Making Graphic-Based Authentication Secure against Smudge Attacks

Emanuel von Zezschwitz, Anton Koslow, Alexander De Luca, Heinrich Hussmann
Media Informatics Group, University of Munich (LMU), Amalienstr. 17, 80333 Munich, Germany
{emanuel.von.zezschwitz, alexander.de.luca, hussmann}@ifi.lmu.de, koslow@cip.ifi.lmu.de

ABSTRACT

Most of today's smartphones and tablet computers feature touchscreens as the main way of interaction. By using these touchscreens, oily residues of the users' fingers, smudge, remain on the device's display. As this smudge can be used to deduce formerly entered data, authentication tokens are jeopardized. Most notably, grid-based authentication methods, like the Android pattern scheme are prone to such attacks.

Based on a thorough development process using low fidelity and high fidelity prototyping, we designed three graphic-based authentication methods in a way to leave smudge traces, which are not easy to interpret. We present one grid-based and two randomized graphical approaches and report on two user studies that we performed to prove the feasibility of these concepts. The authentication schemes were compared to the widely used Android pattern authentication and analyzed in terms of performance, usability and security. The results indicate that our concepts are significantly more secure against smudge attacks while keeping high input speed.

Author Keywords
Mobile; Security; Authentication; Smudge; Attacks.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

General Terms
Human Factors; Security; Design.

INTRODUCTION

The smartphone became a ubiquitous device of the users' daily life. In connection with mobile internet, it empowers its owner to check emails, surf the web or perform financial transactions in almost every situation. As a consequence, various sensitive data is either stored on the device or can be accessed with it [11]. Thus, the access to the mobile device has to be secured and user authentication nowadays is an indispensable part of mobile interaction. Most deployed authentication methods are based on challenge and response, where a secret token (e.g. PIN) has to be entered to authenticate. As mobile devices are often used in public settings, the input can easily be observed by an attacker and the user's password is exposed. Such direct observational attacks are called shoulder surfing. They have been an object of research for many years (e.g. [5, 4, 9, 14]).

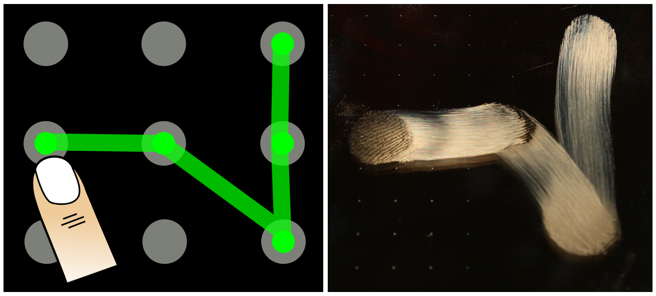

Figure 1. The Android pattern "3 6 9 5 4" (left) and the corresponding smudge (right). The oily residues expose not only the performed pattern, but also the direction in which it was drawn.

In addition to these well-known direct observational attacks, a new security threat was recently discovered [3], which is based on the fact that most smartphones and tablet computers use touchscreens as the main way of user interaction. Every time the user touches the screen, oily residues remain on the device's display. Figure 1 gives an example. When in possession of the device, an attacker can use these residues, called smudge, to deduce the owner's password, even if the input task was not directly observed. This approach is called a "smudge attack".

Graphical authentication methods like Android patterns, which were introduced by Google in 2010[1], seem to be prone to such attacks. At the same time, graphical authentication has advantages compared to alphanumeric methods (e.g. passwords cannot be based on user data) and some approaches exploit the human motor memory [15] or the pictorial superiority effect [12, 16]. In addition, using graphical approaches seems to be more joyful for some users.

The goal of this work was to find usable graphic-based methods, which are particularly secure against smudge attacks. In this paper, we present the design process of such systems and report on two user studies that we performed to prove the feasibility of our concepts.

IUI'13, March 19–22, 2013, Santa Monica, CA, USA.
Copyright 2013 ACM 978-1-4503-1965-2/13/03...$15.00.

[1] Android 2.2 platform highlights: `http://developer.android.com/about/versions/android-2.2-highlights.html`, last accessed: 01/07/2013

RELATED WORK

In general, graphical authentication can be categorized into searchmetric, locimetric and drawmetric systems [13].

Searchmetric systems (e.g. [6, 7]) require the user to identify predefined items from a set of randomly chosen images. While there are concepts, which are secure against shoulder surfing (e.g. [19]), such systems were not yet analyzed in terms of smudge attack vulnerability. However, as most concepts present their challenging sets in a randomized order, we assume that this approach is fairly secure.

Locimetric systems like Passpoints [18] require the user to select specific predefined positions within a picture. When used on touchscreens, these systems seem to be very prone to smudge attacks as oily residues do directly correspond to the selected regions of the image. In addition, it is difficult to select specific regions on a small mobile display and thus this approach seems to be inappropriate for mobile authentication.

Drawmetric systems like Draw-a-Secret [10] and Passdoodles [17] require the user to draw a specific shape, which was defined during enrollment. As Google's pattern concept fits in this category, drawmetric methods seem to be most relevant for graphical authentication on mobile devices. We argue that Android patterns are a restricted and thus usability-optimized version of the Draw-a-Secret authentication. Instead of relying on freely drawn shapes, Android patterns are drawn along a maximum of nine possible dots, which are arranged in a $3*3$ matrix. Aviv et al. [3] analyzed Android patterns in terms of smudge attack vulnerability and revealed that passwords are very often exposed by such an attack. With camera based smudge attacks, 68% of the patterns could be identified under perfect conditions and even after a simulated usage most of the passwords were exposed.

One possible solution to the problem of smudge attacks is the introduction of an additional security layer. De Luca et al. [8] add an implicit authentication layer to an Android like pattern authentication. As a consequence, access is only granted, when the right pattern is entered and the way the authentication is performed matches the stored attributes of the user (e.g. same speed, pressure, etcetera). Thus, such an approach can prevent successful smudge attacks as the right pattern does not necessarily give access to the device. Other approaches add an additional task to the basic authentication. For example, WhisperCore [2] requires the user to wipe the screen after the pattern was entered. By wiping the screen additional smudge is added and the residues of the pattern cannot easily be identified anymore. The drawback of this approach is based on the fact that authentication is never the user's primary goal [1]. Thus adding an additional task to the already cumbersome authentication might lead to frustrated users.

In this paper, we propose three novel input mechanisms, which were developed based on a thorough design process. One system belongs to the drawmetric category and is based on Google's pattern scheme. Instead of adding an additional minimal task or implementing an additional security layer, we add randomization to defend smudge attacks. The other two systems are searchmetric-like systems, but are token-based instead of relying on images. The concepts were evaluated in two user studies using low fidelity and high fidelity prototypes. Using the Android pattern authentication as baseline, we gathered comparable data in terms of security, performance and usability. The security analysis was based on the approach of Aviv et al. [3] and confirmed that Android patterns are very prone to smudge attacks. At the same time, the results indicate that our concepts are significantly more secure against smudge attacks than Android patterns and that authentication speed and error rates are kept in a good range.

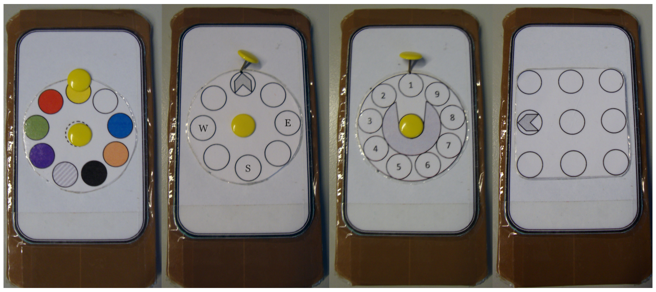

Figure 2. The paper prototypes of the four candidate concepts: *marbles*, *compass*, *dial* and *pattern rotation* (from left to right). Interactivity was achieved using movable elements.

THREAT MODEL

During everyday usage, interaction with mobile devices is far beyond authentication. As authentication takes place multiple times a day and the phone is primarily used for other tasks, there are several residues on the display distracting the deduction of the password. In addition, smudge attacks are hard to accomplish without being in possession of the device and without specific lighting conditions [3].

In this work, we intend to evaluate the security of our approaches in a worst case scenario. Therefore, we assume that *a*) the attacker is in possession of the device and *b*) does have perfect lighting condition as well as a camera to perform the attack. In addition, we assume that *c*) the touchscreen was cleaned before the authentication took place and *d*) the user authenticates only once, before the smudge attack is performed. Thus, there is no smudge on the display apart from the user password. We argue that this is the perfect condition to perform a smudge attack and therefore, the worst case in terms of security.

PAPER PROTOTYPING PRE-STUDY

Based on two brainstorming sessions and the analysis of existing systems, four candidate concepts were designed. We firstly realized them as low-fidelity paper prototypes (see figure 2) and evaluated them in a first user study.

Candidate Concepts

The proposed input mechanisms were explicitly designed in a way to leave smudge traces, which are not easy to interpret. This was achieved due to *a*) randomized distribution of the security tokens, *b*) consecutive blurring of the residues within one authentication or *c*) rotating the view port and thus hamper the deduction of the underlying values.

Marbles

This concept (figure 2, left) is based on the randomized distribution of nine colored "*marbles*". A password consists of up to nine different colors, multiple usage of the same color within one password is allowed. To authenticate, the user is required to drag the colored marbles in the right order into the center of the screen. Each input consists of one distinct dragging operation. The marbles are randomly arranged each time the user authenticates.

Compass

Compass is a drawmetric concept (figure 2, second from left). Instead of a $3 * 3$ matrix known from Android patterns, the dots are arranged in circular order. This order is the same for every authentication. Smudge attack resistance is achieved by randomly rotating the circle of dots. The current orientation is indicated by an arrow and the initials of three cardinal points. A password consists of up to eight connected dots, while each dot can be visited only once. To authenticate, a password is drawn by connecting the dots in the right order.

Dial

This concept (figure 2, second from right) is based on consecutive blurring of residues. A password consists of an arbitrary sequence of the digits one to nine. To authenticate, the user is required to drag the digits in the right order into the center of the screen. The approach uses the metaphor of a dial plate and thus, the numbers have to be dragged within the black margins. To reach the center of the screen, the white opening under the digit "one" has to be used. Thus, the same path is used multiple times within one authentication. While the numbers stay in constant order, the dial (view-port) is randomly rotated within a range of 45°.

Pattern Rotation

This concept (figure 2, right) is based on the Android pattern authentication. Thus, a pattern connects up to nine dots, which are arranged in a $3 * 3$ matrix. To authenticate, the user is required to redraw the pattern. The only modification compared to the original approach is that the matrix is randomly orientated on the screen and the current direction is indicated by an arrow. Two different versions were tested: a 90° version with four different directions and a 360° version allowing an arbitrary orientation on the screen.

Paper Prototype User Study

To gain insights into the feasibility of the concepts and to gather design implications for the final systems, a user study based on the paper prototypes was conducted.

User Study Design

We used a repeated measure within participants design. The independent variables were *authentication system* with five levels (*marbles*, *dial*, *compass*, *pattern 90*, *pattern 360*) and *password origin* with two levels (*given*, *self-selected*).

Authentication system was counterbalanced based on a Latin square design, *password origin* was alternated. Qualitative data was collected via a questionnaire and via video recording.

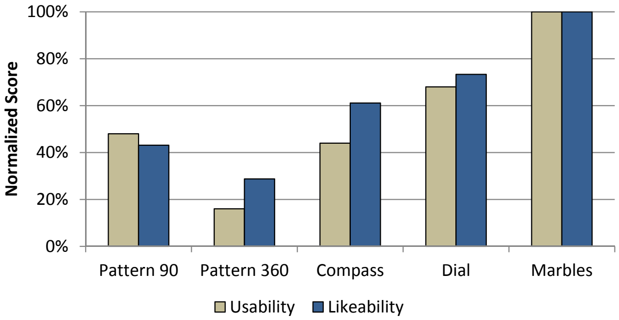

Figure 3. The normalized usability and likeability ranks. *Marbles* scores best in both categories, while *pattern 360* is rated worst.

Procedure

Each user had to use each prototype to create a password and to authenticate using a given password. Each test case started with a short training task and ended with a questionnaire about the usability, likeability and the perceived security of the tested approach. Interactivity of the paper prototypes (e.g. rotation) was simulated by the examiner. The sessions were filmed to get further insights into user behavior and interaction problems. At the end of the session a questionnaire was used to compare and rank the concepts.

Participants

The systems were tested by twelve experienced smartphone users, whose mean age was 22 (19-26) years. Seven participants were female, five male. Ten users stated to use lock screens on their phone, nine of them used Google's pattern authentication, one used PIN. Three (25%) participants stated to be familiar with smudge attacks.

Preliminary Results

The results are based on 6-point Likert scales and user rankings. The Likert scale based questions were answered after each concept was tested, while the ranking was performed after all concepts were used.

Likeability

The likeability ratings are shown in figure 3. The participants ranked the systems according to their willingness to use them on a daily base. The results were normalized for better comparison. Most users would use *marbles*, while both *pattern rotation* approaches scored worst.

In addition, we asked users after each test run if they would use the respective system. The results support the data of the likeability ranking as only three (25%) users would use *pattern 360* and four (33%) would use the *pattern 90* system. The *compass* would be used by eight (66%) participants, nine (75%) users stated, they would like to use the *dial* authentication and eleven (92%) users would like to use *marbles*.

Usability

Usability was subdivided into the two aspects "easy to use" and "easy to understand". Figures 4 and 5 present the respective results. Data was collected using Likert scales ranking from "very good" to "bad".

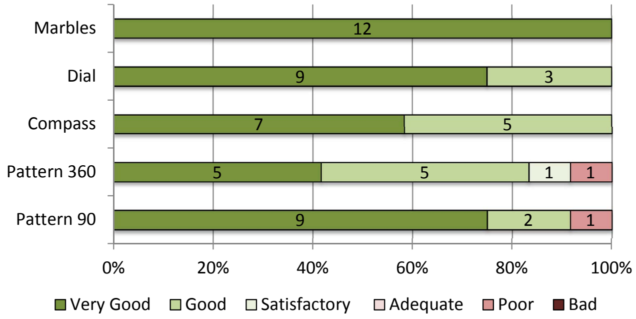

Figure 4. The comprehensibility ratings of the five concepts. *Marbles* is rated best as all participants attested very good understandability.

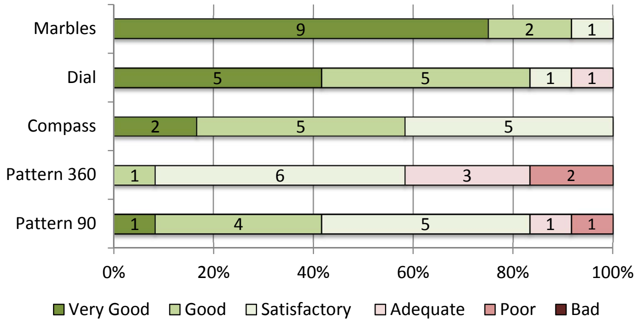

Figure 5. The usability ratings of the five concepts. *Marbles* scores best, while *pattern 360* led to most interaction problems.

Based on the median values, the understandability of all systems is rated "good" or "very good". However, the usability ratings are more diverse. Only *marbles* was attested "very good" usability. According to the participants, the *dial* and the *compass* system still provide "good" usability. Both *pattern rotation* approaches seem to be more difficult to use as usability was rated "satisfactory". Users reported that these systems demand elevated concentration. Similar statements were given according to the *compass* approach. In contrast to the in-between ratings, the *pattern 90* system was preferred to the *compass* approach in the final rankings (see figure 3).

FINAL CONCEPTS AND PROTOTYPES

Based on the results of the first user study, we decided to further analyze the feasibility of the *marbles* authentication approach. In addition, we chose the *pattern 90* approach to be further evaluated. Even if it scored worse than the *dial* system in terms of usability and likeability, we preferred this approach for two reasons: *1*) we were interested in, what impact view port rotation really has on the usability and security aspects, when compared to the Android approach. *2*) We wanted to provide a drawmetric based solution as this category is most widely used on mobile devices and thus, smudge attack protection is very relevant.

In addition to these two systems, a third system was developed during the second design phase. The *marble gap* (see figure 6, right), which is a modification of the *marbles* approach tested in the pre-study. The *marble gap* restricts reuse of identically colored marbles and has therefore the potential to positively influence the creation of more secure passwords. These three novel concepts and Google's pattern approach (baseline) were implemented in high fidelity prototypes using the Android SDK. In the following, the four concepts are presented in detail.

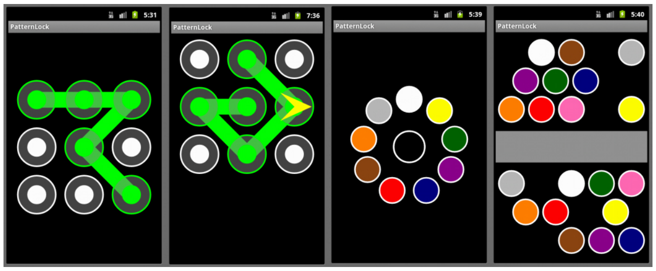

Figure 6. The four prototypes of the user study: *Android pattern* (baseline), *pattern 90*, *marbles* and *marble gap* (from left to right).

Android Pattern

The *Android pattern* approach was implemented according to the Google standards. A password is based on a pattern, which connects a maximum of nine dots. It is not allowed to visit the same dot multiple times. Due to Android specific restrictions, the theoretical password space of this approach is $389,112$ [3]. Figure 6 shows the used prototype on the most left picture.

Pattern 90

The *pattern 90* approach works analog to *Android patterns*. To prevent smudge attacks, the view port is randomly assigned and the position on the screen is alternated. The theoretical password space matches the one of *Android patterns*. The example of figure 6 shows a matrix, which was positioned on top of the screen and clockwise rotated by 90°. The underlying password of the example is "*4 2 6 8 5*".

Marbles

As no relevant shortcomings were detected, the marbles approach was implemented the same way it was tested in the pre-study. A password consists of an arbitrary sequence of marbles (colors). After a marble is dragged in the center, it immediately reappears on its prior position. The positions are kept during one authentication but rearranged at the beginning of each new attempt. The theoretical password space of this approach has no upper bound. The third image from the left of figure 6 shows the interface of the used prototype.

Marble Gap

The *marble gap* is a modification of the *marbles* approach. The most right image of figure 6 shows the interface of this concept. In contrast to the circular arrangement of the *marbles* approach, the user interface of *marble gap* is divided into three sections. The top and the bottom section present the marbles in a randomized order. The section in the center is called gap.

To authenticate, users have to drag the marbles in the right order anywhere into the gap. Marbles with the same colors do have the same values and thus it does not matter for the password, if the intended color is chosen from the top or from the bottom of the screen. After a marble was dragged into the gap, it disappears from the screen and cannot be used anymore. The prototype used in the user study featured ten different colors of which each was present two times. Thus, user passwords could comprise the same color two times maximum and the password length was restricted to 20 characters (theoretical password space $\approx 13.17 * 10^{12}$).

Another interesting aspect, which might lead to more secure passwords, is based on the fact that marbles of the same color are equally distributed over the top and the bottom section. When choosing the same color twice, one has to drag elements from the bottom and the top; in contrast, when choosing different colors one can drag marbles from one side only. This interplay of password complexity and input effort might lead to the selection of passwords, which are based on more different colors. As a consequence, high entropy in user chosen passwords is supported.

USABILITY AND SECURITY USER STUDY

The final concepts were evaluated in a user study to get insights into their usability, performance and security. In this section, we provide details on the design and the conduction of the study and report on the results.

User Study Design

The user study was based on a repeated measures factorial design. The independent variables were *authentication concept* with four levels (*Android pattern, pattern 90, marbles, marble gap*), *password origin* with two levels (*given, self-selected*) and *run* with three levels (*one, two, three*). The order of *authentication concept* was counterbalanced using a Latin square, *password origin* was alternated and *run* represented successful authentications.

Quantitative data was collected via logging mechanisms, qualitative data was collected with a questionnaire. For further analysis, the study was filmed and pictures of the device's touchscreen were taken after successful authentications. As an incentive, 10 Euro vouchers were handed out to the participants.

Experimental Setup

The prototypes were tested on an HTC Google Nexus One device. The smudge attack setup consisted of a camera (Canon EOS 1000D) and a strong light source (Arri 650W spot light) from above. The device was placed in front of the camera using a paperboard platform. The distance to the camera was approximately 20 cm, the angle between object lens and touchscreen was 60°. The setup can be seen in figure 7, it was not modified during the whole study. In addition to the smudge attack setup, the sessions were filmed using a Canon FS 306 camcorder, which was placed behind the participant and targeted on the touchscreen of the device. The web-based questionnaire was filled out using a laptop computer.

Passwords and Procedure

This section describes the guidelines of the passwords, which were used to test the described concepts. In addition, the procedure of the user study is explained in detail.

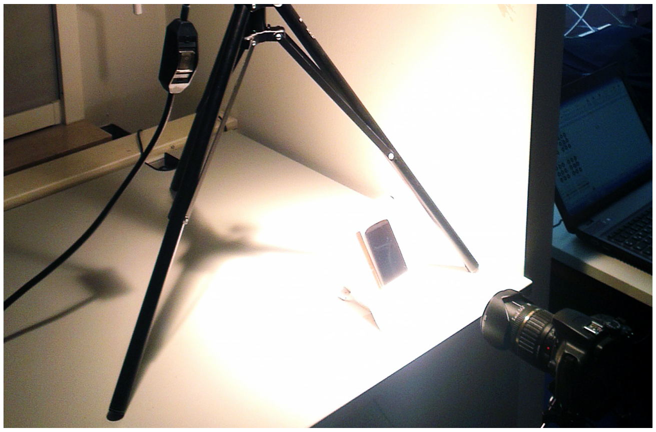

Figure 7. To perform smudge attacks, an HTC Nexus One was placed under an Arri 650W spot light and in front of a Canon EOS 1000D. Under such overexposed conditions, even minor residues become visible.

Passwords

The results of the pre-study revealed that the length of five was most often chosen for the approaches *pattern 90* and *marbles* and that theoretical security is comparable at this length. Therefore, we decided to restrict the password length for all tested systems to the length of five distinctive tokens. The given passwords of *marbles* and the *marble gap* approach were composed with a maximum of one repeated color. The passwords for the pattern approaches were composed with diverse, but counterbalanced input complexity. Half or the patterns were exclusively based on directly adjacent dots, while the other half comprised larger distances between dots. User generated passwords were also restricted to the length of five, but we did not restrict the repetition of same tokens to get insights into the password composition behavior with the respective system.

Based on five activated dots, the theoretical password space of the *Android pattern* and the *pattern 90* approach is 10,672. This is influenced by Android specific restrictions. For example, it is not possible to activate a dot, which is not a direct neighbor without activating the direct neighbor, if all three dots are in line. Given that repeated colors are allowed, the password space of *marbles* with five tokens is 59,049 (15,120 without repeated colors). As the *marble gab* allows each color to be chosen twice at maximum, the theoretical password space is 64,800 (30,240 without repeated colors).

Procedure

After an introduction to smudge attacks, the tasks of the user study were explained to the participant. To preserve privacy, each user was assigned an ID. Based on this ID, the order of *authentication concept* and *password origin* was defined. Each concept was tested twice (alternating *password origin*) using the following procedure: *1*) During a training phase, the user tried out the respective system until she stated to understand the approach. *2*) After the training, the touchscreen was cleaned using a microfiber cloth and the device was given back to the participant. *3*) The user entered the respective password. *4*) If the authentication was correct, a photo of the touchscreen was taken using the described setup. If the au-

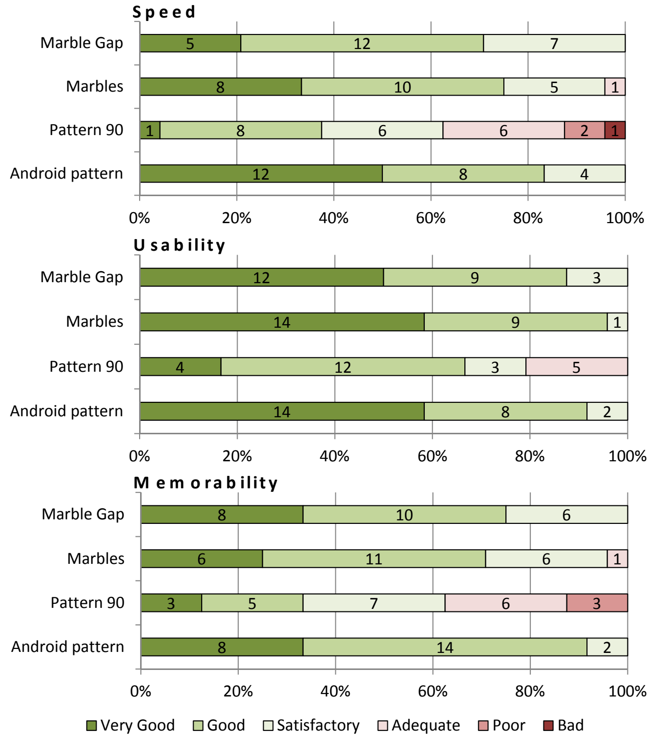

Figure 8. The qualitative results of the Likert scale analysis. While *pattern 90* is rated worst in terms of usability, authentication speed and memorability, both *marble* based approaches are rated comparable to the *Android pattern*.

thentication failed, the process started over with step *2*. *5*) After the picture was taken, the user had to successfully authenticate two more times. Failed authentications were repeated. While the user had the possibility to look up the password for the first and the second authentication, the third authentication had to be performed without such memory aid.

After all concepts were tested, the user filled out a questionnaire. The questionnaire collected data about the usability and performance of the concepts and demographical data of the user. In the end of the study, a 10 Euro voucher for an online shop was handed out to the participant.

Participants

24 participants took part in the user study. The mean age was 25 years (19-33). Eight users were female, 16 users were male. They were experienced touchscreen users as 23 (96%) used a smartphone on a daily base. However, only 13 (54%) users knew about smudge attacks. Seven users used patterns, six used PIN to protect their device. The rest of the group did not use secure lock screens.

USER STUDY RESULTS

In this section, we report on the results of the user study. The results are based on quantitative data of the log files and qualitative data of the questionnaire as well as photos and videos.

Authentication Speed

Authentication speed is distinguished in orientation time and input time. The orientation time is the time span between the start of authentication and the user's first touch event. Thus, it represents the time a user needs to orient herself. With the first touch event, the input time starts. The input time ends when the password is confirmed or the authentication is cancelled. It is very important to analyze the orientation time as the data input only represents one aspect of the whole authentication process.

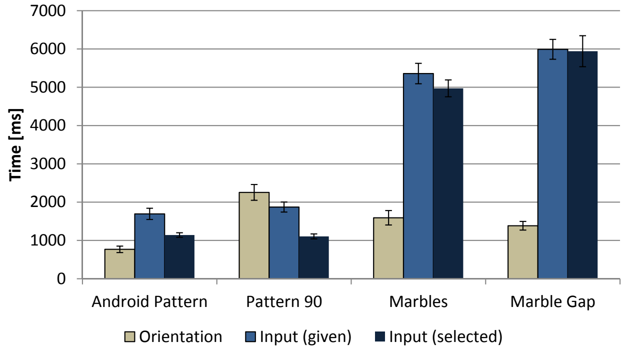

Figure 9. The average orientation time and the average input time of all systems. While input times of the *marble* approaches are significantly higher than the input times of the pattern systems, users of the *pattern 90* system need more time for orienting themselves, than for entering the password.

Data

Authentication time was analyzed based on successfully completed attempts. Input times are therefore based on 576 samples (*24 users* $*$ *2 password origins* $*$ *4 authentication concepts* $*$ *3 runs*). Due to an error in the log files, only the third authentication attempt (*run*) of each user could be used as a basis for the orientation time. Thus, the analysis of orientation time is based on a smaller set of 192 samples (*24 users* $*$ *2 password origins* $*$ *4 authentication concepts* $*$ *1 run*). However, the last run seems to be best suited for this analysis as users were already trained using the specific password two times. We excluded outliers in both data sets using the doubled standard deviation as an upper and lower boundary.

Results

We performed a repeated measures ANOVA of the orientation and input times to analyze the effects of *authentication concept*, *password origin* and *run*. The results are shown in figure 9. For better comparison, the input times of figure 9 were also based on the third run.

According to the orientation times, no significant main effects were found for *password origin* ($p > .05$). In contrast, there are highly significant main effects of *authentication concept* on the orientation time ($F_{2.1,48.4} = 16.64, p < .001$, Greenhouse-Geisser corrected: $\epsilon = .69$). Post-hoc tests reveal that users needed significantly ($p < .001$ for all contrasts) more time using the *pattern 90* ($Mn = 2254ms, SE = 206$), *marbles* ($Mn = 1592ms, SE = 188$) and *marble gab* ($Mn = 1383ms, SE = 113$) systems than using the *Android pattern* approach ($Mn = 768ms, SE = 84$). Furthermore, the *pattern 90* approach led to the highest orientation times and performed significantly worse than *marbles* and *marble gap* ($p < .05$).

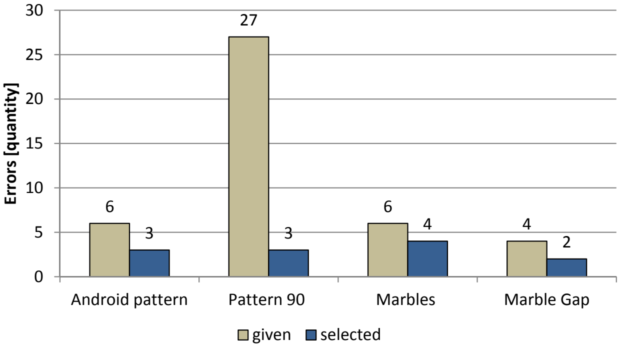

Figure 10. The amount of failed authentications. Given passwords led to significantly more errors than self-selected tokens. The *pattern 90* approach was most affected by errors.

In terms of the input time, highly significant main effects were found for *authentication concept* ($F_{2.1,41.6} = 315.32, p < .001$, Greenhouse-Geisser corrected: $\epsilon = .69$) and *password origin* ($F_{1.0,20.0} = 27.25, p < .001$). No significant main effects of *run* were found ($p > .05$). Post-hoc tests reveal that *marbles* ($Mn = 5233ms, SE = 199$) and *marble gap* ($Mn = 5982ms, SE = 261$) have significantly higher input times ($p < .001$) than *Android pattern* ($Mn = 1611ms, SE = 111$) and *pattern 90* ($Mn = 1664ms, SE = 97$). The *password origin* does not significantly influence these contrasts ($p > .05$). In addition, the input times of marble-based and pattern-based approaches are not significantly different, when compared to each other ($p > .05$). This is true for given and self-selected passwords.

However, the *password origin* has a highly significant effect on the input performance within both pattern-based concepts ($p < .001$). The participants entered passwords significantly faster, when they were self-selected. Both marble-based concepts were not affected by *password origin*.

Interestingly, the participants' perception differs from the results of the quantitative data. Based on the median values, users ranked the *Android pattern* system to be "very fast", both *marble* approaches "fast" and the *pattern 90* approach to be the slowest ("satisfactory") (see figure 8, speed). The same order was found in the final rankings, even though *pattern 90* actually performed second best in terms of authentication speed.

Error Rate and Usability

Failed authentications are distinguished in simple and critical errors. Simple errors are based on authentication sessions, which fail one or two times. Productive systems (e.g. ATM) most often allow a maximum of three attempts to authenticate, before user accounts are locked. Therefore, if an authentication fails three consecutive times, it is interpreted as a critical error.

Data

The data is based on 192 authentication sessions. Each session was finished after three successful attempts. In addition, qualitative data of the questionnaire is analyzed to get further insights into the usability and memorability of the systems.

Results

In summary, 55 authentication attempts failed, thus the error rate based on all authentications was 9.5%. Only one critical error was logged. This occurred in the third run of using the *pattern 90* system with a given password.

We performed a repeated measures ANOVA to analyze the influence of *authentication concept*, *password origin* and *run* on failed authentication attempts. Significant main effects were found for *authentication concept* ($F_{3.0,40.0} = 5.99, p < .05$, Greenhouse-Geisser corrected: $\epsilon = .58$) and for *password origin* ($F_{1.0,23.0} = 8.15, p < .05$). No significant main effect was found for *run* ($p > .05$). In addition, a significant interaction effect was found for *authentication concept* $*$ *password origin* $*$ *run*, ($F_{3.5,80.0} = 2.99, p < .05$, Greenhouse-Geisser corrected: $\epsilon = .58$). Figure 10 shows the amount of failed authentications based on *authentication concept* and *password origin*.

Post-hoc tests regarding the *authentication concept* reveal that significantly more errors were made using the *pattern 90* ($n = 30$) system than using any of the other system ($p < .05$ for all contrasts). Fewest authentications failed with the *marble gap* ($n = 6$). Using the *Android pattern*, nine authentications failed, using *marbles* ten errors were logged.

The contrast of self-selected and predetermined passwords reveals that significantly more errors were made, when the password was given ($p < .05$). 78% ($n = 43$) of all failed authentications were based on given passwords. This is mainly caused by the *pattern 90* approach, where 90% ($n = 27$) of all errors were based on given patterns.

The users' perception supports the quantitative data. Our users stated that the rotation of the *pattern 90* approach was "cumbersome" and "demanded high cognitive load". Our participants' comments reveal that most errors occurred, because the users were confused by the orientation and draw their patterns in the wrong direction. The median values of the usability ranking reflect these problems as *pattern 90* was rated "satisfactory", while all other systems were rated "good" (see figure 8). In addition, our users confirmed that self-selected passwords were easier to use on all systems. While *marble gap*, *marbles* and *Android patterns* were rated "very easy", when used with self-selected passwords, the rating dropped to "easy", when given passwords had to be used. According to the *pattern 90*, users rated the difficulty "satisfactory", when used with self-selected passwords and "poor", when used with given passwords. In the final ranking, *marble gap* was voted the easiest system; *Android pattern* was second, *marbles* third. *Pattern 90* was confirmed to be the most difficult approach.

As memorability is hard to measure in a lab experiment, we have to limit the analysis on the users' perception. In figure 8, the memorability ratings based on 6-point Likert scales are shown. The memorability of the *Android pattern* system was rated best. Based on the median, this aspect was rated "very good", when the password was self-selected and "good", when passwords were given. Both marble-based approaches were rated "good" independently from *password*

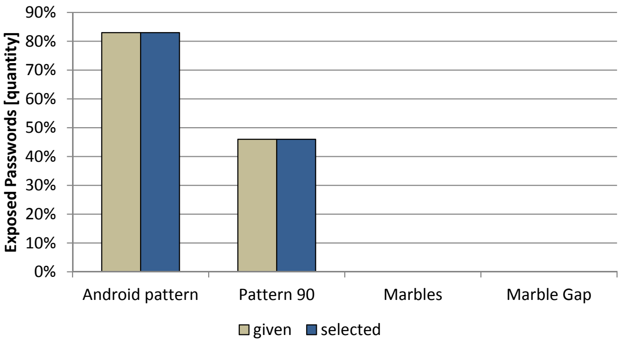

Figure 11. The percentage of exposed passwords. No passwords could be deduced from *marble*-based approaches. The *Android* approach is most prone to smudge attacks, but *pattern 90* is also vulnerable to smudge attacks, especially when the attacker has multiple guesses.

origin. However, memorability was rated slightly better, when self-selected passwords were used and the memorability of *marble gap* seems to be better, than the memorability of *marbles*. *Pattern 90* was rated worst according to this aspect. User stated that memorability was good with self-selected patterns and satisfactory with given patterns.

Security Evaluation

Since the focus of this work is smudge attack secure systems, the security evaluation is restricted to such an attack. We like to point out that there are other security threats like shoulder surfing, which are out of the scope of this work.

Data

Smudge attacks were performed based on the pictures taken during the user study. For each authentication system, one image of the self-selected password and one image of the given password were analyzed. Thus, the results are based on 192 samples. The pictures were not edited (e.g. adjusting contrast) to obtain the same conditions for each participant. Figure 12 shows one example of each concept. It has to be mentioned, that these images show optimal results and there have also been images, which had less oily residues on the screen.

The smudge attacks were performed by a security expert, who was highly familiar with the authentication systems, but had no knowledge about the used passwords. Before each attack, he was informed about the current system and the *password origin* (self-selected, given). Each attack consisted of a maximum of three guesses. The attacker was allowed to zoom and rotate the images, but no other transformations were used.

Results

Figure 11 presents the quantitative results of the security evaluation. The *Android pattern* approach is most vulnerable to smudge attacks. 20 (83%) self-selected and 20 (83%) given passwords were exposed. Eight smudge attacks failed, because none or too little smudge was visible on the display. As one can see in figure 12, left, the oily residues on the touchscreen are clearly exposing the set of activated dots *1 4 5 6 7*. In theory, there are two patterns matching this token set, but due to overlapping smudge, the right order *7 4 1 5 6* can be deduced as well. In detail, 60% of these patterns were exposed during the first attack and the rest was found by a second guess. A third try did not improve the attacker's success rate. This indicates that, whenever enough smudge is visible on the screen and the attacker has multiple guesses, the right password can be deduced. Only two participants left almost no residues on the touchscreen and thus neither the self-selected nor the given pattern was exposed. With the *pattern 90* system, eleven (46%) self-selected and eleven (46%) given passwords could be derived. The smudge patterns of this approach (figure 12, second from the left) can be interpreted in four different directions. Since the attacker had three guesses per image, the chance to find the right pattern was 75%, whenever enough smudge was on the touchscreen. The analysis of single attempts supports this assumption as 36% of the exposed patterns were found in a first guess and another 32% were exposed during the second attack. In contrast to *Android patterns*, the third attempt further improved the success rate as 32% of the patterns were derived at this point. Another aspect, which makes *pattern 90* vulnerable, is due to the fact that in some cases additional smudge allows guessing how the device was grasped, which gives an additional cue, in what direction the pattern was entered.

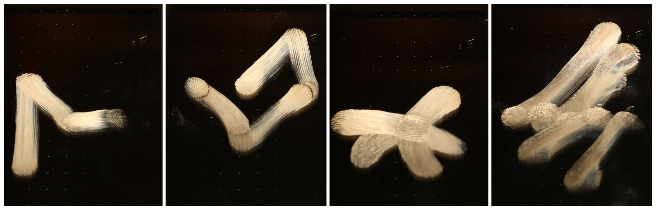

Figure 12. Examples of the pictures used for the smudge attacks. *Android pattern*, *pattern 90*, *marbles* and *marble gap* (from left to right). The images are not edited (except cropping).

Both marble-based approaches are very secure against smudge attacks. Based on the randomized order of the security tokens, the oily residues do not give enough information to deduce the password and thus no password was exposed. However, both approaches allow deriving some information about the composition of the password, which can restrict the password space for further attacks. Since the *marble gap* does not allow selecting marbles of one section multiple times, this information can be used whenever smudge is detected on only one of the two segments. The same is true for *marbles* providing that the attacker knows the length of the password and the amount of distinctive residues matches this value (vice versa repeated colors can be detected).

The analysis of the Likert scale based data reveals that the participants perceived the *Android pattern* as significantly less secure than the other three approaches. The security of the *Android patterns* system was rated "adequate", while the security of *pattern 90*, *marbles* and *marble gap* was rated "good". In the final rankings, *marble gap* was rated most secure, *marbles* was second, *pattern 90* was third and the *Android pattern* approach was ranked last place. This ranking matches the results of the quantitative analysis and indicates that users have a good understanding of security.

User Acceptance

To evaluate the user acceptance, users had to rank the systems according to likeability. In addition, we asked the participants if they would like to use the respective system on a daily base.

Android patterns were placed first according to likeability, the *marbles* approach was ranked second, *marble gap* was set on the third place. The *pattern 90* approach was the least favored concept. Correspondingly, most participants (n = 22, 92%) would use the Android approach on a daily base. The two participants, who refused using this concept, stated that it was not secure enough. Furthermore, 18 (75%) people would use *marbles* and 16 (67%) participants would use *marble gap* on a daily base. Criticism on these approaches was mostly related to the use of color coded tokens. People, who did not want to use these concepts, suggested the use of numbers or symbols instead. Only ten (42%) participants would use the *pattern 90* approach on a daily base. Most users, who did not want to use this approach, stated that entering the pattern needed too much spatial imagination and rotating the mobile device for each authentication was cumbersome.

DISCUSSION

The quantitative analysis of authentication speed reveals that *Android patterns* performed best in the study and thus, randomization does negatively influence the input time and the orientation time. However, marble-based approaches and the *pattern 90* system comprise different aspects of randomization and these aspects affected the authentication speed in different ways. The random orientation of the view-port mainly influences the time a user needs to orient herself, while the randomized arrangements of security tokens had more impact on the input time.

Connected to this, the conflict of the measured speed values and the users' perception is very interesting. While the *pattern 90* approach was actually the second fastest system, it was ranked to be the slowest. In contrast, both marble-based approaches, which actually performed significantly slower than the *pattern 90* approach, were rated faster. This result indicates that high orientation times were more annoying for users than high input times.

In addition, the *password origin* significantly influenced the input speed of both pattern-based approaches. When using these systems, users entered self-selected passwords significantly faster than given passwords. The input time of both marble-based approaches was not affected by this aspect. We assume that this is based on the fact that patterns provide a wider range of input complexity compared to marble-based passwords. When users had the chance to choose their passwords, they selected patterns, which were easier and thus faster to enter than given passwords.

The overall error rate was low, considering that users were not familiar with the systems and not used to the passwords. While both marble-based approaches performed well and error rates were comparable to the *Android patterns*, the *pattern 90* system led to significantly more failed authentications. 90% of these authentication errors using the *pattern 90* approach were based on given passwords. This supports the assumption that patterns cause more interaction problems, when input complexity is not controlled by the user and whenever it is possible, users tend to choose simple patterns. A complex system like *pattern 90* might reinforce this problem.

The security analysis reveals that *Android patterns* are highly vulnerable to smudge attacks and thus confirms the findings of Aviv et al. [3]. However, the risk of smudge attacks is individually diverse as some users did not leave enough residues on the display to be attacked. The *pattern 90* approach was significantly more secure against smudge attacks than *Android patterns*, but still exposed passwords. Thus, the additional benefits in terms of security seem not to justify the drawbacks in usability. The security of both marble-based approaches is excellent as not a single password could be deduced.

Taking all these aspects into account, the *pattern 90* approach seems not usable and not secure enough and thus has to be discarded. However, the analysis of this concept gave valuable insights into the interplay of authentication performance and user perception. In contrast, both marble-based approaches performed well in terms of usability and security. Therefore, we argue that these concepts are promising candidates for a smudge attack secure mobile authentication, which is usable and highly accepted by the users.

LIMITATIONS

Even if the study was carefully designed, there are some limitations concerning the experiment and the collected data, we would like to address.

Since we performed a short term lab study, we were not able to examine training effects and how our concepts affect memorability. All passwords were new to the users and therefore, we assume that performance would improve on all systems when they were used with well-known and memorized passwords. In addition, we assume that password composition was influenced by the fact that the participants did not have to protect real sensitive data. If the respective systems would be used in the wild, users would potentially create more secure passwords and the difference between given passwords and the user-selected ones would be smaller.

In terms of generalizability, we have to mention that the data is based on 24 users only and can therefore only give indications on the performance of the proposed systems. However, based on the thorough design of the study and since multiple samples were collected per user and per system, we argue that our data is valid.

CONCLUSION AND FUTURE WORK

In this paper, we analyzed the vulnerability of *Android patterns* to smudge attacks and proposed alternative more secure graphic-based authentication concepts. The concept development was based on a thorough design and evaluation process. In a first step, candidate concepts were found and paper prototypes were built and evaluated in a user study. In a second step, the results of the paper prototype analysis were used to develop the final concepts, which were implemented for Android smartphones and evaluated in terms of usability, performance and security. The results were set in relation to the *Android pattern* approach.

While one pattern-based concept (*pattern 90*) had to be discarded as the trade-off between usability and security was not sufficient, we found two promising token-based concepts. The systems are significantly more secure against smudge attacks than *Android patterns*. In addition, error rates were low and authentication speed was perceived comparably fast. The likeability scores of both systems support the assumption that our concepts are actual candidates for smudge attack secure and usable graphical authentication systems on mobile devices.

In addition, we presented general findings about user authentication. Firstly, the *marble gap* approach indicates the existence of an interplay of input complexity and password complexity. Based on this approach, the input complexity of stronger passwords seems lower than the input complexity of weak passwords. We argue that this aspect might be used to implicitly teach users to create stronger passwords. Secondly, the results of the distinct analysis of authentication speed indicate that high orientation times are more annoying than high input times.

In future work, we will evaluate the proposed concepts in a long-term field study to gather insights into training effects, password composition behavior and the memorability of such passwords. In addition, we plan to integrate shoulder surfing in our analysis and evaluate our concepts according to such attacks. Furthermore, we like to address the interplay of input complexity and password complexity in our upcoming research.

REFERENCES

1. Adams, A., and Sasse, M. Users are not the enemy. *Communications of the ACM 42*, 12 (1999), 40–46.
2. Airowaily, K., and Alrubaian, M. Oily residuals security threat on smart phones. In *Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on*, IEEE (2011), 300–302.
3. Aviv, A., Gibson, K., Mossop, E., Blaze, M., and Smith, J. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX conference on Offensive technologies*, USENIX Association (2010), 1–7.
4. Bianchi, A., Oakley, I., Kostakos, V., and Kwon, D. The phone lock: audio and haptic shoulder-surfing resistant pin entry methods for mobile devices. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, ACM (2011), 197–200.
5. Bianchi, A., Oakley, I., and Kwon, D. S. The secure haptic keypad: a tactile password system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, ACM (New York, NY, USA, 2010), 1089–1092.
6. Brostoff, S., and Sasse, M. Are passfaces more usable than passwords? a field trial investigation. *PEOPLE AND COMPUTERS* (2000), 405–424.
7. De Angeli, A., Coutts, M., Coventry, L., Johnson, G., Cameron, D., and Fischer, M. Vip: a visual approach to user authentication. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, ACM (2002), 316–323.
8. 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 *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, ACM (2012), 987–996.
9. De Luca, A., Von Zezschwitz, E., and Hußmann, H. Vibrapass: secure authentication based on shared lies. In *Proceedings of the 27th international conference on Human factors in computing systems*, ACM (2009), 913–916.
10. Jermyn, I., Mayer, A., Monrose, F., Reiter, M., Rubin, A., et al. The design and analysis of graphical passwords. In *Proceedings of the 8th USENIX Security Symposium*, Washington DC (1999), 1–14.
11. Karlson, A., Brush, A., and Schechter, S. Can i borrow your phone?: understanding concerns when sharing mobile phones. In *Proceedings of the 27th international conference on Human factors in computing systems*, ACM (2009), 1647–1650.
12. Madigan, S. Picture memory. *Imagery, memory and cognition* (1983), 65–89.
13. Renaud, K., and De Angeli, A. Visual passwords: cure-all or snake-oil? *Commun. ACM 52*, 12 (Dec. 2009), 135–140.
14. Roth, V., Richter, K., and Freidinger, R. A pin-entry method resilient against shoulder surfing. In *Proceedings of the 11th ACM conference on Computer and communications security*, ACM (2004), 236–245.
15. Shadmehr, R., and Brashers-Krug, T. Functional stages in the formation of human long-term motor memory. *The Journal of Neuroscience 17*, 1 (1997), 409–419.
16. Standing, L. Learning 10000 pictures. *The Quarterly journal of experimental psychology 25*, 2 (1973), 207–222.
17. Varenhorst, C., et al. Passdoodles: A lightweight authentication method. *Research Science Institute* (2004).
18. Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., and Memon, N. Passpoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies 63*, 1 (2005), 102–127.
19. Wiedenbeck, S., Waters, J., Sobrado, L., and Birget, J. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of the working conference on Advanced visual interfaces*, ACM (2006), 177–184.