Executive Summary

Google's mobile operating system Android offers a pattern lock mechanism, which lets users draw a pattern on the touchscreen to unlock the device. In this project, we investigate how secure the Android pattern lock mechanism is against soft side-channel attacks, which are attacks that can be employed on human factors, and provide a solution that is immune or more robust against these attacks.

We categorize the soft side-channel attacks into two separate types: physical attacks and psychological attacks. Physical attacks aim to retrieve a pattern using physical traces left by the human user. Psychological attacks aim to detect bias in pattern setting. For the physical attacks, we use an optical camera and a microscope to analyse oily residues left on the screen, and a thermal camera to analyse heat traces left on the screen after drawing a pattern. For psychological attacks, we analyse the lengths of the patterns, the number of direction changes in patterns, start points, end points, and sub-patterns with lengths of one to four dots. To achieve this, we coded and conducted a survey, and collected data from 144 participants. Finally, we propose the auto-hidden numerical wheels lock mechanism. We assess its security, and show that it is more secure than the Android pattern lock whilst maintaining swiping gestures but requiring more time to unlock.

Contributions and achievements can be summarised as follows:

- We made successful attacks with a compact camera (see pages 24-26).
- We made successful attacks with a microscope: no previous study has experimented with this type of attack (see pages 26-27).
- We made limited successful attacks with a thermal camera: no previous study has experimented with this type of attack (see pages 27-29).
- We analysed and found significant results regarding patterns and human behaviour: again, no previous study has studied human set patterns (see pages 30-50).
- We propose a new lock mechanism that is immune to all the attacks above, but prone to psychological PIN attacks (see pages 51-57).

Contents

1 Introduction

In the modern world, passwords are part of our daily routine. We need them for bank accounts, email accounts, websites, companies, information, computation etc. To authenticate ourselves, we use various hardware such as keyboards, ATM keypads, fingerprint readers, or touchscreens. One of the types of device that is in our everyday life that has a touchscreen is the smartphone. We use it to store personal information, to read business mails, or to access any online service. It holds an important amount of information about the owner. For this reason, people tend to lock their phones using the mechanisms provided by their smartphone. This study focuses on new ways to attack a modern lock mechanism.

1.1 The Key Concepts

To better understand this study, it is important to know some key definitions.

In most cases, phone lock mechanisms are either a PIN code, or a password. However, with the arrival of Google's open-source mobile operating system Android, a new type of lock mechanism has been offered. Called the "pattern lock", this mechanism lets the user to draw lines on the touchscreen to form a pattern, and thus to unlock the device.

There are various types of attacks that can be used against a device to retrieve its passcode. Soft side-channel attack, being one of them, depicts any attack that does not deal with the authentication protocol itself but human factors.

1.2 The Android Pattern Lock

With the expansion of touchscreen devices -especially in mobile platforms-, graphical passwords became a real alternative to text-based ones. Android employs such a mechanism under the name of pattern lock (Google, 2012).

Android pattern lock is a locking mechanism alternative to the traditional PIN code keypad. Similar to the keypad, it consists of 9 nodes in 3x3 grid formation. The user starts a pattern by touching one of the dots to make it the start point and the anchor. Then, by swiping his/her finger on the screen towards other dots, he/she creates a dynamic line with one end anchored to the start point, and the other is on the tip of the finger. When the user's finger approaches one of the unselected dots, that dot is added to the pattern, forming a stable link between the previous dot and itself as well as becoming the new anchor point. Ultimately, the user continues to swipe his/her finger to add dots to form a pattern.

There are some constraints while setting a pattern. Those are:

- It takes a minimum of 4 and a maximum of 9 dots to create a lock pattern.
- Each dot can be visited only once.
- A previously not visited node becomes visited if it is on the way of a horizontal, vertical, or diagonal line.

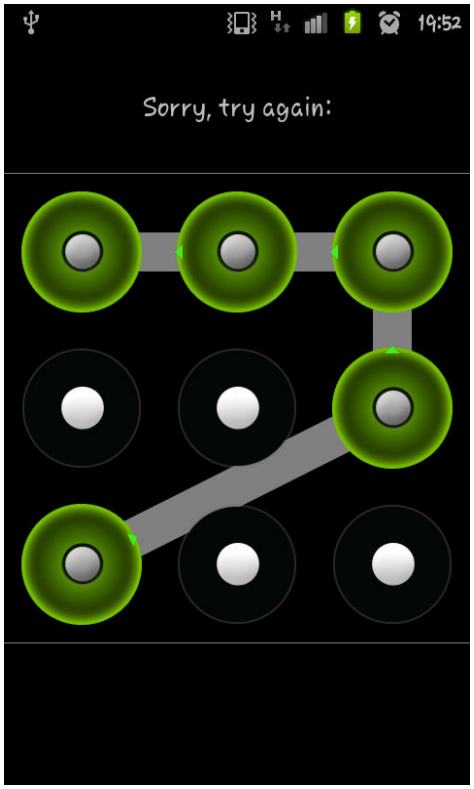

Figure 1. A 30 or 60 degree line is hard to draw.

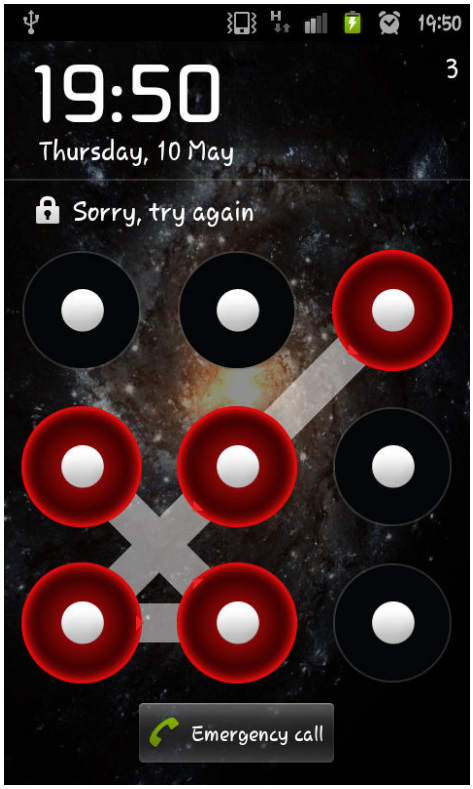

Figure 2. Erroneous patterns are shown on screen no matter what.

Due to these constraints, there are a total of 389112 possible patterns. This value is calculated both by brute force (Aviv, et al., 2010), and by an algorithm provided by a Google engineer (Zhou, Y., 2011). A standard implementation of the system allows user to try 20 patterns before it asks for Google Account for any further action. Additionally, it blocks any input for 30 seconds after each 5 wrong pattern entries, taking at least one and a half minutes to use all the 20 tries.

While not a technical constraint, it is also hard and thus unlikely to visit subsequent nodes which form a 30 or 60 degree angle. This is especially true when the intermediate nodes are not visited previously. Patterns including these types of connections require attention to unlock, so they can hardly be drawn using muscle memory.

There are some options to customise the way the pattern lock works. The first one is to hide the unlock pattern trail to be seen graphically. This option highly increases the security against curious eyes as it eliminates a huge green pattern from the screen while decreasing the convenience for entering complex patterns. However, it does not stop the OS showing the trail when the pattern is entered incorrectly. While this may seem helpful, it can give away the pattern if only one node has been missed and the rest are correct. The second and last option is to enable tactile feedback. This option does not affect security, but increases the ease of use. Especially, in cases where the visible pattern is disabled, tactile feedback gives a good idea of whether the node is being visited as expected or not.

Although it is not crucial to know the cryptographic details of Android pattern lock in our scope of soft side-channel attacks, it may prove useful in the future. The inner workings of the pattern lock are similar to PIN code. Each node is represented by a number starting at top left with 0, continuing horizontally, and ending at bottom right with 8. Android OS takes the SHA-1 hash of the combination entered and stores it in this state, without any salting. The file that holds the hash is stored in system/ directory, which requires root access to the device.

1.3 Aims and Objectives

In this study we aim to:

- Investigate how secure the Android pattern lock mechanism is against soft-side channel attacks.
- Provide an alternative solution if necessary.

To achieve our goal, we will follow the following objectives:

- Analyse oily residues, i.e. smudges, left on the screen using an optical compact camera in a non-clinical environment.
- Analyse smudges using a microscope to observe how a microscope attack performs compared to an optical compact camera attack.
- Perform attacks using thermal imaging in an attempt to retrieve the pattern by observing the heat traces left by the finger.
- Investigate the effect of human behaviour/psychology on pattern setting by conducting a survey on voluntary participants to collect patterns and use this data to find out frequent sub-patterns, average pattern lengths etc.
- Based on the results from the attacks and pattern analyses, provide an alternative solution that is more robust.

1.4 Structure

The report consists of six sections. It begins with background research, continues with physical attacks followed by psychological attacks. Based on the previous sections' results, the fourth section provides an alternative lock mechanism. The report concludes with conclusion and future work. All the sections and their subsections are critically evaluated when possible.

2 Background

2.1 Text-based Passwords vs Graphical Passwords

Text-based passwords and PIN codes are essential to anyone who owns a bank account, an email account, or any computer derivative device. In modern world, an adult has at least several different accounts that he/she needs to remember. As a consequence, they often either recall another account's password (Sasse, Brostoff and Weirich, 2001), or tend to use the same (Biddle, Chiasson and van Oorschot, 2011) or easy to guess passwords (Tryfonas, 2012). Thus, the user has to select between the usability and the security. If a word-like password is chosen, it is easy for an attacker to find it using dictionary based attacks. On the other hand, if a scramble of characters have been set as password, it is highly likely that the user will fail to fully remember the sequence (Sasse, Brostoff and Weirich, 2001). This renders text-based passwords hard for the legitimate user, and easy for the attacker.

Another aspect that makes the text-based passwords hard is the way human brain works. According to Dual Coding Theory, cognition is composed of two separate parts: nonverbal and verbal systems (Paivio, 2007, p.33). Having different systems in the brain to process the verbal and nonverbal information, humans perform differently in these two ways when it comes to remembering. Text requires an additional process of associating symbols with a contextual meaning (Biddle, Chiasson and van Oorschot, 2011). For this reason, it can be concluded that text-based passwords are harder to remember than the graphical ones.

Graphical passwords may come in much more variety compared to text-based solution. It can include clicking/long-clicking some points on an image, or drawing a line or a shape. The most important advantage it provides is the possibility to define a password such that it is memorisable by the user, and yet still hard to guess by the attacker. However, the graphical passwords can also have their weaknesses. We investigate this in the following section.

2.2 Psychology on Setting Graphical Passwords

Previous section has shown that graphical password mechanisms were solid and potentially better alternatives to text-based ones. Yet, taking into consideration the fact that users can select their graphical passwords with respect to some logic, human psychology comes into play when graphical solutions are considered.

Studies on background image based graphical passwords show that humans tend to choose popular points –or "hot-spots"- on the image (Thorpe and van Oorschot, 2007). In their experiment, Thorpe and van Oorschot (2007) collect inputs from 43 users using different images, allowing users to skip images on which they cannot easily remember the points they clicked. They come up with the result that in some pictures 24-31% of the users had chosen the same 5 points as a password. While the results vary from image to image, they nevertheless imply that there are some general hot-spots that people tend to select.

Another study provides 9 different face images to users, and lets them choose 4 of them in a sequence to form a password (Davis, Monrose and Reiter, 2004). For this particular face mechanism, they collect passwords from 79 participants as well as their gender and race data. Using 80% of the collected data as training data, they test the rest 20% to guess. The results are significant: the worst 10% of the passwords set by males can be guessed in 2 attempts. The survey conducted among the participants also indicates that they tend to choose faces similar to their own, be it gender or race.

A study regarding Android pattern lock has been conducted by Angulo and Wästlund (2011) analyses the time spent on dots and in between dots. While their research contributes to human factors on setting graphical passwords to some extent, the data gathered is primarily used to add an extra layer of security, and not to classify humans with respect to their demographics and pattern speed. Further details about how they conducted their research are given on section 2.5.7.

Except the last study, rest of the studies may not seem relevant to the Android pattern lock at first, but the fact that humans have similar preferences on graphical passwords provides a solid enough ground to look for frequent sub-patterns preferred in Android pattern lock. It would also be interesting to see how users

would behave when it comes to Android pattern lock's nodes, which do not differentiate from place to place as in background images or faces.

2.3 Methods of PIN and Pattern Retrieval

Android pattern lock is a touch featured locking mechanism. It relies on users swiping their finger in order to unlock the device. This action leaves behind an oily residue, or in other words, smudges.

In this section, we investigate what has been researched to retrieve lock pattern and PIN code on touch interfaces.

2.3.1 Light and Standard Optics

The first idea that comes to mind for retrieving a lock pattern is to observe the touchscreen for smudges using standard optics. A highly relevant research on retrieving the lock pattern using the mentioned method is conducted by Aviv, et al. (2010). In their paper, it is discussed that recovering smudges using a light source and a digital camera is possible due to the fact that the touchscreen surfaces are reflective rather than diffusive. This allows capturing photographs of the smudges and analysing the pattern. Experimenting with the directional and omnidirectional light sources, and testing from 0° to 180° angle by taking pictures at each 15 degree angles, Aviv, et al. (2010) found out that the smudges were visible in most cases when a directional light source is used, with the exception of complementary angles between the light source and the camera. This exception is caused since such angles result in a full reflection, preventing the retrieval of the details on the surface. While directional light source was useful, omnidirectional light source proves to create the full reflection effect at all angles, rendering this type of light source unusable. Apart from the ideal photograph capturing angles to retrieve the smudges, the experiments focus on various states a touchscreen can be in such as:

- pattern entered using normal or light touches,
- pattern entered before or after phone usage,

- phone put in a pocket and removed after pattern entry, or the same with the addition of intentionally wiping the screen with a cloth prior to usage.

Note that the notions of “normal” and “light” touches are not quantitative in this study, and thus must be intuitively guessed. For this reason, we assume the light touch stands for intentionally low pressure touches to minimise any smudge left behind, whilst the normal touch is the one made without any concern of leaving a smudge behind.

These patterns are tested on two phones, namely HTC G1 and HTC Nexus 1. It is of importance since they give different results in terms of smudge persistence. This indicates that Android phones, even from the same manufacturer, may have different touchscreen surfaces.

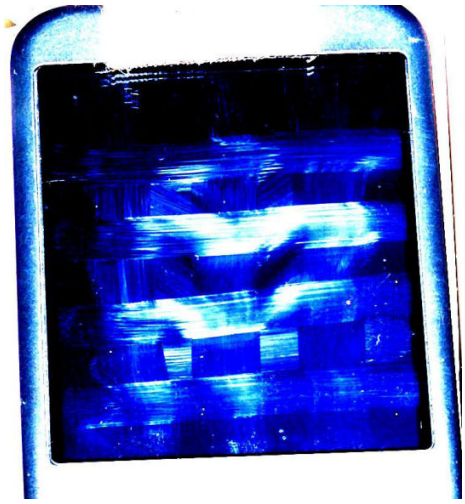

Figure 3. HTC G1 hashed with streaks after pattern entry. (Aviv, et al., 2010, p.7)

Figure 4. HTC G1 with normal touches. (Aviv, et al., 2010, p.6)

When all of the angle setups are taken into consideration, the best result is obtained when the pattern is entered using normal amount of pressure after the phone is held against the face: 68% of the tries resulted in complete retrieval of the pattern while this percentage goes up to 96% when partial retrieval is included. Although this result was achieved with HTC G1, it is not known how well HTC Nexus 1 would perform under the same scenario as it is only involved in normal touch scenario (without contact with the face, thus much less oily residue on the screen). Having said so, it is found out that the best angle to retrieve a pattern was 60

degrees, with 80% of the lighting scenarios resulting in “perfect or nearly perfect retrieval” (Aviv, et al., 2010).

It is also noted that the directionality was discernible as each consecutively drawn line overwrites the previous one right at the node. This is particularly important to decrease the number of attempts to unlock the device. A highlight on their results with the usage scenarios is that tapping on the touchscreen after the entry of lock pattern does not completely eliminate the lock pattern residue, yet on the other hand, when the screen is heavily used with smudge gestures, the pattern can no longer be acquired. Finally, it is deduced that intentionally cleaning with cloth and putting the phone to pocket was not enough to prevent the pattern’s retrieval. As a side note, it is important to note that the researches preferred wording the process as “simple clothing”, which may mean the results may not hold true when the screen is rubbed thoroughly.

Overall this method is particularly efficient as all it requires is a directional light source and a digital camera. An attacker can easily and quickly capture a photo of the touchscreen from a useful angle, then do any necessary contrast/brightness adjustments on the photo to retrieve one’s pattern lock. The pattern will be ready for entry the next time that device is left alone. As the Aviv, et al. (2010) discusses, even if the pattern is only partially retrievable, multiple photos taken in different times may reveal the full pattern.

We have not found any research conducted at microscopic levels for smudges, and believe that it may prove to be a useful method in cases the smudge is casually wiped from the screen.

2.3.2 Thermal Imaging

Although the side-channel attacks to Android pattern lock mechanism is limited to Aviv, et al.’s approach, the use of thermal camera to retrieve the PIN codes is an already existing attack on other devices such as ATM keypads. We believe that this method is applicable to Android pattern lock, and thus investigate how this is executed.

Mowery, Meiklejohn and Savage (2011) examines the efficiency of a thermal camera for obtaining the PIN codes from ATM keypads. To conduct their experiments, they use an A320 FLIR thermal camera with 320x240 resolution. This is a relatively high resolution, which in return allows them to place the camera at 28 inches without losing much data. Although there are two keypads for testing, one being metal and the other plastic, the tests are carried out on the plastic keypad as it is indicated that the metal keypad's conductivity renders it impervious to attack. As our goal is to use thermal imaging on touchscreen devices, we believe that it will provide a much less conductive surface, similar to the plastic keypad used by Mowery, Meiklejohn and Savage (2011). However, according to their research, it is crucial to note that these results differ from (Zalewski, M., 2005) in terms of heat persistency, so we should also expect varying behaviours across different touchscreen devices.

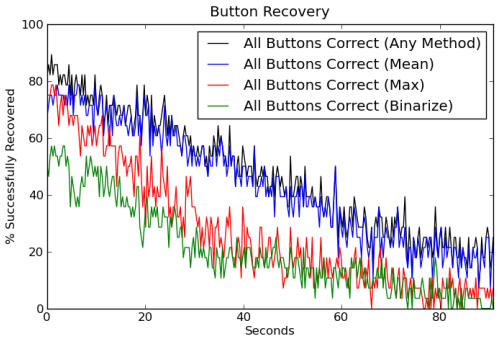

Figure 6. Percentage of correct key combinations found at 28 inches. (Mowery, Meiklejohn and Savage, 2011, p.5)

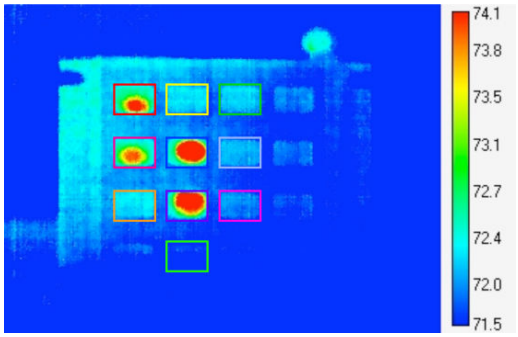

Figure 5. Thermal image right after PIN entry. (Mowery, Meiklejohn and Savage, 2011, p.3)

Data gathered from 21 people and 27 PIN combinations display that the heat transferred to the keypad depends on the amount of pressure exerted to the keys as well as the warmth of the hand. However, the heat of the ATM was not taken into consideration as the keypad is used as an isolated test bed, without being wired to or placed on any electronic device. The thermal images are captured between 71.5°F and 74.1°F: although the ambient temperature is lacking, the image shows a clear distinction between the background and the touched keys. The thermal image captured right after the entry of PIN displays which buttons are pressed and in which order with no difficulty.

To analyse the data, the researchers take the last frame before the hand obscures the key pad for PIN entry as the calibration frame, and compare it with frames after the PIN entry. To evaluate whether a button is pressed or not, the keypad's buttons are taken as Region of Interests (ROIs). Then, three different methods are applied to decide on which buttons are pressed. These are named as max, mean, and binarize. The max method takes the hottest pixel in each ROI. The mean method takes the average temperature of each ROI, making it a better solution against noisy images in which an irrelevant single hot pixel causes the max method to misclassify. Finally the binarize method compares each pixel in a ROI with the pixel in the same coordinate in the calibration frame. Any pixel with increased temperature sets that coordinate as 1, while temperature decrease or indifference leaves it at 0.

After the application of any of these methods, each ROI is assigned with a value with respect to their temperature, and these values are subtracted from the calibration frame's ROIs' values. Then the values are sorted by magnitude, and the ROIs with highest values are assumed as the buttons pressed. However, there is the possibility of having PINs with a number occurring more than once in it. To circumvent this issue, a frame's ROIs are averaged, and only those ROIs that have higher values than the average are considered as pressed buttons.

The results indicate that, when the mean method is applied, in 60% of experiments all the correct keys are guessed. This result applies for both 14 and 28 inch distances from the keypad, rendering the thermal attack feasible in more situations. Although the attempts of recovering the correct PIN sequence at the first try was significantly lower with less than 10% success, we are not limited by such issues as our reading will be on heat traces, and a perfect retrieval of the trace would leave us with the pattern itself, including its direction. Even in the case of partial retrieval, we have the opportunity of 20 tries as opposed to the limit of 3 tries of ATMs.

Another important aspect of thermal images is that it may prove useful in situations where it is not possible to retrieve the smudges using standard optics. An attack in a dimly lit place, for instance, would benefit from thermal imaging. However, the most interesting use of thermal imaging can be when the screen of a smartphone is on. While a normal camera cannot capture smudges at this stage, a thermal camera may overcome this issue, provided that the screen does not immediately warm up.

2.4 Prevention

As previous methods indicate, it is possible and quite easy to attack the pattern lock due to the smudges left on the screen. There are a few attempts to prevent the attacks using different unlock apps.

2.4.1 Vertical PIN

AlRowaily and AlRubaian (2011) mention two mechanisms for this. First solution is the vertical PIN. This application provides the user with 5 buttons -numbered from 1 to 5- to type their PIN code. When the user enters the PIN, the application requires a vertical slide to unlock the device. While this seems a good solution, it is not guaranteed that the swipe would overlap all the fingerprints since the buttons' width allows having fingerprints on a wider area. Another issue with this idea is that it has a very limited amount of possible combinations: with 5 buttons, there are only 5! combinations. Furthermore, it only uses swiping for hiding the entered PIN, making it more of a text-based lock mechanism.

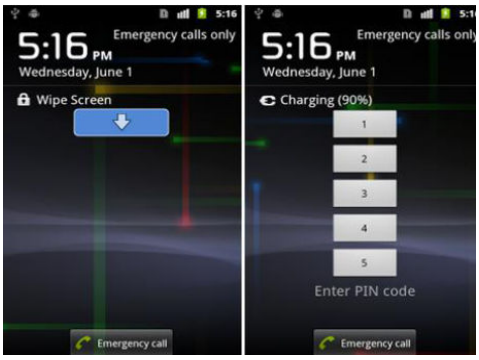

Figure 7. Vertical PIN. (AlRowaily and AlRubaian, 2011, p.301)

2.4.2 WhisperCore

The second solution mentioned is WhisperCore, an application that first lets the user enter the Android lock pattern as usual, but then requires the screen to be completely touched with slide gesture to unlock the device. This application maintains the swiping feature; however it adds a security layer that is not user friendly. Although it heightens the security by overwriting the pattern smudge, it renders the unlocking a long and uneasy procedure.

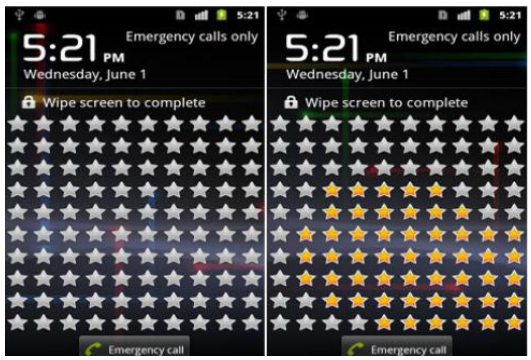

Figure 8. WhisperCore. (AlRowaily and AlRubaian, 2011, p.301)

2.4.3 Fingerprint on Touchscreen

AlRowaily and AlRubaian (2011) propose a fingerprint based unlocking mechanism, such that when the user puts his fingerprint on the indicated circle, the system accepts or rejects depending on if it is the owner or not. While this method forces the device to unlock with a unique key only (compared to other methods which can be easily repeated by strangers), it is neither possible to implement it with the existing smartphone technologies, nor as secure as it is thought. Fingerprints are obtainable from almost any hard surface. Once a fingerprint is obtained, it can be replicated and used as a fake finger. This would let the attacker unlock the device successfully without worrying about a passcode change. Considering we touch everything around us, the proposed method may prove the weakest security of them all.

Figure 9. Fingerprint on touchscreen. (AlRowaily and AlRubaian, 2011, p.302)

2.4.4 Multi-tap Circle Lock

Another alternative to pattern lock is proposed by Shin, Park, Lee and Park (2012). Their idea lies with providing a set of circles which can be clicked multiple times. At each tap, they change their colour to inform the user that input is successfully registered. This allows users to use the same circles multiple times, and renders a flawless attack using standard or thermal optics highly unlikely. Incorporating this lock method with accelerometer, they propose a guest user unlocking by shaking the phone instead of entering the correct sequence. This allows the guest limited access to the resources and apps of the device. While the proposed idea is interesting, it is not clear whether the order of tapping is important or not. In case it is not, the password space is too small, and an attack is feasible. Assuming each dot can be touched at most 4 times, it would make only 4096 possible combinations. Last but not least, it also does not employ any swiping but rather relies on tapping.

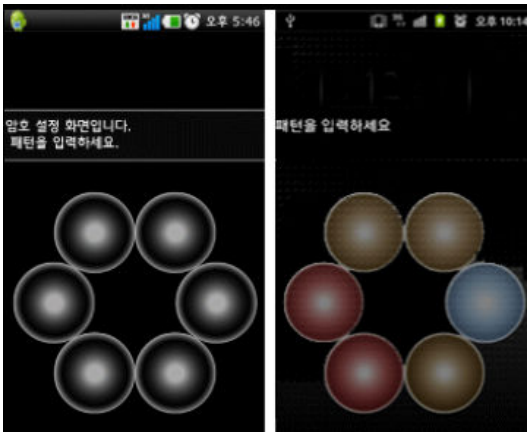

Figure 10. Multi-tap circle lock. (Shin, Park, Lee and Park, 2012, p.705)

2.4.5 Biometric-Rich Gestures

A very recent study by Sae-Bae, Kowsar, Isbister and Memon (2012) proposes a multi tap gesture unlocking mechanism. The mechanism lets user draw a pattern on the screen using all five fingers. This pattern can be counter clockwise movement of the fingers, dragging from top to bottom of the fingers, or one of the other defined patterns by the researchers as well as user defined. A normalization is then applied to the pattern drawn. This includes:

- location and orientation invariance: the pattern drawn is evaluated according to the thumb and index finger's initial position.
- length invariance: the pattern is proportionally normalized as each attempt may not be at the exact same length otherwise.

- dynamic time warping algorithm: “The distance between two time-series signals with different lengths is defined as the sum of the distance at the optimal non-linear path such that the distance or matching cost sum is minimized” (Sae-Bae, Kowsar, Isbister and Memon, 2012).
- dissimilarity score: the pattern is compared to the template and given a score. If its score is lower than the threshold, it is accepted as valid.

This mechanism is tested with 34 participants. The participants experimented with 22 predefined patterns, and they are allowed to skip a pattern if they feel uncomfortable. If comfortable, they are asked to redraw the pattern 10 consecutive times, and this data is collected. According to the results, the accuracy of user defined pattern is the highest with 97%. It is followed by two other methods: counter clockwise pattern and fixed thumb counter clockwise pattern with 93%.

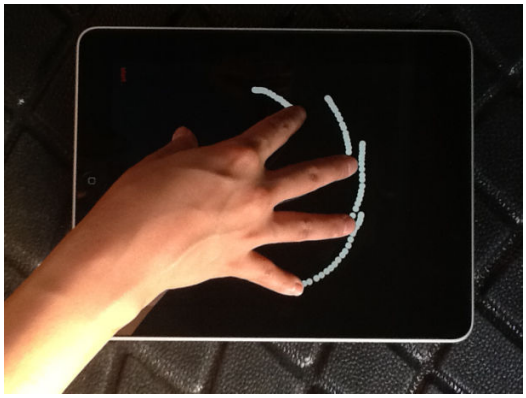

Figure 11. Biometric-Rich Gestures. (Sae-Bae, Kowsar, Isbister and Memon, 2012, p.977

This mechanism is definitely a well thought and tested one, and thus is distinctive. However, it is not stated clearly whether it is secure against mimicking the gesture or not. Another issue is that it requires a big touchscreen, such as tablets, to properly work, as smartphone screens are not large enough to execute five-finger gestures accurately. Therefore, it is not applicable to a large number of devices.

2.4.6 Fingerprint Reader

A hardware based solution to the lock pattern theft due to smudges is implemented by Motorola to their Atrix model (Motorola, 2012). The smartphone has a fingerprint reader on the top back of it, allowing the users to unlock their phone via fingerprint. Similar to the previous case proposed AlRowaily and AlRubaian (2011), this method is only secure until the fingerprint is retrieved, from a glass for example. Once a fake finger is made, the smartphone will always fall victim to the

attacker since its user cannot change their fingerprint. The only solution thus lies in the changing of the lock mechanism.

Figure 12. Motorola Atrix's fingerprint scanner. (GSM Arena, 2012)

2.4.7 Biometrically Enhanced Android Pattern Lock

Another study takes a different approach, and rather than replacing Android pattern lock, enhances it with biometrics data. Angulo and Wästlund (2011), in their research, proposes the idea of measuring the time spent on and in between the nodes of a pattern. They gather 150 pattern drawings (3 different patterns, each drawn 50 times) from 32 participants. For each pattern drawn by a participant, the first 10 are not used in analysis as participants learn the pattern at the beginning, and it may take time to get used to it. Using the Random Forest machine learning algorithm as classifier, the other 40 drawings from each participant are analysed. When 25 of the inputs are used as training set and 15 as test set, an average Equal Error Rate (EER) of 13.84% is obtained. EER is the point where False Acceptance Rate and False Rejection Rate are equal, which are understandable notions from their naming.

While this mechanism provides an extra layer of security, the results indicate that it can have false positives, or the non-user-friendly false negatives. With this mechanism, users have to enter their patterns in a robotic manner, which may prove undesired. Furthermore, the system requires 25 pattern drawings for an EER of 13.84%. It is doubtful that users would like to input their pattern a few dozen times to initialise a secure mechanism on their touchscreen devices.

2.4.8 Combination Lock

There also exist some solutions in Google Play (formerly known as Android Market), the app/movie/music store of Android.

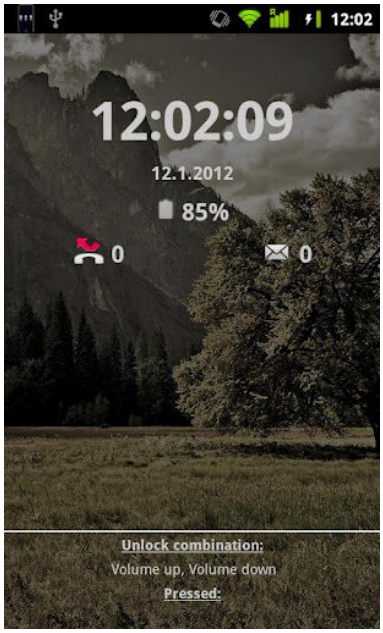

Figure 13. Combination Lock. (Anttalainen, O., 2012)

Combination Lock by Oskari Anttalainen (2012) lets user define lock patterns using the physical or soft buttons outside of the screen on the host device. The keys in use can be those like volume up and down, home button, photo button. This provides a device specific combination since Android phones differ in term of design, and more importantly negates any retrieval of tapping or swiping smudges. Yet once again, this is not a viable solution if the user prefers swiping the screen for unlocking. Furthermore, it is not possible to limit the number of retries since the application checks if the combination entered is correct after each button press. Unlimited number of trials allows a brute force attack, rendering the lock mechanism insecure.

2.4.9 iLockit

iLockit by Appsolute Security LLC (2012) on the other hand, merges the concept of tapping, multi tapping, long pressing, and sliding in its unlocking mechanism. The user selects a background image, and then multi taps, slides etc. on various parts of the image to create their own customised pattern. The main advantage of this lock mechanism is the unpredictability of whether a tapping mark is whether a standard tap, a long tap, or a part of the multi-tap along with another one. While this app may avoid attempts of pattern retrieval with a normal camera in terms of sequence, thermal imaging may prove more successful.

Figure 14. iLockit. (Appsolute Solutions LLC, 2012)

It is important to note is that these apps do not require root access, which, when acquired, lets the application modify system files. The root access is not an official option to turn on and off; it requires a hack, like iOS's Jailbreak. As a result, such non-root apps do not completely replace the default lock patterns. For example, Combination Lock lets user access any notification, and subsequently any app that is triggered by clicking on the notification. Worryingly, it can even be bypassed simply by opening the notification area, and then pressing the Home button in a 2.3 Gingerbread device. Both applications are also useless in 4.0 Ice Cream Sandwich as the Home button directly overrides the applications. For an app to become a placeholder instead of the embedded locking mechanisms, root access and system modification is necessary (Shabtai et al., 2010). However, the ideas are the scope of our research, so we simply ignore the apps' robustness at this point.

3 Physical Attacks

3.1 Smudge Attacks

3.1.1 Optical Camera Attack

The optical camera analysis has already been done by Aviv, et al. (2010) and provides comprehensive testing. Therefore, we have aimed to replicate the results using a more basic camera and a different surfaced smartphone. The Samsung Galaxy S has a Gorilla Glass screen by Corning (Corning Incorporated, 2012). This surface type is widely used on higher end Android smartphones, and thus a high quantity of devices has this surface in the current market.

We used a Panasonic Lumix DMC-TZ5 compact camera to conduct the optical camera attacks. As opposed to the DSLR used in the previous research, this camera has much worse focusing, low light performance, and sensor but it has far superior mobility, allowing the person to carry it in a pocket. Therefore, an optical camera attack is more likely to happen with a pocket-sized, point-and-shoot compact camera since it is easy to carry around unnoticed.

We used hard light to achieve well defined edged shadows as opposed to soft light, which creates soft edged shadows. Thanks to hard light, any shadow cast by the oily residue on the screen results in a sharper image, and consequently easier retrieval of the pattern. The hard light source we used in the experiments is a desk lamp with a spot light, which is by no means a special source but rather a source that can be found at many desks.

Figure 15. Screen surface right after the pattern is drawn.

We performed an optical camera attack on three different surface states. The first one was an attempt to retrieve the pattern drawn on a clean screen without any manipulation. It is evident that the pattern can be retrieved without any difficulty at this stage.

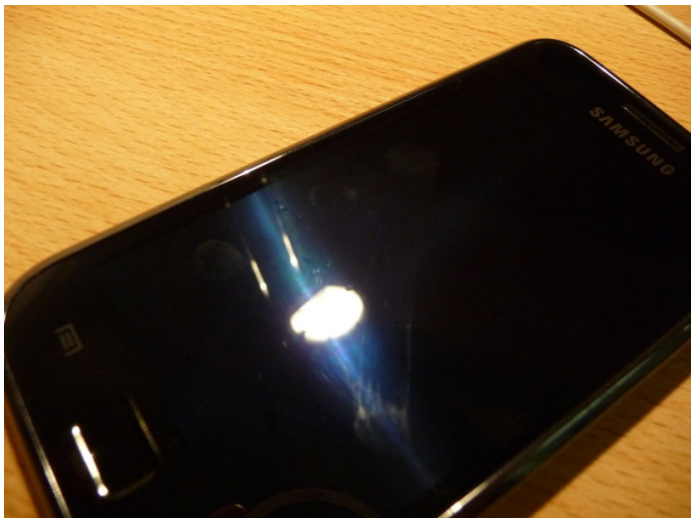

Figure 16. Screen surface after light cleanup.

The second test adds a light cleanup to the first state. The "lightness" of the cleanup is indeed subjective, and in our case, it involves rubbing the smartphone screen against a cotton cloth two times by exerting a force that does not cause arm muscles to contract. This process aims to mimic a person casually cleaning the device's screen, without the specific intent of removing any and all oily residues. The oily residues are quite resistant against simple attempts of cleaning. Although the pattern has faded slightly, it can still be fully retrieved.

The third and final test is conducted on a heavily cleaned up surface. For this test, we rubbed the screen ten times against a cotton cloth while exerting more force than before. This process aims to mimic a person with the determination of getting rid of all the oily residues on the screen at once. In this attempt, most of the pattern is lost, except one diagonal line. Due to increased entropy, it is also not possible to tell the directionality of the pattern.

Figure 17. Screen surface after heavy cleanup.

One can argue that leaving the pattern on the screen without tampering with it may not be a likely event. However, this is a common issue. When a message arrives, a user can unlock the phone, skim the message, and lock the phone again. This would leave an almost perfect pattern trace on the surface, rendering the first case a valid option.

Overall, it is possible to capture patterns using compact optical cameras in most of the cases where the phone is not heavily used or cleaned, and where there is enough lighting.

3.1.2 Microscope Attack

Having seen the results from the optical camera attack on the heavily cleaned up screen, we investigated whether a microscope would achieve better results for the retrieval of a pattern compared to optical camera.

For this experiment, we used a USB microscope with 400x magnification. The microscope employs a manual focus ring, and adjustable LED lights. We captured the images using its own software.

Our experiment included the same three cases of the optical camera attack. However, in the microscope case, we assume that the attacker already retrieved the smartphone, and is able to investigate the screen as long as desired.

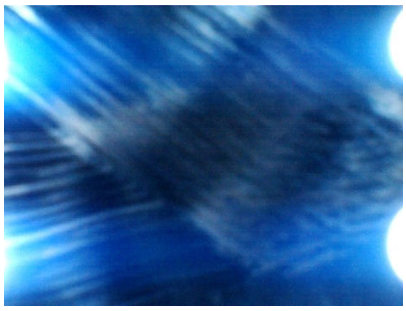

Figure 18. Microscope photo after pattern is drawn.

Similar to optical camera results, the microscope performed well during the first two cases. Lines and directionalities were very easily seen. It is particularly easy to realise the directionality whilst using a microscope as it is possible to see which smudge overwrites the other. There is, however, some loss of details after the first cleanup. Although both the lines and the directionality are visible, the cloth strokes blur the fingerprint's oily residue.

Figure 19. Microscope photo after light cleanup.

For the heavy cleanup case, the microscope performed slightly better than the compact camera, providing more detail of smudge residues. However, assuming that an attacker can make use of a DSLR in a controlled environment, he/she can gather similar results without the need of a microscope.

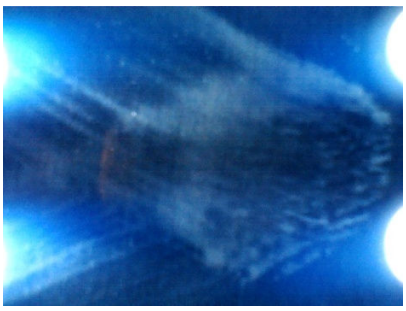

Figure 20. Microscope photo after heavy cleanup.

3.2 Thermal Attack

Although there have been many research papers done with thermal imaging, none that we are aware of has a focus on the Android pattern lock. However, similar research has been conducted on ATM machine keypads (Mowery, Meiklejohn and Savage, 2011), and can be used as a basis.

With this physical side-channel attack, the goal is to retrieve the pattern by examining the heat trace left by the finger on the device surface. Thus, it is only possible to retrieve the pattern within a specific time interval as opposed to oily residues, which can be used to retrieve the pattern any time unless the surface is tampered.

Theoretically, there are cases that a thermal camera can successfully retrieve a pattern where an optical camera is unable to do so. These cases are as follows:

- There is not enough light to capture the oily residues using an optical camera (i.e. the room is dark).
- The camera is fixed (e.g. on a wall), and the lighting/angle prevents the detection of the pattern.

Our thermal camera was FLIR E30, which has a resolution of 160x120. Due to the low resolution, we were not able to mimic a fixed wall camera scenario. Instead the experiments have been conducted from a distance of ≈1 meter.

The test environment for this attack had the following properties:

- The ambient temperature was 26 degree Celsius.
- The ambient light was low, and there was no direct sunlight coming to or

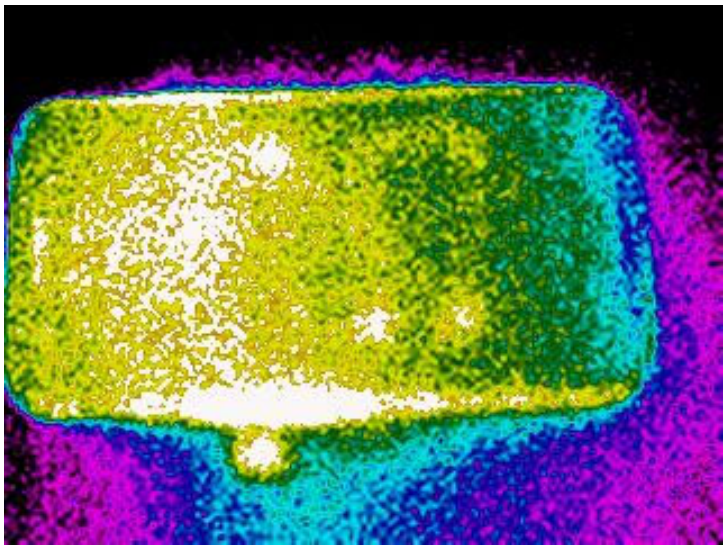

Figure 21. Thermal attack on pattern lock.

near the device. It was not possible to capture the smudges with an optical camera.

- There was one notebook dissipating thermal emission within 30cm of the device.

The test cases for this type of attack were different than the ones used in smudge attacks. Since time and heat are the main factors, the test cases were drawing pattern on an idle device, and drawing pattern on a recently used device. Figure 21 demonstrates an attack of the former case. This attack shows that it is possible to retrieve a pattern via thermal imaging. We managed to observe the heat trace for 3 seconds after the pattern is drawn. However, we were unable to extract the pattern from a recently used device. When the device runs for a short while, its CPU starts to emit a considerable amount of heat radiation. This, in turn, heats up the upper and centre parts of the device, rendering finger's heat untraceable. Even in the idle state, the CPU part of the device is considerably hotter. As a result, top three dots are hard to detect in most circumstances. It is also important to note that the test smartphone, Galaxy S, has an AMOLED display, which emits less heat as opposed to LCD. Therefore, it would be harder to successfully attack LCD screens. While it may not be a preferable attack compared to other options, thermal attacks have more potential in the future as the component manufacturing sizes diminish, chip voltages are lowered, and new display technologies are developed.

4 Attacks on Human Behaviour

While studies have been made to analyse human behaviour on picture based passwords (Thorpe and van Oorschot, 2007), no research has been done that relates to the Android pattern lock. To study the effect of psychological human factors on pattern setting, we have conducted a survey.

4.1 Survey Details

Our initial goal was to create an Android application, and request from participants to install the application in order to join the survey. However, we realized that this would attract much less participants compared to a web based survey that does not need any installation or owning a specific smartphone operating system (i.e. Android). Additionally, creating such an application would not guarantee that it would work on every Android phone flawlessly. A thorough testing stage would be necessary for different phones with different screen resolutions and different Android versions, which simply would not pay off given the effort. Nevertheless, we think that this is still a viable option under controlled data collection: developing and installing the application specifically for one device, and allowing participants to complete the survey face-to-face with the survey conductor would provide highly reliable data collection albeit lower participation.

Due to these given reasons, and the fact that we needed a simulation of the Android pattern lock screen, we preferred to prepare a web survey from scratch. The survey has been coded using JavaScript, PHP, and AJAX. On server side, we held a MySQL database to store survey data. The pattern lock simulation is created by using RaphaëlJS, a vector graphics library for drawing objects. Our survey, along with the pattern lock simulation, is compatible with Firefox 3.0+, Safari 3.0+, Chrome 5.0+, Opera 9.5+, and Internet Explorer 6.0+. The pattern lock simulation is coded for both mouse and touch devices. In this way, we aimed to maximise the number of participants. One limitation was the RaphaëlJS's use of SVG for vector graphics, which is not supported by Android versions older than 3.0. We tested alternative libraries to RaphaëlJS that do not make use of SVG in order to see

whether a transition was feasible. Conducting the test on Samsung Galaxy S, one of the most sold Android devices for version 2.x, we came to conclusion that the device simply cannot handle these JavaScript libraries well enough to provide an acceptable performance. Given the fact that the majority of other smartphones of the Android 2.x era contain similar hardware, we decided to stick with RaphaëlJS.

We hosted our survey on patternsurvey.biz domain. The database contained three additional fields, namely time stamp, IP, and proxy IP. These fields were required to ensure the authenticity of the entries, and to remove duplicates. The results presented in this dissertation are calculated after filtering the database from irrelevant entries.

The survey starts with basic demographics, continues with questions about participant's smartphone experience and his/her opinion on the notion of device locking, and finalises after two pattern entries. The survey questions and their respective options are presented below with corresponding analysis. Extensive pattern analysis follows suit.

We have distributed the survey through social networks and University of Bristol MSc student mail lists. There have been 144 unique participants to the survey.

4.2 Questions

4.2.1 Question 1

Please select your gender:

- Male
- Female

Of the participants, 95 were male, and 49 were female. That is 65.97% and 34.03% respectively.

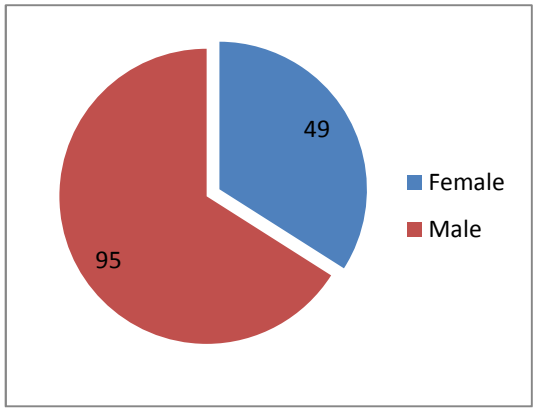

Figure 22. Participants by gender.

4.2.2 Question 2

How old are you?

- Under 18
- 18-29
- 30-49
- 50+

The majority of the participants were aged between 18 and 29 inclusive. The following age bracket is 30-49. These brackets make 81.25% and 15.28% of the participants respectively. The main reason for the high percentage of 18-29 years old participants is a direct result of the social network connections and MSc students being mostly within this age bracket.

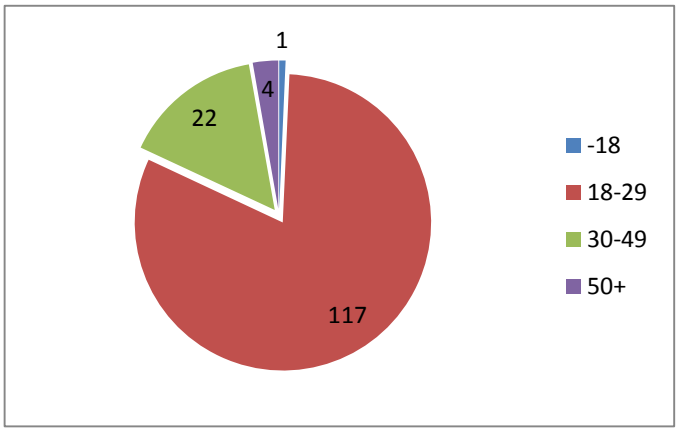

Figure 23. Participants by age brackets.

4.2.3 Question 3

Have you ever owned a touchscreen smartphone?

- Yes
- No

4.2.4 Question 4

Do you currently own a touchscreen smartphone?

- Yes
- No

This question is dependent to Question 3. Participants who have never owned a smartphone cannot answer this question as it is blocked.

79.86% of the participants have owned a smartphone at one time in their lives, and 92.17% of them currently own a smartphone.

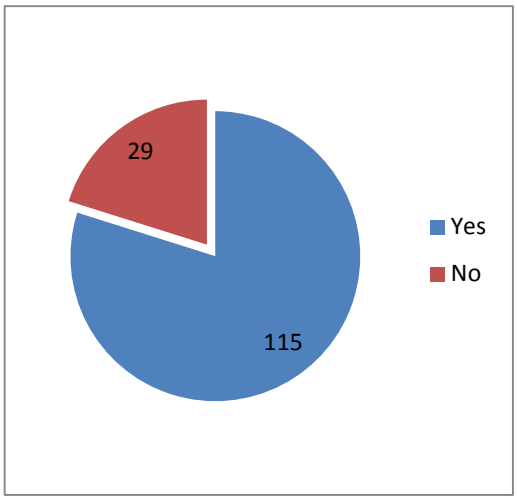

Figure 25. Participants that have ever owned a smartphone.

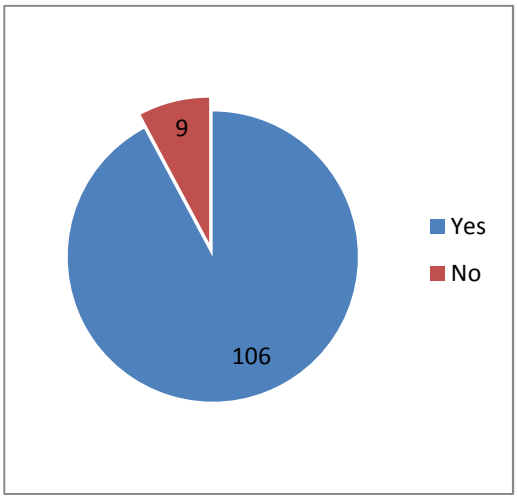

Figure 24. Participants that currently own a smartphone.

4.2.5 Question 5

What is your current smartphone's mobile operating system?

- Android
- Blackberry
- iOS (iPhone)
- Symbian
- Windows Phone
- Other

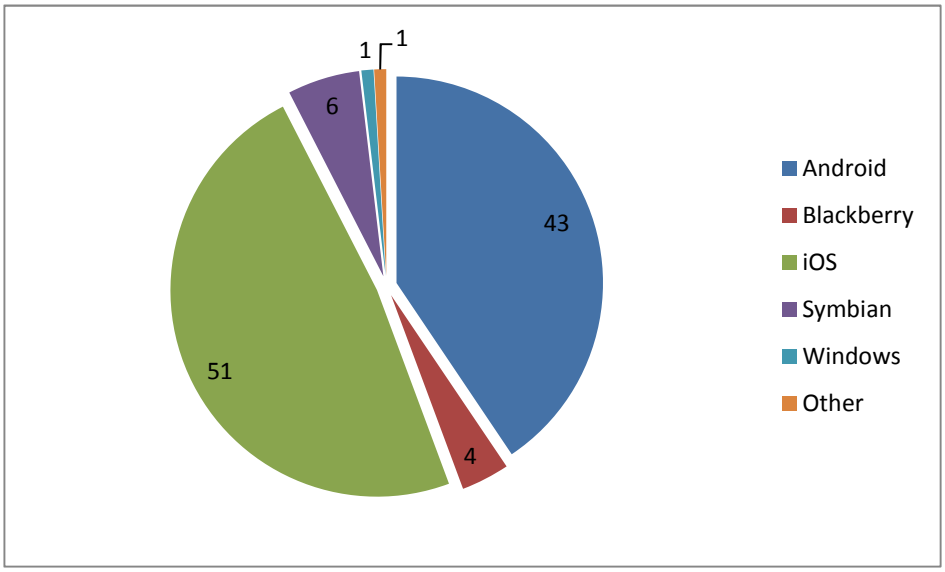

Figure 26. Operating system breakdown.

This question is dependent on the Question 3 and 4: it is only visible if the participant answered "Yes" for both of these questions. Thus, the smartphone operating systems data is collected from participants who currently own a smartphone. One can observe that the results are clustered on two fronts: 48.11% of the participants use iOS, and 40.57% use Android. Symbian and Blackberry follow with 5.67% and 3.78% respectively.

4.2.6 Question 6

How long did/have you been using a touchscreen smartphone?

- Less than a year
- 1-2 years
- 3+ years

This question is dependent on the Question 3 and thus is asked to only participants who owned a smartphone. 20.87% of the participants has less than one year of experience, 36.52% has one to two years of experience, and 42.61% has three or more years of experience. This particular result demonstrates that the smartphone usage has been more than doubled in two years. It is a remarkable growth, which indicates how important smartphone security has become.

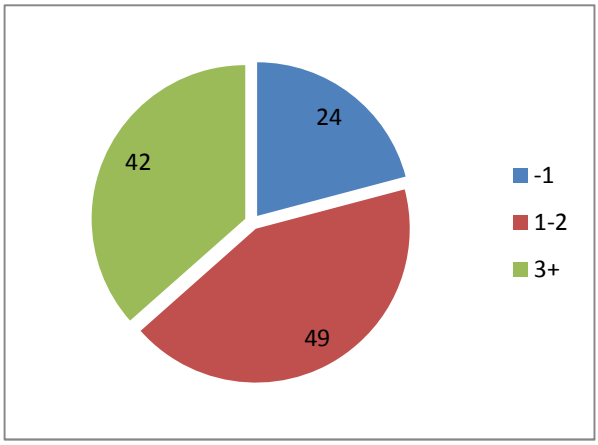

Figure 27. Smartphone experience by # of years.

4.2.7 Question 7

Have you ever used an Android phone, if so, how long?

- I have never used an Android phone before.
- Less than a year
- 1-2 years
- 3+ years

Android experience data is collected from participants who ever owned a smartphone. 58 participants have experienced Android platform. Of those 58,

41.38% has less than a year of experience, 48.28% has 1-2 years of experience, and only 10.34% has 3 or more years of experience.

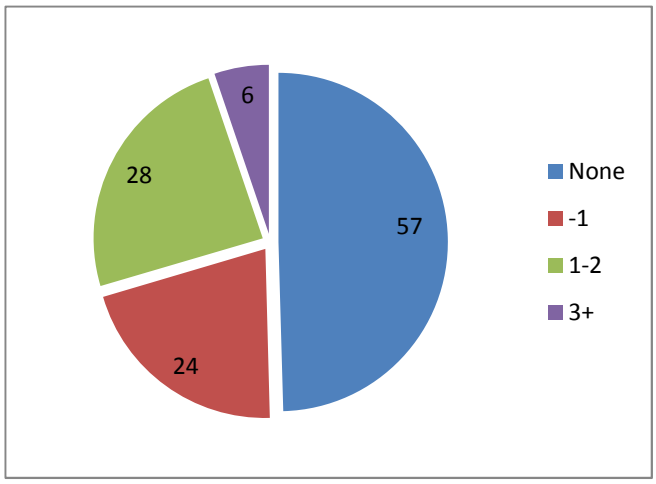

Figure 28. Android experience by # of years.

This result is in correlation with the market share of Android. While Android had only 4.7% of the market share in 2009, it now has 68.1% for Q2 2012 according to Canalys's estimations. This skyrocketing pace in recent years demonstrates the reason why there are much more participants with less experience. Note that the figures are annual for Figure 29, and quarterly for Figure 30.

Worldwide smart phone market by OS vendor
Market shares 2009, 2008

OS vendor	2009 shipments	% share	2008 shipments	% share	Growth 2009/2008
Total	166,271,050	100.0%	143,067,530	100.0%	16.2%
Symbian	78,511,980	47.2%	74,926,550	52.4%	4.8%
RIM	34,544,100	20.8%	23,562,650	16.5%	46.6%
Apple	25,103,770	15.1%	13,727,740	9.6%	82.9%
Microsoft	14,679,720	8.8%	19,945,530	13.9%	-26.4%
Google (Android)	7,786,870	4.7%	663,550	0.5%	1073.5%
Others	5,644,610	3.4%	10,241,510	7.2%	-44.9%

Source: Canalys estimates, © Canalys 2010

Figure 29. Android market share in 2009. Also notice the market penetration rate. (Canalys, 2010)

Global smart phone market

Shipments into the channel, split by platform, Q2 2012, Q2 2011

Platform	Q2 2012 shipments (million)	% share	Q2 2011 shipments (million)	% share	Growth Q2'12/Q2'11
Total	158.3	100.0%	107.7	100.0%	46.9%
Android	107.8	68.1%	51.2	47.6%	110.4%
iOS	26.0	16.4%	20.3	18.9%	28.0%
BlackBerry	8.5	5.4%	12.5	11.6%	-32.1%
Symbian	6.4	4.1%	18.1	16.8%	-64.6%
Windows Phone	5.1	3.2%	1.3	1.2%	277.3%
bada	3.3	2.1%	3.1	2.9%	5.1%
Others	1.2	0.8%	1.1	1.0%	15.2%

Source: Canalys estimates, © Canalys 2012

Figure 30. Android market share in Q2 2011 and Q2 2012. (Canalys, 2012)

4.2.8 Question 8

Are you using any screen lock on your mobile phone which requires PIN, password, pattern etc. to unlock?

- Yes
- No

47.22% of the participants use lock whereas 52.78% do not.

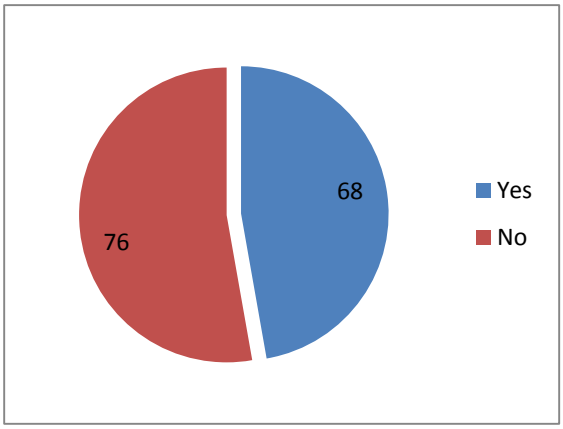

Figure 31. Lock usage of the participants.

4.2.9 Question 9

What are the reasons you use a screen lock?

- I want to protect my personal data (contacts, messages, photos, mail etc.).
- I have sensitive data regarding my business.
- I want to prevent people making calls on my unattended phone.
- I just don't like others fiddling with my phone.
- Other

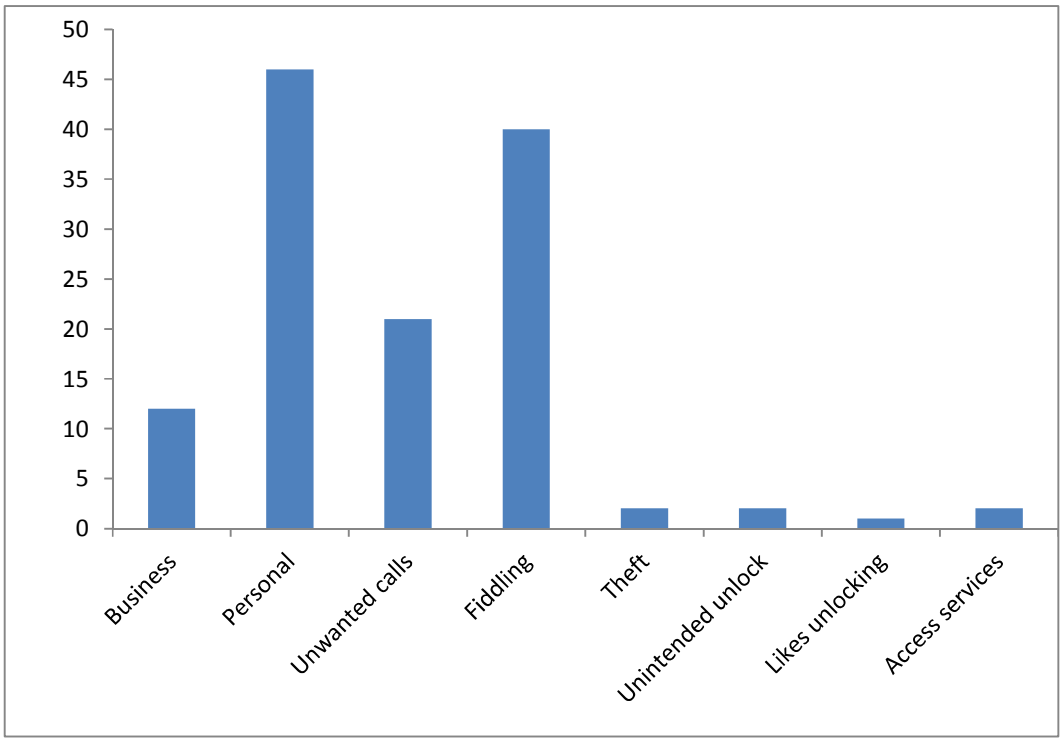

Figure 32. Lock reasons including custom responses.

This question is only asked to participants who lock their phones, since forcing the participants not using a lock mechanism to answer it could lead to pollution of data with random entries. For this question, the participants were allowed to choose multiple options, as well as entering their own reasons for locking their devices. According to the results, most of the participants have their phones locked either to protect personal data or to prevent others fiddling with the device. While these options have been chosen by 46 and 40 participants respectively, the option of

locking to prevent calls while away from the phone has been selected by 21 participants. Only 12 participants ticked the option "I have sensitive data regarding my business".

We also had participants providing additional reasons. One of the locking reasons added is to increase the security in case the phone is stolen. Having a lock would prevent immediate access to the smartphone's content, and may allow the user find the phone using the "find my phone" type of services. Another reason mentioned is to prevent the smartphone against unintended unlocks. This may occur whilst the phone is in a pocket. If the phone's screen is turned on by mistake, the friction caused by walking may unlock the device, and thus the phone would be prone to random commands issued by the friction. Some participants, on the other hand, required to use a lock mechanism to access certain services such as receiving emails from a specific entity (such as university or company). Finally one participant mentioned that he/she uses a lock just because he/she likes the idea of unlocking a device.

4.2.10 Question 10

There are various ways your mobile device security lock can be compromised. These may include the following ones. Please sort them according to your opinion in the order of risk level (1 being highest risk, 3 or 4 lowest one).

- Shoulder surfing (somebody standing behind your shoulder)
- Cameras in the room
- Inspections of smudges left on the touchscreen
- Other

Although we employed thermal and microscope attack, such options are deliberately left out in order to observe if any participant would recognise it as a risk.

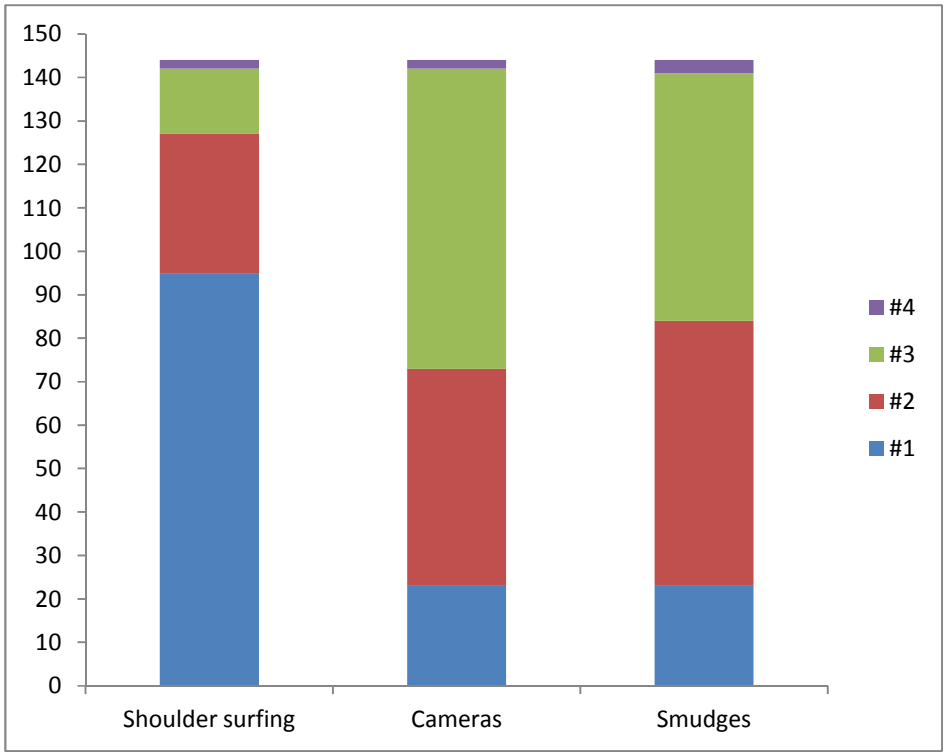

Figure 33. Risks of passcode retrieval by importance: Given choices.

In this graph, we have aggregated the answers given to each option, and stacked the results from the highest risk (#1) to the lowest risk (#4).

According to the gathered data, 65.98% of the participants believe that the highest risk that would compromise a lock is shoulder surfing. Smudges left on the screen and cameras in the room have the same rating of being the highest risk with 15.97%, yet the former has been selected more times as the second highest risk compared to the latter, rendering it the second highest risk after the shoulder surfing.

The other risks mentioned by the participants were just as valid as the options we have provided. Five participants wrote that they could be forced to tell the passcode. According to the aggregated results, he person who requests the passcode can be a family member, a friend, or a police officer. It is followed by the risk of the passcode being written down somewhere. This is a common mistake made by users with text passwords, but the Android pattern lock would be less

likely prone to this type of risk as it is a graphical passcode. Another risk is the reflection. The particular entry talks about reflection through glasses can reveal the passcode. It is possible to expand the idea by adding window glasses and mirrors as reflective surfaces that could be used to retrieve a passcode without direct line of sight to the screen. Finally, trials and weak codes have been shown as risks by the participants. They are linked to each other since having a weak code leads to success on trials. While the weak codes can be "0000" or "1234" type of PINs, they can also be personal numbers like birth date. This notion is also applicable to Android pattern lock, as patterns can be easy to guess like an "L" shape, or a "Z" shape.

There was no mention of a thermal attack or an attack using a microscope by any participant, which means these are not expected type of attacks. Yet as shown, they at least pose a threat to the Android pattern lock.

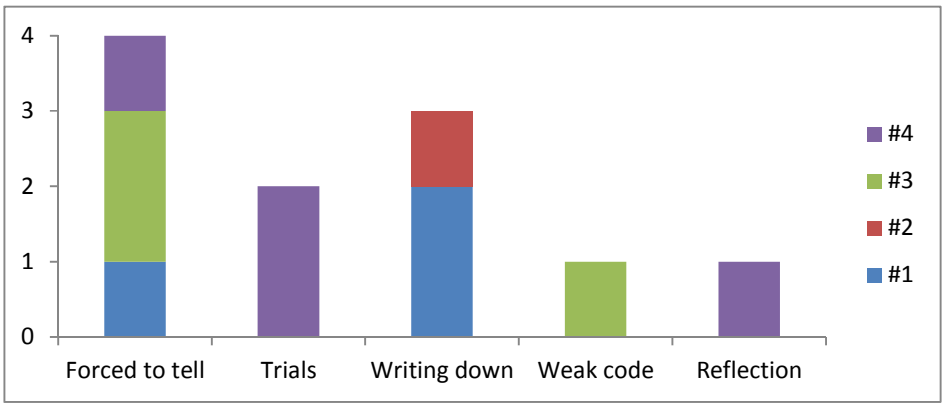

Figure 34. Risks of passcode retrieval by importance: Others.

4.2.11 Question 11

Would you consider using the "secure" pattern you entered in everyday life?

- Yes
- No

57.64% of the participants thought the secure pattern is usable in everyday life, while 42.36% did not.

4.2.12 Question 12

Do you think the "easy to remember" pattern you entered is secure enough?

- Yes
- No

35.42% of the participants thought the easy pattern they entered was secure enough, while 64.58% did not.

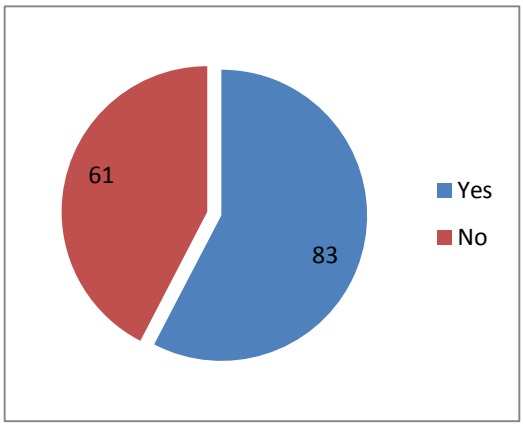

Figure 35. Is secure pattern's usable in everyday life?

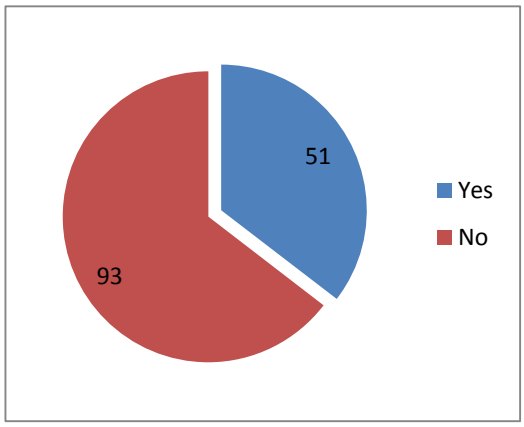

Figure 36. Is easy pattern secure enough?

For Question 11 and Question 12, we deliberately used specific terms such as "secure enough", and "everyday life" in order to let participants impose their subjective opinions.

4.3 Secure Pattern Analysis

4.3.1 Average Pattern Lengths

As part of our analysis, we calculated the average pattern length of the secure and easy patterns. The average length is calculated by summing the number of dots used, and then dividing that value to the number of participants. An interesting result came up when we calculated the average pattern lengths for male and female participants. While the average pattern length for a secure pattern drawn by a male participant is 6.88 dots, female participants averaged 6.16 dots. The same situation can be observed in easy pattern: males' average is 6.32 dots while females' average is 5.94 dots. The average lengths for secure and easy patterns are 6.64 and 6.19 respectively. Considering the fact that at least four dots must be connected to form a valid pattern, the ratio of male secure pattern average to female secure pattern average definitely provides valuable information regarding the perception of security between the two sexes. As a result, it is easier to fully retrieve a female user's pattern compared to a male user's pattern on average.

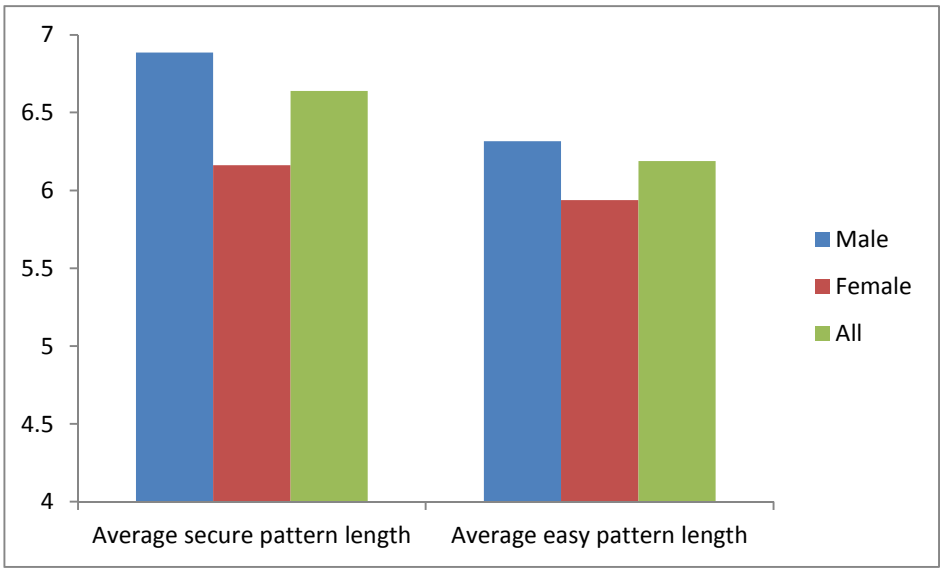

Figure 37. Average pattern lengths by gender.

4.3.2 Average Number of Direction Changes

Another indication of how secure a pattern is the number of direction changes made per pattern. Note that at least one direction change must be done to draw a valid pattern. We assume that, for humans, a direction change is a more difficult move than following a direct line. Consequently, we assume that the more direction changes made in a pattern, the more complex, hence the more secure it gets against guessing a pattern by solely relying on human behaviour study. The average number of direction change made in a secure pattern is 3.57, and the average number of direction change made in an easy pattern is 2.74. The ratios of number of direction change to pattern length are then 3.57:6.64 and 2.74:6.19 for secure and easy patterns respectively. This demonstrates that the secure patterns have more direction changes with respect to their lengths, rendering them more complex. As observed in average pattern lengths, female participants make less direction changes whilst setting a pattern.

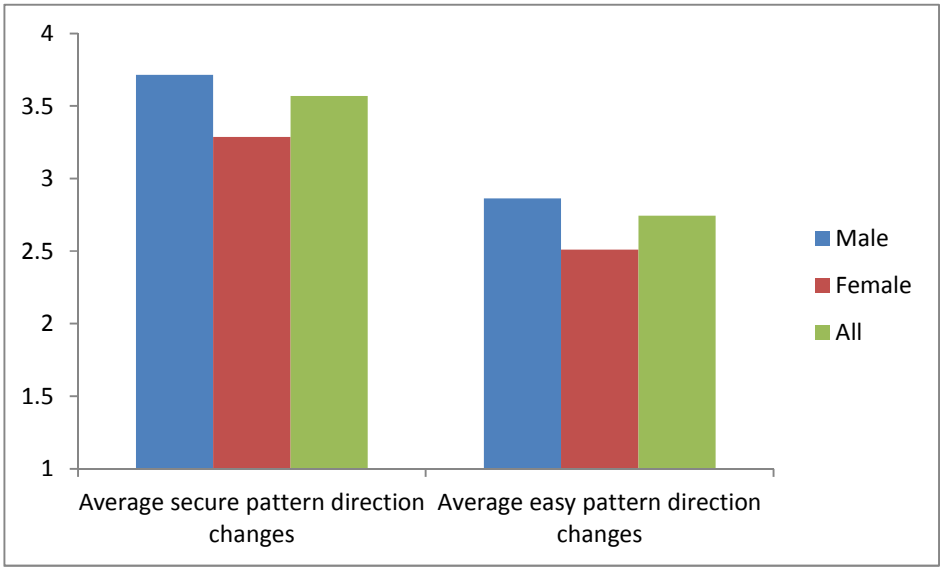

Figure 38. Average number of direction changes by gender.

4.3.3 Entropy

In the following sections of survey data analysis, we calculated the Shannon's entropy while studying sub-patterns, and start and end points for the secure patterns. The sub-patterns' entropies are calculated bearing in mind the conditional probabilities. For the conditional entropy calculations we used the Shannon's formula (Shannon, 1951):

$$F_N = -\sum_{i,j} p(b_i, j) * \log_2 p(b_i, j) + \sum_i p(b_i) \log_2 p(b_i)$$

where b_i is (N-1)-gram,

j is an arbitrary dot that has not yet been chosen, and

$p(b_i, j)$ is the probability of the N-gram b_i, j.

Note that in the case of sub-patterns, we take $p(b_i, j)$ as $p(b_i||j)$, where || stands for concatenation. For instance, if b_i = “01”, and j = “2”, then

$$p(b_i, j) = p(\text{"012"}) = \frac{\#\ occurences\ of\ trigram\ \text{"012"}}{sum\ of\ occurences\ of\ all\ trigrams}.$$

4.3.4 Start Points

One of the major results we obtained is about how the participants preferred to start their secure patterns. More than half, 52.08%, of the participants started their secure patterns from top left dot. Entropy of the start points is 2.35 bits, as opposed to a maximum of $\log_2 9$ = 3.17 bits for which all the dots must have the same probability. This imposes heavy bias, and makes the first dot highly predictable. This surprising result may have a few factors affecting it.

The first one is the shape of the human hand. While holding a phone on the right hand, the thumb naturally idles near the top left of the screen. This fact makes it effortless to start a pattern from the top left dot. It is important to note that the survey did not examine whether the user is right-handed, left-handed, or ambidextrous. Additionally, the survey could be filled using a computer, which

means participants might have used a mouse to draw the pattern. Nevertheless, participants consistently chose the top left dot as the starting point.

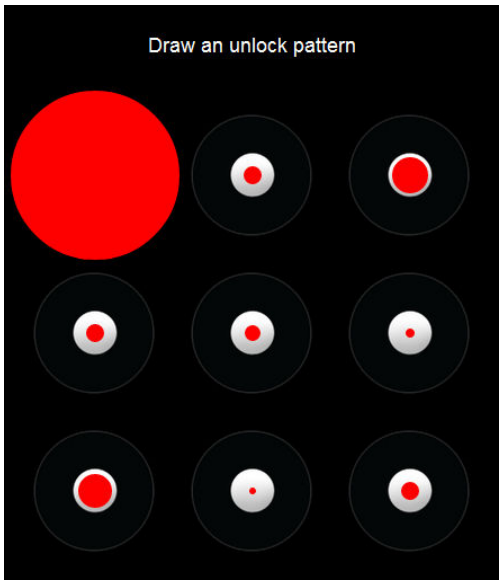

Figure 39. Start point visualisation. Radius depicts the frequency.

Another reason for this clustering can be linked to the participants' geographical positions. Most of the entries originate from United Kingdom, United States, and Turkey. All these countries' native alphabets consist of Latin characters, and consequently their writing starts from the top left, and ends in the bottom right. In addition, the survey's language is English, which may make the participants think in Latin language style even if he/she has a non-Latin based native alphabet. As a result, the participants may be inclined to start from top left, as it is a more natural start point for them. The survey is not conducted on other languages, or in specific regions. Collecting data from participants that have top-to-bottom or right-to-left native languages could provide interesting results.

Apart from the top left dot, the top right and bottom left dots are also noticeable. These two dots have been selected more than the other dots. Once again, these two corners are physically easier for the thumb to reach during normal operation of the phone by a right-handed person as opposed to the bottom right corner.

Finally, the intermediate dots are seldom chosen by the participants. This may be the result of a psychological issue: these dots are just in between other dots, and do not represent a suitable start point.

4.3.5 End Points

We then checked the ending dots for secure patterns. While there was not a single dot on which most participants preferred to end their secure pattern, the bottom right was the most selected one with 20.83%. The entropy calculated is 3.00 bits for

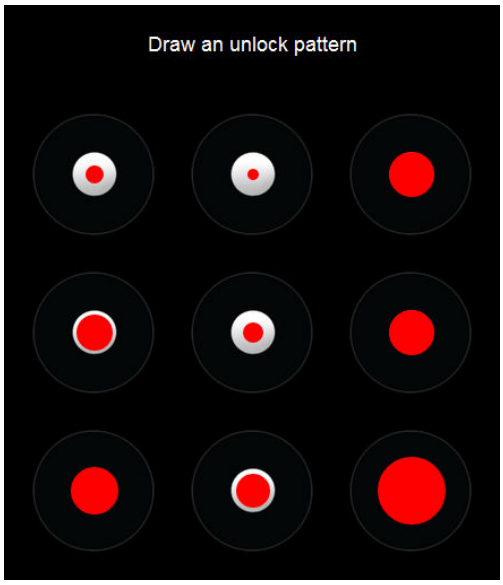

Figure 40. End point visualisation. Radius depicts the frequency.

the probabilities of end points. The ending dots were mostly focused on right and bottom. As expected, this result also conforms to the assumption about the Latin alphabet made on the analysis of start points. Participants tend to finish their patterns on a psychologically suitable end point.

4.3.6 Sub-patterns

One of the main purposes for conducting the survey was to look for popular sub-patterns within the gathered patterns. Extracting these sub-patterns would allow an attacker to guess a partially retrieved pattern's missing parts easier. In other words, an attacker can incorporate the physical attacks with the behavioural attacks to fully retrieve the pattern.

4.3.6.1 Monograms

The first step in the search of sub-patterns is analysing the monograms, which involves searching for the frequency of each dot to see if there is any dot particularly frequently chosen. The result depicts that there is no significant bias towards any of the dots; they are more or less equally used in patterns, hence the entropy of the monograms is 3.16 bits.

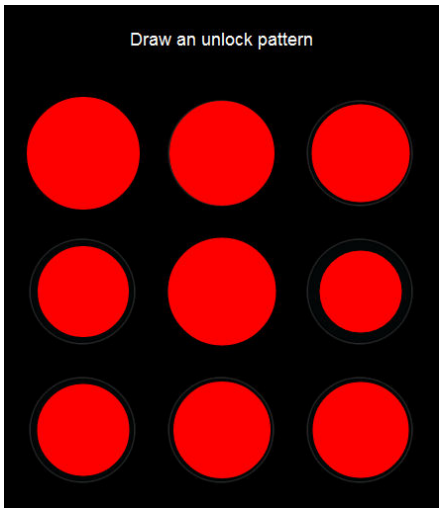

Figure 41. Monogram visualization.

4.3.6.2 Bigrams

We then looked for bigrams, a sub-pattern consisting of two dots. Since bigrams and other longer n-grams create a path, the directionality is taken into account while analysing.

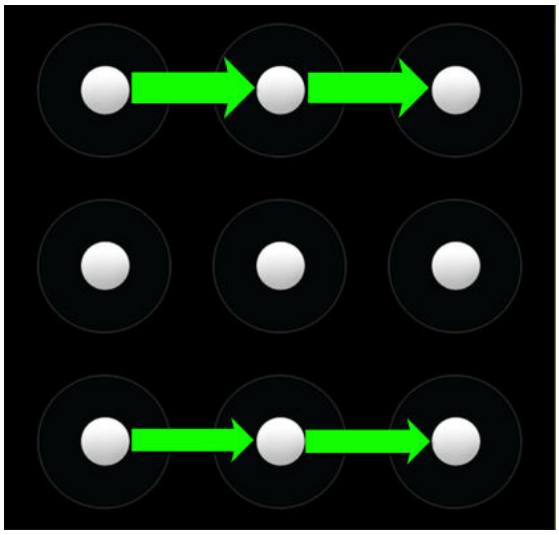

Figure 42. Most frequent bigrams.

There have been some bigrams that occurred especially frequently in patterns collected. Given in the Figure 42, the width of the path depicts the frequency of that particular bigram. In this case, the thickest path represents that 32.64% of the participants drew that bigram, while the thinnest represents 23.61%. Using Shannon's entropy, bigrams' entropy is calculated as 5.47 – 3.16 = 2.31 bits. The maximum entropy for the bigrams is $\log_2 72$ = 6.17 bits. Out of 72 different combinations possible, 64 of them were drawn at least by one participant.

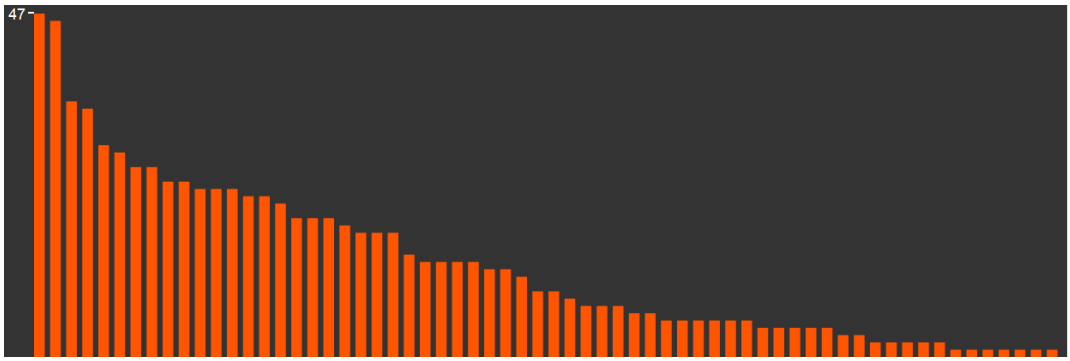

Figure 43. Bigrams sorted by the frequency.

4.3.6.3 Trigrams

Continuing with trigrams, the analysis shows that 18.75% of the participants drew a path from top left dot to top right dot at one point of their patterns. The thinnest path in the figure on the right represents 14.58%. The trigram entropy is 6.99 – 5.47 = 1.32 bits. The maximum entropy for the trigrams is $\log_2 504$ = 8.98 bits. Out of 504 different combinations possible, 203 of them were drawn at least by one participant.

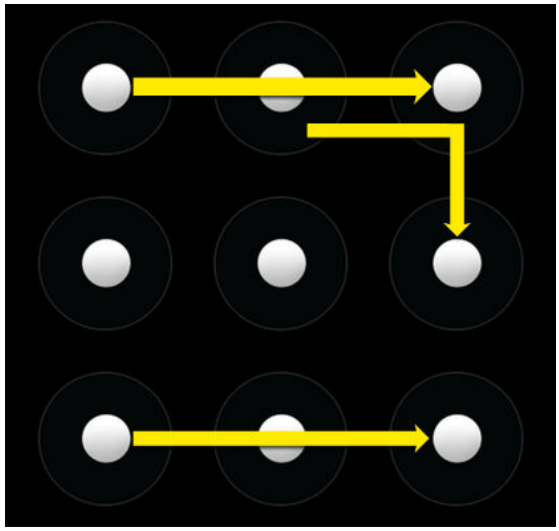

Figure 44. Most frequent trigrams.

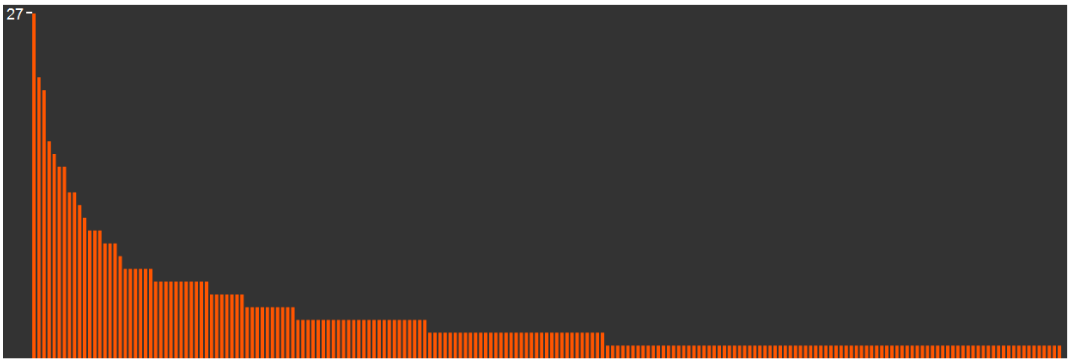

Figure 45. Trigrams sorted by the frequency.

4.3.6.4 4-grams

Finally, we made 4-gram analysis. Three 4-grams particularly stood out of the rest with two of them being drawn by 9.02% of the participants, and the other being drawn by 7.64%. It is easy to spot a trend towards left to right, and top to bottom in these sub-patterns, which contributes to the assumptions made on the psychological behaviour the participants display. The 4-gram entropy is 7.75 - 6.99 = 0.76 bits. The maximum entropy for the trigrams is $\log_2 3024$ = 11.56 bits. Out of 3024 different combinations possible, 283 of them were drawn at least by one participant.

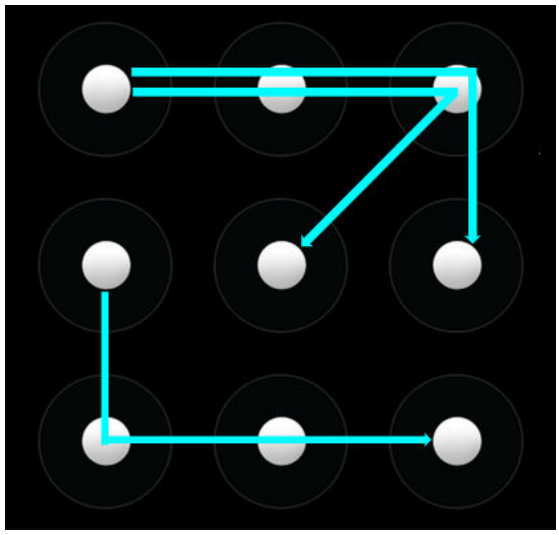

Figure 46. Most frequent 4-grams.

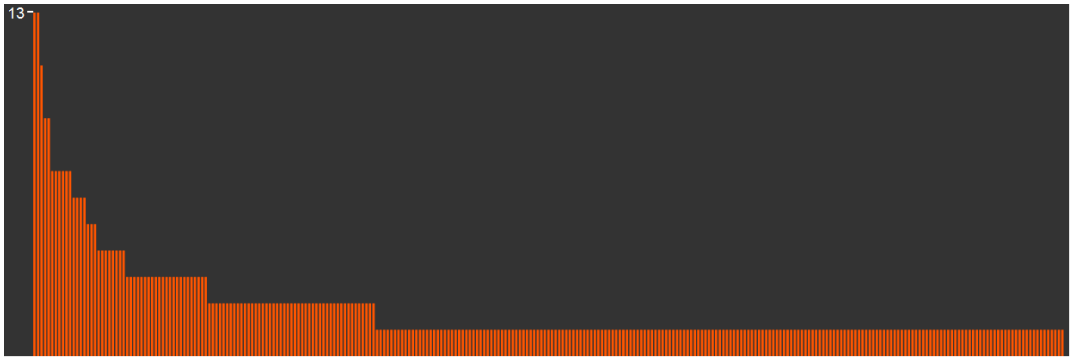

Figure 47. 4-grams sorted by the frequency.

5 A Possible Solution

The results obtained from the physical attacks and the survey demonstrate that Android pattern lock is not secure against smudge attacks, even if the smudge has partially disappeared. Using the gathered data from the results, one can guess the missing parts of a pattern much easily. Thermal attack also maintains its threat level if the phone was idle beforehand. To circumvent these issues, we provide a solution that still incorporates the smudge action as opposed to tapping, but that bypasses any attack presented in this work.

5.1 Extended Pattern Lock

An approach to increase the security while maintaining the pattern lock mechanism would be to relax the constraints of the pattern lock. Removing the constraint of "each dot can be visited only once", we can increase the number of possible patterns, and allow smudge to be overwritten.

Letting the user pass through the same dots more than once, we can render the smudge and thermal attacks to retrieve the pattern harder. For instance, assuming the user goes from top left dot to top right dot, and then back again to top left dot following exactly the same smudge left on the screen, these attacks cannot find out the first path. However, in reality, since the screen does not act as rails, i.e. the finger can roam freely and not in strict lines, the oily residue left on the screen whilst drawing the first path (left to right) would be slightly different than the oily residue left whilst drawing the second path (right to left). This variation would allow smudge attacks to gather some

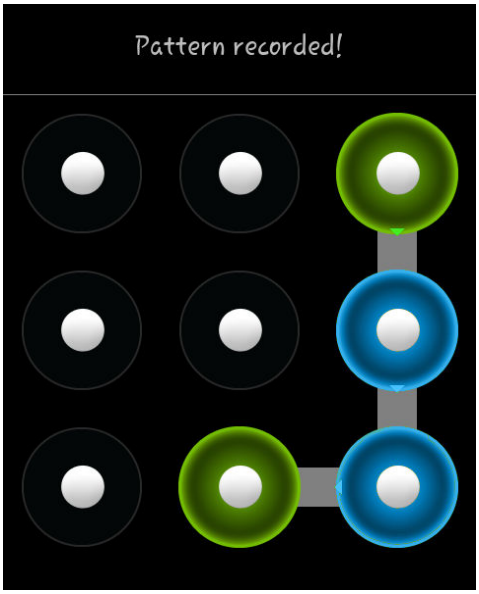

Figure 48. Extended pattern lock mockup.

information about the first path since there would be a part of the first path not overwritten by the second path. While it is also possible to gather the same information using the thermal attack, one may need higher end equipment and a larger difference between the paths to be able to discern the two paths.

Another issue with this approach is that when it is used with visible paths upon unlocking, the overwritten paths need to change colour to indicate that it has been overwritten. Each overwriting would need a new colour, and the mechanism requires having an upper limit on the number of overwriting of the same dot. Therefore, displaying a visible path would make the pattern retrieval by shoulder surfing or video recording as easy as the standard pattern lock. On the other hand, not displaying any path would make it harder to unlock since it may lead to uncertainty of the user about how many times he/she overwritten a particular path, and where to draw next.

Moreover, this extended mechanism would hardly be implemented in an Android operating system (ROM), since adding it separately would cause confusion for having two pattern locks, and replacing the original pattern lock is simply unrealistic.

Overall, the extended pattern lock approach is too complex for the average user, and does not sufficiently increase the security.

5.2 Auto-hidden Numerical Wheel Lock

Our solution is an auto-hidden numerical wheel styled PIN screen. The number wheels are initially hidden, only displaying their respective borders. This allows the user to see where the numerical wheels are, yet prevent any by-passers, or short term shoulder-surfers to see the passcode. Upon the user touching inside the borders of any of the numerical wheels, that particular wheel becomes visible, rendering the user able to scroll the digit for that wheel towards the correct position. The correct position is the centre of the wheel, just as in briefcase locks. The wheel gets hidden again when the user releases his/her finger from the phone. After adjusting all the wheels to their right positions the user presses Unlock button

to unlock the device. If the PIN is correct, the device unlocks, otherwise it asks again for the correct PIN. In case of wrong PIN, the wheels are not reset to their initial position, making corrections easier. The maximum number of trials can be set depending on the implementation, but the default is the same as Android PIN lock.

Figure 49. Initial state of the auto-hidden numerical wheel lock.

Figure 50. State upon touching a wheel.

5.2.1. Options

5.2.1.1 Numerical Wheels' Initial Positions

By default, once the phone is locked, the numerical wheels' initial positions are randomized. In other words, instead of having "0" as the initially selected digit, the wheel is rolled and stopped in a random digit. This randomization occurs once per unlock session, i.e. if the first unlocking trial fails, the wheels are not randomized again but instead stand wherever they were left by the user.

5.2.1.2 Number of Wheels

By default, the number of numerical wheels follows the PIN code standard, and therefore it is four. However, depending on the width of the wheels, this value can be increased up to six.

One can argue that randomizing the standard PIN code number layout (i.e. switching digits' placements) would result in a similar level of security since the fingerprint positions would not reveal the correct digits. While this is indeed true, the usability would be lower compared to a non-randomized PIN code layout. The user would struggle to find where the number is each and every time. This is not the case in numerical wheel case, however. Since the numbers are always in order within the wheel, finding the correct digit in a randomly initialized wheel would be much easier and user friendly than in a randomized PIN code screen. Additionally, it has no graphical element, nor does it incorporate a swiping gesture.

5.2.2 Security Assessment

The proposed solution provides more security than the Android pattern lock on most situations.

5.2.2.1 Smudge Attacks

To unlock the auto-hidden numerical wheel lock, the user has to leave four lines of smudge on the screen. The positions of the smudges do not reveal any information about the PIN, since the wheels are always in the same place on the screen. On the other hand, the direction and the length of the smudges can leak information about the PIN if the user prefers that the wheels have fixed initial positions. For this reason, the wheels are in random positions at the beginning of each new unlock session by default. This randomization invalidates the previous claims about the directionality and length of the smudges, rendering them obsolete for any information retrieval about the PIN.

5.2.2.2 Thermal Attack

Similar to smudge attacks, thermal attacks on screen do not provide any particular hint regarding PIN. Neither the directionality, nor the position, nor the length indicate any information about the PIN due to the exact same reasons listed in the smudge attack assessment. However, since there is also the time dimension in thermal attacks, the attacker can gather an additional knowledge about the order the wheels are used. Obviously, the first used numerical wheel would have the coolest heat trace, while the most recent one would be the hottest. Yet the order of the heat traces only shows the user's arbitrary choice of numerical wheels. Since the user does not know which numerical wheel's initial positioning is closer to its correct digit, he/she cannot inadvertently reflect any psychological behaviour to numerical wheel selection.

5.2.2.3 Attacks on Human Behaviour

Another major advantage of the auto-hidden numerical wheel lock is that it prevents any user behaviour to be imposed on it. The lack of pattern eliminates the chance to analyse any sub-pattern, start or end point. Since the wheels are also randomly initialized, there is no special attack that can be employed to retrieve extra information about PIN via behaviour analysis. However, the known PIN attacks are applicable, like trying the user's birth year or license plate as PIN.

5.2.2.4 Shoulder Surfing/Video Recording Attacks

These types of attacks rely on the observer to be able to see the screen for the whole duration of the unlocking. Our proposed solution is certainly prone to this type of attack. Nevertheless, it is also more robust compared to the pattern lock, standard PIN lock, or password lock. This robustness is granted by two factors: the hidden numerical wheels and the time required to unlock the device. The hidden numerical wheels prevent the retrieval of previously set digits by short glimpses. The fact that it takes longer to unlock the device by setting each wheel separately prolongs the observation duration to fully retrieve the passcode. The longer the observation duration gets, the more likely it is to have some obstruction between the screen and the observer, which would lead to a partial retrieval of PIN.

5.2.3 Disadvantages & Constraints

The auto-hidden numerical wheel lock has its own disadvantages.

Having four to six separate wheels to scroll renders the unlocking process longer than the Android pattern lock. While the pattern lock is drawn in one touch and release of the finger, the wheels require separate touches. Additionally, since the initial positions of the wheels are randomized, the user needs to process the information, and decide which way to scroll for the shortest path to the correct digit. However, the unlocking duration for our proposed solution can be reduced significantly if the user uses two fingers or thumbs to scroll two different wheels.

The other disadvantage is that the lock is prone to standard PIN attacks. If a user sets a guessable PIN, the lock's design cannot help to prevent it. However, easy passcodes are a common issue for all the lock systems where the passcode is something that he/she knows, as opposed to something he/she is (i.e. fingerprint) or something he/she has (i.e. key).

While the lock mechanism we proposed is relatively easy to code as an application, there is an important constraint that needs to be addressed for the sake of a more realistic approach to the distribution and wide usage of such lock.

As previously mentioned, Android is an open source operating system. This allows manufacturers like Samsung, HTC, LG etc. to make custom tailored versions. These custom ROMs contain apps that are embedded into the core system of the Android. This opportunity lets manufacturer to preload undeletable apps, and/or slight variations on the inner workings and options of the operating system. Lock screens are naturally a part of the core system. Therefore, any new lock mechanism needs to be added to the system itself to operate as intended. Publishing a lock application through the app store would not provide any security as a long press of the home screen button pops up task manager, bypassing any standard application.

We can always prepare a custom ROM that includes this lock mechanism, but it would be an unnecessary effort without any support behind the ROM. To make the auto-hidden numerical wheel lock mechanism a viable option, we would require the support of Google or one of the many manufacturers that customize and release

their own official ROMs. Being part of such a ROM, the lock mechanism can reach a large amount of users, and thus be an effective alternative to other lock mechanisms.

Home screens, i.e. the screen that welcomes the user when the device is unlocked, are managed by applications called home launcher. Android's customizability allows users to change their home launcher. One could argue that Android home launchers already provide lock screens, and that our solution could well be bundled with a popular launcher to reach the users. However, although the lock screens provided by the home launchers seem as secure as native ones, there are in fact ways to circumvent them. For instance, the lock screens offered by GO Launcher (Go Launcher Dev Team, 2012) disable the long press of home button, which brings up the task manager screen. Yet, it is possible to make it work by first long pressing the power button, and then long pressing the home button. This would allow the attacker to uninstall the launcher via task manager. It is also possible to do the same by touching the notification bar and long pressing home button simultaneously. Apparently, this occurs since the home launchers cannot completely disable the user interaction of the notification bar. As a result, it is clear that it is not a valid option to implement a lock mechanism in a home launcher.

6 Conclusions

Our background research showed that there was a lack of literature on Android pattern lock analysis. While an optical attack against smudges was studied, attacks using microscope or thermal imaging were not researched. Moreover, there was neither a pattern analysis, nor an attack based on human behaviour. However, related studies demonstrated enough evidence for us to look for sub-patterns in Android pattern lock, and study human behaviour. In contrast with the number of studies on Android pattern lock analysis, there were many alternatives to replace the Android pattern lock. We analysed their strengths and weaknesses in order to come up with a secure and user-friendly lock mechanism.

Both the smudge attacks conducted with optical camera and microscope proved to be effective. Different from the Aviv, et al. (2010)'s research, we demonstrated that a compact camera can also provide feasible results, and thus practically used in real life for low profile attack attempts. The microscope attack displayed a greater amount of detail in the pattern, and performed slightly better than the compact camera in cases where the screen was strongly cleaned.

The thermal attack provided positive results when applied on an idling device, albeit the heat trace was not visible more than a few seconds. The heat radiation of the hardware proved to be too intrusive for the attack to be successful on a running device.

The attacks on human behaviour, i.e. the analysis of patterns, produced significant results. The results showed that start point was highly predictable, that there were outstanding n-grams, that the females were feeling secure with less complex and shorter patterns than males.

In the light of all the attacks we have investigated and their analyses, we offered a possible solution that is immune to the physical attacks. In terms of psychological attacks, our solution is also immune to attacks the Android pattern lock suffers, yet it is prone to standard PIN attacks that rely on personal information. Additionally, it is more robust against shoulder surfing and video recording compared to Android pattern lock.

7 Future Work

Since this study contains many firsts, we believe there is room for future work.

For instance, the picture captured using optical or thermal attacks may be automatically analysed using image processing.

The survey can be converted to an Android application, and a controlled survey can be conducted to better understand the effect of right/left-handedness, having native non-Latin languages, and/or not knowing any Latin based language.

Furthermore, the range of devices can be expanded for physical attacks. There is a plethora of devices including smartphones, tablets, and all the hybrids with different screen sizes and different screen surfaces. Additionally, heat radiation can be less intrusive on bigger devices.

Last but not least, the effectiveness of the smudge resistant oleophobic screen protectors can be analysed against physical attacks.

8 Bibliography

Airowaily, K. and Alrubaian, M., 2011. Oily residuals security threat on smart phones. *International conference on robot, vision and signal processing.* Kaohsiung, Taiwan, 21-23 November 2011. IEEE Computer Society.

Angulo, J. and Wästlund, E., 2011. Exploring touch screen dynamics for user identification on smart phones. *IFIP summer school.* Trento, Italy , 5-9 September 2011.

Anttalainen, O., 2012. *Combination Lock.* [online] Available at: <https://play.google.com/store/apps/details?id=com.aoa.CombinationLock> [Accessed 8 May 2012].

Appsolute Solutions LLC, 2012. *iLockit Lock Screen.* [online] Available at: <https://play.google.com/store/apps/details?id=com.appsolute.ui.paid> [Accessed 8 May 2012].

Aviv , A. J., Gibson, K., Mossop, E., Blaze, M. and Smith, J.M., 2010. Smudge attacks on smartphone touch screens. *Workshop on offensive technologies*. Washington, USA, 11-13 August 2010. USENIX Association.

Biddle, R., Chiasson, S. and van Oorschot, P. C., 2011. Graphical passwords: learning from the first twelve years. *ACM Computing Surveys*, 44(4).

Canalys, 2010. *Majority of smart phones now have touch screens | Canalys*. [online] Available at: <http://www.canalys.com/newsroom/majority-smart-phones-now-have-touch-screens> [Accessed 27 September 2012].

Canalys, 2012. *Stellar growth sees China take 27% of global smart phone shipments, powered by domestic vendors | Canalys*. [online] Available at: <http://www.canalys.com/newsroom/stellar-growth-sees-china-take-27-global-smart-phone-shipments-powered-domestic-vendors> [Accessed 27 September 2012].

Corning Incorporated, 2012. *CORNING® GORILLA® GLASS | SAMSUNG*. [online] Available at: <http://www.corninggorillaglass.com/products-with-gorilla/samsung> [Accessed 27 September 2012].

Go Launcher Dev Team, 2012. *Go Launcher Ex*. [online] Available at: <https://play.google.com/store/apps/details?id=com.gau.go.launcherex> [Accessed 27 September 2012].

Google, 2012. *Android Developers.* [online] Available at: <http://developer.android.com/sdk/index.html> [Accessed 8 May 2012].

GSM Arena, 2012. *Motorola ATRIX Pictures.* [online] Available at: <http://www.gsmarena.com/showpic.php3?sImg=reviewsimg/mwc-11-motorola/atrix/gsmarena_005.jpg&idPhone=3709> [Accessed 8 May 2012].

Mowery, K., Meiklejohn, S. and Savage, S., 2011. Heat of the moment: characterizing the efficacy of thermal camera-based attacks. *Workshop on offensive technologies.* San Francisco, USA, 8-12 August 2011. USENIX Association.

Paivio, A., 2007. *Mind and its evolution : a dual coding theoretical approach,* Mahwah: Lawrence Erlbaum Associates.

Sae-Bae, N., Ahmed, K., Isbister, K. and Memon, N., 2012. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. *Proceedings of the 2012 ACM annual conference on human factors in computing systems.* Austin, USA, 5-10 May 2012. New York: ACM.

Sasse, M. A., Brostoff, S. and Weirich, D., 2001. Transforming the 'weakest link' - a human/computer interaction approach to usable and effective security. *BT Technology Journal,* 19**,** pp.122-131.

Shabtai, A., Fledel, Y., Kanonov, U., Elovici, Y., Dolev, S. and Glezer, C., 2010. Google android: a comprehensive security assessment. *IEEE Security and Privacy,* 8**,** pp.35-44.

Shannon, C. E., 1951. Prediction and entropy of printed English. *The Bell System Technical Journal*, 30, pp.50-64.

Thorpe, J. and van Oorschot, P. C., 2007. Human-seeded attacks and exploiting hot-spots in graphical passwords. *16th USENIX security symposium.* Boston, USA, 6-10 August 2007. USENIX Association.

Tryfonas, T., 2011. A*uthentication and access control.* [lecture notes] September 2011. Bristol: University of Bristol.

Whisper Systems, 2012. *Whisper Systems.* [online] Available at: <http://www.whispersys.com/screenlock.html> [Accessed 8 May 2012].

Zalewski, M., 2005. *Cracking safes with thermal imaging.* [online] Available at: <lcamtuf.coredump.cx/tsafe> [Accessed 8 May 2012].

Zhou, Y., 2011. *How many combinations does Android 9 point unlock have? – Quora*. [online] Available at: <http://www.quora.com/How-many-combinations-does-Android-9-point-unlock-have/answer/Yoyo-Zhou> [Accessed 27 September 2012].

9 Appendices

Monograms

Monogram	# participants who drew it
0	114
1	106
2	99
3	92
4	109
5	83
6	93
7	98
8	97

Bigrams

Bigram	# participants who drew it
01	46
03	29
04	24
05	2
07	1
08	3
10	12
12	47
13	13
14	17
15	9
16	3
17	7
21	19
23	4
24	23
25	26
26	7
27	1
30	13
31	4
32	4
34	14
35	6
36	26
37	12
38	5
40	9
41	19
42	5
43	17
45	21
46	17
47	13
48	22
50	2
51	6
52	23
53	5
54	18
56	1
57	7
58	23
60	1
61	4
62	2
63	19
64	11
65	5
67	34
70	1
71	2
72	2
73	4
74	5
75	8
76	28
78	35
81	5
82	1
83	1
84	13
85	24
87	22

Trigrams

Trigram	# participants who drew it
012	27
013	4
014	9
015	2
016	2
032	1
034	7
035	1
036	13
037	4
038	1
041	3
043	2
045	3
046	2
047	2
048	12
051	1
058	1
074	1
087	1
103	7
104	3
108	1
123	2
124	16
125	21
126	1
132	1
134	2
136	3
137	4
143	4
145	4
146	3
147	3
148	1
152	1
154	3
157	2
158	1
163	1
165	1
167	1
176	3
178	1
210	6
213	3
214	3
215	3
217	3
238	3
240	2
243	3
245	1
246	11
247	6
251	1
254	7
257	2
258	9
260	1
263	1
267	2
275	1
301	6
304	1
305	1
308	1
312	1
314	1
317	2
324	1
325	1
326	1
341	3
342	1
345	8
348	2
352	3
358	3
361	2
362	1
364	5
367	17
374	1
375	2
378	6
381	2
385	1
387	1
401	3
403	5
410	2
412	13
413	2
415	1
417	1
421	2
425	2
430	4
431	1
436	7
437	2
438	1
450	1

451	2
452	6
453	1
457	2
458	5
463	3
467	10
475	1
476	6
478	6
481	1
485	5
487	9
507	1
510	2
512	1
513	2
517	1
521	7
523	1
524	3
526	5
530	1
540	1
541	3
543	6
546	1
547	1
548	3
573	1
576	4
581	1
582	1

584	4
587	10
608	1
610	2
612	1
621	1
630	6
631	2
632	2
634	1
635	5
641	2
642	3
643	1
645	2
647	1
648	2
650	1
652	1
653	1
658	2
670	1
671	1
672	1
674	1
675	4
678	22
712	1
731	1
734	1
736	1
742	1
745	2

751	1
752	3
758	2
761	1
762	1
763	10
765	4
781	1
783	1
784	5
785	15
812	2
814	1
815	1
816	1
830	1
840	6
841	3
843	1
845	1
851	1
852	7
853	3
854	6
856	1
857	1
871	1
872	1
873	1
874	1
876	15

4-grams

4-gram	# participants who drew it
0123	1
0124	13
0125	13
0132	1
0134	1
0136	1
0137	1
0143	2
0145	2
0146	1
0147	3
0148	1
0157	2
0163	1
0167	1
0326	1
0341	2
0345	4
0348	1
0352	1
0361	2
0362	1
0364	2
0367	7
0374	1
0375	1
0378	1
0381	1
0412	2
0415	1
0431	1
0436	1
0452	1
0457	2
0467	2
0475	1
0478	1
0485	4
0487	7
0513	1
0581	1
0876	1
1032	1
1035	1
1036	2
1037	1
1047	1
1048	2
1238	1
1243	3
1245	1
1246	9
1247	3
1254	7
1257	1
1258	6
1267	1
1324	1
1345	1
1348	1
1364	1
1367	2
1375	1
1378	2
1436	3
1437	1
1452	1
1458	3
1467	2
1478	3
1540	1
1546	1
1548	1
1576	1
1587	1
1632	1
1650	1
1672	1
1763	1
1765	1
2103	3
2104	2
2108	1
2136	1
2137	1
2143	2
2145	1
2154	2
2176	2
2381	1
2385	1
2403	1
2430	1
2463	2
2467	6
2476	5
2478	1
2513	1
2543	4
2548	2
2573	1
2584	3
2587	4
2608	1
2631	1
2670	1

2678	1
2758	1
3012	1
3014	2
3015	1
3016	1
3041	1
3058	1
3087	1
3125	1
3176	1
3247	1
3258	1
3412	3
3451	1
3452	3
3458	2
3487	1
3521	1
3587	2
3612	1
3641	1
3642	1
3647	1
3648	2
3671	1
3674	1
3675	1
3678	11
3742	1
3752	1
3758	1
3784	1
3785	2
3814	1
3816	1
3857	1
3876	1

4012	2
4036	3
4037	2
4103	2
4123	1
4125	5
4137	2
4152	1
4178	1
4210	2
4257	1
4258	1
4301	1
4308	1
4317	1
4367	7
4378	1
4387	1
4510	1
4517	1
4521	3
4526	1
4530	1
4576	2
4587	3
4630	1
4635	2
4675	2
4678	7
4752	1
4761	1
4763	2
4765	1
4785	4
4812	1
4852	5
4871	1
4872	1

4876	6
5074	1
5103	1
5124	1
5210	1
5213	2
5217	3
5238	1
5240	1
5247	2
5260	1
5263	1
5267	1
5301	1
5403	1
5412	3
5430	2
5436	2
5437	1
5438	1
5463	1
5476	1
5736	1
5812	1
5840	2
5843	1
5873	1
5874	1
5876	6
6103	1
6104	1
6125	1
6215	1
6301	4
6312	1
6317	1
6325	1
6342	1

6352	2
6358	3
6412	2
6421	2
6425	1
6430	1
6450	1
6452	1
6478	1
6487	1
6507	1
6582	1
6745	1
6751	1
6752	1
6781	1
6783	1
6784	4
6785	9
7125	1
7314	1
7345	1
7364	1

7425	1
7451	1
7512	1
7521	1
7523	1
7526	1
7584	1
7610	1
7621	1
7630	2
7631	1
7632	1
7635	3
7652	1
7653	1
7658	2
7815	1
7830	1
7840	1
7841	3
7851	1
7852	1
7853	3

7854	4
7856	1
8124	1
8126	1
8165	1
8304	1
8401	3
8403	3
8410	1
8412	2
8436	1
8453	1
8510	1
8521	2
8524	1
8526	3
8541	3
8547	1
8576	1
8712	1
8762	1
8763	7
8765	2