SmudgeSafe: Geometric Image Transformations for Smudge-resistant User Authentication

Stefan Schneegass[1], Frank Steimle[1], Andreas Bulling[2], Florian Alt[3], Albrecht Schmidt[1]

[1]University of Stuttgart
HCI Group
first.last@vis.uni-stuttgart.de

[2]Max Planck Institute for Informatics
Perceptual User Interfaces Group
bulling@mpi-inf.mpg.de

[3]University of Munich
Media Informatics Group
florian.alt@ifi.lmu.de

ABSTRACT

Touch-enabled user interfaces have become ubiquitous, such as on ATMs or portable devices. At the same time, authentication using touch input is problematic, since finger smudge traces may allow attackers to reconstruct passwords. We present SmudgeSafe, an authentication system that uses random geometric image transformations, such as translation, rotation, scaling, shearing, and flipping, to increase the security of cued-recall graphical passwords. We describe the design space of these transformations and report on two user studies: A lab-based security study involving 20 participants in attacking user-defined passwords, using high quality pictures of real smudge traces captured on a mobile phone display; and an in-the-field usability study with 374 participants who generated more than 130,000 logins on a mobile phone implementation of SmudgeSafe. Results show that SmudgeSafe significantly increases security compared to authentication schemes based on PINs and lock patterns, and exhibits very high learnability, efficiency, and memorability.

Author Keywords

Graphical passwords; Touch input; Finger smudge traces

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Input devices and strategies*; K.6.5 Computing Milieux: Security and Protection—*Authentication*

INTRODUCTION

Touch-enabled interfaces have become common on mobile phones and tablets, ATMs, or ticket machines and we use these interfaces on a regular basis in our daily life. These devices store and provide personal data that needs to be protected from unauthorized access, such as bank account details, emails, or contact lists. While communication between devices can be secured, for example through encryption, secure authentication and data access with touch-enabled devices remain major challenges. Muslukhov et al. found that while 64% of the users protect their smartphone, the vast majority of them still use PINs and lock patterns that can easily be eavesdropped in public [22]. One solution is the use of physiological or behavioral biometrics [9]. Yet, current approaches suffer from insufficient maturity (e.g., face unlock [18] can easily be fooled by using printed images of the target person [14]) or lack of user acceptance. For instance, TouchID is popular on smartphones but it is questionable whether users would want to give their biometric data to third parties [28].

(a) (b)

Figure 1. Entering a graphical password on a touch-enabled mobile device leaves a smudge trace on the display that may allow an attacker to reconstruct the password (a). SmudgeSafe applies random affine geometric transformations to the image underlying the password for subsequent logins (here: rotation) to increase smudge resistance (b).

Graphical passwords were demonstrated to significantly increase security and usability – they provide a larger password space and make it easier for users to remember their passwords [20, 30]. However, graphical passwords suffer from the same problem as all state-of-the-art touch-based authentication schemes: Finger smudge trails on the display from previous logins may allow attackers to reconstruct the password and access personal data. Previous approaches to address this problem alter a custom login screen each time the user logs in [35]. While such approaches have been shown to increase security, they require custom login procedures that users first

UbiComp'14, September 13 - 17, 2014, Seattle, WA, USA

ACM 978-1-4503-2968-2/14/09...$15.00.
http://dx.doi.org/10.1145/2632048.2636090

have to learn. In contrast, our approach can – if integrated with the device OS – be applied to arbitrary graphical password schemes (e.g., lock pattern, image-based passwords or even PINs). This minimizes the burden for the user since they can continue using their favorite login mechanism.

In this work we introduce SmudgeSafe, a novel authentication system that relies on geometric image transformations to improve security of graphical passwords defined on a single image (Figure 1). These transformations significantly increase password security as the appearance of the underlying image is different for each login. Hence, each login creates an increasingly chaotic pattern of overlapping smudge traces that make it more difficult to guess the original password. It is important to note that while we focus on locimetric cued-recall graphical passwords and touch-enabled mobile devices, the proposed approach is generic and applicable to other graphical password schemes, including cognometric and drawmetric, as well as to arbitrary touch-enabled surfaces.

Our contribution is two-fold. First, we introduce the idea of applying geometric image transformations to increase smudge resistance of graphical passwords, such as translation, rotation, scaling, shearing, and flipping. We describe the design space and present a login screen application for Android that implements these transformations. Second, we evaluate the proposed approach in two user studies. In a security study with a realistic threat model, participants were asked to attack graphical passwords with geometric transformations and commonly used PINs and lock patterns. During an in-the-wild study we collected 130.000 logins from 374 users who downloaded our application from Google Play and used it over five months. We assessed the usability of our system using a built-in questionnaire and logged user performance.

RELATED WORK

Graphical Passwords

Graphical passwords rapidly gained popularity among researchers and users, most prominently in Windows 8 [19]. Graphical passwords are appealing as they can potentially increase security without compromising usability [20, 30]. The underlying assumption is that users can easily remember image features rather than a set of characters and digits. In general, three graphical password schemes can be distinguished.

Cognometric schemes present a set of images (usually a 3x3 or 4x4 grid) from which the user needs to recognize and select the correct image. The approach allows arbitrary images to be used, including faces, random art, or photographs. *Deja vu* requires the user to select 5 out of 25 images in the correct order [10]. The authors found that the failure rate strongly depends on the type of image used. Davis et al. presented *Story Scheme*, an authentication method where users are presented a set of characters and encouraged to come up with a story that would help them remember the order of their password points [8]. Yet, they found that participants often attempt to simply memorize the sequence rather than a story. Everitt et al. provide a comprehensive assessment of the use of multiple graphical passwords, focusing on frequency, interference, and training [13]. *PassFaces* is a commercial web service that uses a set of faces for authentication [26]. To learn cognometric passwords, an initial stage is required where users are shown the images and asked to memorize them, for example, based on something special in the object. However, the challenge is to provide images, that have no extraordinary cues in order not to provide useful hints to potential attackers.

In contrast, *drawmetric schemes* usually rely upon recall and require the user to draw shapes or figures to authenticate, as in the Android lock pattern or the early *Draw-a-Secret* system, presented by Jermyn et al. in 1999 [17]. The evaluation of Draw-a-Secret however showed that it is difficult for participants to accurately draw the shapes. As a solution to this, *Background Draw-a-Secret* requires the user to choose an image and then draw a series of pen strokes over the image [12]. The study showed that choosing the right background image is crucial both for security and memorability.

Locimetric schemes present the user a single image within which the password needs to be defined, for example by subsequently swiping over certain points or areas in the image. An example is *Passpoints*, where passwords are defined through click points in arbitrary images [36]. Microsoft recently employed a combination of locimetric and drawmetric schemes in Windows 8 where users need to draw a combination of shapes (circles, taps, straight lines) onto various locations of the picture to create a password. A major drawback of the approach is the fact that users are in general free to not only choose the image, but also the locations within the image, which are often the most obvious points (so-called hotspots [32]). As a solution, Bulling et al. applied the concept of saliency masks to locimetric passwords to cancel out areas in the image that should not be selected by the user [4]. In this way, a significant increase in security is achieved.

Smudge Attacks and Improving Smudge Resistance

Andriotis et al. investigated the security of the Android lock pattern [2]. They present several approaches that can be used to guess passwords from incomplete smudge trails. Aviv et al. investigated smudge attacks on the Android lock pattern, focusing on conditions under which such attacks can easily be performed [3]. They found that by using the right lighting and camera orientation, the vast majority of (parts of) patterns can be easily recognized. In a similar fashion, Zhang et al. showed that fingerprints on a tablet display can be identified using a latent fingerprint kit, mapped to a keypad and used to reconstruct the corresponding PIN [37].

Work that focused on increasing smudge resistance of mobile devices includes De Luca et al. who suggested the concept of implicit authentication in which users are not only authenticated by drawing their lock pattern but also by how they draw it [9]. Von Zezschwitz et al. presented three approaches to increase smudge resistance for lock (-like) patterns – pattern 90, marbles, and marble gap – that lead to users creating smudge traces that cannot easily be interpreted by an attacker [35]. A limitation of this approach is that the alteration needs to be implemented for every scheme. In contrast, we suggest to apply transformations independently of the background images which makes the approach applicable to any

kind of graphical password. AlRowaily and AlRubaian presented *WhisperCore*, a system that requires the user to wipe parts of the screen at the end of the login procedure to mask the smudge of the actual authentication with a new smudge trail [1]. Finally, Oakley and Bianchi proposed multi-touch passwords to increase password entropy and the difficulty of observation and smudge attacks [24].

Security of Mobile Devices

With the ability to store and access a large variety of personal information on mobile devices beyond contacts and SMS – ranging from emails over account data to online banking credentials – there is an increasing need to restrict access through secure user authentication in case of loss or theft of the device. Chin et al. investigated users' privacy concerns with regard to smartphone applications that access sensitive data (e.g., banking) [6]. They found that users are more concerned with privacy on their phone than on their laptop. As users are afraid of their smartphone being lost or stolen, they tend to minimize such activities. Complementing these findings, Muslukhov et al. found that people are much more concerned about insiders having access to their smartphone than strangers [22]. Work by the same group, furthermore, identified different types of data users tend to store on mobile phones and investigated why a certain data type is considered to be confidential, sensitive, or valuable by the user [21]. Dorflinger et al. investigated the users' view on different novel authentication methods, including biometric authentication as well as 2D and 3D gestures recognition based authentication, focusing on the perceived level of security [11]. They concluded that the PIN is not perceived as being secure enough for most users. Overall, prior work stresses the need to investigate secure and at the same time usable authentication mechanisms.

Summary

Overall, previous work shows that graphical passwords have the potential to increase the security and memorability of passwords on mobile devices, compared to state-of-the-art authentication schemes. At the same time, smudge traces on the display can be analyzed by an attacker to recover the original password and thus represent a major security threat. The approach proposed in this paper aims to globally tackle this problem by applying affine graphical transformations to the underlying password images. While only shown for locimetric graphical passwords, this approach can be applied to cognometric and drawmetric passwords as well.

THE SMUDGESAFE SYSTEM

Skin fat produces a smudge trace whenever a user interacts with a touch-enabled surface (Figure 2). This trace is clearly visible under slant incident light and was shown to allow attackers to reconstruct the original password [3]. This is particularly critical for authentication systems in which the smudge trace can be directly matched to the underlying password, such as lock patterns [34]. In contrast, text passwords, PINs, and image-based passwords are more difficult to reconstruct: While the individual password elements, such as a character or number, may be extracted rather easily, the sequence in which they were entered by the user can typically not be easily deduced from the individual finger smudges alone.

(a) (b) (c)

Figure 2. Smudge traces of three consecutive logins for lock pattern (a), PIN (b), and SmudgeSafe with a different transformation for each login (c). While the smudge traces of the lock pattern and PIN are clearly visible, SmudgeSafe generates a set of different traces overlaying each other, which makes it more difficult for attackers to reconstruct the original password.

This security threat can only be addressed by either cleaning the touch surface carefully after each use or by hiding the password trace within further traces generated while interacting with the phone. However, while users pull out their phone frequently throughout the day, cleaning the display on a regular basis is not practical. In a similar fashion, particularly when users are on the move, interactions are typically very short [25] and, for example, clicking on the mail symbol to check for new emails does not generate a sufficient number of additional smudges to hide the login trace. In addition, interaction traces can also often be distinguished from login traces based on their distinct location.

At the core of SmudgeSafe is the idea of applying affine geometric transformations to the underlying password image. Such transformations may include translations (the image is shown at a different location), rotations (the image is rotated by an angle α), scalings (the image is scaled by a factor S), shearings (the image is sheared by a distance D), or flippings (the image is flipped horizontally or vertically). If these image transformations are applied randomly every time the user logs in, smudge traces from a previous login will not match the current password image, which renders password reconstruction difficult or even impossible. In addition, subsequent logins will result in an increasingly chaotic set of smudge traces overlaying each other, which further increases security.

Transformations are applied to all pixels of the image and take into account the location of the password points. Specifically, we ensure that none of the original password points falls outside of the touch-sensitive display area after applying the transformation. We solve this by calculating the maximum possible parameter value for each transformation from the user-provided password points during runtime. It is important to note that the location of the password points also has an influence on password security and the effective password space. Generally speaking, the closer the password points are to the edge of the image the smaller the transformations can be, for example, the smaller the rotation angle α or the scaling factor S. Transformations impact security as they may

Figure 3. When defining a password on an image, SmudgeSafe allows the user to select password points from a central area of the image. This is to ensure that transformations significantly change password point locations while preserving a reasonably large theoretical password space.

reduce the effective password space and potentially allow an attacker to reconstruct the password more easily. We solve this by restricting password point selection to a central part of the image (see Figure 3). This ensures that transformations still significantly change the location of the password points while at the same time preserving a reasonably large TPS.

Combining several transformations is possible and transformations could also be applied only to certain image parts. We opted to focus on the basic affine transformations described here and leave multiple transformations for future work.

Geometric Image Transformations

We investigate five affine 2D geometric transformations that can be applied to the password images (Figure 4). Mathematically, each pixel of the original image $\vec{v} = \binom{x}{y}$ is multiplied with a transformation matrix to generate the transformed pixel $\vec{v'} = \binom{x'}{y'}$. The matrix is transformation specific.

Translation The translation describes the movement of the image in 2D in x and/or y direction (Figure 4a). Translation requires the offset by which the image is moved as an input parameter. The offset can be both positive or negative, which means that images can be moved to the left and right as well as to the top or bottom. Note, that translations along the x and y axis can be combined which, for example, results in the image to be moved one corner of the screen.

$$\vec{v'} = \begin{pmatrix} 1 & 0 & dX \\ 0 & 1 & dY \\ 0 & 0 & 1 \end{pmatrix} \vec{v}$$

Scaling For scaling, a factor is applied to the x and y dimensions of the image (Figure 4b). Scaling factors between 0 and 1 lead to downsizing the image whereas scaling factors larger than 1 upsize the image. While for upsizing all password points still need to be in the viewport, the challenge for downsizing is to ensure that the image does not become too small and, thus, password points too small to hit.

$$\vec{v'} = \begin{pmatrix} S_X & 0 & 0 \\ 0 & S_Y & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{v}$$

Rotation For rotation, the image is turned around a pivot point (usually the centre) by a certain angle α (Figure 4c). Similar to translations, this transformation moves parts of the image out of the viewport. Furthermore, rotation can either lead to parts of the screen to be left blank, or, in case the original image is larger than the viewport, that new parts of the images become visible.

$$\vec{v'} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{v}$$

Shearing Shearing (sometimes also called transvection) can be applied to both axes. It describes a linear mapping that displaces each point in fixed x or y direction by an amount (D) that is proportional to its signed distance from a line that is parallel to that direction (Figure 4d). Again, this transformation leads for certain areas of the image to become invisible. Note, that either x or y is used.

$$\vec{v'} = \begin{pmatrix} 1 & D_X & 0 \\ D_Y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{v}$$

Flipping For flipping, the image is mirrored along the x axis, the y axis, or both (Figure 4e). This rigid body transformation does not require any limitation of parameters and all parts of the images are visible after the transformation.

$$\vec{v'} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{v}$$

DESIGN SPACE

In the following, we present the design space for transformations in graphical password systems, namely, aspects that need to be considered when implementing and using such systems, because they potentially impact security and usability.

Spatial Dimensions

Transformation can be applied in different spatial dimensions, namely in 2D and 3D. We believe spatial dimensions to be of particular interest as auto-stereoscopic displays enter the market. In general, transformations can be applied both in 2D and 3D but may have unexpected effects. For example, a 3D translation in z-axis is similar to a 2D zoom transformation. Note, that while 3D transformations could enhance security by further increasing the TPS, this may compromise usability, for example, as points are obscured in a 3D scene.

Body Rigidity

An important factor is whether the transformation is body rigid, i.e., the image maintains its form throughout the transformation. Body rigidity may have an impact on how well users can remember a password. For example, a password may include a circle. Through a non-rigid transformation (e.g., shearing), the form might be transformed into an ellipse, making it more difficult to find the password points.

(a) (b) (c) (d) (e)

Figure 4. Affine 2D geometric image transformations studied in this work: translation (a), scaling (b), rotation (c), shearing (d), and flipping (e).

Combination of Transformations

Transformation can be combined by simply applying several transformation, for example first a rotation and then scaling. Mathematically, the transformation matrices are multiplied. While this may lead to higher security this may come at the cost of usability as strongly transformed images may make it difficult to remember a password. Note that applying transformations in different order leads to different results.

Image Context and Viewport

Prior research shows that users sometimes tend to memorize passwords by creating stories around the password points [4]. For example, an image may show a street scene with several persons, a bus stop, passing cars, and a traffic light. A password hint may then be "The man waiting next to the bus stop sign jumps into the red car and passes the traffic light" and the according password are the man, the car, and the traffic light. In this case, the bus stop is not a part of the password but important to remember, which man was chosen as part of the password. Through transformations, for example scaling, the bus stop sign may move out of the viewport, thus making it harder for the user to remember the password. Note, that there are certain transformations, such as flip, that in general preserve the context, but make it otherwise difficult to perceive and interpret content. Examples include flipped text as well as symmetrical or close-to-symmetrical images, like close-ups of faces, that make if difficult to determine whether an image has been flipped.

Image Complexity

Prior work suggests image complexity, for example the number of image features, to influence password security [27]. Low complexity leads to fewer hotspots and makes it easier for attackers to guess the password. Transformations such as zoom may alter the complexity. Note, that complexity may increase for zoom out or rotation in case passwords were only defined on a part of the image.

Type of Background

We envision transformations to be applicable to a wide variety of authentication mechanisms. As a result, one can imagine background images other than a picture taken during holidays or the lock pattern, that are not static. For example, the background may consist of a short video clip from which users could select password points or of a 3D image, such as a rotating dice from which users need to select the correct side. In these cases, the matrix contains one or more variables that change, depending on one or more external factors. Factors could include time, sensor data, or even input by the user.

PROTOTYPE IMPLEMENTATION

We developed a prototype SmudgeSafe authentication system on an Android phone that implements all of the previously described image transformations. To replace the lock screen we used Android's Device Policy Manager. The Device Policy Manager is able to set a password and to lock the phone.

To set up our lock screen, the user has to register our application as a device administrator. The graphical password itself is created with a wizard style dialog. First, the user needs to define a PIN, which is later used by the Device Policy Manager to lock the phone. Furthermore, this PIN can be used as a backup login mechanism in case the authentication with the graphical password fails. Such mechanisms were reported to be perceived as a valuable feature by users [11]. Then, the user has to choose whether a picture provided by our application or a picture from the phone's gallery should be used as password picture. Note, that to apply our approach to the lock pattern, the user could simply choose a lock pattern background image. Finally, the user has to set up a graphical password consisting of a series of password point within the image and enable the lock screen. Once the lock screen is enabled a service is started. A broadcast receiver listens to the intents *ACTION_SCREEN_OFF* and *ACTION_SCREEN_ON*. Once the screen goes off, the phone is being locked and the lock screen is loaded in the background. When the screen is turned on again, the lock screen is brought to the foreground. Subsequently, the user can proceed with the login process. If the user is not able to login to our system, pressing the home button forces the system to show Android's PIN input mask, where the PIN supplied in the setup can be used as a backup for authentication.

SECURITY STUDY

We designed a lab-based user study to investigate the security of geometrically transformed graphical passwords. We hypothesize that such passwords are more secure to smudge attacks compared to PINs and lock patterns, because (a) transformations make it more difficult for attackers to interpret smudge traces and (b) the theoretical password space is increased. We compare SmudgeSafe *graphical passwords* with the most commonly used authentication mechanisms for mobile phones [22], namely *lock patterns* and personal identification numbers (*PIN*). The study consists of two steps: First, we ask one group of participants to generate a set of realistic graphical passwords. We then recruit a second group and train them to attack these passwords by analyzing high-resolution pictures of the corresponding smudge traces captured from the mobile phone display.

Threat Model

The goal of this work is to evaluate the security of SmudgeSafe in a worst case scenario. We therefore assume that the attacker is in possession of the device and has perfect lighting conditions as well as a high resolution camera to perform the attack. In addition, we assume that the touchscreen was carefully cleaned and that the user authenticated only once before the attack. This means there is no smudge trace visible on the display apart from the actual password trace. We argue that this is the perfect condition to perform a smudge attack and therefore the worst case scenario in terms of security.

Password Generation

To obtain realistic passwords we recruited four participants (two female) aged 22 to 42 years ($M = 29.25$, $SD = 9.22$) from University mailing lists. We asked them to generate lock pattern, PINs, and graphical passwords consisting of four points or digits, respectively, as they would do for their personal phone. Graphical passwords were defined on 8 manually selected images of varying complexity and content.

Upon arrival in the lab, participants filled out a brief questionnaire on demographics and were introduced to the concept of graphical passwords and geometric transformations. We then showed them the Android app and asked them to use the wizard to create one graphical password for each of the eight images. Additionally, we asked them to create an Android lock pattern and a four-digit PIN. This resulted in a total set of 32 graphical passwords, four lock patterns, and four PINs.

Smudge Trace Generation

To achieve controlled and optimal attack conditions, we opted to use high resolution images of smudge traces instead of real smudge traces. This is a commonly applied procedure of presenting smudge traces in security assessments [35]. We generated the traces by carefully cleaning the display of a Samsung Nexus S smartphone (screen resolution 800*480 pixels) until not even the slightest amount of grease was visible. A researcher then entered the password on the phone display with his right index finger (see Figure 5). After each trace, we took a high-resolution picture of the trace with optimal lighting conditions using a Canon EOS 7D camera and a 800 watt light source highlighting the smudge trace.

(a) (b) (c)

Figure 5. High-resolution images of smudge traces as used in the lab-based security study: lock pattern (a), PIN (b), and transformed locimetric graphical password (c). Each image was taken under perfect lighting conditions of a single trace entered on a carefully cleaned display.

This three-step procedure was repeated for all passwords defined by the pre-study participants. Passwords were entered by the same researcher to create similar smudge traces in terms of intensity and thickness. To each of the 32 passwords from the pre-study we applied one particular transformation. The assignment of transformations to images/passwords was counterbalanced. In total, we created images of 40 passwords (32 graphical passwords, four lock patterns, and four PINs).

Theoretical Password Space

The theoretical password space (TPS) for the three authentication mechanisms used during the study is as follows. For the *PIN* (four digits, each a number between zero and nine), the TPS is $TPS_{PIN} = log_2(10^4) \approx 13.29$. The *lock pattern* consists of at least four points chosen from a 3x3 grid. Each point can only be used once. Hence, the TPS is $TPS_{Pattern} = log_2(9*8*7*6) \approx 11.56$ at the minimum and $TPS_{Pattern} = log_2(9!) \approx 18.47$ at the maximum. Note, that the Android implementation of lock patterns does not allow intermediate points to be skipped. Finally, the TPS for *graphical passwords* depends on the resolution of the screen and the size of the area users need to touch for the system to correctly recognize the password point. In our case, we use a phone with a screen resolution of 800*480 pixels and a password area diameter of 100 pixels, resulting in a TPS of $TPS_{graphPW} = log_2(32 * 31^3) \approx 19.86$. Note that due to the fact that only the central area is used for password generation (see Figure 3), the TPS is reduced to $TPS_{graphPW} = log_2(18 * 17^3) \approx 16.43$. If we take passwords with more than four points into account, the maximal TPS for graphical passwords can be $TPS_{graphPW} = log_2(18 * 17^{17}) \approx 73.66$.

Note, that due to the different transformations, the TPS will be reduced depending on the used parameters. For instance, when a *translation* with 100 pixels is applied, the TPS will be reduced to $TPS_{Translation100} = log_2(15 * 14^3) = 15.32$. The used parameters can be adapted to the required level of TPS. However, not all transformations modify the TPS. *Flipping*, for instance, has no effect on the TPS since it does not change the size of the area the password is entered on. In contrast, a *rotation* may reduce the TPS.

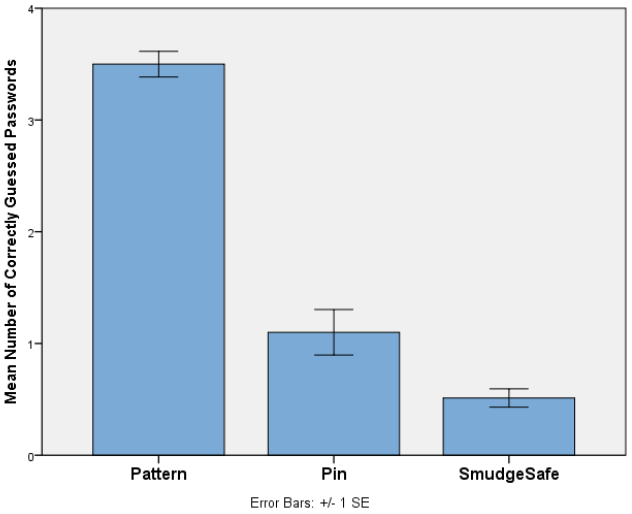

Figure 6. Comparison of the means of three authentication systems: Graphical passwords with affine transformations were significantly more difficult to guess than lock patterns and PINs. The error bars indicate the standard error.

Evaluation

In the second part of the study we assessed the security of the three authentication schemes with regard to the smudge trace threat model. The way we generated the passwords in the first part of the experiment ensures that (1) each password, (2) each image, and (3) each transformation was tested equally often. We grouped the passwords according to the transformations, as we wanted to gain insights into the perceived difficulty of each transformation. Thus, participants had to first perform all attacks for transformation A, then on transformation B, and so on. After each group of transformation, participants got briefed about the used transformation and received a brief questionnaire on each of them. To minimize any potential sequence effects, we counterbalanced the order of the transformations for each participant as well as the order of the passwords within the groups of transformations.

For the study we made sure that the transformation applied to the image for generating the smudge traces and the transformation for which the attack was performed were different. The rationale behind this was that our Android application never shows images with the same transformation applied twice in a row. We believe this increases the security as the trace of the previous login is usually the most visible.

Participants and Procedure

We recruited 20 participants (5 female and 15 male) aged between 19 and 54 years ($M = 26.15$, $SD = 7.04$) from University mailing lists. Upon arrival in the lab, we briefly explained the purpose of the study and asked the participants to sign a consent form as well as to complete a brief demographic questionnaire. We then introduced them to the concept of graphical passwords, login patterns, and PINs. We also showed participants a sample image of a smudge trace together with the underlying image and instructed them to watch out for image features that may have been chosen by the creator of the password.

Participants then started to attack the first group of passwords (i.e., all passwords entered with the same transformation). Note, that participants were not told about the grouping. There was no time limit for the analysis, but participants were limited to three attempts per password. This is a common restriction also for other state-of-the-art authentication systems. After each group, participants completed a questionnaire on the perceived difficulty of the attack and whether they thought that they could have been successful with more attempts. After the last group participants had to complete a final questionnaire and we conducted brief semi-structured interviews. All participants were compensated with a base payment of 10 EUR. To keep motivation high we promised them an additional candy bar for each successful password attack.

Results

We first compared the *overall security* of SmudgeSafe with the PIN and lock pattern conditions in terms of successful attack attempts in one to three trials (see Figure 6). Overall, the lock pattern performs worst (mean number of passwords correctly guessed per participant $M = 3.50$, $SD = 0.51$), followed by PIN ($M = 1.10$, $SD = 0.91$). The graphical transformations implemented in SmudgeSafe performs best ($M = 0.51$, $SD = 0.33$). A Friedman Analysis of Variance (ANOVA) shows statistically significant differences between all three authentication systems, $\chi^2(2) = 33.50$, $p < .001$. Three Wilcoxon tests were used to follow-up on this finding. We applied a Bonferroni correction, hence the effects are reported at $p = .017$. The Wilcoxon tests showed that our approach performs significantly better than the lock pattern, $Z = -4.03$, $p < .001$, $r = -.64$, and better than the PIN, $Z = -2.62$, $p = .010$, $r = -.41$. Furthermore, the PIN performs significantly better than the lock pattern as well, $Z = -3.87$, $p < .001$, $r = -.61$. This shows that the SmudgeSafe approach outperforms state-of-the-art authentication systems with regards to the smudge threat model.

Furthermore, we evaluated the *interplay of different transformations* in more detail, that is, whether passwords of images with a certain transformation (source transformation) are more difficult to be entered correctly on images with another transformation (target transformation) than others. The aim was to identify un-secure combinations of transformations that should not be used after each other. First, we ran two Friedman ANOVAs on the *source transformation* and the *target transformation*. An overview of the mean number of successful password attacks per transformation is shown in Figure 7. The success rate of each combination can be found in Figure 8. The figure shows that participants are able to guess the lock pattern with three trails correctly in 87.5% of all cases. For PINs, they correctly enter the password in 27.5% of the cases. Looking at our approach we found that in cases where translation is the source transformation and scaling the target transformation, SmudgeSafe performs worst (30% correctly guessed passwords), while the combination rotation and scaling is most secure (0% correct). A Friedman ANOVA and pairwise comparisons of the transformations using Bonferroni corrected Wilcoxon tests showed no statistically significant difference. Though we cannot claim that there is no difference, the findings suggest a considerably small effect.

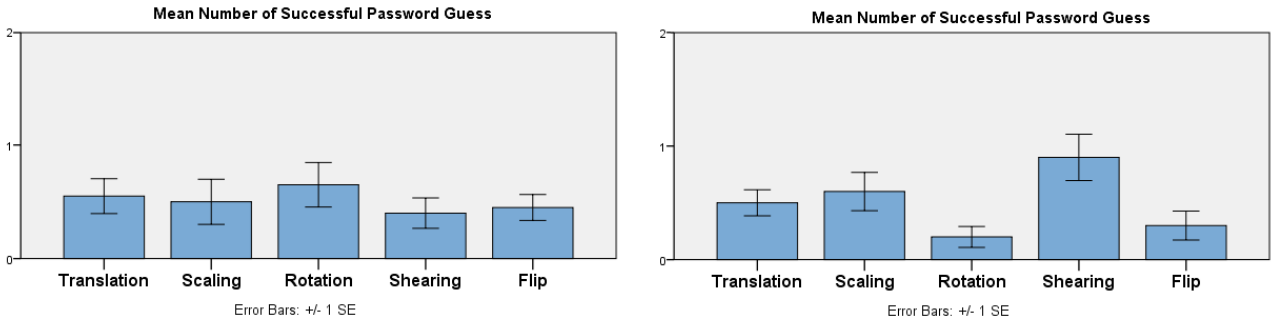

Figure 7. Results of the security study. The results grouped by transformations used for creating the smudge traces (left) and for transformations used for attacking the system (right) are shown. The error bars indicate the standard error.

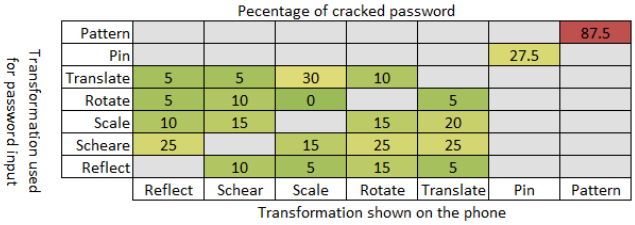

Figure 8. Percentages of successful attacks depending on the input of the smudge trace and the authentication system shown to the attacker. Different graphical transformations are used for smudge trace generation and attack trails.

USABILITY STUDY

To analyse usability we modified our prototype implementation so that it saves every login attempt to a log file on the phone. This file is sent to our server. Each log file entry consists of a timestamp, the transformation used and its parameters, the entered and the original graphical password, and whether the login attempt was successful. We released our application in the Google Play store to create insights with regard to the usability in the wild, which we consider more ecologically valid than lab studies [16]. All following analyses cover a period of five month. 632 users actively used the application over the reporting period according to Google Play store statistics. For privacy reasons we allowed users to turn off the logging functionality and, hence, to not share data with us. We received data from two different sources. First, we logged the user's authentication attempts and, second, we embedded a questionnaire within the app.

Log Data

During five months 374 users shared their data. This includes only users that authenticated at least once per transformation.

Login Attempts

We calculated the rate of successful login attempts using SmudgeSafe. We first removed all login attempts that were generated accidentally. This can happen, for example, when the screen turns on while the mobile phone is still in the user's pocket or when the user accidentally touches the screen. Such login attempts differ in length from a true login attempt. We removed all login attempts that were either too short (i.e. less than half the length of the actual password) or too long (i.e. more than twice the length of the password). After this cleanup we recorded a total of 129,538 login attempts, 98,130 of which were successful (74% success rate).

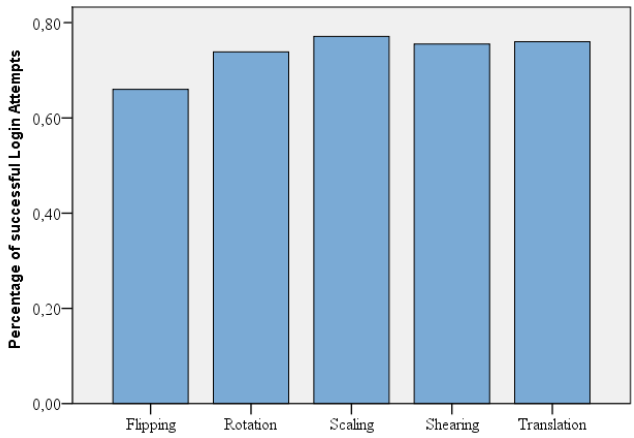

Figure 9. Comparison of the five different transformations and their percentage of successful login attempts.

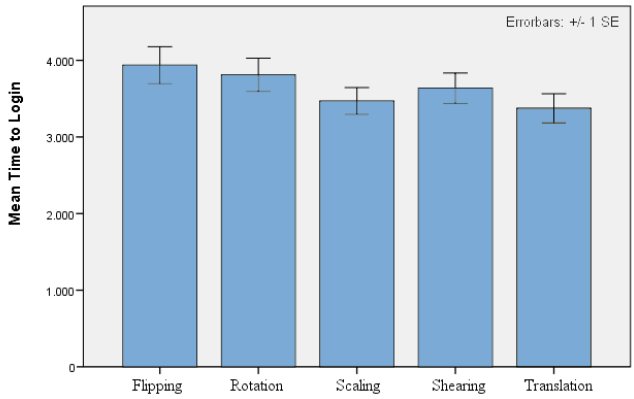

Figure 10. Comparison of the five different transformations and their mean login times.

Figure 9 depicts the number of successful attempts per transformation. Since Mauchly's test indicates that the assumption of sphericity had been violated, $\chi^2(3) = 41.603$, $p < .001$, degrees of freedom are corrected using Huyn-Feldt estimation of sphericity, $\epsilon = .954$. Results of a repeated measures analysis of variance shows statistically significant differences between transformations, $F(3.816, 1423.465) = 23.803$, $p < .001$, $\eta^2 = .060$. Bonferroni post-hoc tests revealed *flipping* to perform significantly worse than the other transformations.

Usability Metric	Median	Std. Dev.
Learnability (grasp)	5	1.42
Learnability (easy to learn)	5	1.29
Efficiency	5	1.25
Memorability	5	1.21
Errors	3	1.34
Satisfaction	4	1.11

Table 1. Results of usability assessment.

Login Time

In addition, we explored the time needed for users to login into the system. We measured the time from pressing the unlock button until the user was successfully authenticated. Since we received some artificial values (e.g., login times of more than an hour), we eliminated such outliers using the Tukey method [33]. Overall, in the reporting period, users needed an average of 3.64 seconds to login ($SD = 1.66s$). This is roughly in line with work by von Zezschwitz et al. who reported an average login time of $1.50s$ for PINs and $3.17s$ for lock patterns [34]. Note, that our users could choose passwords with arbitrary length, whereas the aforementioned study limited users to 5 points per password.

Figure 10 shows the mean login times per transformation. A repeated measures analysis of variance reveals statistically significant differences between transformations, $F(4, 176) = 3.777$, $p < .006$, $\eta^2 = .079$. Bonferroni post-hoc tests revealed rotating to perform significantly worse than scaling.

Questionnaire

We released a new version of the application into the Google Play store with an embedded questionnaire at the end of the fourth month of the study. After the users updated their application, they were prompted after their seventh successful authentication trail with the questionnaire as a pop-up. Since users usually authenticate with a particular task in mind, we allowed them to skip the questionnaire. After users skipped the questionnaire they got prompted two more times after another seven successful authentications.

The questionnaire consists of six questions on system usability. Following Nielsen's definition of usability [23], we assessed learnability, efficiency, memorability, errors, and satisfaction. In total, 49 out of 487 users (those who received the updated version of the application) completed the questionnaire. In the questionnaire participants were asked to rate using a five point Likert scale (1=totally disagree and 5=totally agree).

To assess *learnability* of our approach we asked participants whether 'the system is easy to grasp/understand' and whether 'the authentication system is easy to learn'. Users strongly agreed that the system is easy to grasp/understand ($Med = 5$, $SD = 1.42$) and easy to learn ($Med = 5$, $SD = 1.29$). With regard to *efficiency*, we presented users the following statement: 'I am able to log in quickly.' The results showed that participant attribute a very high efficiency to the system ($Med = 5$, $SD = 1.25$). One of the main benefits of graphical password is that they provide a higher *memorability* than PINs and passwords. This was confirmed by the answers of the participants who strongly agreed that 'the password trace is easy to remember' ($Med = 5$, $SD = 1.21$). To assess the perceived usability with regard to *errors*, we presented the following statement. 'I often do not manage to log in successfully.' Participants rate this statement moderately ($Med = 3$, $SD = 1.34$). Finally, we were interested in the overall *satisfaction* of the users with the system. We found that users are satisfied with our approach and rate it as easy to use ($Med = 4$, $SD = 1.11$).

	Entry Time			Error Rate			Security	
System	Lab	Field	Wild	Lab	Field	Wild	Smudge	Shoulder
PIN	3.9s	1.5s		0%	4.3%		27.5%	60%
Lock Pattern	2.4s	3.2s		2%	16.6%		87.5%	
Multitouch	1.8s			8%				
Pattern 90	4s			2%			46%	
Marbles	6.8s			3%			0%	
Marbles Gap	8.2s			1%			0%	
PassGo	4s			0%				20%
UYI	8.7s			10%				20%
TAPI	4.8s			8%				20%
CCP	4.6s			34%				20%
MIBA	3.9s			18%				8.5%
Smudge Safe			3.6s			25%	12.7%	

Table 2. Comparison with other approaches. SmudgeSafe achieves a low input time, moderate error rates, and high security.

DISCUSSION AND COMPARISON TO PRIOR WORK

This section compares input times, error rates, and level of security of SmudgeSafe with 11 alternative approaches [5, 7, 15, 24, 29, 31, 34, 35] to put the results into perspective. All approaches aim to overcome common threats like smudge attacks or shoulder surfing. The results are summarized in Table 2. Note, that this summary ignores the context in which the different studies have been performed (see discussion at the end of this section) and that different threat models were applied.

In summary, graphical passwords with affine 2D image transformations are significantly more difficult to attack than lock patterns and PINs, but are outperformed by approaches such as Marbles or Marbles Gap. At the same time, our approach preserves the low login times of lock patterns and, hence, are faster to apply than the majority of other approaches. Finally, the login success rate of 74% is slightly lower, yet comparable, with that of lock patterns, as found by von Zezschwitz et al. [34]. They also found that the reason why such rather low success rates are acceptable for users is that a false login comes at only marginal additional costs. Compared to approaches such as PIN, where users first need to click away information on the remaining attempts, users can instantly proceed with re-entering the lock pattern after an unsuccessful attempt. The high / very high ratings with regard to satisfaction and efficiency back up the assumption that this is also true for our approach that uses a similar mechanism.

Selecting the best approach from the comparison is hardly possible, simply because authentication systems face the immanent trade-off between high usability and high security. As all other systems, SmudgeSafe provides a compromise – yet one that has, in comparison to the other approaches, been proven to be acceptable by users through our public release and user uptake. Furthermore, our results provide evidence, that security could be further improved by smartly choosing the order in which transformations are applied (successful attacks could be brought down to 0%, cf., Figure 8).

The comparison of the different approaches is difficult for a number of reasons.

Lab vs. Field vs. In-the-Wild. There is no commonly accepted best practice for evaluating authentication systems. Methods range from controlled lab environments or field tests, where authentication is the primary task, to in-the-wild studies, which are less controlled but embed authentication tasks into users' daily routine and, therefore, achieve high ecologic validity. Our comparison provides evidence of a strong influence on the results. Most importantly, the numbers reveal a striking increase in error rates in the real world, where authentication is not the primary task (cf. Table 2).

Security assessment. There are different approaches to assess security. Prior work relies upon one researcher performing attacks [34]. While a researcher is certainly among the most skilled persons for such an attack it could be subject to discussion whether this is sufficient to comprehensively assess security. In this regard, we believe our approach with 20 motivated and trained participants attacking the passwords to be a particular strength of our work.

Overall, our observations suggest, that research on secure authentication mechanisms could benefit from a stronger focus on ecologic validity obtained from in-the-wild deployments. While we do not question the value of controlled lab studies for certain aspects, we believe data obtained from authentic situations to be a valuable complement.

Limitations

The studies have some limitations. First, we focused on smudge traces in our threat model to evaluate security. Hence, we cannot draw any conclusions on other threats, such as shoulder surfing. Second, although we provided participants with an introduction and hints to attacking passwords, their abilities to extract passwords based on transformation may have been different. Third, background images used during the study might have had an influence on attack performance. However, as we chose a wide variety of common types of images and counter-balanced passwords and transformations across images, we expect a rather small effect. Fourth, we acknowledge the general limitations of an in-the-wild study, foremost the lack of internal validity. For instance, we did not have any control over the situations in which participants entered their passwords (while walking, driving, biking) and how they chose their password. At the same time, conducting the study in-the-wild allowed our approach to be studied in a natural setting, which we believe to be another strength of this work.

FUTURE WORK

The current work aims to lay the groundwork for image transformations on graphical passwords and for this reason focuses on a sample set of 2D image transformations. It will be interesting to see which other 2D transformations the research community will come up with. One obvious avenue for future research is to investigate more sophisticated and/or non-affine 2D transformations, for example, fisheye effects. As mentioned before, a second promising extension of this work is the application of multiple transformations to the same image to further increase smudge resistance. Finally, while the current work focuses on static 2D transformations, the advent of powerful smartphones with auto-stereoscopic displays also raises the question of whether and how the proposed approach could be applied to 3D as well as to dynamic interfaces.

Furthermore, we focused on touch-enabled mobile devices in our user studies. Nevertheless, the results are equally applicable to other touch-enabled devices using graphical passwords. Additionally, transformations can be applied to other password systems such as PINs entered on a (digital) keypad as used at ATMs or ticket machines. By applying a transformation, for instance a translation, smudge attacks on these keypads can be prohibited. The usefulness of each transformation needs to be investigated in detail for different password systems, since in cases where the screen cannot easily be rotated, usability may be seriously compromised.

CONCLUSION

In this paper we presented SmudgeSafe, an authentication system for touch-enabled devices that increases security by applying random geometric transformations to the image underlying graphical passwords. Results from our user studies show that SmudgeSafe is significantly more secure than state-of-the-art authentication schemes based on PINs and lock patterns. Furthermore, an in-the-wild study attributes high usability with regard to learnability, efficiency, memorability, errors, and satisfaction. These results underpin the significant potential of this approach, particularly as it is also applicable beyond locimetric passwords. In general, any password schemes that are based on a series of password points can benefit from our approach. Even though we focused on mobile phones as one particular use case for our approach, we see large potential in applying it to other ubiquitous systems, such as tablets, terminals, and public displays.

ACKNOWLEDGMENTS

The research leading to these results has partly received funding from the German Research Foundation within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

REFERENCES

1. AlRowaily, K., and AlRubaian, M. Oily residuals security threat on smart phones. In *Proceedings of the 1st International Conference on Robot, Vision and Signal Processing* (2011), 300–302.

2. Andriotis, P., Tryfonas, T., Oikonomou, G., and Yildiz, C. A pilot study on the security of pattern screen-lock

methods and soft side channel attacks. In *Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (2013), 1–6.

3. Aviv, A. J., Gibson, K., Mossop, E., Blaze, M., and Smith, J. M. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies* (2010), 1–7.
4. Bulling, A., Alt, F., and Schmidt, A. Increasing the security of gaze-based cued-recall graphical passwords using saliency masks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), 3011–3020.
5. Chiasson, S., van Oorschot, P. C., and Biddle, R. Graphical password authentication using cued click points. In *Computer Security–ESORICS 2007*. Springer, 2007, 359–374.
6. Chin, E., Felt, A. P., Sekar, V., and Wagner, D. Measuring user confidence in smartphone security and privacy. In *Proceedings of the 8th Symposium on Usable Privacy and Security* (2012), 1:1–1:16.
7. Citty, J., and Hutchings, D. R. Tapi: touch-screen authentication using partitioned images. In *Elon University*, Citeseer (2010).
8. Davis, D., Monrose, F., and Reiter, M. K. On user choice in graphical password schemes. In *Proceedings of the 13th USENIX Security Symposium* (2004), 11–11.
9. 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 SIGCHI Conference on Human Factors in Computing Systems* (2012), 987–996.
10. Dhamija, R., and Perrig, A. Deja vu: a user study using images for authentication. In *Proceedings of the 9th USENIX Security Symposium* (2000), 4–4.
11. Dörflinger, T., Voth, A., Krämer, J., and Fromm, R. "my smartphone is a safe!" - the user's point of view regarding novel authentication methods and gradual security levels on smartphones. S. K. Katsikas and P. Samarati, Eds. (2010), 155–164.
12. Dunphy, P., and Yan, J. Do background images improve "draw a secret" graphical passwords? In *Proceedings of the 14th ACM Conference on Computer and Communications Security* (2007), 36–47.
13. Everitt, K. M., Bragin, T., Fogarty, J., and Kohno, T. A comprehensive study of frequency, interference, and training of multiple graphical passwords. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), 889–898.
14. Findling, R. D., and Mayrhofer, R. Towards face unlock: on the difficulty of reliably detecting faces on mobile phones. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, ACM (2012), 275–280.
15. Hayashi, E., Dhamija, R., Christin, N., and Perrig, A. Use your illusion: Secure authentication usable anywhere. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, SOUPS '08, ACM (New York, NY, USA, 2008), 35–45.
16. Henze, N., Shrazi, A. S., Schmidt, A., Pielot, M., and Michahelles, F. Empirical research through ubiquitous data collection. *IEEE Computer 46*, 6 (2013), 74–76.
17. Jermyn, I., Mayer, A., Monrose, F., Reiter, M. K., and Rubin, A. D. The design and analysis of graphical passwords. In *Proceedings of the 8th USENIX Security Symposium* (1999), 1–1.
18. Kelion, L. Google facial password patent aims to boost android security. BBC News – Technology, June 2013. `http://www.bbc.co.uk/news/technology-22790221`, last accessed: March 8, 2014.
19. Microsoft. Personalize your PC. `http://windows.microsoft.com/en-us/windows-8/picture-passwords`, last accessed: March 8, 2014.
20. Moncur, W., and Leplâtre, G. Pictures at the atm: exploring the usability of multiple graphical passwords. In *Proceedings of the SIGCHI International Conference on Human Factors in Computing Systems*, CHI ' 07 (2007), 887–894.
21. Muslukhov, I., Boshmaf, Y., Kuo, C., Lester, J., and Beznosov, K. Understanding users' requirements for data protection in smartphones. In *Proceedings of the 28th IEEE International Conference on Data Engineering*, IEEE (2012), 228–235.
22. Muslukhov, I., Boshmaf, Y., Kuo, C., Lester, J., and Beznosov, K. Know your enemy: the risk of unauthorized access in smartphones by insiders. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services* (2013), 271–280.
23. Nielsen Norman Group. Usability 101: Introduction to Usability. `http://www.nngroup.com/articles/usability-101-introduction-to-usability/`, last accessed: March 8, 2014.
24. Oakley, I., and Bianchi, A. Multi-touch passwords for mobile device access. In *Proceedings of the ACM Conference on Ubiquitous Computing*, Ubicomp '12 (2012), 611–612.
25. Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2005), 919–928.
26. Passfaces Corporation. Passfaces: Two Factor Authentication for the Enterprise. `http://www.passfaces.com/`, last accessed: March 8, 2014.

27. Peters, R. A., and Strickland, R. N. Image complexity metrics for automatic target recognizers. In *Automatic Target Recognizer System and Technology Conference* (1990), 1–17.

28. Riha, Z., et al. Toward reliable user authentication through biometrics. *IEEE Security & Privacy 1*, 3 (2003), 45–49.

29. Ritter, D., Schaub, F., Walch, M., and Weber, M. Miba: Multitouch image-based authentication on smartphones. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, ACM (New York, NY, USA, 2013), 787–792.

30. Schaub, F., Walch, M., Könings, B., and Weber, M. Exploring the design space of graphical passwords on smartphones. In *Proceedings of the 9th Symposium on Usable Privacy and Security*, SOUPS'13 (2013), 11:1–11:14.

31. Tao, H., and Adams, C. Pass-go: A proposal to improve the usability of graphical passwords. *IJ Network Security 7*, 2 (2008), 273–292.

32. Thorpe, J., and van Oorschot, P. C. Human-seeded attacks and exploiting hot-spots in graphical passwords. In *Proceedings of 16th USENIX Security Symposium* (2007), 8:1–8:16.

33. Tukey, J. W. *Exploratory data analysis*. Addison-Wesley, Reading, Mass. [u.a.], 1977.

34. 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.

35. von Zezschwitz, E., Koslow, A., De Luca, A., and Hussmann, H. Making graphic-based authentication secure against smudge attacks. In *Proceedings of the International Conference on Intelligent User Interfaces* (2013), 277–286.

36. Wiedenbeck, S., Waters, J., Birget, J.-C., Brodskiy, A., and Memon, N. Passpoints: design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies 63*, 1-2 (July 2005), 102–127.

37. Zhang, Y., Xia, P., Luo, J., Ling, Z., Liu, B., and Fu, X. Fingerprint attack against touch-enabled devices. In *Proceedings of the 2nd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices* (2012), 57–68.