ColorSnakes: Using Colored Decoys to Secure Authentication in Sensitive Contexts

Jan Gugenheimer[1] **Alexander De Luca**[2,3] **Hayato Hess**[1]
Stefan Karg[1] **Dennis Wolf**[1] **Enrico Rukzio**[1]
[1]Institute of Media Informatics, Ulm University, Ulm, Germany
[2]University of Munich (LMU), Munich, Germany; [3]DFKI GmbH, Saarbrücken, Germany
[1]<*firstname*>.<*lastname*>@*uni-ulm.de*, [2,3]*alexander.de.luca@ifi.lmu.de*

ABSTRACT
In this paper we present *ColorSnakes*, a PIN-based authentication mechanism for smartphones which uses fake paths on a grid of numbers to disguise user input. In a lab study (n=24), we evaluated variations of *ColorSnakes* in terms of usability and security. In comparison to direct input, indirect input significantly reduced the risk of shoulder surfing (10.5%) without increasing the input time. In a follow up real-world study (n=12), we compared *ColorSnakes* with PIN entry and Android's Pattern Unlock over the course of three weeks. Although authentication time for *ColorSnakes* was higher than for the other two mechanisms, participants valued the security benefit over its slightly higher error rate and increased authentication time. We argue that *ColorSnakes* could be used as an additional authentication mechanism alongside current mechanisms, thus providing the user with the choice of changing to *ColorSnakes* for certain applications or when there is an observer.

Author Keywords
Authentication; security; shoulder surfing; smartphone.

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User Interfaces - Input devices and strategies, evaluation.

INTRODUCTION
Today's smartphones are no longer just devices for making phone calls but are also devices which offer a variety of personal services and which contain private information. There are several unlock mechanisms which keep the phone and its content (e.g. pictures, emails, social media) secure. The most common and widespread knowledge-based mechanisms to secure the phone are Pattern Unlock (Android), PIN and passwords [14]. Since the smartphone has become such an important and integral device in people's lives, it is used in both private and public locations. Unlocking a smartphone with the aforementioned mechanisms in a public location provides almost no protection against shoulder surfers [19, 9]. In addition, current mechanisms provide little or no protection against so-called "smudge attacks" [2]. Furthermore, in light of upcoming ubiquitous cameras such as Google Glass, video attacks in which the attacker can record the whole authentication procedure are increasingly becoming a topic of more interest and relevance [24].

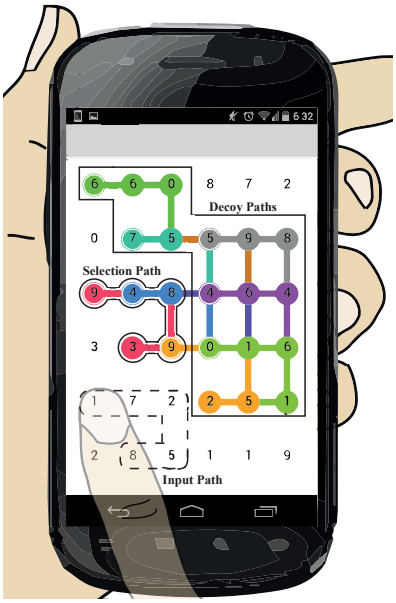


Figure 1. The final screen of a successful authentication using *ColorSnakes* in combination with offset input (the PIN being "red 3 + 9849"). The doted line indicates the physical input location (input path), while the red line shows the actual selection (selection path). During the input, the red path was partly covered by the blue and orange paths (decoy paths).

We introduce *ColorSnakes*, an authentication mechanism based solely on software modification which provides protection against shoulder surfing and to some degree to video attacks. A *ColorSnakes* PIN consists of a starting colored digit and is followed by four consecutive digits (for instance "red 3 + 9849" in figure 1). From the starting colored digit, users indirectly draw a path (selection path) consisting of their PIN. The input path can be drawn anywhere on the grid (see figure 1). As the user is inputting their PIN, different colored decoy paths will be generated simultaneously from other starting colored digits, imitating the selection path in order to disguise the input (figure 1). The underlying grid of numbers is randomly generated after each successful input to counter smudge attacks.

We argue that *ColorSnakes* can be used to provide additional security to specific applications such as pictures, emails or banking. We implemented and evaluated *ColorSnakes* as a phone unlock mechanism to be able to collect a big amount of data.

 MobileHCI '15, August 24 - 27, 2015, Copenhagen, Denmark 2015 ACM. ISBN 978-1-4503-3652-9/15/08$15.00 DOI: http://dx.doi.org/10.1145/2785830.2785834

However, we envision *ColorSnakes* more as a high security authentication mechanism which can either protect certain applications or activated by the users when needed (e.g. having the feeling of being observed or showing someone something on his phone and unlocking it in front of them).

We ran two user studies to evaluate the usability and security of *ColorSnakes*. In a lab study, we investigated the influence of *Decoy Paths*, *Grid Size* and *Input Method* on the usability of *ColorSnakes* (n=24). In a further evaluation, we demonstrated how these factors influence the resistance to shoulder surfing and video attacks using three experts. In a final real-world study (n=12), we compared *ColorSnakes* with Android's Pattern Unlock and PIN over the course of three weeks to investigate its real-world use. The main contributions of this paper are:

- *ColorSnakes*, an authentication mechanism which counters shoulder surfing, smudge and partially video attacks, for touch interaction on smartphones which is solely based on software and does not require any additional hardware.
- We provide data on the influence of *Decoy Paths*, *Input Methods* and *Grid Size* on the usability and security of *ColorSnakes*. Specifically, we show the high level of security of indirect input (only 10.5% successful shoulder surfing attacks).
- Real-world data from a three week comparative study on the use of *ColorSnakes*, Android's Pattern Unlock and PIN. We show the high user acceptance rate for *ColorSnakes*. Overall, 92% of the participants rated *ColorSnakes* as the most secure mechanism out of the three and 83% wanted to use *ColorSnakes* as an authentication mechanism to secure sensitive data.

RELATED WORK

There have been many different proposals for improving smartphone authentication. One of the aims (besides improving usability and memorability) of these proposals is to increase protection against shoulder surfing, video and smudge attacks. Proposals include the implementation of biometric mechanisms and additional hardware as well as indirect input mechanisms.

Biometric smartphone authentication has received a lot of attention in recent years. In addition to biometric features of the human body (e.g. face recognition [12] or fingerprints like in the iPhone 6), behavioral biometrics has also become an important area of research. These mechanisms exploit the way users behave as well as the existence of multiple sensors on current smartphones. For instance, extensions to Android's Pattern Unlock were proposed [1, 7] which use the pattern as well as the way it is performed to identify a person. By adding a second, invisible layer of security, the interaction itself is not compromised. Further examples of behavioral biometric approaches include the identification of key stroke dynamics [6, 11] and gait patterns [13].

While biometric mechanisms have multiple advantages, we opted for developing a knowledge-based mechanism due to two main reasons: a) many users have privacy issues with having their biometric data stored and out of their control [18]; b) modern smartphones are in many cases shared with others [15], even in an ad-hoc manner which is hard or even impossible with current biometric approaches.

Sharing secret information with the user over an invisible channel is a useful way to secure the input against onlookers. In most cases, additional hardware is used to provide these channels to the user [3, 4]. For instance, the phone lock by Bianchi et al. [3] uses an external motor to provide tactile feedback to the user. The authentication is then adapted based on this feedback. In our work, we wanted to avoid additional hardware in order to create an authentication mechanism that can be used on current off-the-shelf smartphones by using simple software updates.

ColorSnakes uses visual distraction [16, 21] as well as indirect input [17, 21] to secure the authentication process. Visual distraction mechanisms mainly focus on providing protection against shoulder surfing and function based on the provision of additional actions that confuse an onlooker. Despite focusing on desktop environments, cursor camouflage [23] and fake cursors [10] highlight this idea well. In both mechanisms, additional mouse pointers realistically move over the screen to hide the actual input (the real mouse pointer). In *ColorSnakes*, distraction is achieved by different colored lines (decoy paths) moving across the screen in unison with the input (selection path).

Finally, indirect input refers to mechanisms which decouple the input from the actual information. More specifically, this means that the interaction which takes place on the screen does not reveal the true data input. For instance, Picassopass by van Eekelen et al. [21] encodes several attributes such as color, position etc. into one icon. Several of these icons are then presented to the user but clicking on them does not reveal which icon is the true input.

THREAT MODEL

In our threat model, we assume an observer who is standing close to the user in a public environment. To show how shoulder surfing attacks are countered, we are demonstrating the worst case scenario.

Under perfect lighting conditions, the user enters their PIN by holding the device in one hand and using their second hand to input the data. The observer is standing close behind the user on the opposite side of the inputting hand to avoid visual obstruction.

CONCEPT

To authenticate in *ColorSnakes*, a PIN consisting of a starting color and four digits is used (e.g. "red 3 + 9849"). In an initial process, a random grid of numbers is generated and the user's PIN is randomly placed inside that grid. In this grid, ten digits from zero to nine (starting colored digits) will be highlighted using a specific color (resulting overall in ten unique colors). The specific color of a starting colored digit will always remain the same (e.g. "3" will always be red) allowing the user to memorize either a five digit PIN or a color and four digits (in this example "39849" or "red + 9849").

When authenticating, a user has to find their starting colored digit on the grid (e.g. red 3). From the starting colored digit, they select their path on the grid corresponding to their remaining PIN (e.g. 9849). During the input, the path created by the user (selection path) will be drawn from every other starting colored digit to disguise the user input (decoy paths). In figure 1, the user selected the path "red 3 + 9849" using *Offset Input* (explained later, see *Input Methods*). During the input, decoy paths (e.g. "orange 2

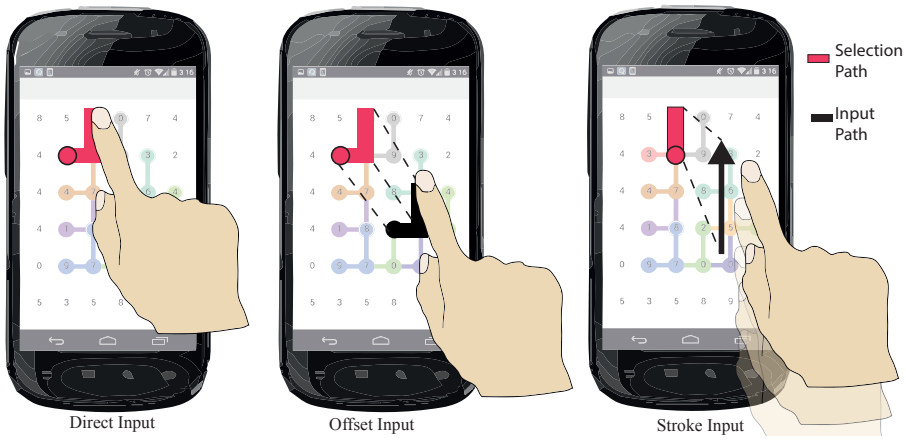


Figure 2. A visualization of the different *Input Methods* which were used in the lab study: Direct Input, Offset Input and Stroke Input. The red path indicates the selection path while the black path shows the actual input path. In this figure, the selection path started at the circled location.

+ 5109") imitated the same path. In the case that a decoy path runs outside of the grid, a random function will generate a new direction at the edge of the grid so as to redirect the decoy path back inside. Although this will not help during a video attack, it will help increase the protection against shoulder surfers since following the finger movement and the drawing simultaneously is a difficult task (see section Security Analysis).

After every successful input, the entire grid is once again randomly generated and the starting colored digits together with the user's PIN are randomly placed onto the grid.

In several brainstorming sessions, different *Input Methods* and modifications of this basic concept were discussed. Three *Input Methods* were evaluated in a lab study (see figure 2): The initial concept of *ColorSnakes* was built around *Offset Input*. Using *Offset Input*, the user can draw the input path anywhere on the grid. The absolute movement of their finger is mapped to the user's starting colored digit. Another indirect input method was *Stroke Input*, where the user draws several directional strokes (in one of eight possible directions: up, down, left, right and the diagonals) anywhere on the grid to draw the selection path. To measure the impact of indirect input methods on *ColorSnakes*, *Direct Input* was further used as a baseline. In *Direct Input*, the user directly draws the selection path on the corresponding digits. *Offset Input* and *Stroke Input* are both indirect input methods. Both of them separate the input from the actual selection path. Further variations included the size of the grid (*Grid Size*), whether the selection path should be visible or not (*Path Visibility*) and whether the selected digits should be masked using an asterisk (*Star Visibility*). All of these variations were combined, implemented and evaluated in a lab study as described in section Lab Study.

Theoretical Security

Brute Force/Guessing Attacks: The different *Input Methods* have different levels of security. When assuming that a PIN consists of a "colored starting digit + four digits", the *Direct Input* method results in a theoretical password space of 10 (picking one color) $*(8)^4$ (selecting one of eight directions four times) $= 40,960$ possibilities. This is around four times bigger than a four-digit PIN (10,000 possibilities) and almost the same as an Android Pattern Unlock with five strokes (32,768 possibilities [22]).

By using the *Offset Input* or *Stroke Input* method with one color and four digits as the PIN, the theoretical password spaces are reduced to $8^4 = 4{,}096$ possibilities, since the starting color is no longer selected but it is assumed that the user knows it. However, even when three guessing attempts are taken into account, there is still a very low probability of guessing the correct PIN; approximately 0.07% with *ColorSnakes* in comparison to a four-digit PIN 0.03% and a five stroke Android Unlock Pattern 0.02%. Similar to [8], we argue that observation attacks are a much more serious threat in the mobile context than brute force attacks.

Smudge Attack: ColorSnakes is by design resistant to smudge attacks since the numerical grid and the placement of the PIN is always randomly generated after each successful authentication. Therefore, even if an attacker could steal the correct smudge path, a new grid would have already been generated.

Video Attack: ColorSnakes is partially resistant to a one time video attack but can be broken using repetitive video attacks of different authentications. However, *ColorSnakes* focuses on more casual everyday life threats and does not protect against professional attackers.

LAB STUDY

We conducted a lab study in order to evaluate the advantages and disadvantages of all the modifications of *ColorSnakes* described earlier in the Concept section. The goal was to evaluate the influence of each variable on usability and security.

Study Design

The study was conducted using a repeated measures factorial design. The independent variables were *Input Method*, *Path Visibility*, *Star Visibility* and *Grid Size*. The *Input Method* had three levels: *Direct Input*, *Offset Input* and *Stroke Input* (see section Concept and figure 2). *Path Visibility* was a boolean variable since the path could either be drawn in the corresponding color of the starting colored digit or it could be invisible. *Star Visibility* was also a boolean variable since it could either overwrite each digit of the selection path and decoy paths with a star or it could not (see figure 3). Finally, *Grid Size* also had two levels, namely a 6x6 grid size or a 8x8 grid size. This resulted overall in 24

Figure 3. A visualization of the combination of the variables *Path Visiblity* and *Star Visibility*.

Input Method	Direct		Offset		Stroke		
Grid Size	6x6	8x8	6x6	8x8	6x6	8x8	*Mean*
Path On Stars Off	6.94	2.78	13.89	12.50	2.78	4.17	*7.18*
Path Off Stars Off	2.78	6.94	56.94	45.83	2.78	2.78	*19.68*
Path On Stars On	5.56	1.39	16.67	26.39	5.56	11.11	*11.11*
Path Off Stars On	2.78	1.39	62.50	50.00	4.17	6.94	*21.30*
Mean	*4.51*	*3.13*	*37.50*	*33.68*	*3.82*	*6.25*	*14.81*

Figure 4. Percentage of the average failed authentications within the three authentications for each variant of the *ColorSnakes* mechanism.

different variants of *ColorSnakes*, which were presented to the participants using a 24x24 Latin square for counterbalancing.

The dependent variables were authentication time, error rate and cognitive load. The latter was measured using three questions picked from the NASA TLX (mental demand, physical demand and frustration). The protection against shoulder surfing/video attack was collected in an expert attack study.

Procedure

All participants performed the authentication task on the same smartphone (Samsung Galaxy Nexus) having the same conditions (e.g. same room). To begin with, the participants were introduced to the study and were shown a video demonstrating each variant. Participants were instructed to hold the phone in their non-dominant hand and to input with their dominant hand.

The authentication phase started with a training of the current variant. For each variant, a new random PIN consisting of one color and four digits was generated. This PIN was written down and kept in front of the user during the whole authentication phase. Participants could practice using the variant until they felt comfortable and until they were able to successfully input their PIN at least two times. After each training, the participants had to authenticate three times. For each authentication participants had three attempts until one authentication finally failed. Once the authentication phase was completed, the participants answered three questions picked from the NASA TLX (mental demand, physical demand and level of frustration).

Participants

We recruited 24 participants (female=8, male=16) with an average age of 29 years (range: 19 to 45). The participants were mostly students or people with a university education. All participants were smartphone users, 16 used a screen lock mechanism (Pattern Unlock: 9, PIN: 6 , Password: 1, None: 8). The overall study lasted 1.5 hours. Participants could choose between receiving 6 Euro or 5 bars of chocolate as a reward.

Results

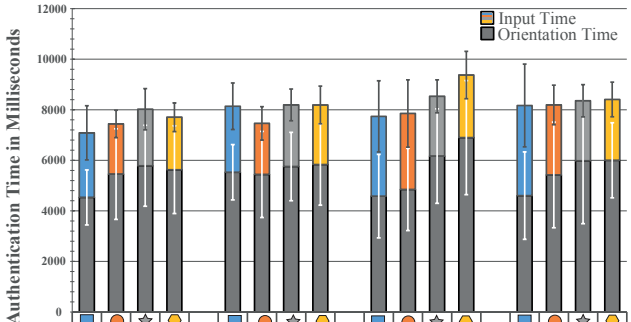
Since there were 24 participants who authenticated at least three times with 24 variants, our analysis was based on 1,728 (1,474 successful) authentications. *Error rate* was measured by counting how many failed authentications occurred during the three authentications of a participant (value from 0 to 3 failed authentications). If a participant was able to authenticate within three attempts, this counted as one successful authentication. The three attempts were chosen on account of their real life appliance in systems such as ATMs or mobile phones. Figure 4 gives an overview of the failed authentications from all 24 variants.

Since the error rate was not normally distributed, a Friedman ANOVA and post hoc analysis with Wilcoxon signed-rank test was conducted. There was a statistically significant difference in the error rate according to the *Input Method* used ($\chi^2(1)=151.04$, $p<.001$). Using a Wilcoxon signed-rank test with a Bonferroni correction resulted in a significance level of $\alpha<.017$. Post hoc analysis using Wilcoxon signed-rank tests showed significantly higher errors using *Offset Input* vs *Direct Input* (Z=-8.96, $p<.001$) and *Offset Input* vs *Stroke Input* (Z=-8.49, $p<.001$). However no significant differences between *Direct Input* and *Stroke Input* were found. One surprising finding was that there were some participants who were not able to authenticate even once using *Offset Input*.

Furthermore, *Path Visibility* was a significant factor for the error rate. The Wilcoxon signed-rank test revealed a significantly higher error rate when paths were not enabled (Z=-5.10, $p<.001$). This indicates that users had more difficulty authenticating when paths were not visible. The *Star Visibility* and *Grid Size* had no significant influence on the error rate.

The Authentication time was divided into two times: *orientation time* and *input time*. The orientation time was measured from the moment the users were presented with the initial screen (random grid and ten colored digits, (see figure 8)) up until the first touch on the touch screen. The input time was measured from the first touch until the last touch of an authentication. Only successful authentications were used for the time analysis (1474 of the 1728). Furthermore, since some participants using the *Offset Input* were not able to authenticate successfully even once, *Offset Input* was not taken into account in this analysis. An overview of the overall authentication times can be found in figure 5.

A $2x2x2x2$ (*Input Method* x *Path Visibility* x *Star Visibility* x *Grid Size*) repeated measures ANOVA of the total time showed significant main effects for *Star Visibility* ($F(1,23)=16.778$, $p<.001$, $\eta^2=0.42$). Pairwise comparisons showed that with *Stars On* (M=7.88s, SD=2.96s) participants authenticated significantly slower ($p<.001$) than with *Stars Off* (M=7.32s, SD=2.85s). The same analysis using the orientation time resulted also in a significant effect for *Star Visibility* ($F(1,23)=23.320$, $p<.001$, $\eta^2=0.50$). Participants needed significantly more time ($p<.001$) using *Stars On* (M=5.73s, SD=2.38s) than *Stars Off* (M=4.81s, SD=2.03s).



	Direct 6x6		Direct 8x8		Stroke 6x6		Stroke 8x8	
	Orientation Time	Input Time	Orientation Time	Input Time	Orientation Time	Input Time	Orientation Time	Input Time
Path On Stars Off	4532	2555	5530	2608	4583	3157	4602	3567
Path Off Stars Off	5454	1986	5441	2021	4847	3006	5420	2771
Path On Stars On	5785	2235	5755	2437	6166	2364	5985	2372
Path Off Stars On	5628	2075	5833	2357	6898	2475	6001	2406

Figure 5. Average Authentication Time using the different levels of independent variables.

A $2x2x2x2$ (*Input Method* x *Path Visibility* x *Star Visibility* x *Grid Size*) repeated measures ANOVA of the input time showed significant main effects for *Input Method* ($F(1,23)=8.928$, $p<.01$, $\eta^2=0.28$), *Path Visibility* ($F(1,23)=13.073$, $p<.001$, $\eta^2=0.36$) and *Star Visibility* ($F(1,23)=7.164$, $p<.05$, $\eta^2=0.24$). We also found significant interaction effects for *Input Method x Star Visibility* ($F(1,23)=28.598$, $p<.001$, $\eta^2=0.55$) and *Path Visibility x Star Visibility* ($F(1,23)=14.666$, $p<.001$, $\eta^2=0.39$). Pairwise comparisons revealed that participants inputted significantly faster ($p<.01$) using the *Direct Input* (M=2.09s, SD=1.50s) compared to *Stroke Input* (M=2.47s, SD=1.27s). Also participants entered significantly slower ($p<.01$) using *Path On* (M=2.39s, SD=1.47s) than *Path Off* (M=2.17s, SD=1.29s). Finally, participants inputted significantly faster ($p<.05$) using *Stars On* (M=2.18s, SD=0.97s) compared to *Stars Off* (M=2.38s, SD=1.70s).

Analyzing the internal consistency of all three questions (Frustration, Mental Demand and Physical Demand) for each variable resulted in an internal consistency (Cronbach's α) of: *InputMethod* (Direct: α=0.87, Offset: α=0.87, Stroke: α=0.92), *Path Visibility* (On: α=0.90, Off: α=0.92), *Star Visibility* (On: α=0.91, Off: α=0.90) and *Grid Size* (6x6: α=0.84, 8x8: α=0.78). Therefore the three questions were combined in one task load scale (a high score represents a high task load, on a scale from one to ten).

A Friedman ANOVA and post hoc analysis with Wilcoxon signed-rank test was conducted for the task load. There was a statistically significant difference in the task load depending on the *Input Method* used ($\chi^2(1)=204.02$, $p<.001$). Using Wilcoxon signed-rank test with a Bonferroni correction resulted in a significance level of $\alpha<.017$. Post hoc analysis using Wilcoxon signed-rank tests showed a significantly higher task load using *Offset Input* (M=5.77, SD=2.59) vs *Direct Input* (M=2.81, SD=1.64) (Z=-10.82, $p<.001$) and *Offset Input* vs *Stroke Input* (M=2.95, SD=2.01) (Z=-10.64, $p<.001$). Furthermore, a Wilcoxon signed-rank test showed that *Path Visibility* (*Path On* M=3.59, SD=2.25; *Path Off* M=4.09, SD=2.74) (Z=-5.16, $p<.001$) and *Star Visibility* (*Stars On* M=4.21, SD=2.60; *Stars Off* M=3.48, SD=2.38) (Z=-7.04, $p<.001$) resulted in a significantly higher task load. The *Grid Size* had no significant influence (6x6 M=3.90, SD=2.47; 8x8 M=3.79, SD=2.19) (n.s.).

Security Analysis

In a security analysis, we wanted to investigate the influence of the different variables on the shoulder surfing and video attack success rate. Therefore, we video recorded every authentication of each participant during the lab study. For the security analysis, the video material was cut down to one successful authentication for each participant with each variant. Since the *Offset Input* had such a poor performance, it was not considered in the security analysis. Overall this resulted in 384 videos (24 participants * *Input Method* (2) * *Path Visibility* (2) * *Star Visibility* (2) * *Grid Size* (2)).

Three members of our institution who were not involved in this work were recruited to simulate attackers. Each of them was introduced to all the variants and educated to a point that they could be considered experts. Each expert watched and attacked a different subset of eight of the 24 participants (128 videos). Since watching and attacking one participant took around one hour, two sessions of four hours, spread over two days were needed. The experts were placed in a lab environment with a laptop and a 24 inch display.

Before starting the video, the experts were told which variant they were about to see and could use it on a smart phone beforehand. Then, they watched the authentication once and were allowed three attempts to guess the right PIN (shoulder surfing). If the experts were not able to figure out the right PIN, they could start the same video again and were allowed to operate and navigate the recording and were again allowed three guessing attempts (video attack). All videos were played without sound. The videos were recorded from an over the shoulder perspective, allowing to perfectly see the display and the user's finger. The recordings where done in 1080p (25fps) using a Sony Alpha 57. During the entire procedure, the experts were encouraged to think aloud, so the experimenter was able to understand and write down the attack methodology being used.

The experts were all inclined to use the same approaches for the attacks. For the *Direct Input*, the shoulder surfing method depended on whether stars were visible or not. In case *Stars Off*, the experts could mostly follow the whole input path. In case *Stars On*, the experts tried to estimate the starting colored digit based on the hand position and tried to memorize as many digits in its vicinity as possible. For the video attack, the experts could almost always recreate the PIN by first recreating the path and then rewinding the video to see the grid before the stars appeared.

For *Stroke Input* in the shoulder surfing attack, experts picked one starting colored digit, close to the input finger, and tried to follow that path. If the path was not enabled (*Path Off*), they also tried to estimate the directions of the strokes. In the video attack, experts mostly used the following three-step approach. Firstly, they recreated the selection path. If the path was invisible (*Path Off*), experts could mostly recreate it from watching the finger movement (84%). Secondly, experts watched or retraced the discovered path from every starting colored digit to see if one would run out of bounds. This helped them to expose some decoy paths. In a last step, the position of the finger was set in relation to what could be a possible input. In that step, effects such as

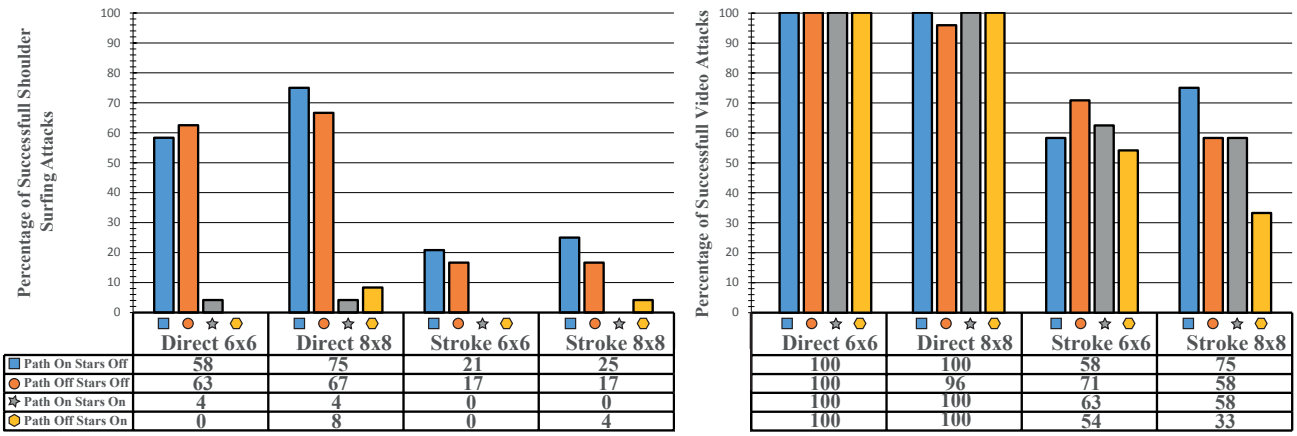


Figure 6. Successful shoulder surfing and video attacks for all combinations of each variable in percent.

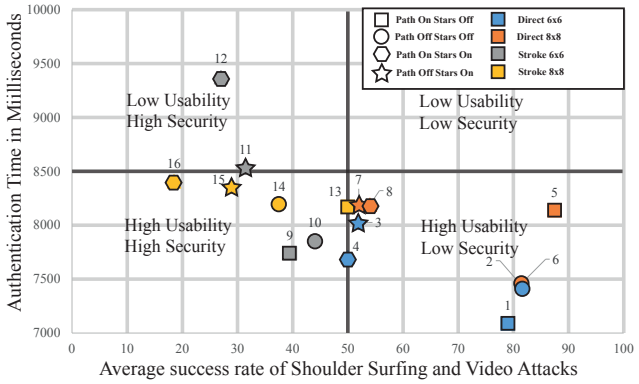


Figure 7. Arrangement of each variant (numbered from 1 to 16) along the two axis: authentication time and the mean of successful shoulder surfing and video attacks, resulting in a usability/security matrix with the desired position in the lower left corner

obscuration and distance to the finger were taken into account to expose other decoy paths. After these steps were taken, experts attempted to guess one of the remaining possible paths.

An overview of the successful shoulder surfing and video attacks can be found in figure 6. Since six participants directly drew their strokes onto their selection path, and thereby revealed the selection path, the success rate of *Stroke Input* was negatively influenced. Nevertheless, *Stroke Input* resulted in an overall lower shoulder surfing and video attack success rate. Shoulder surfing and video attacks were measured in a boolean variable for each variant of one participant (successful in three attempts or failed). Therefore, the variables were considered dichotomous and continuity-corrected McNemar's tests with Bonferroni correction revealed that significantly more shoulder surfing attacks succeeded in an attack on *Direct Input* (35%) than on *Stroke Input* (10.5%) (χ^2=38.47, p<.001, ϕ=0.45). In addition, significantly more shoulder surfing attacks succeeded with *Stars Off* (43%) variants than with *Stars On* (2.5%) variants (χ^2=71.31, p<.001, ϕ=0.61). Looking at the video attacks, *Direct Input* (99.5%) led to a significantly higher video attack success rate compared to *Stroke Input* (58.75%) (χ^2=76.01, p<.001, ϕ=0.63). All the other variables were not significant.

Discussion

Input Method: The study revealed that using *Offset Input* led to a significantly higher level of task load and error rate. During the study, participants had problems controlling the offset line drawing. The offset was implemented by mapping direct movement one to one from the physical input location to the selection path. Furthermore, when close to a digit, the path would snap with a certain pixel offset. The snap was signaled with a vibration of the phone. Even though this principle was explained in full detail and participants were able to practice, it probably led to a wrong mental model during the input of the participants and resulted in overdrawing. Once the path was invisible (*Path Off*), users struggled to realize where their current tip of the selection was, which then resulted in a wrong input and led to an overall higher error rate (see figure 4). Since the goal was to design an authentication mechanism having a high usability, we decided not to pursue the *Offset Input* method in the real-world study.
Direct Input resulted in the shortest overall input time. However, in the shoulder surfing and video attack experiment, it was the most vulnerable *Input Method*. Video Attacks were successful almost every time (figure 6). Since the goal was to create a mechanism which counters shoulder surfing, we did not continue using the *Direct Input* method for the real-world study.
Regarding the overall authentication time, *Stroke Input* had almost the same authentication time as *Direct Input*. Furthermore, *Stroke Input* had also the highest level of protection against shoulder surfing and video attacks. By using *Stroke Input* and therefore not revealing the selection path, attackers were forced to guess one of the possible paths. Therefore, the *Stroke Input* was the most promising *Input Method* and was enhanced and used in the real-world study.

Path Visibility: Path Visibility had no negative influence on the overall authentication time. However, it resulted in a slower input time when it was enabled. This can possibly be explained with the distraction the decoy paths create when they are visible. While inputting the PIN, decoy paths can cross the selection path and as a result overdraw a color. However, with the *Stroke Input*, paths were considered very useful since the input is conducted in several steps and the path shows the current state and progress. Considering the security analysis with the *Stroke Input*, hiding the paths led to a higher security. Not all paths could be recreated during the video attack and following the users

finger and watching the grid (during the shoulder surfing attack) was considered a hard task. Therefore disabling the paths can be seen as one step in increasing the protection against shoulder surfing but it may decrease usability for the *Stroke Input*.

Star Visibility: As described earlier in the paper, authentication times have been sub-divided into orientation time and input time. Enabling the stars led to a higher orientation time, since participants had to find the entire path before starting the input, as it was possible that one decoy path would cross the selection path and hide a digit which is needed. As a result, the input time was decreased, since the path was already known by the user and they did not have to think or pay attention during the input. When looking at the overall authentication time, it became clear that this trade-off was not high enough since the overall authentication time was still higher compared to *Stars Off*. Considering the security analysis, attackers stated that using stars highly increased the protection for shoulder surfing. Using the attack methods previously stated, attackers tried to memorize a large amount of numbers which surrounded one digit. This mostly did not work and several digits were forgotten. In the video attack however, stars did not increase the security since attackers could easily rewind the video and see the whole grid again. Nevertheless, *Star visibility* led to a higher level of protection against shoulder surfing but resulted in a lower usability, in terms of overall authentication time and a higher task load.

Grid Size: The different sizes had no significant effect on authentication time, error rate or cognitive load. In the security experiment it also showed no effect. The reason therefore could be that both sizes were picked to still be usable for the participants. However, we assume that larger grids would result in a higher authentication time since the orientation time would probably increase with the higher amount of targets to search trough to find the colored starting digit. For the final mechanism we therefore decided to use the smaller grid since the verbal feedback during the study benefited the 6x6 grid.

REAL-WORLD STUDY

In a final user study, we wanted to investigate the real-world use of *ColorSnakes* and compare it to widespread unlock mechanisms currently being used (PIN and Android's Pattern Unlock). Therefore, we conducted a three week long real-world study by deploying all three mechanisms on users' phones. We wanted to investigate how authentication times develop over this period, what learning effects arise and how people would use the mechanism.

The variant which was chosen for the real-world study was *Stroke Input, Path On, Stars Off, 6x6*. This variant was considered the best trade-off between usability and security (figure 7). We decided against *Stars On* and *Path Off* variants (e.g. variant 3, 4, 9) since they resulted in a significant higher task load, which possibly could lead to a higher level of frustration and dropout in a real-world study. Furthermore, based on the insights gained from the lab study and the security analysis, the final version was modified to further improve security. To prevent people from directly drawing the strokes on their path, and therefore giving away their PIN, a dedicated area was created on the bottom of the screen (figure 8).

To ensure we collected a large amount of authentications we opted for implementing *ColorSnakes* as a phone unlock and not

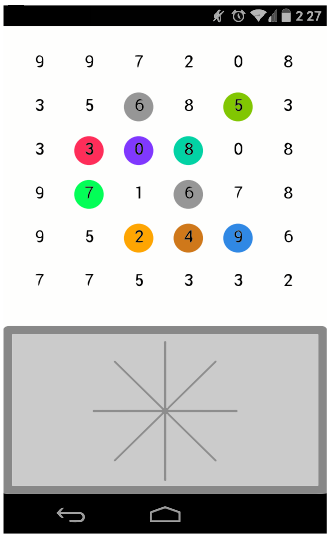


Figure 8. Initial start screen of the modified *ColorSnakes* mechanism, which was used in the final real-world study.

a sensitive application unlock. Therefore, we implemented the selected *ColorSnakes* variant as well as PIN and Pattern Unlock as the actual unlock screen for Android phones. This allowed for being able to log every input on the phone (only during the unlocking). To enhance everyday usability, the implementation did not run in full screen and showed the status bar of the phone. This enabled the participants to check for new messages or the time without unlocking the phone (which frequently happens [14]). In case of a problem with the unlocking application, the user could press the back button to switch back to the native PIN application of Android, where features such as the alarm clock could be operated without unlocking the phone. These modifications to the unlocking service ensured that the unlock applications could be operated in daily life without any drawbacks.

Study Design

The final user study was conducted using a repeated measures factorial design. The independent variable was *Unlock Mechanism* and had three levels: PIN, Pattern Unlock and *ColorSnakes*. The dependent variables were authentication time, error rate and perceived usability which was measured using the System Usability Scale (SUS) questionnaire [5]. The SUS generates a score on a scale from 0 to 100 (0 being worst and 100 best). The study took place within a three week period, in which each *Unlock Mechanism* was installed for every user for one week using counterbalancing.

Procedure

For one week participants used one of the mechanisms as their *Unlock Mechanism* for the phone. Each participant was visited at home or contacted in a video call and one *Unlock Mechanism*, based on the counterbalancing, was installed. Participants were introduced to the study and were instructed on how their *Unlock Mechanism* works. For each *Unlock Mechanism* a PIN (four digits), *ColorSnakes* (color + four digits) or Pattern Unlock (five strokes) was randomly generated (the PINs had the constraint of two consecutive numbers not being the same to avoid benefiting the PIN mechanism). We randomly generated these to avoid

too simple user choices (e.g. 1212) and thereby "skipping" the authentication process out of convenience. Furthermore, each participant selected one fallback PIN which could be used in case the generated PIN was forgotten. After one week of use, participants were revisited and were asked to fill out a questionnaire on the use of the unlock mechanism. The new mechanism was subsequently installed. At the end of the three weeks all participants were again asked to fill out one last questionnaire to compare, rank and comment on all three *Unlock Mechanisms*.

Participants

For the real-world study, 12 participants (female=4, male=8) with an average age of 25 years (19 to 33) were recruited using word-of-mouth. All participants were Android users and owned a device with at least Android version 4.2.2 and almost all were university-educated. On average, participants were using smartphones for around 4 years (range: 1 to 7) and used a touch screen for around 5 years (range: 2 to 11). Furthermore, 10 of the participants used an unlocking mechanism for their smartphone (PIN: 2, Pattern Unlock: 7, Password: 1). As an incentive, 30 Euro was given to each participant after the three weeks.

Results

To evaluate the results, data of each participant with each *Unlock Mechanism* was pruned down to seven full days. The pruning was done by selecting the first full seven days of use. Outliers (e.g. accidental activation of the smartphone inside the pocket) were eliminated using the Tukey method [20]. Since this led to an unbalanced amount of authentications per participant per day (some authenticated more often during one day some less), the means per participant per day were used for the analysis. During the 21 days, we collected approximately 8,000 authentications from the 12 participants, which were used for the analysis. On average, each participant activated (pushed the power button) the phone approximately 58 times a day (range from 18 to 204) and unlocked the phone 32 times a day (range from 12 to 107). The difference in these values can be explained by participants activating their phone without unlocking it to see the current time or check for notifications, similar to the findings of Harbach et al. [14].

The average number of successful and failed authentications per day can be found in figure 10. An authentication was considered as failed, in case the participant failed to authenticate once. The error rate was calculated using the mean failed authentications per day. A $3x7$ (*Authentication Method x Days of Use*) repeated measures ANOVA revealed significant main effects in error rate for *Unlock Mechanism* (F(2,22)=33.292, p<.001, η^2=0.75). An analysis of contrasts showed that *ColorSnakes* had a significantly higher error rate compared to PIN (F(1,11)=54.421, p<.001, η^2=0.83) and Pattern Unlock (F(1,11)=28.436, p<.001, η^2=0.72).

The authentication times of the participants during the seven days are depicted in figure 9. Similarly to the lab study, we divided the total authentication time into orientation time and input time. For all measurements (day one to seven), *ColorSnakes* was slower in terms of orientation time, input time and total authentication time compared to PIN and Pattern Unlock. A $3x7$ (*Authentication Method x Days of Use*) repeated measures ANOVA revealed a significant difference for the orientation time (F(1.162,12.786)=206.300, p<.001, η^2=0.95), input time (F(1.130,12.433)=35.895, p<.001, η^2=0.77) and total authentication time (F(1.125,12.371)=85.873, p<.001, η^2=0.89) between all three *Unlock Methods*. An analysis of contrasts showed that *ColorSnakes* had a significantly higher total authentication time compared to PIN (F(1,11)=85.893, p<.001, η^2=0.89) and Pattern Unlock (F(1,11)=92.674, p<.001, η^2=0.89).

By calculating the difference of the authentication times between the first day and the last day, a new variable for each mechanism was created which indicates the improvement of participants over the seven days. An ANOVA revealed a significant difference for each mechanism in regards to total authentication time (F(1.227,13.494)=33.910, p<.001, η^2=0.76), input time (F(1.132,12.456)=8.815, p<.001, η^2=0.45) and orientation time (F(1.179,12.965)=17.724, p<.001, η^2=0.62). Pairwise comparisons revealed a significantly higher drop in authentication time for *ColorSnakes* (M=3364ms, SD=1692ms) vs PIN (M=699ms, SD=890ms) (p<.001) and *ColorSnakes* vs Pattern Unlock (M=558ms, SD=411ms) (p<.001), a significantly higher drop in orientation time for *ColorSnakes* (M=2439ms, SD=524ms) vs PIN (M=534ms, SD=238ms) (p<.001) and *ColorSnakes* vs Pattern Unlock (M=260ms, SD=59ms) (p<.001) and a significant higher drop in input time for *ColorSnakes* (M=962ms, SD=340ms) vs PIN (M=81ms, SD=36ms) (p<.001) and *ColorSnakes* vs Pattern Unlock (M=212ms, SD=70ms) (p<.001).

Using the System Usability Scale (SUS), participants ranked PIN (84.5 points) the highest, Pattern Unlock second (79.3 points) and *ColorSnakes* third (70 points). In further questions, 33% of the participants stated they would use *ColorSnakes* as their new unlock mechanism (figure 11). 83% of the participants said that they would like to use *ColorSnakes* to unlock sensitive applications such as online banking, picture galleries and certain emails or messages. 92% of the participants ranked *ColorSnakes* as the most secure mechanism and saw the benefit in it. In total, 50% of the participants remembered five digits and the color (color + 4 digits: 25%, five digits: 25%) as their *ColorSnakes* PIN. The color was mainly used for fast orientation and the five digits were remembered as the PIN. During the real-world study, none of the participants forgot their PIN using any one of the mechanisms.

To further investigate the potential of *ColorSnakes* to decrease the authentication time, one of the authors conducted an informal self-study. The author used *ColorSnakes* on his private phone as an unlock screen for over three weeks. His total authentication time decreased from an average of 5.2 seconds on the first day to an average of 3.4 seconds on the last (approximately one second orientation and two seconds input). Overall, he authenticated around 300 times (average of 18 times per day) with an average of 80% successful authentications.

DISCUSSION

Acceptance and Use: The three week real-world study showed that in spite of the slightly higher error rate and authentication time, participants (92%) valued the high level of security of *ColorSnakes*. The fact that 83% of the participants said they wanted to use *ColorSnakes* on their phone to secure sensitive data and 33% were even willing to change their current unlock mechanism to *ColorSnakes* showed that the security benefits exceed the usability deficits. Even though participants learned the

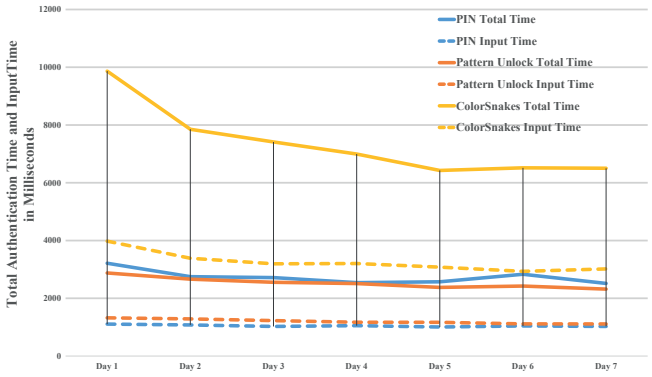
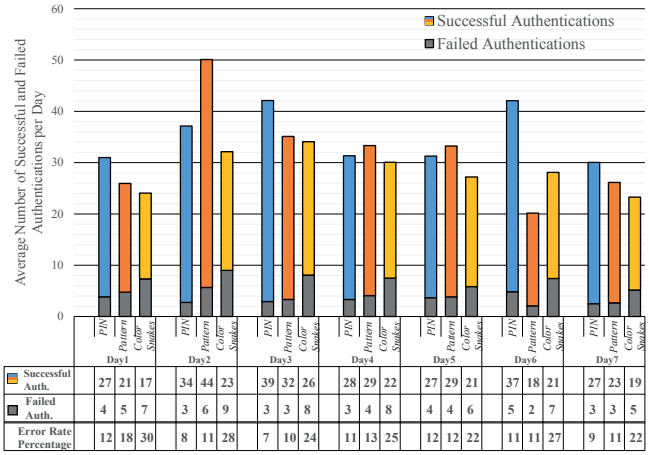


Figure 9. Authentication Time and Input Time for each mechanism over the deployment of seven consecutive days.



	Day1			Day2			Day3			Day4			Day5			Day6			Day7		
	PIN	Pattern	Color Snakes	PIN	Pattern	Color Snakes	PIN	Pattern	Color Snakes	PIN	Pattern	Color Snakes	PIN	Pattern	Color Snakes	PIN	Pattern	Color Snakes	PIN	Pattern	Color Snakes
Successful Auth.	27	21	17	34	44	23	39	32	26	28	29	22	27	29	21	37	18	21	27	23	19
Failed Auth.	4	5	7	3	6	9	3	3	8	3	4	8	4	4	6	5	2	7	3	3	5
Error Rate Percentage	12	18	30	8	11	28	7	10	24	11	13	25	12	12	22	11	11	27	9	11	22

Figure 10. Rounded mean values for successful and failed authentications with each mechanism over the deployment of seven consecutive days.

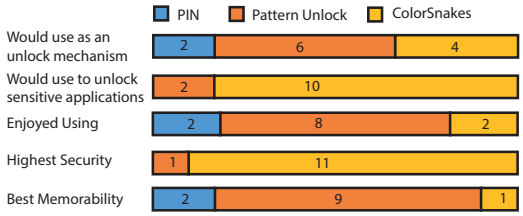


Figure 11. Results from the comparison questionnaire handed out after the three week real-world study.

interaction concept quickly, it still required a certain amount of concentration. This indicates that *ColorSnakes* is highly suitable for less frequently occurring situations that require a high level of security. A possible scenario would be to secure an image gallery with sensitive/private photos. A further benefit would be its implicit security concept. If a user plans to show some photos secured with the *ColorSnakes* mechanism, then they would not have to cover the authentication input from onlookers (in this case friends) which would directly indicate a level of mistrust.

Performance and Improvement: The almost equally distributed usage of all three mechanisms over the three weeks (figure 10) indicates that *ColorSnakes* is feasible to use on a daily basis. Although the authentication time was significantly higher compared to PIN and Pattern Unlock, *ColorSnakes* had by far the highest decrease in authentication time over the period of seven days. Therefore, we assume that authentication time can decrease further over a longer period of use. A further indicator for this assumption are the times (orientation time: 1.2 seconds, input time: 2.2 seconds) one author achieved over the period of three weeks using *ColorSnakes* on his private device.

The generally higher error rate of all three mechanisms compared to other lab studies (e.g. [9]) can be explained with the use of the real-world study approach. In such studies, the login process is less controlled and is more influenced by the user's daily routine, therefore achieving a higher ecologic validity. Participants stated, that the initial orientation part of *ColorSnakes* forced the users to fully concentrate on the authentication process, whereas PIN and Pattern Unlock will work almost exclusively using muscle memory. This could be one reason for the higher error rate of *ColorSnakes*, since unlocking in real life does not always offer a scenario which allows the user to fully concentrate on the unlocking process.

Perceived Security The qualitative feedback from the 21 day study demonstrated that when using *ColorSnakes*, participants felt more comfortable and reassured inputting their PIN in public setting i.e. in a setting which is vulnerable to shoulder surfing. Participants also felt they were able to avoid a level of awkwardness as they did not have to cover up their input in front of friends and family since *ColorSnakes* already covers up the true input by generating decoy paths. This partially explains why participants enjoyed using *ColorSnakes* in spite of the higher error rate and increased authentication time.

LIMITATIONS

For both studies, we used a self-recruited sample of participants living close to our location. In the real-world study, this allowed us to be able to closely monitor the participants and support them in case of a problem. In addition, the low number of attackers for the shoulder surfing experiment could have biased the results. However, since one attacker needed 8 hours to finish the whole sample, the recruitment of more than three attackers was not practicable. Furthermore, the modification of the regular four-digit PIN to a four-digit + color or five-digit + color PIN could have influenced the memorability. However, during the entire real-world study no participant forgot their PIN.

CONCLUSION AND FUTURE WORK

In this paper we introduced *ColorSnakes*, a novel concept enabling an authentication mechanism which is solely based on software modification and which provides protection against shoulder surfing and to some degree to video attacks. We investigated the influence of several factors on input time, error rate and shoulder surfing/video attack (n=24). The lab study revealed that using *Stroke Input* reduced the shoulder surfing attack down to 10.5% successful attempts without significantly increasing the authentication time and error rate. In a next step, we modified one promising combination (*Stroke Input*, *Path On*, *Stars Off*, *Grid 6x6*) to conduct a field study (n=12). We compared *ColorSnakes* to PIN and Pattern Unlock in a three week long real-world study and collected qualitative and quantitative data on the daily use of unlock patterns.

In spite of higher authentication times and error rates when using *ColorSnakes* in comparison to PIN and Pattern Unlock, 92% of the participants stated that even after having used the mechanism for 7 days they valued the added security *ColorSnakes* provided. Furthermore, 83% stated that they would start using the mechanism to secure sensitive data. This indicates that participants were willing to accept the higher authentication times and error rates to access sensitive data in a more secure and less socially awkward manner.

To collect more data, we are planning to offer *ColorSnakes* as an application on the Android Market. Additionally, we are planning to investigate the use of *ColorSnakes* in scenarios such as an ATM machine or debit payment to focus on application scenarios with a high risk of shoulder surfing/video attacks and a low frequency of daily use.

ACKNOWLEDGMENTS

This work was conducted within the projects "CompanionTechnology for Cognitive Technical Systems SFB/TRR 62" and "Mobile Interaction with Pervasive User Interfaces" both funded by the German Research Foundation (DFG).

REFERENCES

1. Angulo, J., and Wästlund, E. Exploring touch-screen biometrics for user identification on smart phones. In *Privacy and Identity Management for Life*. Springer, 2012, 130–143.
2. Aviv, A. J., Gibson, K., Mossop, E., Blaze, M., and Smith, J. M. Smudge attacks on smartphone touch screens. In *Proc. WOOT 2010*, USENIX Association (Berkeley, CA, USA, 2010), 1–7.
3. Bianchi, A., Oakley, I., Kostakos, V., and Kwon, D. S. The phone lock: Audio and haptic shoulder-surfing resistant pin entry methods for mobile devices. In *Proc. TEI 2011*, TEI '11, ACM (New York, NY, USA, 2011), 197–200.
4. Bianchi, A., Oakley, I., and Kwon, D. S. Counting clicks and beeps: Exploring numerosity based haptic and audio pin entry. *Interacting with Computers 24*, 5 (2012), 409–422.
5. Brooke, J. Sus-a quick and dirty usability scale. *Usability evaluation in industry 189* (1996), 194.
6. Burgbacher, U., and Hinrichs, K. An implicit author verification system for text messages based on gesture typing biometrics. In *Proc. CHI 2014*, ACM (New York, NY, USA, 2014), 2951–2954.
7. De Luca, A., Hang, A., Brudy, F., Lindner, C., and Hussmann, H. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *Proc. CHI 2012*, ACM (2012), 987–996.
8. De Luca, A., Harbach, M., von Zezschwitz, E., Maurer, M.-E., Slawik, B. E., Hussmann, H., and Smith, M. Now you see me, now you don't: Protecting smartphone authentication from shoulder surfers. In *Proc. CHI 2014*, ACM (New York, NY, USA, 2014), 2937–2946.
9. De Luca, A., von Zezschwitz, E., Nguyen, N. D. H., Maurer, M.-E., Rubegni, E., Scipioni, M. P., and Langheinrich, M. Back-of-device authentication on smartphones. In *Proc. CHI 2013*, ACM (New York, NY, USA, 2013), 2389–2398.
10. De Luca, A., von Zezschwitz, E., Pichler, L., and Hussmann, H. Using fake cursors to secure on-screen password entry. In *Proc. CHI 2013*, ACM (New York, NY, USA, 2013), 2399–2402.
11. Draffin, B., Zhu, J., and Zhang, J. Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction. In *Mobile Computing, Applications, and Services*. Springer, 2014, 184–201.
12. Findling, R. D., and Mayrhofer, R. Towards pan shot face unlock: Using biometric face information from different perspectives to unlock mobile devices. *International Journal of Pervasive Computing and Communications 9*, 3 (2013), 190–208.
13. Gafurov, D., Helkala, K., and Søndrol, T. Biometric gait authentication using accelerometer sensor. *Journal of computers 1*, 7 (2006), 51–59.
14. Harbach, M., von Zezschwitz, E., Fichtner, A., De Luca, A., and Smith, M. It's a hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *Proc. SOUPS 2014* (2014).
15. Karlson, A. K., Brush, A. B., and Schechter, S. Can i borrow your phone?: Understanding concerns when sharing mobile phones. In *Proc. CHI 2009*, 1647–1650.
16. Kim, S.-H., Kim, J.-W., Kim, S.-Y., and Cho, H.-G. A new shoulder-surfing resistant password for mobile environments. In *Proc. ICUIMC 2011*, ACM (New York, NY, USA, 2011), 27:1–27:8.
17. Kwon, T., and Na, S. Switchpin: Securing smartphone pin entry with switchable keypads. In *Proc. ICCE 2014*, IEEE (2014), 23–24.
18. Pons, A. P., and Polak, P. Understanding user perspectives on biometric technology. *Commun. ACM 51*, 9 (Sept. 2008), 115–118.
19. Tari, F., Ozok, A., and Holden, S. H. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proc. SOUPS 2006*, ACM (2006), 56–66.
20. Tukey, J. W. Exploratory data analysis. Addison-Wesley, Reading, Mass. (1977).
21. van Eekelen, W. A., van den Elst, J., and Khan, V.-J. Picassopass: A password scheme using a dynamically layered combination of graphical elements. In *EA CHI '13*, ACM (New York, NY, USA, 2013), 1857–1862.
22. von Zezschwitz, E., Dunphy, P., and De Luca, A. 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, ACM (New York, NY, USA, 2013), 261–270.
23. Watanabe, K., Higuchi, F., Inami, M., and Igarashi, T. Cursorcamouflage: Multiple dummy cursors as a defense against shoulder surfing. In *SIGGRAPH Asia 2012 Emerging Technologies*, SA '12, ACM (2012), 6:1–6:2.
24. Yue, Q., Ling, Z., Liu, B., Fu, X., and Zhao, W. Blind recognition of touched keys: Attack and countermeasures. *arXiv preprint arXiv:1403.4829* (2014).