# Now You See Me, Now You Don't – Protecting Smartphone Authentication from Shoulder Surfers

**Alexander De Luca[1], Marian Harbach[2], Emanuel von Zezschwitz[1], Max-Emanuel Maurer[1], Bernhard Slawik[1], Heinrich Hussmann[1], Matthew Smith[2]**
[1]Media Informatics Group, University of Munich (LMU), Munich, Germany
[2]Usable Security and Privacy Lab, Leibniz University Hannover, Hannover, Germany
{alexander.de.luca, emanuel.von.zezschwitz, max.maurer, bernhard.slawik}@ifi.lmu.de,
hussmann@ifi.lmu.de, {harbach, smith}@dcsec.uni-hannover.de

## ABSTRACT
In this paper, we present XSide, an authentication mechanism that uses the front and the back of smartphones to enter stroke-based passwords. Users can switch sides during input to minimize the risk of shoulder surfing. We performed a user study ($n = 32$) to explore how switching sides during authentication affects usability and security of the system. The results indicate that switching the sides increases security while authentication speed stays relatively fast ($\leq 4$ seconds). The paper furthermore provides insights on accuracy of eyes-free input (as used in XSide) and shows how 3D printed prototype cases can improve the back-of-device interaction experience.

### Author Keywords
Back-of-device interaction; Authentication; Security

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

## INTRODUCTION
Due to increased computing capabilities, modern smartphones hold massive amounts of private and potentially sensitive user data. In many cases, additional remote data is accessible through apps on the smartphone. This is very convenient for the users but becomes problematic should they ever lose their phone or if it gets stolen. Many users are aware of this problem and thus, want to protect access to their devices [16].

The most common authentication mechanisms for smartphones are password, PIN and the Android unlock pattern. While the latter was specifically designed for touchscreens, PIN and password were adopted from different contexts (ATMs and desktop computers). It comes as no surprise that they do not perform well on mobile devices. Especially passwords are hard and slow to enter on smartphones [21]. Furthermore, the security of all three systems suffers from different attacks like smudge [2] and shoulder surfing [13, 21].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*CHI 2014*, April 26–May 1, 2014, Toronto, Ontario, Canada.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2473-1/14/04..$15.00.
http://dx.doi.org/10.1145/2556288.2557097

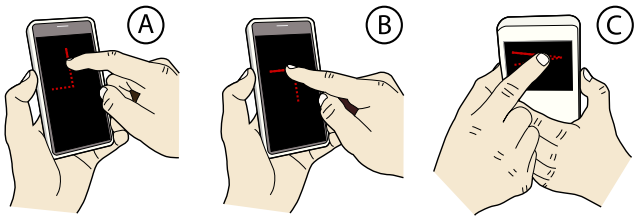



**Figure 1. A user enters a password consisting of three shapes: "Down, Left" (A), "Right, Down" (B) and "Left, Right" (C). Shapes can be entered on the front or on the back of the device. Important: the strokes are only visualized in this figure, but not in the actual user interface.**

In this paper, we present a system that we designed to overcome these problems, but that also avoids the usability impact most countermeasures to these attacks usually have. XSide leverages recent technological and research advances to provide ease-of-use while also protecting against observation attacks. To authenticate using XSide, users draw shapes (or gestures) on the back and/or on the front of the device. XSide was specifically designed for mobile touchscreen devices. It can be used eyes-free, provides increased protection against smudge attacks and improves shoulder surfing resistance. The main contributions of this work are:

1. A flexible authentication system that is adaptable to a user's context. This is a very desirable property for a smartphone, a device that is constantly carried around. The study showed that XSide has good resistance against shoulder surfing attacks and is easy and fast to use.
2. We present insights on how well relative horizontal and vertical strokes can be performed on both sides of the smartphone. Specifically, we identified how switching sides during the input influences usability and security.
3. We improved the hardware prototype, creating a single unit with a configurable back touch entry window using 3D printing. We show how this, together with an enhanced algorithm significantly influenced user experience. Thus, this work offers evidence of the importance of more professional prototypes for HCI research.

## RELATED WORK

Stroke-based passwords are representatives of drawmetric authentication mechanisms, which provide a graphical alternative to alphanumeric passwords [20]. Instead of typing an alphanumeric string, the user draws a simple pattern, which was chosen during enrollment. Drawmetric passwords have various advantages when compared to alphanumeric passwords, especially when focusing on current touchscreen-based mobile devices. While direct touchscreen interaction makes the use of randomized alphanumeric passwords cumbersome and error prone [21], it makes graphical passwords more usable and natural [4]. In addition, drawmetric passwords are easier to memorize as they exploit the human motor-memory [11] and benefit from the picture-superiority effect [22].

The first drawmetric approach was proposed in 1999 with Draw-a-Secret [15]. By analyzing the position and the temporal order of freely drawn images, the concept has a big theoretical password space. At the same time, this also makes it hard to reproduce complex drawings. Consequently, less complex modifications of Draw-a-Secret were proposed in the following years to improve memorability and usability (e.g. [12, 23]). With the introduction of Android's pattern unlock by Google in 2008, a usability-optimized drawmetric password concept was finally widely deployed on mobile devices. Instead of relying on free drawings, the Android pattern unlock is based on simple patterns, which result from connecting dots of a $3 \times 3$ matrix.

Despite being more usable, the downside of direct input of graphical passwords is that authentication is very prone to shoulder-surfing attacks [13]. This is particularly critical in the context of mobile devices, where authentication often takes place in public spaces. In addition, Aviv et al. [2] showed that Android patterns are vulnerable to smudge attacks. Smudge attacks are based on oily residues, which remain on the touchscreen, when a user enters her pattern. This "smudge" is used to deduce the user's credentials.

While investigations on reducing the risk of smudge attacks are reasonably new (e.g. [25]), several shoulder-surfing resistant approaches have been proposed in the recent years. The Convex-Hull-Click (CHC) authentication by Widenbeck et al. [26] makes use of indirect selections. The user can click into the convex-hull of specific icons, but never clicks on the specific icon itself. A similar approach was proposed by Zhao et al. [29]. The Phonelock [3] adds audio and haptic cues to fight observational attacks. Other approaches make use of complex selection chains [18], which are hard to observe or are association-based [14, 19]. None of these concepts is following the drawmetric approach, but all comprise either randomization or demand multiple challenges to authenticate. As a consequence, shoulder-surfing resistance comes with higher input times and decreased usability.

To date, only a few stroke-based solutions have been proposed. One approach is based on adding a second security layer to the basic pattern unlock [1, 9]. Those concepts analyze biometric cues like pressure and input speed to validate the entered pattern. Since the actual authentication mechanism stays the same, authentication speed is maintained.

Like all biometric approaches, these concepts comprise false detection to some extent, affecting either security (false positives) or usability (false negatives). Zakaria et al. [28] propose to make use of decoy strokes or disappearing strokes to secure the Draw-a-Secret [15] concept. The user study showed that both approaches are not significantly improving shoulder surfing resistance. EyePassShapes [8] uses gaze-based input, which is highly resistant to shoulder surfing. Since this approach needs accurate and costly eye tracking hardware, it is not yet applicable to mobile devices.

In contrast to previous concepts, XSide does not use randomization or the risk of false detections. By utilizing simple strokes on the front and on the back of the device, users are able to adapt the input to current shoulder surfing risks. Additionally, our system is the first approach specifically designed for eyes-free interaction, which improves usability and security at the same time. Authentication simply takes place out-of-sight of the attacker and therefore, no additional noise is needed. Since strokes are entered over and over at the same position, we furthermore claim that XSide is more secure against smudge-attacks than other drawmetric concepts.

The most relevant work with respect to XSide is the back-of-device authentication system [10], in which all input is performed on the back of the device. The goal of XSide was to allow a more user-friendly and flexible approach, enabling users to intuitively adapt the required level of security to their current context as both sides can be used even within a password. For instance, if people are standing around the user and can see the front screen, the user can perform the input on the back. Both, the front and the back of the device can be used in combination to offer enhanced security in crowded places.

## CONCEPT AND PROTOTYPE

XSide utilizes the touchscreen on the front as well as a touch sensitive area on the back of the device for password entry. We assume that mobile devices with touch sensitive back sides will hit the market soon as companies like DoCoMo are working on prototypes [17]. In addition, more and more smartphone manufacturers are using the rear to place interaction elements like buttons [7]. Therefore, concepts like XSide have high potential for future smartphone authentication.

To authenticate with XSide, users enter $n$ shapes. Shape entry can take place on the front, on the back or on a combination of both sides (see for instance figure 1). Shapes are made up of an arbitrary combination of horizontal and vertical strokes. The same stroke cannot be used twice in a row (e.g. "Up, Up" is not possible). Different shapes are distinguished by lifting the finger. In the remainder of this paper, a set of shapes will be referred to as a user's "password".

The left-hand side of figure 4 shows the four possible strokes a shape in XSide can consist of, together with their internal representation. For instance, an upward stroke is represented as "U". On the right-hand side, an exemplary password is depicted consisting of three shapes with two strokes each. The color code (black and red shapes) was required for the user study and will be explained in the user study design section.

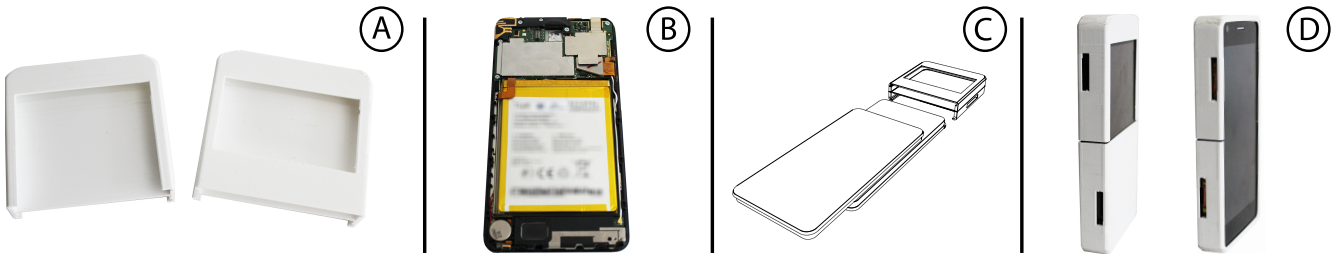



**Figure 2. XSide prototype. (A) Two design alternatives for the back cover, leaving 50 % and 33 % of screen real estate open. (B) One of the smartphones used for the prototype. The back cover and the camera were removed. (C) Cover mechanism. The covers are slid on the smartphones from two sides. This enables easy replacement of the smartphones and the covers. (D) The final prototype using the 50 % screen space cover.**

The system uses relative horizontal and vertical strokes. This decision was made as feedback-less pointing accuracy on the back of a smartphone is very low and since diagonal strokes are hard to perform without visual feedback [10].

To minimize the form factor as a source of bias, we created an advanced prototype for XSide. We used two Alcatel One Touch Idol Ultra since it was advertised as the thinnest smartphone on the market. As shown in figure 2, B, we removed the back cover as well as the camera to further reduce the thickness and weight. To give the prototype a look and feel as close as possible to a normal smartphone, we created 3D printed cases to combine the two devices. Using a sliding mechanism to connect the two halves of the case (see figure 2, C), the devices can be removed and replaced at any time. The final prototype is shown in figure 2, D. It is 1.2 cm thick and weighs 247 grams. This is 1.2 cm and 22 grams less than the prototype used in previous work [10].

The most important design decision for the prototype was on how much interaction space to provide on the back. A large space increases the chance of unwanted touches, e.g. by touching the screen with the palm. If the area is too small, it is harder to input the strokes. As a consequence, we experimented with different designs. Two of them are shown in figure 2, A. The left shell leaves 50 % of the screen open while the right one covers two thirds. We performed informal evaluations trying shapes of different complexity and comparing short to long shapes. Based on this, we decided to use the 50 % design for the final prototype (see figure 2, D).

### Informal User Statements

The prototype presents a significant improvement over the setup that was used in previous work [10]. It is thinner, more lightweight and most importantly, it feels like one device.

To get some preliminary feedback, we gave the device to people that took part in the user studies with the old prototype from [10] and asked them to comment on it. All of them had a very positive reaction to the redesigned device. The most common comments were that the new prototype 1) feels better and is easier to hold, 2) looks much better and 3) gives the impression that it is an actual device rather than two smartphones stuck together. Even though we did not formally evaluate the design and thus only got qualitative statements, we believe that this supports the design choice to create an enhanced prototype.

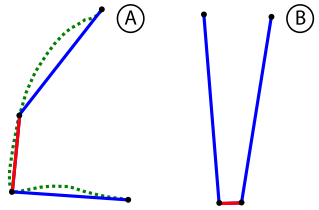



**Figure 3. Prototype algorithm: (A) The user's input (dotted green) is translated into strokes using the ShortStraw algorithm. Duplicates "Up, Up" or "Left, Left" are removed. (B) To avoid unwanted strokes, all strokes that are shorter than 25 % of the average stroke in a shape are removed from the results.**

### Stroke Recognition

To identify horizontal and vertical strokes, we used the ShortStraw algorithm [27]. Every time the finger is lifted from the touchscreen, all touch points are analyzed and the strokes are extracted. For instance, strokes with an angle less than 45 % from a perfect vertical line are counted as "Up" or "Down" respectively. Due to this, our algorithm can create duplicates based on the data created by the ShortStraw algorithm (see figure 3, left) that are removed from the result.

The resulting strokes are further processed to minimize errors. We set the minimum length of a stroke to 60 pixels to avoid single taps from being counted. Furthermore, when reading the shape from the path reduced by the ShortStraw algorithm, we ignore strokes that are shorter than 25 % of the average length of all strokes in the shape. This way, small fragments do not lead to input errors. An exemplary fragment is shown on the right-hand side of figure 3.

### Theoretical Security Analysis

A shape in XSide consists of an arbitrary number of the four possible strokes shown in figure 4, left. One rule is that the same stroke cannot be used two times in a row. For instance, "Up, Up" is not possible and the algorithm will make a single "Up" of it. This means that each stroke can be followed by one of the three remaining strokes. For instance, for shapes with a maximum of two strokes as used in the study, there are thus $4$ single-stroke shapes and $4 * 3$ double-stroke shapes resulting in $4 + 12 = 16$ possible shapes per input. If a password consists of three shapes, the theoretical password space of XSide is $16^3 = 4.096$. However, this combination was chosen for the study to control for the influence of password length. In a real world implementation, the number of strokes

should be more diverse. Assuming shapes with a maximum of three strokes per shape, there are additionally $4 * 3 * 3$ triple-stroke shapes, resulting in $4 + 12 + 36 = 52$ possible shapes per input (password space: $52^3 = 140,608$). For comparison, a four-digit PIN has $10,000$ possible combinations and an eight-stroke Android pattern has a password space of $106,160$, if each point in the grid is only allowed once (the standard Android configuration). We would like to note here that even though password space is important, we argue that observation attacks are a much more serious threat in the mobile context than brute force attacks.

Since shapes, and thus the passwords of XSide are stored as strings, standard cryptographic approaches can be used to appropriately protect the stored passwords from on-device theft. Thus, they are never stored in clear text.

XSide improves shoulder surfing resistance (and resistance against video attacks) as the user can adapt the input to the current situation. For instance, if the user is standing (e.g. in a public train) and other people in the vicinity are sitting, the whole authentication process can take place on the front of the device. If the user is sitting and everyone else is standing, the back would be the input area of choice. Finally, in a mixed situation, switching sides during the input increases security since no single shoulder surfer should be able to see both sides at once. Monitoring XSide passwords, therefore, would require shoulder surfers (or cameras) on both sides of the device, one of them facing upwards from the ground.

Android unlock patterns are always performed in the same way on the same spot, not overwriting previous patterns. Thus, they are highly susceptible to smudge attacks [2]. XSide provides advanced smudge resistance, as the shapes of one password overwrite each other during input. Even if sides are switched during the input, two shapes will take place on the same side if at least three shapes are used.

## THREAT MODEL

In our threat model, the user is located in a (semi-)public setting, an environment which is not or only slightly under the control of the attacker [5]. Additionally, the attacker has the possibility to get in possession of the user's mobile device in our scenario (e.g. by theft).

The proposed system increases resistance against two kinds of possible attacks: 1) "Shoulder surfing", that is, people spying on the input from either side of the device and 2) smudge attacks [2], using residues on the screen to find out the original input once the device is in possession of the attacker.

## USER STUDY

The main focus of the user study was to analyze how switching sides during input affects usability and security. The study was also used to gain insights into differences of performing strokes on the front in comparison to doing them on the back.

### User Study Design

We used a repeated measures factorial design with two independent variables: *Begin* (front, back) and *Switches* (0, 1 Start, 1 End, 2). *Switches* refers to how often the side is switched during authentication. For instance, *Begin* "front" and *Switches* "1 Start" means that there is one switch after the first shape, i.e. the first shape is performed on the front, the second and third on the back. "1 End" respectively means that there is one switch performed before the last shape.

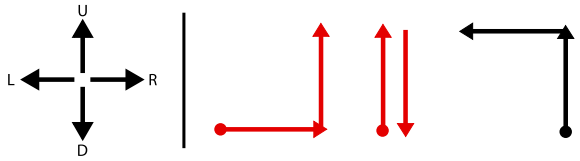



**Figure 4. Left: Passwords consist of different shapes. Shapes are made up of an arbitrary number of horizontal and vertical strokes. Right: An example of how the passwords were presented to the user study participants. Red passwords (the first two from the left) had to be input on the back, black ones on the front.**

Taking the two independent variables into account, the study had two baselines: all input on the front (Begin front + Switches 0) and all input on the back (Begin back + Switches 0). In the remainder of this paper, these will be called *Front Only* and *Back Only*. It should be noted that since we already know how the system performs in these conditions compared to PIN and Android patterns [10], those baselines were excluded from the current study. This way, the workload for the study participants could be reduced to minimize fatigue.

All study passwords consisted of three shapes with two strokes each. They were provided to the participants instead of allowing self-selection to control the passwords and draw unbiased conclusions about the influence of side switches. We also did not want to extend the study length to keep fatigue low. Each participant had a different set of passwords.

To minimize learning effects, we used an $8 \times 8$ Latin square design. That is, a multiple of eight was required as an optimal number of participants. The passwords for all eight inputs were provided on a piece of paper, which was positioned in front of the participants. The sides were color-coded on this list, as shown on the right-hand side of figure 4. Black shapes had to be input on the front and red shapes on the back. As opposed to the general concept presented above, the study prototype enforced using specific sides for the passwords (e.g. front, front, back). This was necessary to counterbalance the variables. Usability consequences of this decision will be discussed later in the results and discussion sections.

The study took place in an isolated study room on our premises. Only the participants and the experimenter were present during the study. Interaction was filmed with three cameras to analyze the participants' interactions, but mainly to perform a security analysis (see figure 5). Three cameras were chosen to simulate different positions of shoulder surfers. One camera was located behind the participant in the classical shoulder surfing position (to attack the "Front Only" input). The second camera was located in front of the user to film the back (to attack "Back Only"). The last camera was positioned either to the left or right of the users, depending on how they were holding the device (to see if this position would enable a shoulder surfer to perform the attack on inputs including side switches).

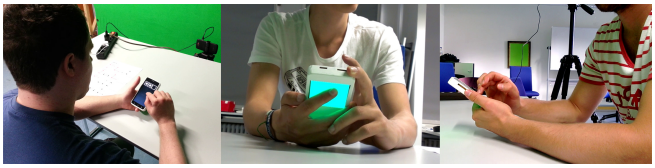


**Figure 5. The study was filmed from three angles for "Front Only", "Back Only" and interactions with side switches.**

### Procedure

At first, we provided a detailed explanation of the prototype and the concept to the participants. We told them that the cameras would be used to film the main phase of the study as well as for further evaluation after the study. The word "security" was never mentioned during the explanation.

After the introduction, a training phase was conducted to ensure that the participants were familiar enough with the system to avoid beginner mistakes. We provided a list of ten passwords for training, one for each authentication session. For each session, three tries were possible, a maximum of three wrong entries or one correct entry. During the training phase, all combinations of back and front entries were allowed. However, we encouraged the participants to at least once perform the whole input on the back, on the front and perform at least one time a single side switch and one time a double side switch. All participants followed this proposal. The training sessions took around 2-4 minutes.

In the main phase, the participant had to perform three authentication sessions for each of the eight combinations of the independent variables. For each session, the participants had three tries. That is, each session had a minimum of one input (correct) and a maximum of three inputs (three times incorrect or two incorrect and one correct). Summed up, each combination was used between three and nine times. With three authentication sessions, this made $8 * 3 = 24$ authentication sessions per participant. It should be noted once more that during the main phase, the expected input side was predefined and provided to the participants on a piece of paper.

Based on how the prototype was held by the participants, the side camera was repositioned for an optimal view on the input. The participants were free to choose their hand position as they preferred. One-handed input was allowed as well.

After the practical task, the participants were asked to complete a questionnaire, collecting demographic information as well as opinions about the system. Overall, the study took around 30 minutes per participant.

### Participants

We recruited 32 participants using mailing lists, word of mouth and social networks. Their average age was 25 years, (range: 19-38). 14 participants were female, 18 were male. 19 had a university-entrance diploma, 8 had a graduate degree, 3 were PhDs and 2 held a certificate of secondary education. The participants were of different professions and majors (if students). None of them was working in security. We ensured that none of them had taken part in any back-of-device interaction study before.

| Front | | Back | |
|---|---|---|---|
| 23.3 px | | 30.1 px | |
| 90 Degree | Back and Forth | 90 Degree | Back and Forth |
| 21.5 px | 27.2 px | 28.6 px | 32.7 px |

**Figure 6. Average deviations in pixels of participants' strokes from perfect horizontal and vertical lines.**

28 participants were touchscreen device owners, some of them possessing several such devices. They included smartphones (28), tablets (5), laptops with touchscreens (7), Nintendo DS (2), cameras (1) and convertibles (1). On average, they had been using them for 3 years (SD=1.95 years).

In general, the study participants were rather concerned about the data on their smartphones. 21 of the 28 stated to use lock screens. Furthermore, many additionally protect access to services on the devices (like Facebook or e-mail) using the integrated password functionalities of the respective apps. The top three protection mechanisms were PIN (15) Android pattern (11) and password (3). None of the participants used face recognition, even though some of their devices supported it.

In addition to using PINs, patterns and passwords, seven participants reported to use techniques to avoid shoulder surfing (four of them were victim to shoulder surfing before). Measures included covering the input (3) and moving the device out of the sight of onlookers (3). "Being fast" was mentioned one time as being an appropriate way to avoid shoulder surfing. Each participant received 5 Euro at the end of the study.

## RESULTS

Having 32 participants, the quantitative results of the study are based on $(8 * 3) * 32 = 768$ authentication sessions. All data was normally distributed and therefore analyzed with repeated measures ANOVAs.

### Accuracy

The results confirm the appropriateness of relative strokes for back-of-device authentication. We were also interested in whether there are differences in accuracy when performing the shapes on the back and on the front. Due to a logging error that was found halfway through the study, only the data of the last 16 participants could be used for this analysis.

The assumption for this analysis is that participants tried to make horizontal and vertical strokes as instructed. That is, in the best case, they would draw perfect lines. For horizontal strokes, this means that the Y-coordinate does not change. For vertical strokes, the X-coordinate should stay unchanged. Our measure for input accuracy was therefore the average deviation from these perfect lines, measured in pixels.

For the analysis, accuracy for each password was averaged over all inputs of a participant. The average deviations for all inputs were quite low. For all sides and shapes, the horizontal and vertical deviation was 26.7 pixels on average. Given the pixel density of 316 pixels per inch for our prototype, this equals a deviation of 2.15mm or about the fifth of the touch area of an index finger. Figure 6 depicts the detailed values for front, back, 90 degree (e.g. "Down, Left") and back-and-forth strokes (e.g. "Left, Right").

A 2 x 2 (*Side* (front or back) x *Shape* (90 degree or back and forth)) within participants analysis of variance revealed a significant main effect for *Side* ($F_{1,15} = 16.85, p < .001$). *Shape* did not have a significant effect and there were no interaction effects (all $p > .05$).

### Error Rates

In previous work [10, 25], it has been shown that it is beneficial to distinguish between two kinds of errors: basic and critical errors. A basic error refers to an authentication session that was successful overall, but took the participant two or three times to correctly authenticate. A critical error means that the complete authentication session failed, i. e. the password was wrongly entered three times in a row. Distinguishing these errors makes sense as for many systems like ATMs or mobile phones, there is an upper limit for the maximum number of tries and this number oftentimes is "3".

Figure 7 depicts the number of errors that occurred while entering the different combinations of the independent variables. Overall, basic errors happened in 93 out of 768 authentication sessions (12 %). "Back, Back, Front" was the combination with most basic errors (16). For the statistical analysis, the errors for each password were accumulated for each participant. A 2 x 4 (*Begin* x *Switches*) within participants analysis of variance of basic errors revealed no significant main effects as well as no significant interaction effects (all $p > .05$).

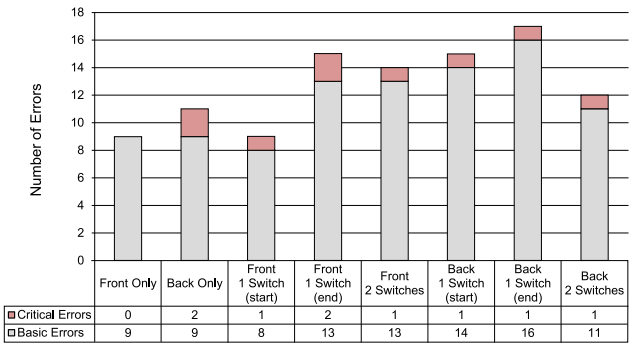
Critical errors were created in 9 instances out of 768 (1.2 %). "Front, Front, Back" and "Back Only" had the most critical inputs with 2 each. "Front Only" was the only combination with no critical error. As for the basic errors, a 2 x 4 (*Begin* x *Switches*) within participants analysis of variance showed no significant main or interaction effects (all $p > .05$).

#### *Error Categories*

Due to the fact that each session could consist of a maximum of three attempts and that basic errors could consist of two incorrect entries and critical errors always consisted of three failed attempts, the overall number of wrong inputs is bigger than basic plus critical.

Overall, there were a total of 140 erroneous authentication attempts. Having a closer look at the logs and the videos, we could identify five main error categories:

1. *Wrong order/mirrored shapes* caused errors in 55 instances (39.3 % of all wrong inputs). This means that participants mixed up which shape they intended to draw. In some cases, they mixed up left and right or up and down, for example drawing "left, right" instead of "right, left". Other shapes were the mirror image of the desired shape, drawing "left, up" instead of "right, down".
2. Drawing a *shape on the wrong side* of the prototype caused 29 errors (20.7 %). Please note that these are only counted due to the strict study design with pre-defined sides.
3. *Slips* caused 27 errors (19.3 %). For instance, slips happened when participants lifted their finger in the middle of a shape, did not put their finger down early enough or reached the edge of the screen while drawing.




| | Front Only | Back Only | Front 1 Switch (start) | Front 1 Switch (end) | Front 2 Switches | Back 1 Switch (start) | Back 1 Switch (end) | Back 2 Switches |
|---|---|---|---|---|---|---|---|---|
| Critical Errors | 0 | 2 | 1 | 2 | 1 | 1 | 1 | 1 |
| Basic Errors | 9 | 9 | 8 | 13 | 13 | 14 | 16 | 11 |

**Figure 7. Basic and critical errors for all combinations of the independent variables.**

4. 20 shapes (14.3 %) were incorrect because *extra* strokes were detected. This happened mostly at the end or the beginning of shapes but there were also instances with additional strokes in the middle of a shape.
5. *Other*: Lastly, 9 errors (6.4 %) were caused by unexplained touches during the input of a shape. These inputs were also not visible on the video and were mostly caused by the prototype casing not shielding the back device completely or by accidental touches from the participant.

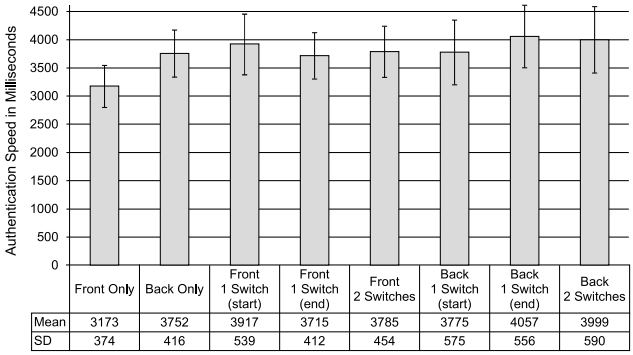
### Authentication Speed

Authentication speed was measured from the first touch to the last touch of the authentication session. To make sure that switching between sessions did not influence the time, the participants had to confirm the start of a session by clicking a button.

For the authentication speed analysis, only successful authentication sessions were counted. There was no instance in which a participant created critical errors for all three authentication sessions of the different combinations of *Begin* and *Switches*. Thus, all participants contributed to the authentication speed calculation for all eight cases. For each user and each combination of the independent variables, the average authentication speed of all successful tries were used.

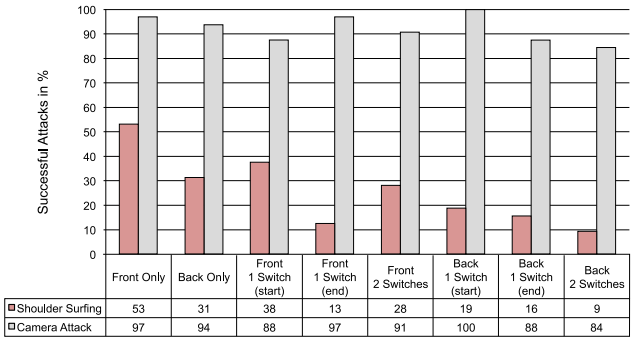
Figure 8 shows the average authentication speed. "Front Only" was the fastest input (M=3.1s) and starting the input on the back with one switch at the end was the slowest (M=4.1s). The numbers show that while switching the side adds time to the input, all combinations are rather fast to use. It is also interesting to have a look at the fastest participants of all combinations. Again, "Front Only" performed best with a participant achieving 2.1 seconds. All other top inputs were between 2.1 and 2.5 seconds (beginning at the front with two switches being the slowest).

A 2 x 4 (*Begin* x *Switches*) within participants analysis of variance of authentication speed revealed highly significant main effects for *Begin* ($F_{1,31} = 14.673, p < .001$) and *Switches* ($F_{2.310,71.601} = 12.18, p < .001$; Greenhouse-Geisser corrected). We also found a significant interaction effect for *Begin* x *Switches* ($F_{3,93} = 5,438, p < .05$).




| | Front Only | Back Only | Front 1 Switch (start) | Front 1 Switch (end) | Front 2 Switches | Back 1 Switch (start) | Back 1 Switch (end) | Back 2 Switches |
|---|---|---|---|---|---|---|---|---|
| Mean | 3173 | 3752 | 3917 | 3715 | 3785 | 3775 | 4057 | 3999 |
| SD | 374 | 416 | 539 | 412 | 454 | 575 | 556 | 590 |

**Figure 8. Average time required to authenticate with the different levels of the independent variables.**




| | Front Only | Back Only | Front 1 Switch (start) | Front 1 Switch (end) | Front 2 Switches | Back 1 Switch (start) | Back 1 Switch (end) | Back 2 Switches |
|---|---|---|---|---|---|---|---|---|
| Shoulder Surfing | 53 | 31 | 38 | 13 | 28 | 19 | 16 | 9 |
| Camera Attack | 97 | 94 | 88 | 97 | 91 | 100 | 88 | 84 |

**Figure 9. Successful shoulder surfing and video attacks in percentage.**

Post-hoc tests (Bonferroni corrected) revealed that passwords begun on the front were significantly faster than those started on the back (around 0.2 seconds, $p < .001$). Furthermore, post-hoc tests for *Switches* showed that passwords with zero side switches were significantly faster than those having one switch at the start ($p < .05$), one switch at the end ($p < .001$) or two switches ($p < .001$). The remaining differences are not statistically significant.

### Security

For the security analysis, we recruited four volunteers (two male, two female) to perform the attacks and gave them a monetary incentive to guess as many passwords as possible. Attackers had high security awareness and were all involved in authentication projects before. They received a 20 Euro basic salary plus 30 Cents for each successful shoulder surfing attack and 20 Cents for successful camera attacks. It took an attacker around four hours to attack eight participants (not counting breaks etcetera). Thus, we split the attacks into two two-hour sessions to reduce mental load (they reported getting tired after around two hours). 16 hours of video analysis and shoulder surfing attacks were performed by the volunteers. None of the attackers had participated in the user study. Each attacker was responsible for eight of the 32 study participants. The sequence of the attacks was counterbalanced.

We used the video material recorded during the study to perform the attacks. The last successful input for each password was used and cut in a way that it only showed the authentication itself, skipping all additional input before and after authentication. The attackers knew that all passwords consisted of three two-stroke shapes. They were also told the order of sides on which the password was input (e.g. "back, front, front"). This way, the attack can be considered a worst-case-scenario for the users. All audio and video information that would have given away the password was removed from the material. Since each participant performed each combination of the independent variables correctly at least once, none of them had to be excluded from the security analysis.

At first, XSide was explained to all attackers in detail. After that, they performed the same training task with the same prototype as the study participants. This allowed them to get familiar with the system.

There were two types of attacks: a simulated shoulder surfing attack as well as a camera attack. For the shoulder surfing attack, the video material was played to the attackers once. After this, they had three tries to correctly guess the input. If they failed, they could use the video material in which way and however long they wanted (e.g. forward, pause, repeat) and got another three guesses to find out the password. This was called a camera attack. If the shoulder surfing attack was successful, it was also counted as a successful camera attack. Throughout the whole session, attackers were allowed to take notes using pen and paper.

The results of the attacks are depicted in figure 9. As can be seen, video attacks were highly successful no matter which combination of the independent variables was used. With respect to shoulder surfing, "Front Only" was the easiest combination to attack with a 53 % success rate (17 out of 32). It is followed by passwords beginning at the front with one switch at the start (12; 38 %) and "Back Only" (10; 31 %). The graph further shows that passwords starting on the back show a tendency to be safer than passwords that start on the front. Authentication sessions that started on the back and had two side switches were the safest for both, camera attacks (27; 84 %) and shoulder surfing attacks (3; 9 %).

During the task, the attackers continuously commented on how and why they could or could not figure out passwords. In several occasions, attackers mentioned that the users were performing additional movements with their fingers between the shapes. These movements made it harder to distinguish between the actual input and the fingers moving while not touching the screen. One attacker called this behavior "finger-mumbling" as a reference to unclear finger movements. The attackers also stated several times that slow passwords were very easy to spy on and fast inputs were very hard to see. With respect to the input on the back vs. on the front, the attackers were in unison about the fact that it was harder to correctly distinguish angles for back input. Finally, there were some shoulder surfing attacks during which the attackers successfully guessed a shape that they did not see. This was made easier by the fact that due to the study setting they knew that the shape had to consist of two strokes. As such this represents a worst-case-scenario, as in real life they would not have this information.

### Questionnaire Results

In the post study questionnaire, we asked the participants to rank four different combinations: "Front Only", "Back Only", "One Switch" and "Two Switches". We did not differentiate all eight combinations of the independent variables as we thought that distinguishing between their fine nuances would have been too complex to ask in the questionnaire. The ranking was completed for ease-of-use, speed and security.

For security, the majority of participants (22) said that passwords with two side switches were the most secure ones. Seven voted for "Back Only", two for "One Switch" and one participant for "Front Only". "Front Only" was voted the least secure by 28 participants.

When ranking ease-of-use and speed, the picture looks different. Almost all participants ranked "Front Only" as number one (30 for ease-of-use and 31 for speed). One participant ranked "One Switch" to be the fastest and easiest approach and one thought that "Back Only" was the easiest password. "Two Switches" ranked last in both categories (27 votes for ease-of-use and speed).

Additionally, the participants were asked to rate speed, ease-of-use and security on 5-point Likert scales. The results are similar to the rankings, with "Front Only" getting most positive votes for speed and ease-of-use and the most negative votes for security. For the detailed ratings see figure 10 (ease-of-use is not listed as it was almost identical to speed).

We also asked the participants whether they would use an authentication system that uses both sides of their smartphone. 25 answered this question with "Yes". Among the seven participants that answered "No", four currently do not use any form of authentication mechanism on their smartphone.

## DISCUSSION

### Stroke-Based Authentication on Smartphones

As mentioned before, the stroke-based passwords used in XSide were chosen since absolute pointing or movements on the back are very hard without feedback. Their main advantages are that they work eyes-free, do not require accurate pointing or dragging and they are, to a certain extent, more robust against smudge attacks.

The study showed that using stroke-based passwords was an appropriate decision. They work well on both sides of the device (both in terms of accuracy and speed). Some participants were rather skeptical at the beginning of the study and were surprised how well it actually worked. One participant even stated: "I am amazed about how easy it is to input the 'code' on the back with the index finger. This is totally different from what I expected."

In general, we argue that authentication mechanisms based on strokes, be it the Android pattern or extensions of it [6] are very well suited for smartphones with touchscreens. Current studies showed that even if they have disadvantages compared to PIN with respect to error rates and speed, users highly prefer them [24]. This can be partially attributed to their playful character and them being more suitable for smartphones.

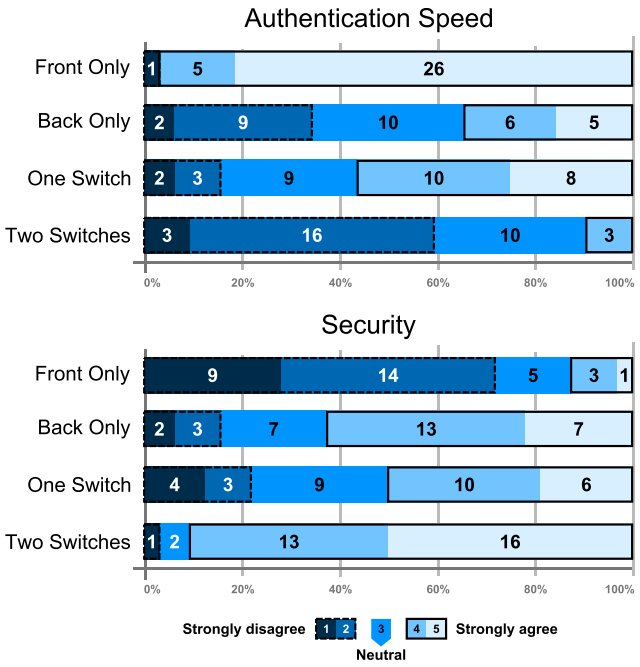



**Figure 10. Likert scales ratings for "X was fast to use" and "X is secure".**

### Performance

Switching the sides naturally adds a certain amount of time to the input as the hand has to be moved when using two-handed input or rearranged when using one-handed input. We were surprised to find that this influence was smaller than expected. For instance, front entry with one switch to the back at the end was even slightly faster than "Back Only". At the same time, it was more resilient to shoulder surfing attacks than "Back Only". This is a great success for secure and usable authentication on smartphones.

With respect to error rates, the differences were also much smaller than expected. The absolute numbers of speed and error rates are even more surprising when considering that all study participants were first time users of the system. We can therefore expect performance to improve once the system is used over a longer period of time and with "real" devices.

Overall, the study results are highly encouraging and we argue that XSide presents a good alternative for authentication on mobile devices. As mentioned earlier, it is only a matter of time until smartphones with touch-sensitive rear sides will hit the market. Thus, the requirements for deploying XSide are likely to be met in the nearer future.

### Minimizing Errors

Overall, error rates for all combinations of XSide were rather low. There was not a single authentication session which failed completely due to three critical errors. However, looking at the detailed error analysis, it can be expected that XSide will perform even better in real world use.

Firstly, 20.7 % of all errors happened as the shapes were drawn on the wrong side. This error was only an error in the context of our study, in which we forced the participants

to use specific combinations. The actual XSide concepts allows input to take place wherever the user wants it to be. This means that these errors will not be possible outside the lab study.

Closely related to this is the category of wrongly ordered or mirrored shapes (39.3 % of all errors). We argue that such problems will be reduced by motor memory effects [11].

While the algorithm was already highly improved as compared to the original system [10], there is still space for improvement. This could help to minimize the errors in which extra strokes were detected.

Finally, slips cannot be avoided but they are a kind of error category that seems to be rather acceptable for stroke-based authentication systems [24]. Directly comparing the results with the error rates from [10] is not possible due to different variables and study designs but we can informally state that there was an improvement in slip rates. We attribute this mainly to the improved form factor of the prototype. An even more lightweight and thinner device thus has the potential to further reduce the chance for slips.

### Switching Sides

The results of the security analysis show that passwords during which the sides were switched were in most cases more resilient to shoulder surfing attacks than "Front Only" or "Back Only" input. At the same time, authentication speed remains high, error rates remain low and accuracy stays high. Including the results from the questionnaire, we find that "Front Only" was rated the fastest and easiest. However, input with one and two side switches received good ratings as well. The quantitative results back this as "Front Only" was slightly faster and less error-prone than the alternatives.

Looking at security, participants were aware that switching the sides increases security. The quantitative results support this and show that even with one switch, security can be improved. This great level of understanding of the security-speed trade-off makes us believe that there is a good chance that people will actually adapt the security when using the system in the wild.

A major benefit of XSide is that the flexible side switching capability allows users to adapt the security of the input to their current context. Thus, they can use the easy and fast "Front Only" when in safe environments, but can spontaneously use switches to increase security when they feel threatened. To the best of our knowledge, this is the first system that allows the user to increase the security of the authentication mechanism on-demand. This opens up many interesting avenues for future research. The main question will be whether people will actually employ side switches in the wild. It is promising that in the questionnaire, a majority of participants stated that they would use such a system. However, these results have to be handled with care as they might be influenced by lab study effects, such as demand characteristics.

### Real World Security

We have several reasons to believe that the real world security of XSide will be higher than what we found in the study: None of the participants was trying to hide the password entry. Even more, due to the fact that they were in a user study, they even tried to hold the device in a way that it was easier for us to film the input (this was mentioned by several participants during and after the study). This behavior differs from what we expect from real life use.

With respect to password diversity, all passwords in the study consisted of three shapes with two strokes each. This provided the attackers with an advantage they would not have in the real world as they could more easily guess shapes they did not see. In a real world situation, the structure of the passwords would be unknown to an attacker. Related to this, the attackers had another advantage as the video material was cut in a way that it only showed the password entry. All interactions before and after it were removed.

## LIMITATIONS

Due to the lack of a long-term study, we do not know how XSide will be used over time. There is a chance that users will use only the front or always the same combination of sides. This would reduce the security of the system (especially if "Front Only" is used). Furthermore, as memorability can only seriously be evaluated within long-term settings, we cannot make certain predictions how memorable the passwords of XSide really are. The fact that we are using a stroke-based approach speaks for the system but a real evaluation will be required in the future.

For the experiment, we used a self-recruited sample. The questionnaires show that the technology affinity of these users was higher than would be expected from a general population. Thus, the results might differ for other demographics.

## CONCLUSIONS AND FUTURE WORK

In this paper, we presented XSide, a stroke-based authentication mechanism for smartphones that uses the back and the front of the device to enter passwords. We were specifically interested in how switching the sides during input affects security and performance of the system. The study results show that switching only slightly decreases speed ($\leq$ 4 seconds per authentication session) and that the same applies for error rates. Security, on the other hand, is increased if side switching is applied. To the best of our knowledge this is the first system in which a user can spontaneously choose a usability/security trade-off. The results from our questionnaire indicate that there is the desire for more protection in dangerous situations. XSide can offer that without impacting the usability of all other (safe-environment) authentication sessions. This capability opens up a new and highly interesting research area. In future work, we will look into how people utilize this kind of adaptive ability and whether it increases acceptance of authentication technology in smartphones.

With the improved form factor of the prototype, the next step will be to conduct real-world studies. This will also provide insights into memorability properties of the system and allow us to explore real password composition strategies. In addition to improvements to the prototype, we believe it is likely that commercial products with capacitive sensing on the back side will hit the market soon.

**ACKNOWLEDGMENTS**
This work was partially funded by a Google Research Award.

**REFERENCES**

1. Angulo, J., and Wästlund, E. Exploring touch-screen biometrics for user identification on smart phones. In *Privacy and Identity Management for Life*. Springer, 2012, 130–143.
2. Aviv, A., Gibson, K., Mossop, E., Blaze, M., and Smith, J. Smudge attacks on smartphone touch screens. In *Proc. USENIX 2010*, USENIX Association (2010), 1–7.
3. Bianchi, A., Oakley, I., Kostakos, V., and Kwon, D. S. The phone lock: audio and haptic shoulder-surfing resistant pin entry methods for mobile devices. In *Proc. TEI 2011*, ACM (2011), 197–200.
4. Biddle, R., Chiasson, S., and Van Oorschot, P. C. Graphical passwords: Learning from the first twelve years. *ACM CSUR 2012 44*, 4 (2012), 19.
5. Carr, S. *Public Space*. Cambridge Univ Pr, 1992.
6. Chiang, H.-Y., and Chiasson, S. Improving user authentication on mobile devices: a touchscreen graphical password. In *Proc. MobileHCI 2013*, ACM (2013), 251–260.
7. Cunningham, A. Hands-on with LGs G2 smartphone (and the buttons on the back). `http://arstechnica.com/gadgets/2013/08/hands-on-with-lgs-g2-smartphone-and-the-buttons-on-the-back/`, Aug. 2013. Accessed: September 8, 2013.
8. De Luca, A., Denzel, M., and Hussmann, H. Look into my eyes!: Can you guess my password? In *Proc. SOUPS 2009*, ACM (2009), 7.
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 *Proc. CHI 2012*, ACM (2012), 987–996.
10. De Luca, A., von Zezschwitz, E., Nguyen, N. D. H., Maurer, M.-E., Rubegni, E., Scipioni, M. P., and Langheinrich, M. Back-of-device authentication on smartphones. In *Proc. CHI 2013*, ACM (2013), 2389–2398.
11. Fleishman, E. A., and Parker Jr, J. F. Factors in the retention and relearning of perceptual-motor skill. *Journal of Experimental Psychology 64*, 3 (1962), 215.
12. Gao, H., Guo, X., Chen, X., Wang, L., and Liu, X. Yagp: Yet another graphical password strategy. In *Proc. ACSAC 2008*, IEEE (2008), 121–129.
13. Hafiz, M. D., Abdullah, A. H., Ithnin, N., and Mammi, H. K. Towards identifying usability and security features of graphical password in knowledge based authentication technique. In *Proc. AICMS 08*, IEEE (2008), 396–403.
14. Hayashi, E., Dhamija, R., Christin, N., and Perrig, A. Use your illusion: secure authentication usable anywhere. In *Proc. SOUPS 2008*, ACM (2008), 35–45.
15. Jermyn, I., Mayer, A., Monrose, F., Reiter, M. K., Rubin, A. D., et al. The design and analysis of graphical passwords. In *Proc. USENIX 1999*, Washington DC (1999), 1–14.
16. Karlson, A. K., Brush, A. B., and Schechter, S. Can I borrow your phone?: Understanding concerns when sharing mobile phones. In *Proc. CHI 2009*, ACM (2009), 1647–1650.
17. Kennedy, D., and Osuga, R. Transparent double-sided touchscreen display Android smartphone prototype. `http://www.diginfo.tv/v/12-0099-r-en.php`, May 2012. Accessed: September 9, 2013.
18. Kim, S.-H., Kim, J.-W., Kim, S.-Y., and Cho, H.-G. A new shoulder-surfing resistant password for mobile environments. In *Proc. ICUIMC 2011*, ACM (2011), 27.
19. Li, Z., Sun, Q., Lian, Y., and Giusto, D. D. An association-based graphical password design resistant to shoulder-surfing attack. In *Proc. ICME 2005*, IEEE (2005), 245–248.
20. Renaud, K., and De Angeli, A. Visual passwords: cure-all or snake-oil? *Communications of the ACM 52*, 12 (2009), 135–140.
21. Schaub, F., Deyhle, R., and Weber, M. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proc. MUM 2012*, ACM (2012), 13:1–13:10.
22. Standing, L. Learning 10000 pictures. *The Quarterly journal of experimental psychology 25*, 2 (1973), 207–222.
23. Tao, H., and Adams, C. Pass-go: A proposal to improve the usability of graphical passwords. *IJ Network Security 7*, 2 (2008), 273–292.
24. 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 *Proc. MobileHCI 2013*, ACM (2013), 261–270.
25. von Zezschwitz, E., Koslow, A., De Luca, A., and Hussmann, H. Making graphic-based authentication secure against smudge attacks. In *Proc. IUI 2013*, ACM (2013), 277–286.
26. Wiedenbeck, S., Waters, J., Sobrado, L., and Birget, J.-C. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proc. AVI 2006*, ACM (2006), 177–184.
27. Wolin, A., Eoff, B., and Hammond, T. Shortstraw: A simple and effective corner finder for polylines. In *Proc. Eurographics 2008* (2008), 3340.
28. Zakaria, N. H., Griffiths, D., Brostoff, S., and Yan, J. Shoulder surfing defence for recall-based graphical passwords. In *Proc. SOUPS 2011*, ACM (2011), 6.
29. Zhao, H., and Li, X. S3pas: A scalable shoulder-surfing resistant textual-graphical password authentication scheme. In *AINAW 2007*, vol. 2, IEEE (2007), 467–472.