern technique will be slightly slower and less accurate than the PIN and pattern techniques, but significantly more shoulder surfing resistant. We received approval from our university’s research ethics board for this experiment involving human participants (ORE# 22114).

We recruited six graduate student participants from the university community via word of mouth. Participants each received a gift card valued at \$5 for their participation in the 30-minute session.

6.1 Participants

We recruited 6 participants (5 male, 1 female) with average age 24 (SD=2) to enter PINs on a mobile phone. All participants were right-handed and used their right hand to authenticate using the three implemented techniques (PIN, pattern, and 3D Pattern). Participants reported using a diverse range of locks on their personal phones, with one participant using a PIN keypad, another using Android pattern lock, three using fingerprint reading, and one using iris scanning.

¹<https://unity3d.com>

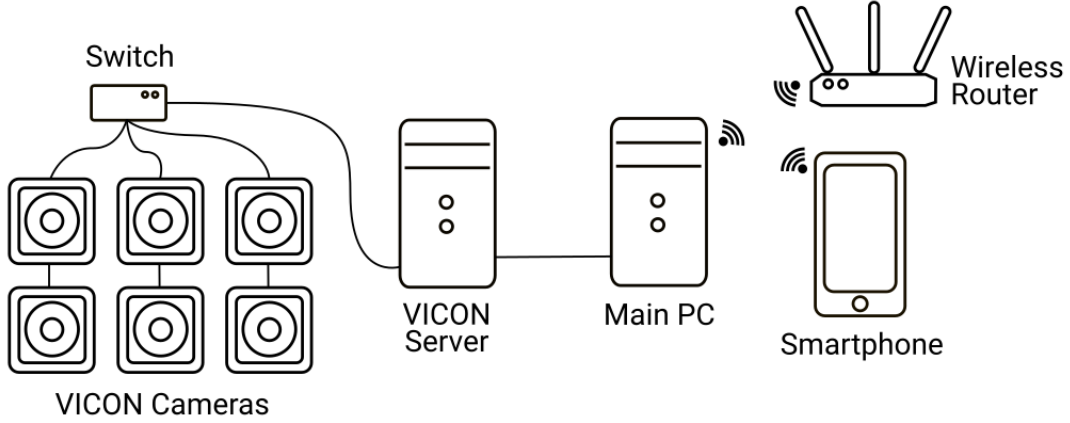


Figure 4: System architecture diagram of the prototype smartphone authentication techniques. Information flow is generally from the left to the right. The Vicon cameras capture the position of fiducial markers in 3D space, and forward this information to a Vicon server through a network switch. The Vicon server calculates the absolute positions of the smartphone and the user’s finger, and forwards this over a link-local connection to the main PC. The main PC establishes a bidirectional Wi-Fi connection with the smartphone through a router. It uses input data from the smartphone in conjunction with motion capture object positions to operate the techniques. In practice, when pre-touch technology becomes commercially available, all computation and sensing will be performed on the smartphone itself.



Figure 5: Room setup used for the experiment. Six Vicon cameras were placed throughout the room. Participants sat on a chair against the far wall, with a video camera pointing over their left shoulder.

6.2 Apparatus

The previously described implementation was used for the experiment. Figure 5 shows the general layout of the experiment room. Participants sat against a wall with no motion capture cameras, to minimize occlusion of the cameras. Participants authenticated on a Google Pixel 3 smartphone. A 1080p video camera was mounted on a tripod, aiming over the left shoulder of the participant. This camera provided a clear view of the phone screen and was used to mimic the view of a shoulder surfer.

6.3 Task

The experiment was divided into two sections. In the first section, the participant was instructed to authenticate using each of the three techniques. Input events on the smartphone were logged on the main PC and videos of participants authenticating were recorded.

All authentication techniques were four “digits” long; that is, PINs had four numbers and patterns involved connecting four points. Each point in the pattern corresponded to a digit. For 3D Pattern lock, the top-left point of each layer corresponded to the digits 0, 9, and 18. Within the same layer, the digits increased left to right, top to bottom. For example, in the layer closest to the screen, the top-left point corresponded to 0, the top-middle point corresponded to 1, and so on.

For the second section, the participant was asked to shoulder surf the videos of the previous participant authenticating; the last successful (most practiced) authentication of each PIN or pattern for each technique was shown. The first participant performed shoulder surfing once the last participant finished authenticating using all techniques. The participant had up to 20 guesses to correctly determine the PIN or pattern entered. While guessing, participants were allowed to consult reference images of each authentication technique (e.g., see Figure 2a). We chose to have our participants, who used our 3D Pattern technique, be the shoulder surfers because they were familiar with this novel technique.

6.4 Design and Procedure

The study was a within-subjects design with `TECHNIQUE` and `TRIAL NUMBER` as independent variables. Technique *Entry Time*, technique *Error Rate*, and shoulder surfing *Guesses* were measured as dependent variables. `TECHNIQUE` had 3 levels: PIN, Pattern, and Pattern3D, the last of which corresponded to our 3D Pattern design.

Techniques were presented to participants in a Latin square arrangement. For each technique, there were two PINs or patterns. For each PIN or pattern, there were two blocks of authentication trials, each with five trials. The first block of each PIN or pattern was a practice round, and the data was not analyzed. The practice round of Pattern and Pattern3D rendered in *with feedback* mode, which rendered lines between connected points on the screen. However, during the second block, the technique rendered in *without feedback* mode.

PINs and patterns were randomly generated. During the first section of the experiment, participants were allowed to look at a separate computer monitor on which the reference PINs and patterns were displayed. The pattern and 3D Pattern reference images were rendered as they would be seen after being entered on the phone screen (see Figure 2b). As described in Section 4.1, to improve usability, patterns for the Pattern3D technique were controlled to always start on the top (furthest from screen) layer.

7 RESULTS

A repeated measures ANOVA with Greenhouse-Geisser sphericity correction found a significant main effect of **TECHNIQUE** on log-transformed *Entry Time* ($F_{1,36,6.84} = 22.79, p < 0.01, \eta_G^2 = 0.71$). Post hoc paired t-tests with Holm correction show with significance that Pattern3D was slower than PIN ($p < 0.0001$) and Pattern ($p < 0.0001$), and that PIN was slower than Pattern ($p < 0.001$). The median time to authenticate using PIN was 3.0 seconds (IQR=1.2), Pattern was 2.0 seconds (IQR=1.5), and Pattern3D was 8.0 seconds (IQR=7.1). We also measured the “time from first digit”, or the difference in time between the first input towards authenticating and finishing authentication. The median time from first digit using PIN was 2.0 seconds (IQR=1.0), Pattern was 1.3 seconds (IQR=0.8), and Pattern3D was 4.7 seconds (IQR=4.4). These results are depicted in Figure 6.

A Friedman rank sum test shows a significant effect of **TECHNIQUE** on *Error Rate* ($\chi_3^2 = 17.72, p < 0.001$). Post hoc paired Wilcoxon signed-rank tests with Holm correction show with significance that Pattern3D has a higher error rate than PIN ($p < 0.0001$) and Pattern ($p < 0.0001$), but do not indicate any significant difference between PIN and Pattern ($p = 0.57$). The mean error rate for PIN was 3% (SD=18%), Pattern was 2% (SD=13%), and Pattern3D was 52% (SD=50%). These results are depicted in Figure 7.

An empirical CDF representing the number of guesses needed to correctly guess a PIN or pattern from a shoulder-surfing video is depicted in Figure 8. PIN and Pattern are similar in shoulder-surfing resistance, whereas Pattern3D appears to have a slight advantage. A Friedman rank sum test shows a significant effect of **TECHNIQUE** on *Guesses* ($\chi_3^2 = 9.4, p < 0.05$). Post hoc paired Wilcoxon signed-rank tests with Holm correction show with significance that Pattern3D was harder to guess correctly than PIN ($p < 0.01$), but shows no other significant effects. The mean number of guesses needed for PIN was 1.3 (SD=1, median=1), Pattern was 2.0 (SD=1, median=1.5), and Pattern3D was 5.3 (SD=6, median=2.5). One participant was not able to guess one 3D Pattern within the given 20 trials.

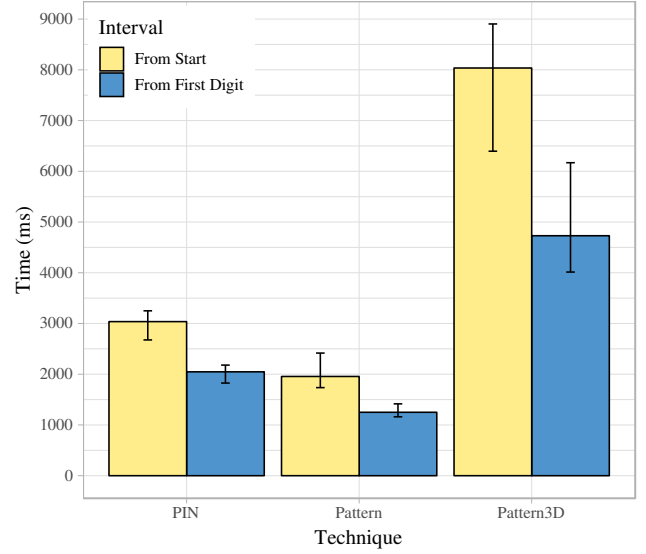


Figure 6: Median times taken to authenticate for each of the authentication techniques. “From Start” indicates the time from the user pressing the start button to finishing authentication; “From First Digit” indicates the time from the user entering the first digit to finishing authentication. Error bars indicate 95% confidence intervals.

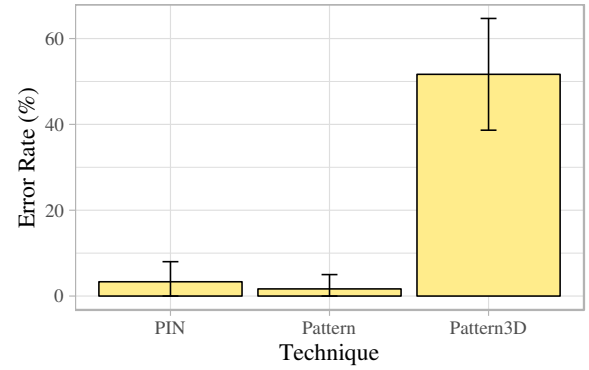


Figure 7: Mean error rates for each of the authentication techniques. Error bars indicate 95% confidence intervals.

8 DISCUSSION

Based on our experience designing and evaluating the 3D Pattern technique, we discuss the shoulder-surfing resistance of the technique, and possible future directions for exploration.

8.1 Shoulder-Surfing Resistance

Statistical analysis shows that the shoulder-surfing resistance of the 3D Pattern technique is higher than that of PIN locks. The empirical CDF of shoulder-surfing guesses (Figure 8) might indicate that the 3D Pattern technique is more shoulder surfing resistant than the pattern technique. Because users hover their fingers in 3D space,

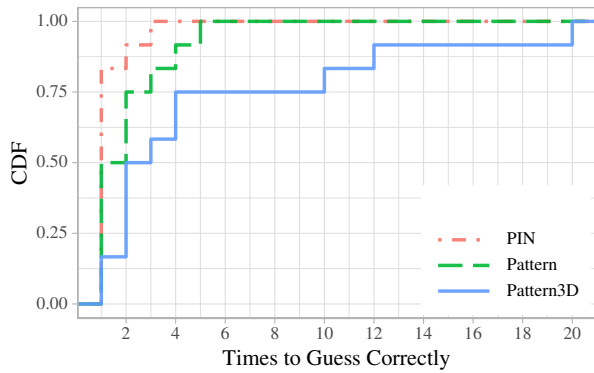


Figure 8: Empirical CDF of number of guesses needed to correctly identify the PIN or pattern in the shoulder surfing video.

it is hard for a shoulder surfer to guess the layer in which the user’s finger hovers. Moreover, due to the fact that the pattern and 3D Pattern locks rendered in *without feedback* mode, no lines were rendered on the screen. This could explain why pattern lock appeared slightly more shoulder surfing resistant than the PIN technique.

Previous work [21, 22] has used graphical visualisation to provide more memorable passwords. Our technique could be expanded to combine the ideas of both picture passwords and pre-touch information. This would allow for the creation of meaningful and easily remembered passwords that are less position sensitive.

8.2 Equipment Resolution and Latency

The Vicon tracking system has the potential to be very accurate when tracking, but this depends on a number of environmental factors such as lighting, camera placement, and so on. In the study at hand, there was some visible jitter in the position of the tracked objects, likely slowing down participants when using the 3D Pattern technique. While not reported by participants, wearing the fiducial marker with double-sided tape on the finger may have caused discomfort and also slowed down authentication time. Further, due to network latency and the high frequency of messages being sent across the network, user interface latency was low but apparent with all three techniques, potentially affecting the external validity of our study. In the future, smartphones with built-in pre-touch support would eliminate the need for motion capture and its associated limitations.

8.3 Visualization of the Cube

Representing the 3D cube on the 2D phone screen is important for our technique to be effective. Our choice to use orthogonal projection may have negatively affected our results both in terms of authentication speed and error rate.

One possible way to extend our design would be to use different kinds of depth cues. Our implementation uses a cursor with varying colours and sizes based on depth. We could instead visualize shadows to give an impression of finger height. The distance between the cursor and its shadow and shadow size could be varied with

finger height. Alternatively, a depth-of-field blurring effect based on the finger’s distance from the screen could be used. Another possible extension would be to investigate if the use of perspective projection instead of our implemented orthogonal projection would have better performance. It could also be effective to slightly rotate the projection of the cube depending on the position of the finger or orientation of the smartphone.

8.4 Experimental Protocol

Our experiment allowed the user to practice each technique five times. Given the novelty of pre-touch interfaces, this might not have been enough practice rounds for users to become accustomed to this new paradigm. Further in support of this argument is the fact that participants took several seconds between starting authentication and entering the first digit for the 3D Pattern technique. Users needed to adjust the position of their fingers to find the correct position of the top layer.

Compared to previous studies [10], both our implementations of PIN authentication and pattern authentication were slower. There are several factors that could have contributed to these results. First, the main PC, rather than the smartphone, performed all calculations, resulting in a possible small effect of network latency. We also found that participants sometimes referred back to the secondary computer monitor to recall which PIN or pattern to enter.

9 CONCLUSION

In this work, we have proposed a novel approach to smartphone authentication using pre-touch information, called 3D Pattern. We have implemented a prototype of the technique by simulating a pre-touch-capable smartphone using a motion capture system. We have also evaluated the 3D Pattern technique in a pilot study, in comparison to two popular existing techniques, finding that authentication times were longer, but that the technique could be more resistant to shoulder-surfing attacks, while being immune to smudge attacks. We attribute the longer authentication times to environmental conditions adversely affecting motion capture and the novelty of pre-touch to participants. We believe these limitations could be easily overcome as pre-touch becomes mainstream.

ACKNOWLEDGEMENTS

This work was made possible by NSERC Discovery Grants 2016-03878, 2017-03858, and 2018-05187, the Canada Foundation for Innovation Infrastructure Fund “Facility for Fully Interactive Physio-digital Spaces” (#33151), and Ontario Early Researcher Award #ER16-12-184.

REFERENCES

- [1] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. 2010. Smudge Attacks on Smartphone Touch Screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies (WOOT’10)*. USENIX Association, Berkeley, CA, USA, 1–7. <http://dl.acm.org/citation.cfm?id=1925004.1925009>
- [2] Andrea Bianchi, Ian Oakley, Vassilis Kostakos, and Dong Soo Kwon. 2011. The Phone Lock: Audio and Haptic Shoulder-surfing Resistant PIN Entry Methods for Mobile Devices. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI’11)*. ACM, New York, NY, USA, 197–200. <https://doi.org/10.1145/1935701.1935740>
- [3] Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors*

- in *Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1393–1402. <https://doi.org/10.1145/2702123.2702252>
- [4] Peng Cheng, Ibrahim Bagci, Utz Roedig, and Jeff Yan. 2018. SonarSnoop: Active Acoustic Side-Channel Attacks. <https://arxiv.org/abs/1808.10250>
 - [5] Hsin-Yi Chiang and Sonia Chiasson. 2013. 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 (MobileHCI '13)*. ACM, New York, NY, USA, 251–260. <https://doi.org/10.1145/2493190.2493213>
 - [6] Alexander De Luca, Roman Weiss, and Heinrich Hussmann. 2007. PassShape: Stroke Based Shape Passwords. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces (OZCHI '07)*. ACM, New York, NY, USA, 239–240. <https://doi.org/10.1145/1324892.1324943>
 - [7] Malin Eiband, Mohamed Khamis, Emanuel von Zeischwitz, Heinrich Hussmann, and Florian Alt. 2017. Understanding Shoulder Surfing in the Wild: Stories from Users and Observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4254–4265. <https://doi.org/10.1145/3025453.3025636>
 - [8] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. 2013. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *IEEE Transactions on Information Forensics and Security* 8, 1 (January 2013), 136–148. <https://doi.org/10.1109/TIFS.2012.2225048>
 - [9] Meriem Guerar, Alessio Merlo, and Mauro Migliardi. 2017. Clickpattern: A pattern lock system resilient to smudge and side-channel attacks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* 8 (January 2017), 64–78.
 - [10] Marian Harbach, Alexander De Luca, and Serge Egelman. 2016. The Anatomy of Smartphone Unlocking: A Field Study of Android Lock Screens. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4806–4817. <https://doi.org/10.1145/2858036.2858267>
 - [11] Marian Harbach, Emanuel von Zeischwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. 2014. It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. USENIX Association, Menlo Park, CA, 213–230. <https://www.usenix.org/conference/soups2014/proceedings/presentation/harbach>
 - [12] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. 2016. Pre-Touch Sensing for Mobile Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2869–2881. <https://doi.org/10.1145/2858036.2858095>
 - [13] Hassan Khan, Urs Hengartner, and Daniel Vogel. 2016. Targeted Mimicry Attacks on Touch Input Based Implicit Authentication Schemes. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '16)*. ACM, New York, NY, USA, 387–398. <https://doi.org/10.1145/2906388.2906404>
 - [14] Hassan Khan, Urs Hengartner, and Daniel Vogel. 2018. Evaluating Attack and Defense Strategies for Smartphone PIN Shoulder Surfing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 164, 10 pages. <https://doi.org/10.1145/3173574.3173738>
 - [15] Insu Kim, Keunwoo Park, Youngwoo Yoon, and Geehyuk Lee. 2018. Touch180: Finger Identification on Mobile Touchscreen Using Fisheye Camera and Convolutional Neural Network. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings (UIST '18 Adjunct)*. ACM, New York, NY, USA, 29–32. <https://doi.org/10.1145/3266037.3266091>
 - [16] Geetika Kovelamudi, Jun Zheng, and Srinivas Mukkamala. 2016. Scramble or not, that is the question a study of the security and usability of scramble keypad for PIN unlock on smartphones. In *2016 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, Chengdu, China, 1–6. <https://doi.org/10.1109/ICCCChina.2016.7636862>
 - [17] Takuro Kuribara, Buntarou Shizuki, and Jiro Tanaka. 2014. Vibrainput: Two-step PIN Entry System Based on Vibration and Visual Information. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHIEA '14)*. ACM, New York, NY, USA, 2473–2478. <https://doi.org/10.1145/2559206.2581187>
 - [18] Taekyoung Kwon and Sarang Na. 2014. TinyLock: Affordable defense against smudge attacks on smartphone pattern lock systems. *Computers and Security* 42 (2014), 137 – 150. <https://doi.org/10.1016/j.cose.2013.12.001>
 - [19] Toan Van Nguyen, Napa Sae-Bae, and Nasir Memon. 2017. DRAW-A-PIN: Authentication using finger-drawn PIN on touch devices. *Computers & Security* 66 (2017), 115 – 128. <https://doi.org/10.1016/j.cose.2017.01.008>
 - [20] DaeHun Nyang, Hyoungshick Kim, Woojoo Lee, Sung bae Kang, Geumhwan Cho, Mun-Kyu Lee, and Aziz Mohaisen. 2018. Two-Thumbs-Up: Physical protection for PIN entry secure against recording attacks. *Computers & Security* 78 (2018), 1 – 15. <https://doi.org/10.1016/j.cose.2018.05.012>
 - [21] Ilesanmi Olade, Hai-Ning Liang, and Charles Fleming. 2018. SemanticLock: An authentication method for mobile devices using semantically-linked images. *CoRR abs/1806.11361* (2018).
 - [22] Tetsuji Takada and Hideki Koike. 2003. Awase-E: Image-Based Authentication for Mobile Phones Using User's Favorite Images. In *Human-Computer Interaction with Mobile Devices and Services*, Luca Chittaro (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 347–351.
 - [23] Desney S. Tan, Pedram Keyani, and Mary Czerwinski. 2005. Spy-resistant Keyboard: More Secure Password Entry on Public Touch Screen Displays. In *Proceedings of the 17th Australia Conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future (OZCHI '05)*. Computer-Human Interaction Special Interest Group (CHISIG) of Australia, Narrabundah, Australia, Australia, 1–10. <http://dl.acm.org/citation.cfm?id=1108368.1108393>
 - [24] Desney S. Tan, Pedram Keyani, and Mary Czerwinski. 2005. Spy-resistant Keyboard: More Secure Password Entry on Public Touch Screen Displays. In *Proceedings of the 17th Australia Conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future (OZCHI '05)*. Computer-Human Interaction Special Interest Group (CHISIG) of Australia, Narrabundah, Australia, Australia, 1–10. <http://dl.acm.org/citation.cfm?id=1108368.1108393>
 - [25] Emanuel von Zeischwitz, Alexander De Luca, Bruno Brunkow, and Heinrich Hussmann. 2015. SwiPIN: Fast and Secure PIN-Entry on Smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1403–1406. <https://doi.org/10.1145/2702123.2702212>
 - [26] Haijun Xia, Ricardo Jota, Benjamin McCanny, Zhe Yu, Clifton Forlines, Karan Singh, and Daniel Wigdor. 2014. Zero-latency Tapping: Using Hover Information to Predict Touch Locations and Eliminate Touchdown Latency. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 205–214. <https://doi.org/10.1145/2642918.2647348>
 - [27] Xing-Dong Yang, Tovi Grossman, Pourang Irani, and George Fitzmaurice. 2011. TouchCuts and TouchZoom: Enhanced Target Selection for Touch Displays Using Finger Proximity Sensing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2585–2594. <https://doi.org/10.1145/1978942.1979319>