FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Stop to Unlock - Improving the Security of Unlock Patterns on Mobile Devices

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Alexander Suchan, BSc
Registration Number 1028089

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Assistance: Univ.Lektorin Dr.techn. Katharina Krombholz

Vienna, 6th April, 2018

Alexander Suchan	Edgar Weippl

Erklärung zur Verfassung der Arbeit

Alexander Suchan, BSc
Kopalgasse 36/10 1110 Vienna

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 6. April 2018

Alexander Suchan

Acknowledgements

This master thesis would have never been written in this form without the help from the following people:

First and foremost I would like to thank all the participants from the lab and field study for their time, engagement and constructive feedback during their interviews and study participation. Their discussions helped to improve our *StopUnlock Patterns* and revealed future research questions as well as limitations for our solution.

I further want to thank the whole team of SBA-Research, especially my co-supervisor Katharina for her support and encouragement in this scientific research. Numerous discussions and suggestions from her helped me to improve and refine this master thesis.

Special thanks goes to my family and girlfriend for their ongoing moral and financial support during my study time at the TU-Vienna.

Abstract

In this master thesis, we propose a security improvement for traditional unlock patterns as found in modern off-the-shelf smartphones. With our *StopUnlock Patterns* we propose to add a *stop* component which lets users define nodes where they deliberately stop for a limited amount of time before swiping to the next node of their unlock pattern. We argue that this stop component lets users select higher entropy patterns with only a minimal impact on usability metrics (such as authentication time and error rate) in comparison to simply increasing the length of the pattern.

To evaluate *StopUnlock Patterns*, we conducted a lab study (n=40), a field study (n=14) and a security evaluation. Our findings suggest that *StopUnlock Patterns* (i) are easy to use, (ii) are memorable, and (iii) improve security.

Contents

CHAPTER 1

Introduction

This master thesis from the field of usable security proposes an improvement for the traditional (Android) unlock patterns. In this chapter we describe the motivation, problem definition and why we are in need for new authentication methods for mobile devices. We define the aim of the work as well as the contributions in this master thesis; this chapter is concluded with the structure of the work.

1.1 Motivation and Problem Definition

Harbach et al.'s [11] research showed that every smartphone user unlocks their phone up to 47 times a day and a slow authentication method results in a significant amount of interaction time (up to 9%) spend only with authenticating. Therefore, besides a high entropy, any new authentication method should offer the user fast authentication times, low error rates and a good memorability when unlocking the smartphone. Different attack scenarios like shoulder surfing or smudge attacks should also be considered while evaluating the security of authentication methods.

Six years after Bonneau et al.'s quest to replace passwords [5], knowledge-based authentication has still not given way to other means such as biometrics or implicit authentication. While most commercially available smartphones come with built-in fingerprint sensors or sophisticated face recognition features, PINs and unlock patterns are still used by a large fraction of users and are still the fallback methods if biometric authentication fails. Hence, low-entropy unlock patterns pose a major vulnerability to the potentially sensitive user data stored on the device.

1.2 Aim of the Work

As no other mechanism has managed to fully replace unlock patterns, we argue that iterative improvements of the well-established authentication methods are necessary to

improve the security of today's smartphone ecosystem. Similar to Krombholz et al.'s Force-PINs [15] and De Luca's XSide [7] we propose an improvement for unlock patterns as found in commercially available Android phones. Our *StopUnlock Patterns* do not require additional hardware and let users choose highly memorable unlock patterns with a practically invisible stop component.

Figure 1.1 shows an example of a *StopUnlock Pattern* with one stop node (marked in red). If nodes are expressed as numbers, starting at one on the top left corner, the entered pattern in this case will be: *7 - 4 - 1 - 5 (Stop) - 9 - 6 - 3*. This practically invisible timing component offers additional possibilities to create a pattern while keeping it simple and easy to learn.

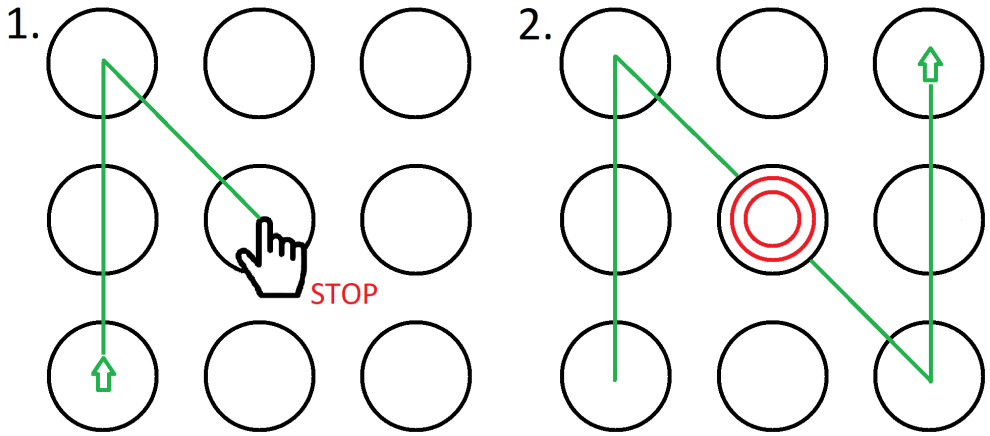


Figure 1.1: An exemplary unlock process for the presented *StopUnlock Pattern*, enhancing the traditional unlock pattern with a *stop* component marked with a red circle.

1.3 Methodology

To evaluate our *StopUnlock Patterns* we conducted two user studies each with an implemented prototype Android application and a security evaluation. The studies were accomplished with user selected patterns and included a memorability experiment. The detailed contributions of this master thesis are:

- We propose *StopUnlock Patterns* as an enhancement of traditional unlock patterns. Our approach enables users to stop at 1 to n nodes in their unlock pattern.
- We implemented a prototype application and performed an evaluation of the *StopUnlock Pattern* in a lab study with 40 participants and a field study with 14 participants.
- Our collected data provides evidence for security and usability metrics, such as practical entropy, memorability, authentication speed and error rate.

The results from our lab and field study suggest that *StopUnlock Patterns* improve security with only a minimal impact on usability. *StopUnlock Patterns* allow users to select higher entropy patterns with fast authentication times and a low error rate. Furthermore, our *StopUnlock Patterns* are highly memorable.

1.4 Structure of the work

The remainder of this master thesis is structured as follows:

In Chapter 2 we present the related work concerning unlocking behavior, attacks and improvements for authentication methods on mobile devices. In Chapter 3 we introduce besides the attacker model the *StopUnlock Pattern* design, our prototype application and our conducted pilot study. In Chapter 4 and 5 we present the design and results of the lab study and the field study. In Chapter 7 we discuss our work, its limitations and future improvements; we conclude this work in Chapter 8.

CHAPTER 2

Related Work

The user authentication process on mobile devices has been covered in several papers, including research about the theoretical and practical entropy, different attacking scenarios and the general unlocking behavior of the users. Existing and new authentication solutions were analyzed in respect to different attack scenarios, usability and security. In this chapter we present selected related work that is relevant to this master thesis. We divided this chapter into different sections for unlocking behavior, attacks and introduced improvements for authentication methods on mobile devices.

2.1 Unlocking Behavior

Research about unlocking behavior helps to gain insights about the unlocking process of users in the real world. With these studies we could recognize improvements and pitfalls of existing unlocking solutions like PIN codes, patterns or the simple slide-to-unlock.

Harbach et al. [11] conducted a comprehensive one-month field study with 52 participants and an online survey with 260 participants to gain insights into real-world unlocking behavior. They especially answered the question how much overhead unlocking and authentication adds to the daily smartphone usage. Additionally, Harbach et al. [11] investigated (i) why some participants refrained from using any unlocking method at all, and (ii) how they cope with security risks like shoulder surfing. Interestingly their participants estimated the risk of a shoulder surfing attack as low, more precisely in only 11 out of 3410 sampled situations shoulder surfing was perceived as a relevant risk. They found that their participants spent on average 2.9% of their smartphone interaction time with authenticating, in the worst case they even spent 9% of the interaction time authenticating. Every participant unlocked their phone on average 47.8 times, this translates to 3.0 unlocks per hour. The average authentication time for unlock patterns was 3.0 seconds which was faster than for numeric PINs (4.7 seconds) and only a little bit slower than without any unlocking method (2.67 seconds). This results suggests that any

security improvement for unlock patterns should keep the additional usability overhead as low as possible.

Zezschwitz et al. [20] performed a 21 days real world field study to compare the usability and memorability of Android like patterns with numeric PINs on mobile devices. Their quantitative results suggested that PINs outperform the pattern lock to a certain extent in respect to authentication speed and error rates. However the qualitative results showed that users accept this small drawback because patterns were rated better in terms of ease-of-use, feedback, efficiency and memorability. In comparison with our research where the participants could select their own pattern, they used predefined patterns in their field study. Zezschwitz et al. [20] additionally conducted a memorability experiment where according to the participant statements PIN users needed significantly less time to memorize their given credentials than pattern users. But according to the spontaneous memorability test after 14 days both groups (pattern and PINs) could remember their credentials equally.

Harbach et al. [10] presented a detailed analysis of different smartphone unlocking mechanisms through a month-long field study (n = 134) where they logged events on instrumented smartphones. They were able to show that lock screen mechanisms like slide-to-unlock, PINs or patterns provide the users distinct tradeoffs between usability and security. Harbach et al. [10] additional found that PIN users take longer to unlock but commit fewer errors than pattern users, also pattern users unlock their smartphone more frequently and are very prone to errors. But overall PIN and pattern users spent the same amount of time unlocking their devices on average. Interestingly, pattern performance seemed unaffected by enabling the stealth mode, where the visited cells are not highlighted while unlocking the smartphone. In their study they identified areas where smartphone unlocking mechanisms could be improved in respect to usability while also maintaining security. For example reducing the error rate of unlocking patterns would allow users to save up to a fifth of their unlocking time and simultaneously make the patterns more attractive for new or inexperienced users. They also found some potential in finding new unlocking solutions which are quicker than existing solutions, this would be interesting for users currently using slide-to-unlock. Finally they proposed a PIN lock screen which *teaches* the user one or two additional digits over time as well as the possibility to leave the phone unlocked while being in the car or riding a bike after the user successfully authenticated once.

2.2 Attacks

Attacks on well established authentication methods have been studied by different researchers in the last years. In this section we present selected solutions which are relevant to our own research.

For example from Aviv et al. [3] who researched *smudge attacks* on smartphones with Android unlock patterns. *Smudges* are oily residues on the touch screen surface which could give away the used authentication pattern. They first investigated the conditions in

which the smudges are easily extracted, this includes the lightning and camera orientation. Different settings that sometimes may interfere with the identification process were emulated to show that patterns could be recognized in these situations. The results were very promising as they were able to detect partially 92% and fully 68% of the patterns in their simulated scenarios. Even in their worst-case scenarios where the conditions were less than ideal they could fully extract 14% of the patterns. In their scenarios they simulated real usage of the smartphone, this includes for example making a phone call after authenticating and putting the mobile phone back in the pocket after usage. This research showed that *smudge attacks* are a possible attack vector and that unlock patterns need strengthening in this regard.

Abdelrahman et al. [1] investigated thermal attacks and showed that heat traces left on the screen can be used to successfully guess a secret if the attack was performed immediately after a successful authentication session. They investigated the possibility of these attacks with the help of thermal cameras for PIN and pattern authentication methods on mobile devices. In their attack scenario the user only enters the pattern or PIN one time and then leaves the mobile phone idle on the table and moves away without further interacting with the phone. When using a pattern Abdelrahman et al. [1] found that overlapping patterns decrease the success rate for thermal attacks from 100% to only 16.67%, while PINs still remain vulnerable to thermal attack with a success rate of greater than 72%, even with duplicated digits. This kind of attack has to be performed shortly after entering the credentials because the success rate decreases significantly when performed 30 seconds after authenticating. Abdelrahman et al. [1] stated that unlike smudge attacks, thermal attacks could recover information about the order of entry for PINs and patterns. These attacks were also shown to be very tolerant to the angle and distance of the thermal camera, which makes the attack less obvious for the victim. To resist thermal attacks they recommended amongst other things to include an overlap movement possibility in patterns and simple physical measures as placing the hand on the display after usage. They where aware that in real world usage it is most likely that the victim interacts with the mobile phone after authenticating and creating further traces which decreases the success rate of an attack.

Song et al. [16] evaluated the strength of real-world patterns and proposed a strength meter to help users find a stronger pattern. They collected and analyzed patterns with an application where users can encrypt their Dropbox files, a portion of the participants were provided with the meter support when creating their pattern. The meter provides the users immediate feedback on the security level (weak, medium and strong) of their selected pattern they are about to use. They found that about 10% of the patterns that were generated without the meter support could be compromised through 16 guessing attempts. Patterns that where generated with the meter support could be guessed with up to 48 attempts which is a significant improvement in terms of security. The partial guessing entropy (with $\alpha = 0.10$) for patterns generated with the meter support had 8.96 bits and patterns without meter support had only 7.38 bits. Besides the statistical analysis of the pattern strengths Song et al. [16] performed a shoulder surfing experiment where

71.29% of the patterns were successfully guessed by an attacker. Here again patterns generated with the meter support categorized as *strong* are harder to compromise through shoulder surfing attacks than patterns categorized as *medium* or *weak*. In conclusion Song et al. [16] recommended to implement a strength meter where users are in need of creating a pattern to secure their mobile phone.

Zezschwitz et al. [19] performed an evaluation on shoulder surfing resistance with Android unlock patterns. With their conducted online study (n = 298) they found that line visibility and pattern length are the most important parameters in terms of observation resistance. The number of overlaps, number of intersections and knight moves also influences the observability of unlock patterns. A knight move describes a connection of two cells that are not direct neighbors. With the presented linear regression model they predicted the observability of any given pattern. This regression model can help to proactive measure the security of unlock patterns, similar to the presented strength meter from Song et al. [16]. The study was conducted with patterns generated by an algorithm with distinct difficulties in respect to length, overlaps, line visibility, knight moves and intersections. Zezschwitz' et al. [19] results suggest that 51.7% of all tested patterns were successfully attacked with only one observation. They argue that, even if this means only half of the patterns were completely observed, participants were able to partly recognize most of the patterns and therefore concluded that Android patterns are easy to attack.

Eiband et al. [9] likewise performed a shoulder surfing survey (n = 174) in which they investigated real world user stories from a user and observer standpoint. Their results showed that shoulder surfing occurs mainly in a non-malicious opportunistic way, which evokes negative feelings for both parties and resulting in different coping strategies. The observed data had a wide range from personal information about interests or hobbies to login data and intimate details from third parties. The analysis of the user stories revealed no indication of observations out of malicious intent and or with the help of technical equipment, but was most common for strangers in social situations. However, a shoulder surfing attack often went unnoticed, in only 7% of the incidents users were aware of the attack reported by the observer. Eiband et al. [9] also proposed design implications in regards to privacy protection for improving the user experience on mobile devices in public. For example, privacy protection should not have to be initiated by the user, because they are mostly not aware of the attack.

Uellenbeck et al. [17] analyzed the guessability of unlock patterns. They performed a large-scale study with actual user selected patterns and showed that users are biased in their pattern choice and therefore only use a small fraction of the theoretical password space. Uellenbeck et al. [17] documented a high bias in the selection process e.g. upper left corner and three-point long straight lines are typical selection strategies. Their results suggest that the security of patterns is less than for three digit randomly-assigned PINs, the calculated estimated partial guessing entropy (with $\alpha = 0.20$) is 9.10 bits. Based on the study results they suggest some small but still effective changes in the grid layout to make patterns more secure. Uellenbeck et al. [17] showed that some of the changes

improve the security more than doubling the space of the used patterns. These results support our idea of iterative improvements of existing authentication methods to make unlock patterns more secure without the need of increasing the grid size.

Finally, Krombholz et al. [14] presented a microbiological attack based on bacterial growth on smartphone touchscreens. However, they were unable to show that the attack can be conducted successfully.

2.2.1 Summary of Attack Scenarios

In the following Table 2.1 we present selected attacking scenarios with the respective defensive strategies according to our cited papers. As described by Song et al. [16] and Zezschwitz et al. [19] who proposed a strength meter and a linear regression model for unlock patterns, meter support could improve security against a wide range of attacks. These meters can help the users to select higher-entropy patterns which are more secure against different kinds of attacks. They propose for example overlaps which makes smudge attacks more difficult.

Attack Scenario	**Defense Strategies**
Shoulder Surfing Attack	remove line visibility, increase pattern length
Smudge Attack	meter support
Pattern Guessing	changes in grid layout, increase entropy
Thermal Attack	possible overlap movement, physical measures
Petri Dish Attack	no real word usage for this attack

Table 2.1: Attack scenarios against unlock patterns and their defensive strategies.

2.3 Improvements

Currently there are different suggestions for improving the authentication process for existing unlock solutions on mobile devices. In this section we present some selected solutions that are relevant to our work.

Krombholz et al. [15] developed a new type of PIN with the use of Apples' *3D-Touch*. They integrate pressure-sensitive touch screens interactions into knowledge-based authentication PINs. These so-called *force-PINs* let users select higher entropy PINs with an (practically) invisible pressure sensitive component. *Force-PINs* have not only a higher entropy, but are also harder to observe for a shoulder surfer. Krombholz et al. [15] calculated the practical entropy gain for the binary force component as 3.41 bits. This is an entropy gain of 23% compared to four-digit PINs. To evaluate *force-PINs* they conducted a within-subjects design lab study (n = 50) where they compared them against standard four-digit and six-digit PINs in terms of usability and security. Their results suggest that *force-PINs* are not significantly slower than six-digits PINs, but still significantly slower than four-digit PINs. The error rate is rather low despite most participants never having

experienced pressure sensitive touchscreen interaction. Additionally, they performed a field study where they demonstrated that *force-PINs* authentication speed improves over a long period of time. This research showed us that small enhancements in already known well-established authentication methods allow the users so select higher entropy PINs while keeping the impact on usability metrics low.

Zezschwitz et al. [18] presented *SwiPIN*, an approach that combines traditional PINs with simple touch gestures like up or down. *SwiPIN* assigns for every unlock process each number a touch gesture and the user has to enter the gestures for the respective numbers in his PIN. They implemented distinct versions where the user has to enter the gestures on different locations of the screen for example on the bottom, in a random predefined area or free on the screen. They showed that *SwiPINs* were significantly more secure against shoulder surfing attacks, while being perceived as easy-to-use as PINs. To evaluate *SwiPINs* they present two user studies in which they evaluate the different versions and compared their methods against traditional PIN authentication. The results of these studies showed that *SwiPINs* offer a fast authentication time (3.7 seconds) and low error rate (3.1%) to serve as a real alternative for PINs in risky situations. In a shoulder surfing experiment for the different versions they showed that between 35.6% and 59.5% of the patterns were successfully attacked. Because they are very similar to standard PINs, switching between *SwiPINs* and PINs feels very natural and makes them easy to learn. In comparison with *force-PINs*, *SwiPINs* don't need any special hardware, they work with every off-the-shelf smartphone. Therefore, Zezschwitz et al. [18] argues that *SwiPINs* have the potential to be widely accepted as an alternative authentication mechanism in risky situations.

Another method to improve the input of standard PINs was researched by De Luca et al. [8] who presented *ColorPIN*, an authentication mechanism that uses indirect input to provide security-enhanced PIN entry. *ColorPIN* is a combination of a standard PIN with a color (black, red or white). For example a four-digit *ColorPIN* could look like: 1 (black) - 2 (red) - 3 (white) 4 - (black). To input the credentials, the user has to press the respective colors for the digits in his *ColorPIN*. With this method they could achieve a higher security but still have an one-to-one relationship between the length of the PIN and the required number of key presses. The conducted user study showed that *ColorPIN* was significantly stronger than standard PINs while enabling good authentication speed. The results showed that the mean authentication time of user generated *ColorPINs* are 13.33 seconds, which were significantly longer than traditional PINs with 1.23 seconds. *ColorPINs* offer a much higher theoretical password space and were in the conducted studies more resistant against observation attacks. Only two out of 48 authentication sessions could be successfully identified.

De Luca et al. [7] also presented *XSide*, an authentication mechanism which uses the front and the back of a smartphone to enter patterns. *XSide* lets the user draw simple shapes or gestures, i.e. is a system that can be used eyes-free and provides increased protection against smudge or shoulder surfing attacks. Because it can be used eyes-free, users can switch sides during the input to further improve the risk of shoulder surfing.

They performed a user study (n = 32) to show the effects of switching sides during authentication on usability and security of the system. Their results suggest that *XSide* increases security while the authentication speed stays relatively fast (under 4 seconds). *XSide* uses the standard smartphone touchscreen as well as a touch sensitive area on the back of the smartphone. This means the mobile phone needs special hardware to utilize the full functionality of this system. To authenticate the user can enter for example a pattern consisting of a combination of strokes (i.e. up and down) on the front and then another pattern on the back. The theoretical password space of *XSide* is 52^3 and the authentication speed lies between 2.1 and 2.5 seconds. Out of 768 authentication sessions they had a total of 140 erroneous sessions which included different error categories. According to De Luca et al. [7] this is the first authentication system where the user can spontaneously select a usability/security tradeoff during authentication.

Summarized we found many proposals for improving the authentication process on mobile devices, most of the presented solutions were evaluated with similar usability metrics compared to our research. Finally, most proposals are improvements of existing authentication methods, which makes them easy to use and learn because the users are already familiar with some parts of the unlock process.

CHAPTER 3

Concept and Prototype

The proposed *StopUnlock Pattern* adds a timing dimension to the traditional Android unlock pattern. This approach does not require additional hard- or software. In order to offer the user a simple new unlocking method, we tried to make this new version as uncomplicated as possible by keeping all the known rules from the traditional (Android) patterns. In this chapter we describe the attacker model, design and prototype of the *StopUnlock Pattern* as well as the conducted pilot study.

3.1 Attacker Model

We assume that the attacker has physical access to the phone. As shown by Song et al. [16], Android users mostly select simple and easy-to-predict unlock patterns with low entropy. Our *StopUnlock Patterns* offer a much higher theoretical entropy and pattern space by design and are therefore more difficult to guess. Because of the seemingly invisible stop component we expect that the risk of successful shoulder surfing and smudge attacks will be equal to or slightly higher than in traditional patterns. In this master thesis we focus on the practical entropy gain and the impact of higher entropy *StopUnlock Patterns* on usability.

3.2 Design

The *StopUnlock Pattern* was designed to improve the traditional unlock pattern provided by the standard Android environment. Our design offers an additional invisible component to create higher-entropy and harder-to-predict patterns. As most Android users are already familiar with similar authentication methods, the learning process is quite easy. Our approach is easy to deploy and works with every off-the-shelf Android smartphone (there is no need for external hard- or software as e.g. for similar concepts like Force-PINs [15]).

In this new version the user can stop at each node on the grid for a short time or run through it quickly when entering the unlock pattern. This stop component offers additional possibilities to create higher entropy patterns while keeping them simple and memorable. Figure 1.1 shows an example of a *StopUnlock Patterns* with one stop node. In this case, the user selected a "N" shaped pattern where they stopped at the center node for a short time and entered the rest of the pattern uninterrupted. For better memorability and in order to train muscle memory (cf. [13, 12]), the user receives a vibration feedback when the application recognizes a *stopping* point. Apart from that, the rules for user-selected patterns remain the same, as described in Uellenbeck et al. [17]. For *StopUnlock Patterns* we also use the standard 3x3 grid, the four rules for standard Android unlock patterns are as follows:

1. At least four connected nodes,
2. Already connected nodes cannot be connected again,
3. Only straight lines are allowed and
4. One cannot jump over points not visited before.

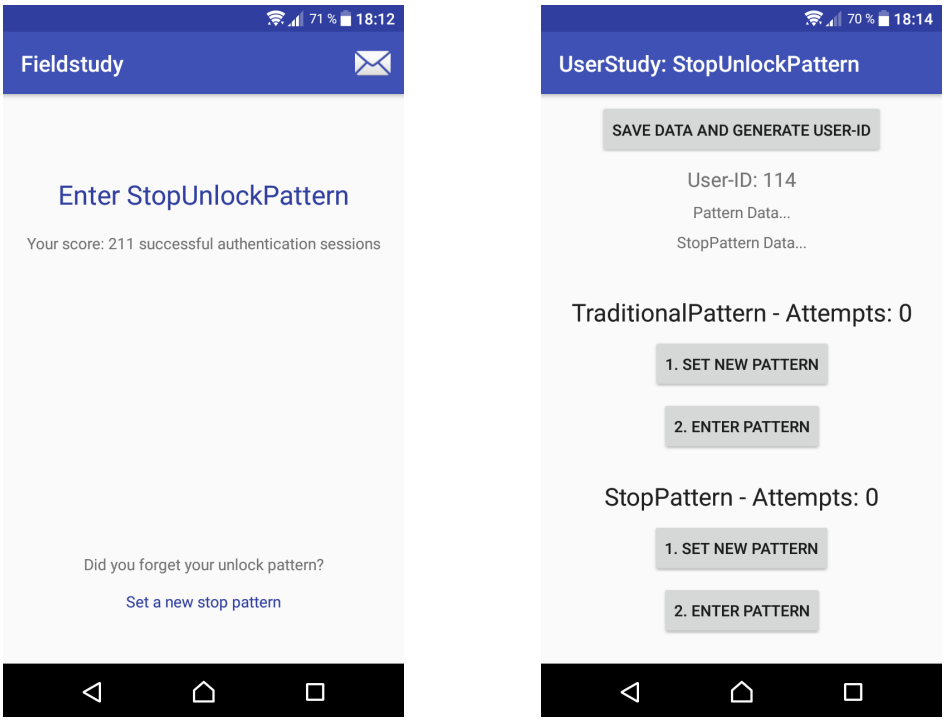
StopUnlock Patterns offer a larger pattern space by design. For every existing pattern, the stop component expands the space by 2^k, whereby k is the length of the pattern. Considering the smallest pattern with four visited nodes, the number of possible combinations grows by the factor 2^4=16.

3.3 StopUnlockPattern Prototype

We developed a prototype application on Android which simulates a login screen with *StopUnlock Patterns* and offers different methods to measure the usability metrics for our evaluation. This includes measurements of authentication time, error rate and failed authentication sessions. Additionally we collected the user-chosen *StopUnlock Patterns* for later entropy calculations. The application let users select a new pattern and presents an unlock screen similar to the one of common Android smartphones and simulates an unlock process during which the user has up to three attempts to unlock the phone with the correct pattern. Our goal was to make the unlock process and design similar to those of e.g. banking applications which require an authentication before usage. For comparison with the traditional unlocking method, we implemented the simulation for standard Android patterns as well.

Additionally to the lab study application we developed a slightly different application for our field study. This second application offers notifications to remind the study participants to enter their patterns three times a day in order to achieve a minimum number of login attempts throughout the day. Again we let the participants select a *StopUnlock Pattern* and offer the possibility to change the pattern anytime throughout the field study.

In Figure 3.1 we present the main screens of our lab- and field study applications. In the main screen for our field study 3.1a we added two buttons to either change the saved pattern and enter the user selected pattern. We present the user a score which should motivate and inform the user about the already completed successful authentication sessions. With the email function users were able to send us observations, questions and finally their collected data (i.e. the basic and critical errors, authentication time and the selected pattern). In the main screen for the lab study 3.1b we were able to



(a) Field study application main screen. (b) Lab study application main screen.

Figure 3.1: The different main screens for our two developed applications.

generate a new user id, select and enter patterns for either the traditional or *StopUnlock* authentication version. For the study supervisor we also displayed the collected data which will be saved, this includes the generated unique user id, to avoid errors in the collection of the data. The generated user id was used to connect the collected data to the answers in the later conducted questionnaire.

In Figure 3.2 we present the login screen for both applications during a pattern input. This simulated login screen closes either after successful authentication or after three unsuccessful authentication attempts.

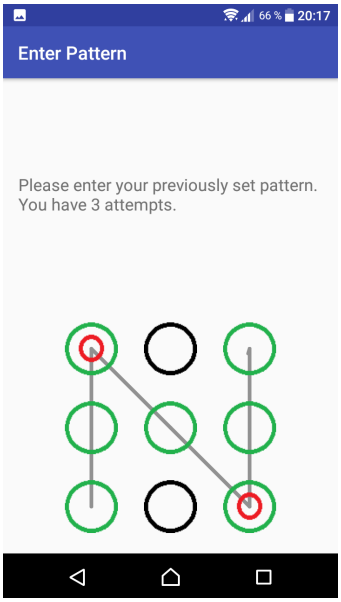


Figure 3.2: Simulated login screen for the lab study and field study application.

3.4 Pilot Study

We conducted a small pilot study with five participants to determine the optimal stop time. The goal was to find a stop time that only minimally compromises usability. The target group were people who are already familiar with the traditional Android authentication pattern. Before each session we explained the participants the goal of the study, and the users had some time to get acquainted with the new unlocking system. Before the study was carried out, we selected three different patterns with distinct difficulties. In Figure 3.3 we present all our selected patterns. In order to complete the study, every participant has to enter all three patterns with their different stopping times correctly. For each pattern and each stopping time we measured the failed attempts until the user entered the pattern successfully. Additionally we saved the mean authentication time for the correct and failed attempts.

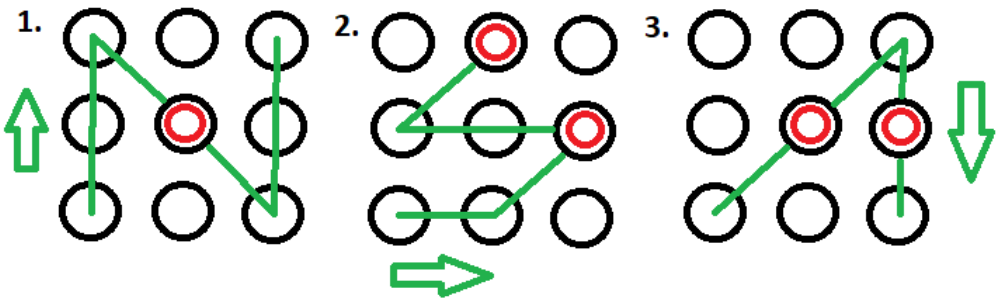
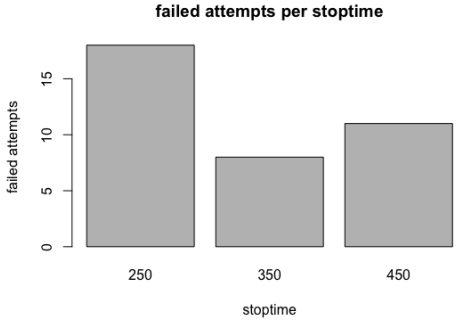


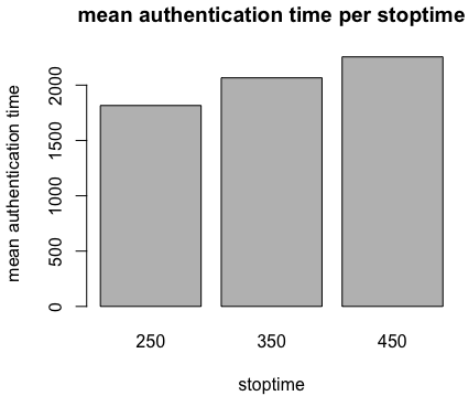
Figure 3.3: Selected patterns with distinct difficulties for the pilot study.

In our first tests the estimated best stopping time was between 200ms and 400ms, therefore we let the users conduct the study with 450ms, 350ms and 250ms. Almost all participants told us that they are most comfortable with a stopping time of 350ms. Reasons for that were (i) the increased error rate at the edges of the pattern with slower stopping times, and (ii) the overall increased authentication time with slower stopping times.

This was further confirmed via our collected data: out of all tested stopping times, 350ms had the fewest failed attempts. In Figure 3.4a we present the failed attempts per selected stopping time. For all tested stopping times, the mean authentication times were very close together. As expected, the authentication time increases in accordance with increasing the stopping time and was for all three tested times around 2000ms. In Figure 3.4b we present the mean authentication time per selected stopping time.



(a) Failed authentication attempts per tested stopping time.



(b) Mean authentication time per tested stopping time.

Figure 3.4: Pilot study plots; failed attempts and mean authentication time per tested stopping time. (n = 5)

CHAPTER 4

Lab Study

We conducted a lab study with our previously developed application and did a comparison with the traditional Android unlock patterns. Important research questions concern the comparison of (i) authentication time, (ii) error rate, (iii) memorability, and (iv) the entropy between the traditional and the new authentication method. Similar studies have been conducted [15, 18, 7, 20] with these kind of metrics. Additionally, we used the lab study to collect user-chosen *StopUnlock Patterns* for our security evaluation including entropy calculations. In this chapter we present the methodology and results from our lab study.

4.1 Design and Procedure

Every participant was exposed to two conditions in an random order to minimize learning effects. The conditions for every participant were as follows: C1 represents the traditional unlock pattern with the same rule set as in the standard Android environment, and C2 represents the advanced method with an additional stopping possibility.

- (C1) traditional pattern
- (C2) stop unlocking pattern

The participants (with academic and non-academic backgrounds) were recruited at the TU-Vienna and via personal contacts or social networks. We recruited 40 participants for our lab study. Table 4.1 shows the demographics of our study participants, most of them frequent smartphone users (Android or iPhone). The users are mostly familiar with PINs and traditional unlock patterns. We refrained from studying memorability in the course of the lab study as the setting is not suitable to provide ecologically valid results.

We started each session by explaining the goal of the lab study, i.e. that we want to compare different authentication methods for smartphones. Our application generated a unique user-id for every participant. We tried not to give the participants ideas about the security and usability of our new method in order to keep the results unbiased. We then introduced the two conditions (C1 and C2) and gave the users some time to familiarize themselves with the two chosen authentication methods. Thanks to this short training session, we reduced the knowledge difference between the traditional pattern and our new *StopUnlock Pattern*. For each of the conditions the participant had to define a new pattern or *StopUnlock Pattern*; we instructed the participants to select a pattern that they could remember and was, in their opinion, secure. Finally we simulated three authentication sessions during which the user had to correctly enter the previously selected pattern or *StopUnlock Pattern*. Three failed attempts in one authentication session meant that the whole session had failed.

The metrics we used for our usability evaluation were (i) authentication speed and (ii) error rate as defined by De Luca et al. [7]. For the *error rate* they distinguish between basic and critical errors. A *basic error* is an overall successful authentication session in which it took the user one or two tries to enter the pattern correctly. A *critical error* on the other hand is a completely failed authentication session, i.e. the user doesn't enter the correct pattern at all (they reach the allowed three failed attempts, hence the complete authentication session fails). We designed our study like that because many authentication systems (e.g. in ATMs) allow a maximum three attempts. We measured the *authentication speed* from the first to the last touch of an authentication session. In our user study the participants initiated the authentication session by pressing a button; the session ended either when entering the correct pattern or after three failed attempts.

In addition to the data collected through the Android application we asked the participants a series of questions about the tested authentication system. Apart from some demographic data about the participant (age and gender) we did not ask for any personal information. In order to link the technical data to the questionnaire, we assigned each participant an id number. Further questions were in relation to the used mobile phone and preferred authentication methods; to be able to rank and compare our *StopUnlock Pattern* with already existing methods, we asked the participants to sort unlocking methods by their speed and security. The questionnaire can be found in the appendix A.

4.2 Results

Our sample includes 40 participants who under two conditions had to complete three successful authentication sessions. Hence the quantitative results are based on 40 * 2 * 3 = 240 authentication sessions.

Table 4.1 shows the demographics of our participants, including their preferred authentication methods. Most of our participants use 4-digit PINs and fingerprint sensors; unlock patterns were used by just nine participants. Overall, we reached a wide range of people divided nearly evenly between male and female; the ages of our participants were between

Demographic	Number	Percent
Gender		
Male	21	52.5%
Female	18	45%
No Information	1	2.5%
Age		
Min.	20	
Max.	57	
Median	27	
Mean	29.73	
Used Smartphone		
Android	23	57.5%
iPhone	16	40%
Other	0	0%
None	1	2.5%
Used Authentication Method		
4-digit PINs	25	62.5%
Password (character and digit)	3	7.5%
Unlock Pattern	9	22.5%
Fingerprint Sensor	19	47.5%
Face Unlock	0	0%
None	1	2.5%

Table 4.1: Participant demographics from the lab study. (n = 40)

20 and 57. It was important for us to assess *StopUnlock Patterns* with people of different ages, genders and background knowledge.

4.2.1 Authentication Speed

To measure the authentication speed we used the metrics defined by De Luca et al., see section 4.1. For our statistical analysis we only considered successful authentication sessions, i.e. all sessions with a maximum of two failed authentication attempts. In order to reduce the impact of measurement errors, we removed two outliers from the *StopUnlock Pattern* samples. We took out two authentication sessions that lasted longer than 14 and 21 seconds, since they occurred because participants were distracted from the task at hand. We did not remove any measurement errors from the traditional unlock pattern samples.

For the first interpretation of the distribution of our sample dataset we used boxplots. In Figure 4.1 we present the generated boxplot for the *StopUnlock Pattern* sample. As explained before, we removed two outliners shown in this boxplot which took exceedingly longer than comparable authentication sessions; this measurement errors arose because the user was not focused on the task at hand. In Figure 4.2 we present the generated boxplot

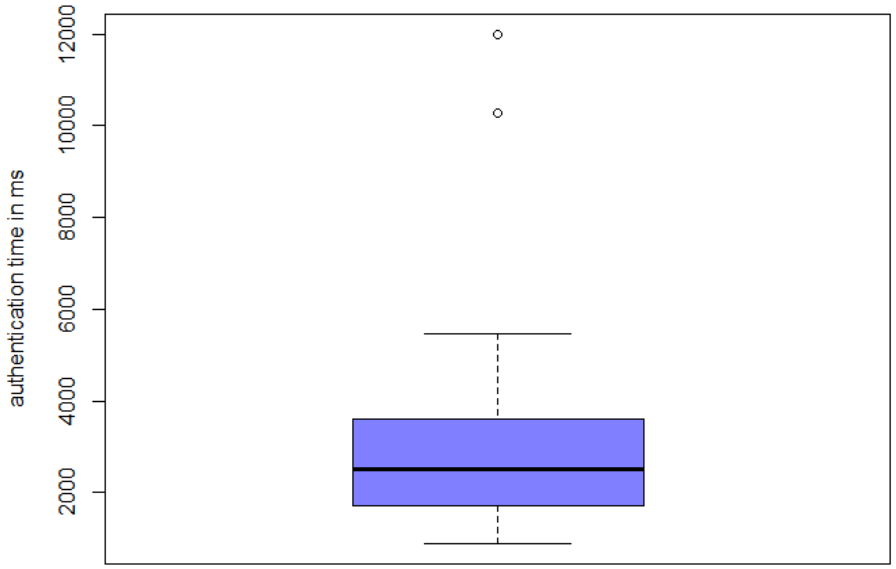


Figure 4.1: Lab study boxplot over the authentication time in milliseconds for the *StopUnlock Pattern* sample. (n = 40)

for the traditional sample. We did not remove the shown outliner in the traditional pattern sample, because we could not found any evidence that this was an measurement error.

Overall our participants did not select duplicated patterns in the *StopUnlock Pattern* sample (i.e. 40 different patterns); in the traditional pattern sample the pattern *1-2-3-5-7-8-9* occurred four times and the pattern *1-4-7-8-9* occurred two times (i.e. 36 different patterns).

Both the *StopUnlock Pattern* samples and the traditional pattern samples are normally distributed and have a homogeneity of the variances. Further, by study design the two conditions are independent from one another. In Table 4.2 we show the mean authentication speed and error occurrences for both conditions. An independent sample t-test suggests significant effects regarding the difference in authentication time between the *StopUnlock Pattern* and the traditional unlock pattern ($t(40) = 5.4117$, $p < 0.05$)).

Table 4.2 further includes comparisons with existing research. *Force PINs* introduced by Krombholz et al. [15] which used the same metrics for their research as well as authentication speed comparison from Harbach et al. [10] for traditional unlock patterns.

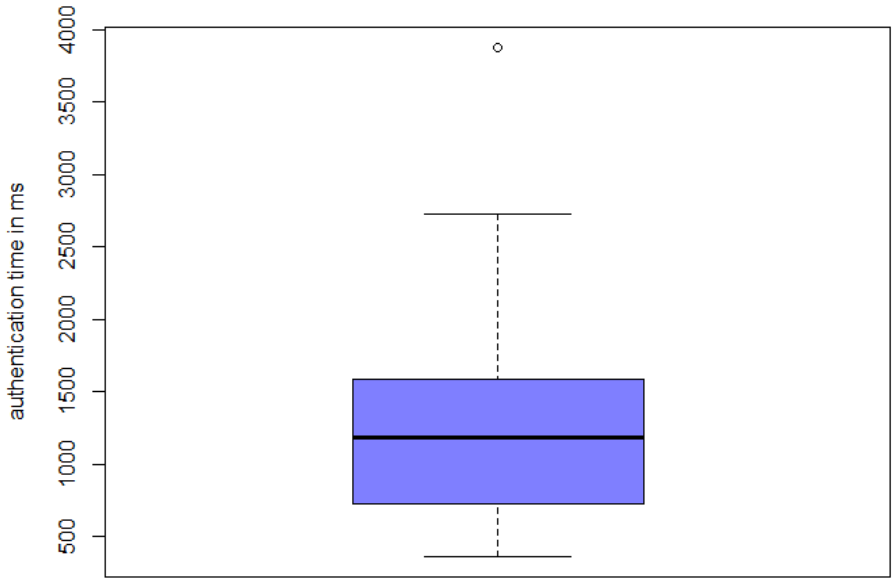


Figure 4.2: Lab study boxplot over the authentication time in milliseconds for the traditional pattern sample. (n = 40)

Harbach et al. [10] did a comprehensive user study regarding unlocking behavior for different authentication methods. They found that the average duration of a successful unlocking session for traditional Android patterns is 0.91 seconds. Our proposed solution adds 350ms for every stop node of a selected pattern. To calculate the additional time the stopping points add to the unlocking process, we counted the stops of every selected pattern in our study and multiplied the time for the stops. Through this approach we added an average 0.76 seconds solely via the stopping points. Therefore, the lowest possible authentication time for *StopUnlock Patterns* is approx. 1.67 seconds (consisting of 0.91 seconds calculated by Harbach et al. [10] + 0.76 added via the stopping points).

4.2.2 Error Rate

We measured the error rate with the same metrics as defined by De Luca et al. and described in section 4.1. As shown in Table 4.2, out of all 240 authentication sessions only one session with an *StopUnlock Pattern* failed completely (i.e. three failed authentication attempts). Furthermore 9 (3.75%) failed attempts (basic errors) occurred with *StopUnlock Patterns* and 5 (2.08%) failed attempts occurred with the traditional unlock patterns.

Authentication Speed	**Mean**	**SD**
StopUnlockPattern	2.79	1.80
Traditional Unlock Pattern	1.32	0.76
Harbach et al. Traditional Pattern	0.91	0.62
Krombholz et al. Force PIN	3.66	1.96
Krombholz et al. 4-digit PIN	2.34	1.21
Error Occurrences	**Basic**	**Critical**
StopUnlockPattern	9	1
Traditional Unlock Pattern	5	0
Krombholz et al. Force PIN	36	4
Krombholz et al. 4-digit PIN	21	0

Table 4.2: Mean authentication time in seconds and error occurrences for basic and critical errors, in comparison to research results for existing authentication methods.

4.2.3 Perceived Usability and Security

To find out how they participants perceived this new authentication method, we asked them a series of questions regarding our *StopUnlock Pattern*. One central question was: *Would you use the StopUnlockPattern on your own smartphone to unlock it?*. It was interesting to observe that some iPhone users declared that they don't want to use this new pattern. The main reason for this was that they were really satisfied with the options offered by iOS (fingerprint sensor and 4-digit PINs) and are therefore not in need of a new authentication method. As a consequence of that observation, we divided the given answers to this question with respect to Android users and other smartphone users. As expected, Android users expressed more often that they would use our authentication method to unlock their own smartphone. More precisely, 70% of the participants with Android devices stated that they would use *StopUnlock Patterns*, 9% don't want to use it, and 22% said maybe. In comparison, only 56.25% of participants with iPhones or other smartphones stated that they would use *StopUnlock Patterns*, 18.75% don't want to use it, and 25% said maybe.

To determine how the participants perceived the *StopUnlock Patterns* compared to other authentication methods, we asked them to rank some methods, including our *StopUnlock Pattern*, according to authentication speed and security. In Table 4.3 we summarize in descending order the perceived security and authentication speed as reported by our participants. Interestingly, the participants agreed that the fingerprint sensor is the securest and fastest authentication method. During the completion of the questionnaire, some users stated that they feel that the fingerprint sensor is only the fastest if the recognition works on the first try; often the users need two or more tries to unlock the phone, and in that cases the PIN was perceived as faster. Regarding security, most of the users acknowledged that our *StopUnlock Pattern* is be more secure than traditional unlock patterns and 4-digit PINs. When considering authentication speed, participants perceived the *StopUnlock Pattern* as slower than traditional unlock patterns and the

fingerprint sensor. Finally, most users are unsure about the unlock speed regarding *StopUnlock Patterns* and 4-digit PINs.

	Participant Votes			
Method	First	Second	Third	Fourth
Security				
Fingerprint Sensor	**26**	6	4	4
StopUnlockPattern	7	**21**	12	0
4-digit PIN	7	11	**13**	9
Unlock Pattern	0	2	11	**27**
Auth. Speed				
Fingerprint Sensor	**30**	5	2	3
Unlock Pattern	4	**21**	13	2
StopUnlockPattern	2	5	**14**	**19**
4-digit PIN	4	9	**11**	**16**

Table 4.3: User-based ordering of authentication methods according to their security and authentication speed. (n = 40)

All this votes are of course highly subjective; the questions were intentionally asked in a very open way to give the participants room to interpret the security and speed based on their own observations and experiences. Nevertheless, we think this question gives valuable insights in the reception of the *StopUnlock Pattern* compared to existing solutions.

4.2.4 Informal Participant Statements

In this section we present informal participant statements gathered via the open-ended questions asked after the lab study. During the user study we got the impression that many participants liked this new authentication method or are at least open to it. They mentioned the increased possibilities to create a pattern while keeping it easy to remember and use. Ten participants especially commended the hepatic (vibration) feedback when the system recognizes a stop at one node. In regard to security, many participants acknowledged that this should make the *StopUnlock Pattern* more secure that the traditional pattern. 20 participants liked the increased security and thought that this system is not easily observable by other persons. In the following, we present selected participant comments. They mentioned the somehow rhythmic component and the additional possibility to make patterns more secure.

- *This works better than I thought at first.* (ID-91)
- *This new pattern includes a somehow rhythmic component, this maybe helps some people to recognize their patterns better.* (ID-94)

- *In the real world you don't have to use the stop function, but the user has the possibility to increase the security of his phone with this expansion.* (ID-103)

We not only asked what people liked about the pattern, but also wanted feedback on how we might improve this authentication method. Most frequently, participants mentioned the increased time (nine people) and concentration (five people) they needed to enter the pattern. In the beginning, some people needed a few tries to get used to the *StopUnlock Patterns*, some found this a bit frustrating. Some participants mentioned that they need to stop exactly in the circle with no possibility to go back after leaving the node.

In the following, we again present selected participant comments.

- *People who have already problems with the usage of their smartphone (or are overwhelmed with some of the functionality), would have problems with this unlocking method.* (ID-103)
- *This method could be difficult to enter if I want to unlock the phone with one hand in the car with navigation setup.* (ID-108)
- *Difficult to stop exactly in the circle. After I left the circle there is no going back to stop again, pattern must be entered again.* (ID-91)

To summarize, most of the participants were interested in keeping their smartphone secure, often they asked about what authentication method they should use and why. Although they see some difficulties with *StopUnlock Patterns* they understand the need for a more secure unlocking method and are willing to test them.

4.3 Statistical Evaluation

In this section we describe the statistical R scripts which were used in order to evaluate the lab study. In Algorithm 4.1 we present some example data we collected with the first Android application described in Chapter 3. Three failed attempts during the login simulations results in a critical error and the respective authentication time is set to zero.

- (V1) Participant-ID.
- (V2-V4) Failed authentication attempts.
- (V5-V7) Authentication times in milliseconds.
- (V8) User selected (stop) pattern.

Algorithm 4.1: Some example data for the lab study scripts.

```
V1  V2  V3  V4  V5    V6    V7    V8
73  0   0   0   2826  2788  2923  147L5L9L63
74  0   3   1   2573  0     13783 5236L947
75  0   0   0   1821  1736  1584  123L69L87
77  0   0   0   3177  3121  3204  123L5L7L89
78  0   0   1   2291  2220  6500  741L59L63
...
```

This data was used to run the following R scripts, presented in Algorithm 4.2. This script was used to generally analyze the lab study data for mean authentication times, standard deviation and the variances. Another important part in this script was a paired samples t-test to show if there is a significant difference between the two examined conditions (i.e. the traditional patterns and our *StopUnlock Patterns*). A paired samples t-test was used because we had a *within-group-design*, meaning that there is one group of participants doing both conditions.

TradiData holds the data for the traditional condition and *StopData* holds the data for the stop pattern condition. In Part 1 we only read the data from the CSV files and remove with the help of boxplots some outliners (measurement errors). In Part 2 we extract the general information about the collected data. This includes the mean authentication time for both conditions. Before we could perform a t-test we need to check if the data is normally distributed and the variance is homogenous. Both tests are in Part 3 of the script realized with a *sharpio* and a *barlett* test. In Part 4 we conduct the paired samples t-test with both datasets. Finally, in Part 6 we calculate the average time which was added solely by the stopping component, this was later used to calculate the minimal authentication time.

Algorithm 4.2: The R script for the lab study.

```
# paired samples t-test
install.packages("stringr")
library(stringr)

# 1. read in and calculate means for each participant
TradiData <- original.TradiData[5]
TradiData[2] <- original.TradiData[6]
TradiData[3] <- original.TradiData[7]

StopData <- original.StopData[5]
StopData[2] <- original.StopData[6]
StopData[3] <- original.StopData[7]

StopData[4] <- rowMeans(StopData, na.rm = FALSE, dims = 1)
names(StopData)[4] <- "mean"
```

```
16
17 TradiData[4] <- rowMeans(TradiData, na.rm = FALSE, dims = 1)
18 names(TradiData)[4] <- "mean"
19
20 # boxplot to check for outliners
21 boxplot(TradiData[,4],
22         col = rgb(0,0,1,0.5),
23         ylab = "authentication time in ms")
24 boxplot(StopData[,4],
25         col = rgb(0,0,1,0.5),
26         ylab = "authentication time in ms")
27
28 # remove 2 outliners in stop sample
29 StopData[2,4] <- StopData[2,1]
30 StopData[36,4] <- (StopData[36,2] + StopData[36,3]) / 2
31
32 # 2. general information about the data
33 mean_stop <- mean(StopData$mean)
34 mean_traditional <- mean(TradiData$mean)
35
36 sd_stop <- sd(StopData$mean)
37 sd_traditional <- sd(TradiData$mean)
38
39 var_stop <- var(StopData$mean)
40 var_traditional <- var(TradiData$mean)
41
42 # 3. pre tests:
43 # normal distributed:
44 # p < 0.05 then the sample deviates from normality
45 shapiro.test(StopData$mean)
46 shapiro.test(TradiData$mean)
47
48 # homogeneity of variance:
49 # if p < 0.05 then the variance is homogeneous
50 homogenious_variance <- rbind(StopData, TradiData)
51 homogenious_variance[5] <- as.factor(c(rep(1, 40), rep(2, 40)))
52 names(homogenious_variance)[5] <- "group"
53 bartlett.test(mean ~ group, data = homogenious_variance)
54
55 # 4. t-test
56 t.test(StopData$mean, TradiData$mean, paired=TRUE)
57
58 # 6. calculate the average time added by the stopping points
```

```
59 countL <- str_count(original.StopData$V8, "L") * 350
60 meanTimeAddedByStopPattern <- mean(countL)
```

The result of the script 4.2 indicates that the data is normally distributed, the variances are homogenous between the two conditions and the t-test suggests a significant difference in authentication time between the *StopUnlock Pattern* and the traditional unlock pattern ($t(40) = 5.4117$, $p < 0.05$)).

CHAPTER 5

Field Study

In order to see how *StopUnlock Patterns* perform in a real-world setting, we conducted a field study with a second Android application that was installed on the private smartphones of the study participants. We measured authentication time and error rate of *StopUnlock Pattern* over a longer period of time. We furthermore studied how *StopUnlock Patterns* perform with respect to memorability. In this chapter, we present the methodology and results including the debriefing questions for the conducted field study.

5.1 Design and Procedure

We asked all participants from the lab study who use an Android smartphone to participate in our field study and were able to recruit 14 participants.

During the installation of our application we explained our participants that the goal of our study was to analyze the performance of our *StopUnlock Pattern* in many different situations over a longer period of time. Therefore, we asked the participants to enter their *StopUnlock Patterns* at least three to five times a day. We did not mention that we were going to study memorability in the course of the field study. To remind the participants to use their *StopUnlock Patterns* multiple times throughout the day, we issued notifications three times a day (morning, midday, and afternoon).

We encouraged the participants to enter the pattern as often as possible. After the installation the users had to choose a new pattern, and we mentioned again that this pattern should be secure, easy to remember and include at least one stop node. The field study was conducted over a period of two weeks. In this time, the participants were expected to accomplish between 100 and 200 successful authentication sessions. In case the participant couldn't remember the chosen pattern or wanted to change it to a more secure one, we implemented the possibility to switch the pattern anytime. We

also included an email function through which the participants could send us comments or inform us about problems. It was important to make sure that the application did not override the standard Android authentication method (i.e. the lock screen), since we didn't want to change the preferred security settings of the user, which was a great concern of some participants at the beginning of the field study. These participants would not have taken part if the application had overridden their selected unlocking method. Nevertheless, we think that our study process simulated rather well the usage of an authentication screen throughout an average day.

For the calculation of the authentication time and error rate we used the same metrics and procedures as in our lab study, see section 4.1.

After the two weeks were up, we asked the participants some informal questions about their experiences during the field study. We wanted to know if the user encountered problems in specific situations, for example on public transport or when walking fast. Another question was how often and why the user decided to change his pattern. We were also interested if the stopping time was selected correctly when the participants had some time to get familiar with *StopUnlock Patterns*. All asked questions in detail can be found in appendix A.

Additionally we conducted a memorability experiment where we asked participants approx. two months after completion of the field-study if they could remember their selected pattern. The participants were not informed beforehand about this memorability experiment.

5.2 Results

Overall, the 14 participants (P1-P14) completed 3,223 authentication sessions. Every participant achieved on average 230.21 authentication sessions, i.e. about 16 sessions per day. In sum, the participants produced 405 basic errors and 30 critical errors. The detailed results can be found in Table 5.1. Furthermore, we included the error rate for basic and critical errors (i.e., up to two failed authentication sessions, three basic errors), the pattern length, the stops from every selected pattern and the completed study days, i.e. the days on which a participant entered the pattern at least one time. Participants P5, P11 and P13 changed their pattern at some point during the field study.

As we had requested, most of the participants distributed their authentication sessions as evenly as possible over the entire two weeks. In Figure 5.1 we depict the overall successful authentication sessions per day, i.e. between 150 and 250 sessions. Some of our participants exceeded the requested study time as well as the authentication sessions, whereas others skipped a few days. In the following, we list the participants who did not enter the pattern on more than one day in a row. Subject P3 and subject P12 skipped six days at the end of the study time due to failed notifications. Subject P5 skipped four days and subject P7 three days. Nevertheless, these participants fulfilled the requested number of sessions.

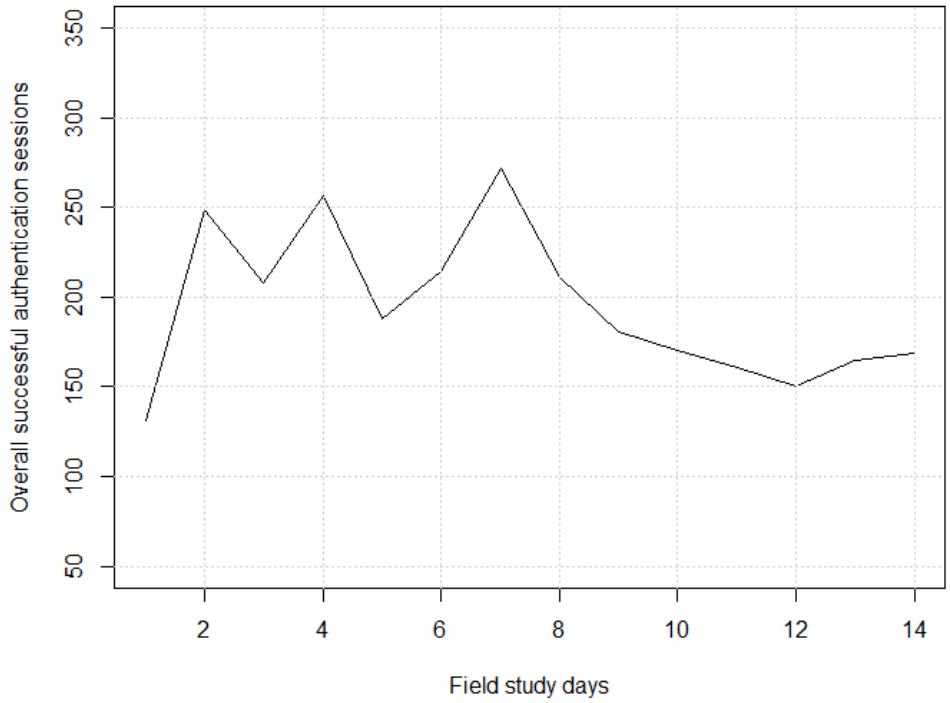


Figure 5.1: Overall successful authentication sessions per day.

We removed 14 authentication sessions due to measurement errors. These sessions took significantly longer than comparable ones, therefore we expect that the participants were not focused on the given task. In order to detect these sessions, we calculated the authentication time without the predefined time for every stop and removed outlying authentication sessions, i.e. those that are longer than 10 seconds. In Figure 5.2 we present the boxplot for the authentication time over all participants from the field study. In this Figure are many outliners due to one participant with significantly more stops than other participants and therefore has a much bigger authentication time than all other participants. As explained before we only removed measurement errors with an authentication time longer than 10 seconds without the stopping time.

The mean authentication speed over all successful authentication sessions was 2.41 seconds (median=2.06, SD=1.56), which is an improvement over the results from the lab study. The shortest authentication session was 0.696 seconds, the longest was 11.59 seconds.

To visualize the development of authentication time and error rate, we grouped the results in 16 bins (one bin per study day). We selected 16 bins because the average study duration was 15.5 days, and we believe that this is a good way to simulate a trend and

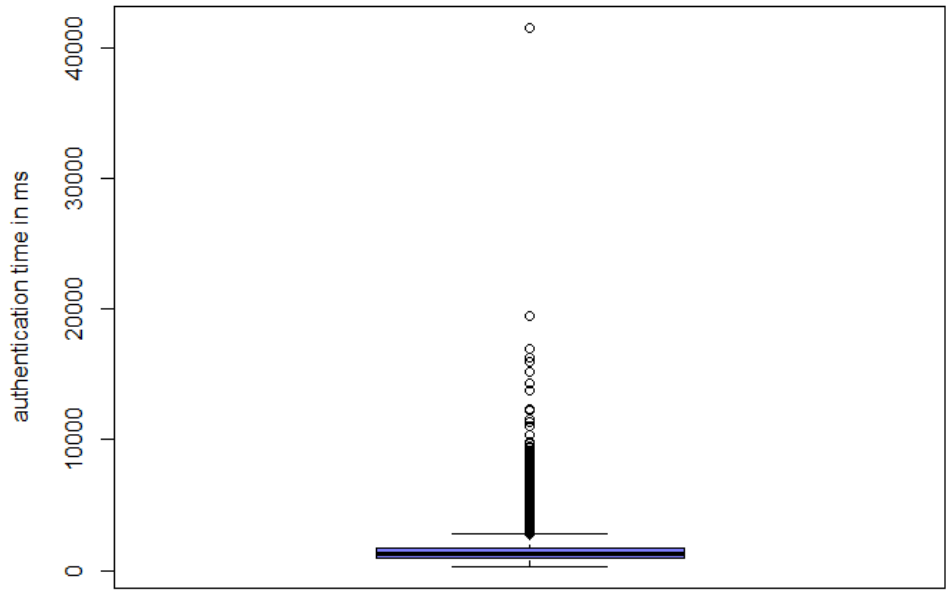


Figure 5.2: Field study boxplot over the authentication time in milliseconds for all participants. (n = 3,223)

make the results comparable.

Three participants changed their patterns halfway through the field study and increased the number of stops, which yields an increased overall authentication time and can therefore not be compared with that of other participants. Consequently, we split the results according to participants who changed the pattern (interrupted lines) and participants who did not change the pattern (solid line). In the first sample (participants who changed patterns) we considered every pattern change as a new start of the study (i.e. starting at day one). Despite the user being already familiar with the new unlocking system, we think that a newly selected pattern requires some learning time. We are confident that this solution utilizes both samples in a comparable and clean way without loosing the data from participants who changed their patterns.

In Figure 5.3 we present the development of the authentication time across the 16 bins, each representing one day from the field study period. Our results suggest that the mean authentication time decreases over time. In Figure 5.4 we present the error rate across the study duration. In the sample of participants who didn't change the pattern,

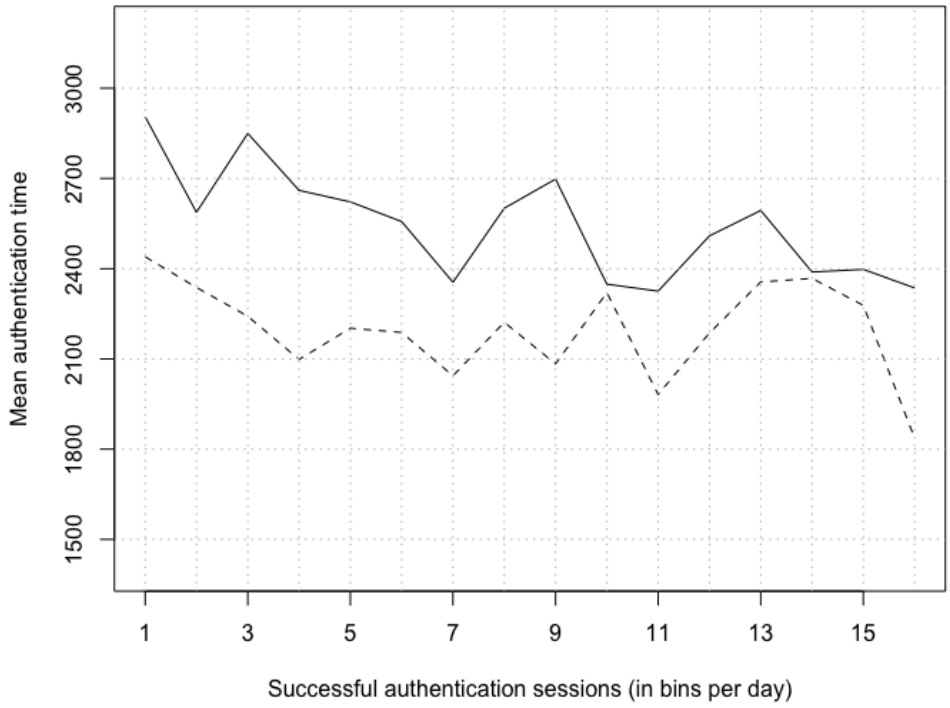


Figure 5.3: Authentication time development based on the successful authentication sessions across all participants. We combined the sessions in bins per day.

the error rate varies between 20% and 5%. In the sample of participants who changed the pattern the error rate varies between 11% and 4%. We didn't find any increase or decrease of the error rate during the field study duration.

5.2.1 Debriefing Questionnaire

Every participant who finished the field study (i.e. completed around 150 successful authentication sessions over a period of two weeks) took part in the debriefing questionnaire which consists of six questions. Most of the questions where open-ended to leave the participants room to express their opinion of this new authentication method. Because of the rather small number of participants in this study, we display the results in absolute numbers and not percentages.

We again asked the participants if they would use *StopUnlock Patterns* to secure their own smartphones after having used the method on a daily basis in a real-world setting. Seven users can imagine themselves to use this unlocking method, four told us they maybe want to use it, and three claimed that they would not like to use it.

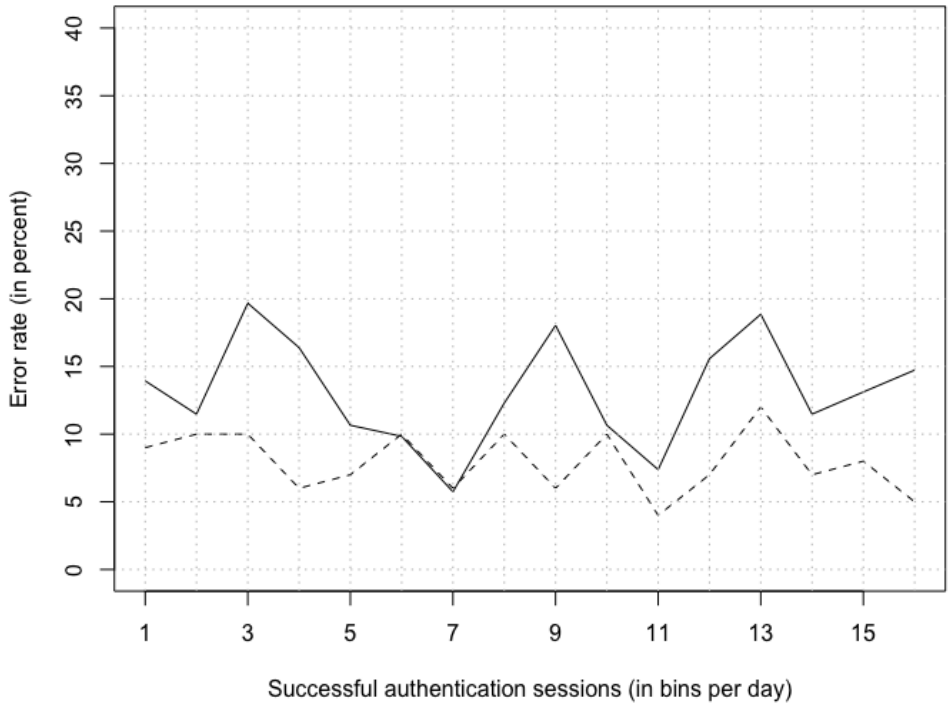


Figure 5.4: Error rate development based on the successful authentication sessions across all participants. We combined the sessions in bins per day.

The next question was: *Where you able to enter the pattern faster and with less errors after a few days, compared to the beginning of the study?*. Eight participants claimed that they perceived a decrease in authentication time; the remaining six did not feel any difference compared to the beginning of the study.

5.2.2 Informal Participant Statements

In this section, we present the open-ended questions from the debriefing questionnaire. Three participants changed their pattern during the field study, and we asked them why they decided to do so. Two participants stated that they wanted to try out a pattern with more than one stop node; one participant just wanted to try something different. One participant tried to change the pattern, but immediately changed it back without having it entered correctly. The latter did not provide reasons for this behavior.

We further asked which problems occurred during the authentication process in the real world (for example when using public transport). A few users stated that they sometimes had problems hitting the stopping point, i.e. they did not stop exactly on the point and

the stop was not recognized by the system. Six people told us that they experienced these problems more often while they were on public transport or busy with something else. In the following, we present selected participant comments.

- *Sometimes I had difficulties when entering the pattern with a wet display, gloves, or in jerky environments (fast walk, bus ride,...) in comparison to the traditional version. I think I make more mistakes in these situations.* (ID-4)
- *Like with traditional patterns, I often visited the wrong nodes. With the stopping feature I had no difficulties.* (ID-1)
- *After a few tries I had no problems.* (ID-11)

We furthermore asked our participants if they thought that the stopping time should be reduced, increased or remain as it is after the two-week trial. Most of the participants stated they where more or less satisfied with the selected stopping time. One participant requested a possibility to select an individual stopping time. More precisely, eight users thought the stopping time is well selected, five said the stopping time could be shorter, and one person wanted a longer stopping time.

The last question asked the participants to give general feedback; most participants repeated observations e.g. concerning vibration and visual feedback. In the following, we present two final comments from users.

- *In my opinion this pattern can be used just as the traditional one. If this new pattern design offers a better security for my smartphone than it's worth to getting used to it.* (ID-4)
- *Short time getting used to. I imagine this can be a good alternative.* (ID-3)

In summary, we found that most participants were generally satisfied with the everyday usage of *StopUnlock Patterns*; some stated that it required more concentration in public environments compared to traditional patterns.

5.3 Memorability

To assess the memorability of the user-chosen *StopUnlock Patterns*, we sent out emails approx. two months after completion of the field study and asked the participants whether they still remembered their patterns. In order to prevent the participants from writing their pattern down, we did not tell them that we would contact them once more in this regard.

13 out of 14 participants immediately replied to our request and sent us their selected patterns; nine participants correctly remembered the complete *StopUnlock Pattern*. Four participants remembered it only partially. One user could not recall the last two nodes,

but managed to draw the rest of the pattern, including all stopping points. Another user drew one pattern in a mirrored way and the other pattern correctly, but made a small mistake with the location of the stopping points. Finally, two participants were not able to draw the pattern from memory, but were able to locate the stopping nodes when shown the pattern.

Some participants who could not draw their selected pattern completely from their memory stated that they used for this field study a pattern which they normally wouldn't select. They told us that they wanted to try out a very complex pattern, in the real world they would have used a more memorable pattern.

Our results suggest that *StopUnlock Patterns* are very memorable, even after two months not using them. Even complex patterns with up to five stopping points were remembered.

5.4 Statistical Evaluation

In this section we describe again the statistical R scripts which were used to evaluate the field study. In Algorithm 5.1 we present some example data we collected with the second Android application as described in Chapter 3. Three failed attempts during the login simulations results in a critical error and the respective authentication time is zero.

- (V1) Number of the authentication session.
- (V2) Day of the study process.
- (V3) Date and time of the authentication session.
- (V4) Failed authentication attempts.
- (V5) Authentication times in milliseconds.
- (V6) User selected stop pattern.
- (V7) Participant-ID.

Algorithm 5.1: Some example data for the field study scripts.

```
1 V1 V2               V3 V4    V5             V6 V7
2  1  1 22.11.2017 17:27  1  9206 1L4L5L7L89632L P1
3  2  1 22.11.2017 21:13  1 10528 1L4L5L7L89632L P1
4  3  1 22.11.2017 21:13  0  4851 1L4L5L7L89632L P1
5  ...
6  1  1 17.11.2017 15:27  0  1524       1235L789 P2
7  2  1 17.11.2017 19:20  0  1347       1235L789 P2
8  3  1 17.11.2017 19:20  0  1314       1235L789 P2
9  ...
```

```
  1  1 26.11.2017 16:29  0  1557       12357L89 P3
  2  1 26.11.2017 16:29  0  1706       12357L89 P3
  3  1 26.11.2017 16:29  0  1637       12357L89 P3
  ...
```

This data was used to run the following R scripts, presented in Algorithm 5.2. This script was used to create plots for the mean authentication time and error rate per study day. The authentication sessions were divided into 16 bins, because the average study duration was 15.5 days.

In Part 1 we read in the data and select only the successful authentication sessions for our analysis. We remove the outliners with the help of a boxplot which have a authentication time over 10 seconds. Then we create some basic summary about the data, this includes mean authentication time and the standard deviation. To split the data into 16 bins we add in Part 2 a new column with IDs from 1 to 16 for each participant, based on these IDs we calculate the average authentication time per bin. More precisely in this new column every authentication session for every participant was divided into 16 bins. In this Part we also split the data into participants who changed the pattern during the field study and participants who used only one pattern. In Part 3 we draw the plot for the authentication time, based on the beforehand aggregated means per bin. Before we could draw in Part 3 the plot for the error rate, we calculate the error rate for each bin.

The created plots suggest that there is some kind of learning curve in respect to the authentication time. For the error rate plot we didn't find any learning effect based on our data.

Algorithm 5.2: The R script for the field study.

```
library(stringr)

# Part 1: read in data, without failed sessions
succSessions <- original.SplitUp[!(original.SplitUp$V4==3),]

# add auth. time without stopping time
countL <- str_count(succSessions$V6, "L") * 350
succSessions[,8] <- succSessions[,5] - countL

# remove outliners, with the help of an boxplot
boxplot(succSessions[,8],
        col = rgb(0,0,1,0.5),
        ylab = "authentication time in ms")
succSessions[(succSessions[,8]>10000),]
succSessions <- succSessions[!(succSessions$V8>10000),]

# summary of the data
summary(succSessions[,5])
```

```
sd(succSessions$V5)

# Part 2: add the 16 bins to the dataset
d_concat <- data.frame(a=NULL)
names <- unique(succSessions$V7)
for(i in names) {
  print(i)
  rows <- nrow(succSessions[succSessions$V7==i,])
  times <- round(rows/16)
  d <- data.frame(a = c(rep(1, times),rep(2, times),rep(3,
   ↪ times),rep(4, times),rep(5, times),rep(6, times),rep(7,
   ↪ times),rep(8, times),rep(9, times),rep(10, times),rep(11,
   ↪ times),rep(12, times),rep(13, times),rep(14, times),rep
   ↪ (15, times),rep(16, 100)))
  d <- head(d,rows)
  d_concat <- rbind(d_concat,d)
}
succSessions <- cbind(succSessions,d_concat)

# split the data up (one pattern, changed pattern)
samePattern <- succSessions[!(succSessions$V7=="P1_1"),]
samePattern <- samePattern[!(samePattern$V7=="P1_2"),]
samePattern <- samePattern[!(samePattern$V7=="P2_1"),]
samePattern <- samePattern[!(samePattern$V7=="P2_2"),]
samePattern <- samePattern[!(samePattern$V7=="P2_3"),]
samePattern <- samePattern[!(samePattern$V7=="P3_1"),]
samePattern <- samePattern[!(samePattern$V7=="P3_2"),]

changPattern <- succSessions[!(succSessions$V7=="P4"),]
changPattern <- changPattern[!(changPattern$V7=="P5"),]
changPattern <- changPattern[!(changPattern$V7=="P6"),]
changPattern <- changPattern[!(changPattern$V7=="P7"),]
changPattern <- changPattern[!(changPattern$V7=="P8"),]
changPattern <- changPattern[!(changPattern$V7=="P9"),]
changPattern <- changPattern[!(changPattern$V7=="P10"),]
changPattern <- changPattern[!(changPattern$V7=="P11"),]
changPattern <- changPattern[!(changPattern$V7=="P12"),]
changPattern <- changPattern[!(changPattern$V7=="P13"),]
changPattern <- changPattern[!(changPattern$V7=="P14"),]

# calculate the mean auth. time per bin
samePatternMean <- aggregate(samePattern$V5,
        by=list(repedition=samePattern$a), FUN=mean)
```

```
58 changPatternMean <- aggregate(changPattern$V5,
59         by=list(repedition=changPattern$a), FUN=mean)
60
61 # Part 3: draw the auth. time plot
62 plot(samePatternMean$repedition, samePatternMean$x,
63         type='l',
64         xlim=c(1, 16), ylim=c(1400, 3200),
65         xlab="Successful authentication sessions",
66         ylab="Mean authentication time",
67         xaxt="n", yaxt="n")
68 axis(side=1, at = seq(1,16,2), labels = seq(1,16,2))
69 axis(side=2, at=seq(1500,3200,300),labels = seq(1500,3200,300))
70 lines(changPatternMean$repedition, changPatternMean$x, lty = 2)
71 abline(h=seq(1500,3200,300), v=1:16, col="gray", lty=3)
72
73 library(plyr)
74 # calculate the error rate for each bin
75 samePatternError <- aggregate(samePattern$V4,
76         by=list(repedition=samePattern$a), FUN=sum)
77 changPatternError <- aggregate(changPattern$V4,
78         by=list(repedition=changPattern$a), FUN=sum)
79 samePatternError <- cbind(samePatternError, count(samePattern,
    ↪ "a"))
80 changPatternError <- cbind(changPatternError, count(
    ↪ changPattern, "a"))
81
82 samePatternError[,5] <- ((samePatternError$x/samePatternError$
    ↪ freq) * 100)
83 changPatternError[,5] <- ((changPatternError$x/
    ↪ changPatternError$freq) * 100)
84
85 # Part 4: draw error plot
86 plot(samePatternError$repedition, samePatternError$V5,
87         type='l',
88         xlim=c(1, 16), ylim=c(0, 40),
89         xlab="Successful authentication sessions",
90         ylab="Error rate (in percent)",
91         xaxt="n", yaxt="n")
92 axis(side = 1, at = seq(1,16,2), labels = seq(1,16,2))
93 axis(side = 2, at = seq(0,40,5), labels = seq(0,40,5))
94 lines(changPatternError$repedition, changPatternError$x, lty =
    ↪ 2)
95 abline(h=seq(0,40,5), v=1:16, col="gray", lty=3)
```

Subjects	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
Completed UnlockSessions	183	235	212	271	155	157	200	104	180	190	502	100	571	163
Basic Errors	47	25	23	25	43	25	30	15	15	5	48	14	38	52
Basic Error Rate (%)	25.7	10.6	10.8	9.2	27.7	15.9	15.0	14.4	8.3	2.6	9.6	14.0	6.7	31.9
Critical Errors	3	0	1	1	4	4	0	1	0	0	2	0	1	13
Critical Error Rate (%)	1.64	0.00	0.57	0.37	2.58	2.55	0.00	0.96	0.00	0.00	0.40	0.00	0.18	7.98
Pattern Length	14	8	8	9	6 8	5	9	8	7	10	6 12 6	12	7 10	11
Stops in Pattern	5	1	1	2	1 3	1	2	1	2	3	1 4 2	5	1 2	4
Completed Study Days	17	15	9	16	11	16	12	18	12	16	12	11	16	13

Table 5.1: Participant results in detail. (n = 14)

CHAPTER 6

Security Evaluation

Based on the collected data from our lab and field study, we evaluated the security of this new authentication method. This includes the evaluation in relation to the practical and theoretical entropy of the *StopUnlock Pattern* as well as the basic pattern space. In this section, we additionally want to discuss the impact of the pattern length to the mean authentication time and error rate.

6.1 Entropy

Password entropy measures the probability distribution of passwords over the entire search space. These measures are based on different mathematical models, for example the guessing entropy measures the average number of guesses that an optimal attack needs in order to find the correct password [17].

Obviously, the password space of *StopUnlock Patterns* is larger than for traditional patterns, because they offer a larger pattern space by design. There is no easy way to calculate the number of possible patterns using combinatorics, but one can simply enumerate all possible combinations and find that there are $389.112 \approx 2^{19}$ [17, 3] possible patterns. For every existing pattern of length k the stopping component expands these possibilities by 2^k, whereby k is the length of the pattern. Considering the shortest pattern with four visited nodes, the possibilities grow by the factor 2^4=16.

As explained by Cherapau et al. [6] zero-order entropy measures the entire search space of all possible secrets of a given length and the size of a given alphabet. Zero-order entropy assumes that each character is selected randomly and represents the effort an attacker needs to spend on guessing it. Zero-order entropy is measured in bits and calculated as $L * log_2(N)$, whereby L is the length of the secret and N the size of the character set. For *StopUnlock Patterns* we have a length between 4 and 9 connected nodes and a character set of 18 (9 times 2, for stop/no stop). Therefore, the lower bound for zero-order entropy

is 16.68 bits, and the upper bound is 37.53 bits. While traditional unlock patterns have a zero-order entropy between 12.68 bits and 28.52 bits, 4-digit PINs have an entropy of 13.28 bits [6] and force-PINs have an entropy of 17.28 bits [15].

In theory *StopUnlock Patterns* are a major improvement over traditional unlock patterns because of the extended pattern space and the higher zero-order entropy. However, this statement assumes that user-chosen patterns are equally distributed across the pattern space, which barely holds in practice. Hence, real-world patterns are distributed over a much smaller search space and are therefore easier to guess for an attacker. To provide a rough estimate, we calculated the partial guessing entropy gain of our stopping component based on the patterns (n = 58) from our field and lab study.

6.1.1 Partial Guessing Entropy

In contrast to the guessing entropy, the partial guessing entropy [4] (or α-guesswork) calculates the average number of guesses that an attacker needs to break a certain fraction of passwords. The partial guessing entropy was calculated as explained by Uellenbeck et al. [17].

Let

$$\mu_\alpha = \min \{j \mid \sum_{i=1}^{j} p_i \geq \alpha\} \tag{6.1}$$

be the minimal number of guesses to cover at least a α fraction of the patterns (i = 1 is the pattern with the highest probability). While $\lambda_\alpha = \sum_{i=1}^{\mu_\alpha} p_i$ represents the actual fraction covered, which is greater or equal to α. The partial guessing entropy is defined as follows:

$$G_\alpha(X) = (1 - \lambda_\alpha) \cdot \mu_\alpha + \sum_{i=1}^{\mu_\alpha} i \cdot p_i \tag{6.2}$$

In equation 6.2 the first term represents the values that were not guessed in the given fraction, and the second term represents those that were guessed [17]. We express entropy in *bits* to make this estimate comparable with other measurements. This can be achieved with equation 6.3.

$$\tilde{G}_\alpha(X) = \log\left(\frac{2 \cdot G_\alpha(X)}{\lambda_\alpha} - 1\right) + \log\frac{1}{2 - \lambda_\alpha} \tag{6.3}$$

Table 6.1 shows the occurrences of user-selected *StopUnlock Patterns* from the lab- and field study (n = 58). The table is sorted by the probability for every pattern, and the S represents the selected stopping point. Because there are 39 different stop pattern possibilities, we omitted some patterns which occurred only one time. We included the patterns from both studies, although participants were able to use the pattern from the lab study in the field study. We told the participants in both cases to select a pattern that they think is secure and easy to remember, since in the real world many users have different devices and some users will select the same pattern for all their devices.

Nevertheless, most participants selected different patterns. We think this approach represents real usage of patterns on different devices.

i	Stop Pattern	Number	Probability
1	- - - S - - -	4	0.069
2	- - S S S - -	4	0.069
3	- - S - S - -	3	0.052
4	S - - - -	3	0.052
5	S - - - S	3	0.052
6	- - - - S - -	2	0.034
7	- - - - S - S	2	0.034
8	- - S - -	2	0.034
9	- - S S S	2	0.034
10	S - - - - - S	2	0.034
11	S - S - S - S	2	0.034
12	S S S S - - - - S	2	0.034
13	- - - - - S	1	0.017
14	- - - - S - - -	1	0.017
15	- - - S	1	0.017
16	- - - S -	1	0.017
17	- - - S S	1	0.017
18	- - S -	1	0.017
19	- - S - - -	1	0.017
20	- - S - - - -	1	0.017
...	...	...	...

Table 6.1: Stop patterns as chosen by the participants in the lab- and field study. n = 58 user-selected patterns, whereby S represents the stopping point. For reasons of brevity, some patterns with less than 2 occurrences were omitted.

Based on the probabilities in Table 6.1, we calculated the previously discussed partial guessing entropy. The results of our calculations for an $\alpha = 0.50$ are as follows: $\mu_\alpha = 12$, $\lambda_\alpha = 0.532$, $G_\alpha(X) = 2.973$, $\tilde{G}_\alpha(X) = 1.3423$ bit. This means that the practically hidden stop component from our proposed *StopUnlock Pattern* offers an additional partial guessing entropy of 1.34 bit.

In comparison to our solution, Aviv et al. [2] presents in Table 3 different real-world entropies for traditional Android unlock patterns as well as PINs. For example, the self-reported real-world traditional 3x3 pattern offers an entropy of 9.94 bits (with $\alpha = 0.50$). Together with our estimated entropy gain of 1.34 bits, we are close to the entropy of traditional 4x4 patterns which is 11.61 bits [2].

6.2 Pattern Length

In this section, we discuss the impact of the pattern length on the mean authentication time and error rate. Our *StopUnlock Patterns* adds additional time to the unlocking process with every selected stopping node. For example, if the user selects a pattern with 3 stops, the stopping time alone is 1.05 seconds long (350ms per stop). In our lab- and field study we observed that the authentication time can become quite long when the pattern includes more stops, especially if the participant made an error during the authentication process. Consequently we wanted to know at which point it becomes unfeasible for the user to select more nodes with stopping points.

In Figure 6.1 we visualized the mean authentication time with the respective pattern length. In the pattern length we included all selected nodes with stopping points, for example the pattern: *1-2L-3L* equals a length of 5. The mean time is based on all 58 selected patterns from both the lab- and field study and the first three successful authentication attempts (n = 173). One can easily observe that for *StopUnlock Patterns* the authentication time increases with a longer pattern. Harbach et al. [10] found in

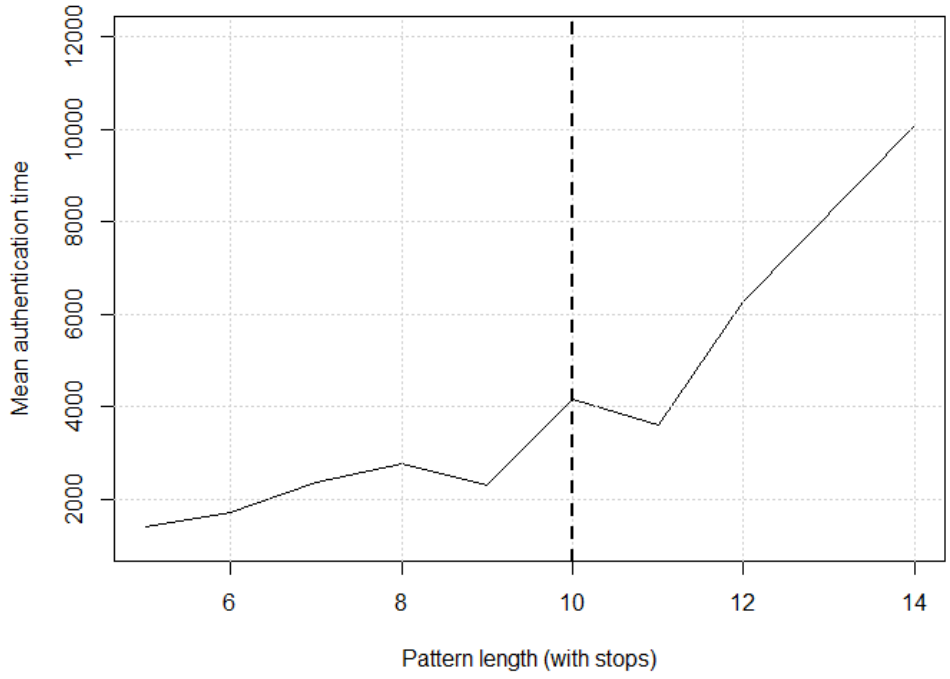


Figure 6.1: Authentication time development based on the length of the selected pattern (n = 173).

their comprehensive study that the average pattern length for traditional unlock patterns

is 5.9 nodes and the average authentication time is 0.91 seconds. They also found that every additional node increases the successful login time by on average 147ms.

We argue that *StopUnlock Patterns* with a pattern length over 10 inclusive stopping points are not feasible. In this case the mean authentication time reaches 4 seconds which is way over the average authentication time for traditional patterns. Besides authentication time, the pattern length will be longer than the mean pattern length of 8.43 inclusive stopping points. The average number of selected nodes for our *StopUnlock Patterns* is 6.2, which is a little bit more than for traditional patterns.

In Figure 6.2 we provide the same evaluation for the error rate. In regards to the selected pattern length we didn't measure any significantly higher error rate if the pattern is longer. The observed error rate varies between 0.0% and 5.0%, whereby the patterns with a length of 4 to 7 nodes have an error rate under 2.5%.

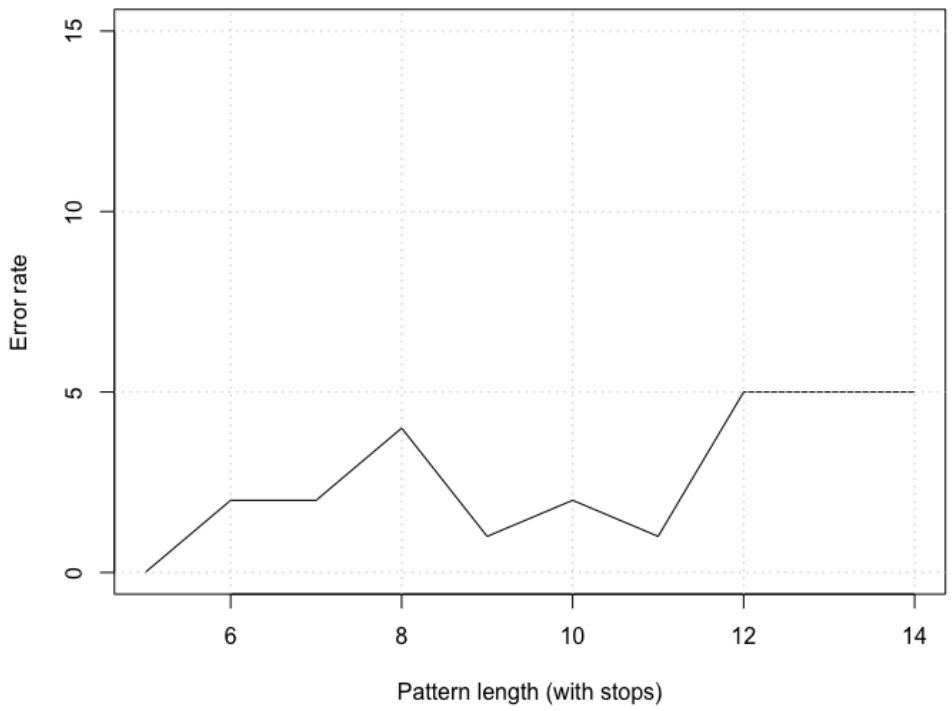


Figure 6.2: Error rate development based on the length of the selected pattern (n = 173).

Summarized, we found that the pattern length has an impact on the authentication time; therefore, this factor should be taken into consideration if we select the length restriction of *StopUnlock Patterns*.

6.3 Statistical Evaluation

In this section we describe again the statistical R scripts which were used for the security evaluation plots. In Algorithm 6.1 we created the plots for authentication time and error rate in comparison with the pattern length. The input data were the first three authentication sessions for every selected pattern in the field and lab study. Additionally we added to the dataset two columns for the length of the pattern (inclusive the stopping points) and the number of stops in the pattern. Before we could create the plot we aggregated the mean authentication time based on the respective pattern length.

Algorithm 6.1: The R script for the security evaluation.

```
1  library(stringr)
2  # calculate the means for every pattern length
3  allPatterns <- original.AllPatterns
4  allPatterns[,4] <- str_length(original.AllPatterns$V3)
5  allPatterns[,5] <-  allPatterns$V4 - str_count(allPatterns$V3,
       ↪ "L")
6
7  # calculate the auth. time for every pattern length
8  allPatternMean <- aggregate(allPatterns$V2, by=list(repedition=
       ↪ allPatterns$V4), FUN=mean)
9
10 # draw the plot for the auth. time
11 plot(allPatternMean$repedition, allPatternMean$x,
12         type='l',
13     xlab="Pattern␣length␣(with␣stops)",
14         ylab="Mean␣authentication␣time",
15     xlim=c(5, 14), ylim=c(1100, 12000))
16 grid()
17 abline(v=10, col="black", lwd=2, lty=2)
18
19 # calculate the error rate for every pattern length
20 allPatternError <- aggregate(allPatterns$V1, by=list(repedition
       ↪ =allPatterns$V4), FUN=sum)
21 allPatternError[,3] <- (allPatternError$x/3)*100
22
23 #draw the plot for the error rate
24 plot(allPatternError$repedition, allPatternError$x,
25         type='l',
26     xlab="Pattern␣length␣(with␣stops)",
27         ylab="Error␣rate",
28     xlim=c(5, 14), ylim=c(0, 15))
29 grid()
```

CHAPTER 7

Discussion

In this section, we discuss some observations and limitations for *StopUnlock Patterns* we discovered during both our studies. Some of the observations and ideas were brought up from participants during interviews and discussions.

StopUnlock Patterns are designed to be easily used and memorized by the user. This was achieved by using the same rule set as traditional unlock patterns and only add the invisible stop component to the unlocking process. During the lab- and field study no participant forgot their pattern, which gave us the impression that users didn't have major problems memorizing their patterns. Another interesting observation by one participant was that *StopUnlock Patterns* implement some kind of rhythmic component which helps to memorize the selected pattern.

StopUnlock Patterns provide the possibility to select 1 to n stopping points for every pattern. We observed in our study that some users selected long patterns with many stopping points. Our data showed that for these users the authentication time gets exceedingly long, especially if the user makes one or more errors during the authentication process. Unfortunately this is a shortcoming of *StopUnlock Patterns*, because at some point it becomes unfeasible to select more stopping points. Additionally we expect that if the user select a pattern with too many stops, it can get frustrating for them to unlock their phones. One solution for this problem could be to implement an upper bound for stopping points per pattern.

After the lab- and field study we learned that some participants added some stopping points to the unlock pattern of their private smartphones. We think that this is a good way to make already familiar patterns more secure without having to learn a complete new pattern or even unlocking system.

A limitation found during both the lab- and field study was that if the user does not hit the last stopping node correctly (i.e. moves slightly outside of the node), the stop would not be recognized when re-entering the last visited node. This should be improved in

further implementations, for example by recognizing the stopping point even if the user re-enters the last visited node.

Although we didn't measure shoulder surfing or smudge attacks, we expect that these are equally easy or slightly harder to perform in comparison to traditional unlock patterns. To further improve the resistance against shoulder surfing attacks, there is the possibility to remove the visual aids in *StopUnlock Patterns*, i.e. remove the red markers for stopping points, lines between nodes or, like in some custom Android modifications, all visual elements. The implications of such improvements on security might be studied in further research.

Another modification one could consider is to remove the restriction that a pattern must connect at least 4 nodes. Because *StopUnlock Patterns* offer a bigger pattern space by design this requirement might be interesting to adjust.

Generally speaking, *StopUnlock Patterns* might make the login process more difficult in otherwise challenging situations, for example if the display is not working properly or even slightly broken. Similar situations would be entering the pattern while driving a the car or on a hands-free device. We are aware that such circumstances make every unlocking process harder, but it might be interesting what impact they have on *StopUnlock Patterns*.

CHAPTER 8

Conclusions

In this master thesis we presented *StopUnlock Patterns*, a concept which adds a practically hidden stopping component to traditional Android unlock patterns. This timing concept enables the user to select higher entropy patterns with an only minimal impact on usability. We evaluated *StopUnlock Patterns* through a lab study and observed real-world usage in a field study as well as evaluated the theoretical entropy gain and the impact of pattern length to the authentication time. Fast authentication times (2.97 seconds) and a low error rate (3.75% for basic errors) combined with a good memorability indicate that *StopUnlock Patterns* could be a potential alternative for traditional patterns.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] Yomna Abdelrahman, Mohamed Khamis, Stefan Schneegass, and Florian Alt. Stay cool! understanding thermal attacks on mobile-based user authentication. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3751–3763. ACM, 2017.

[2] Adam J. Aviv, Devon Budzitowski, and Ravi Kuber. Is bigger better? comparing user-generated passwords on 3x3 vs. 4x4 grid sizes for android's pattern unlock. In *Proceedings of the 31st Annual Computer Security Applications Conference*, ACSAC 2015, pages 301–310, New York, NY, USA, 2015. ACM.

[3] Adam J Aviv, Katherine L Gibson, Evan Mossop, Matt Blaze, and Jonathan M Smith. Smudge attacks on smartphone touch screens. *Woot*, 10:1–7, 2010.

[4] J. Bonneau. The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In *2012 IEEE Symposium on Security and Privacy*, pages 538–552, May 2012.

[5] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 553–567. IEEE, 2012.

[6] Ivan Cherapau, Ildar Muslukhov, Nalin Asanka, and Konstantin Beznosov. On the impact of touch id on iphone passcodes. In *SOUPS*, pages 257–276, 2015.

[7] Alexander De Luca, Marian Harbach, Emanuel von Zezschwitz, Max-Emanuel Maurer, Bernhard Ewald Slawik, Heinrich Hussmann, and Matthew Smith. Now you see me, now you don't: Protecting smartphone authentication from shoulder surfers. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2937–2946, New York, NY, USA, 2014. ACM.

[8] Alexander De Luca, Katja Hertzschuch, and Heinrich Hussmann. Colorpin: securing pin entry through indirect input. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 1103–1106, New York, NY, USA, 2010. ACM.

[9] Malin Eiband, Mohamed Khamis, Emanuel von Zezschwitz, Heinrich Hussmann, and Florian Alt. Understanding shoulder surfing in the wild: Stories from users and observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4254–4265. ACM, 2017.

[10] Marian Harbach, Alexander De Luca, and Serge Egelman. The anatomy of smartphone unlocking: A field study of android lock screens. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4806–4817, New York, NY, USA, 2016. ACM.

[11] Marian Harbach, Emanuel Von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. Its a hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *Symposium on usable privacy and security (SOUPS)*, pages 9–11, 2014.

[12] Bernhard Haslinger, Peter Erhard, Eckart Altenmüller, Andreas Hennenlotter, Markus Schwaiger, Helga Gräfin von Einsiedel, Ernst Rummeny, Bastian Conrad, and Andrés O Ceballos-Baumann. Reduced recruitment of motor association areas during bimanual coordination in concert pianists. *Human brain mapping*, 22(3):206–215, 2004.

[13] Dong-Eog Kim, Min-Jung Shin, Kyoung-Min Lee, Kon Chu, Sung Ho Woo, Young Ro Kim, Eun-Cheol Song, Jun-Won Lee, Seong-Ho Park, and Jae-Kyu Roh. Musical training-induced functional reorganization of the adult brain: Functional magnetic resonance imaging and transcranial magnetic stimulation study on amateur string players. *Human brain mapping*, 23(4):188–199, 2004.

[14] Katharina Krombholz, Adrian Dabrowski, Peter Purgathofer, and Edgar Weippl. Poster: The petri dish attack - guessing secrets based on bacterial growth. In *Proceedings of the NDSS Symposium 2018*. Internet Society, 2018.

[15] Katharina Krombholz, Thomas Hupperich, and Thorsten Holz. Use the force: Evaluating force-sensitive authentication for mobile devices. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 207–219. USENIX Association, 2016.

[16] Youngbae Song, Geumhwan Cho, Seongyeol Oh, Hyoungshick Kim, and Jun Ho Huh. On the effectiveness of pattern lock strength meters: Measuring the strength of real world pattern locks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2343–2352. ACM, 2015.

[17] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. Quantifying the security of graphical passwords: The case of android unlock patterns. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 161–172, New York, NY, USA, 2013. ACM.

[18] Emanuel von Zezschwitz, Alexander De Luca, Bruno Brunkow, and Heinrich Hussmann. Swipin: Fast and secure pin-entry on smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1403–1406, New York, NY, USA, 2015. ACM.

[19] Emanuel von Zezschwitz, Alexander De Luca, Philipp Janssen, and Heinrich Hussmann. Easy to draw, but hard to trace?: On the observability of grid-based (un)lock patterns. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2339–2342, New York, NY, USA, 2015. ACM.

[20] Emanuel von Zezschwitz, Paul Dunphy, and Alexander De Luca. Patterns in the wild: a field study of the usability of pattern and pin-based authentication on mobile devices. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, MobileHCI '13, pages 261–270, New York, NY, USA, 2013. ACM.

Appendix A

User Study Questionnaire

1. What was your ID during the lab experiments?
2. Gender (male/female/no information)
3. Age
4. What kind of smartphone are you currently using?
   (single- choice: iPhone, Android, Other, I don't use a smartphone)
5. What methods are you currently using to unlock your smartphone? (multiple-choice: 4-digit PINs, password with character and digit, traditional unlock pattern, fingerprint sensor, face unlock, none)
6. Order the following methods according to their security. (4-digit PINs, fingerprint sensor, *StopUnlock Pattern*, traditional unlock pattern)
7. Order the following methods according to their authentication speed. (4-digit PINs, fingerprint sensor, *StopUnlock Pattern*, traditional unlock pattern)
8. Would you use the *StopUnlock Pattern* on your own smartphone to unlock it? (yes/no/maybe)
9. What did you like about *StopUnlock Pattern*? (optional, open-ended)
10. What did you NOT like about *StopUnlock Pattern*? (optional, open-ended)

Field Study Questionnaire

1. Would you use the *StopUnlock Pattern* on your own smartphone to unlock it? (yes/no/maybe)
2. Would you say you could enter the *StopUnlock Pattern* faster and with fewer errors after a few days of using our app? (yes/no/no difference)
3. Did you at one point change the *StopUnlockPattern* and why? (open-ended)

4. What problems did occur when entering the *StopUnlock Pattern* in the real word, for example on public transport? (open-ended)

5. What do you think of the stopping-time of the *StopUnlock Pattern*, should it be faster or slower? (open-ended)

6. Is there anything else you would like to let us know? (optional, open-ended)