Back-of-Device Authentication on Smartphones

Alexander De Luca[1], **Emanuel von Zezschwitz**[1], **Ngo Dieu Huong Nguyen**[1], **Max-Emanuel Maurer**[1], **Elisa Rubegni**[2], **Marcello Paolo Scipioni**[2], **Marc Langheinrich**[2]
[1]Media Informatics Group, University of Munich (LMU), Munich, Germany
[2]University of Lugano (USI), Lugano, Switzerland
{alexander.de.luca, emanuel.von.zezschwitz, max.maurer}@ifi.lmu.de, nguyenn@cip.ifi.lmu.de
{elisa.rubegni, marcello.paolo.scipioni, marc.langheinrich}@usi.ch

ABSTRACT
This paper presents *BoD Shapes*, a novel authentication method for smartphones that uses the back of the device for input. We argue that this increases the resistance to shoulder surfing while remaining reasonably fast and easy-to-use. We performed a user study ($n = 24$) comparing BoD Shapes to PIN authentication, Android grid unlock, and a front version of our system. Testing a front version allowed us to directly compare performance and security measures between front and back authentication. Our results show that BoD Shapes is significantly more secure than the three other approaches. While performance declined, our results show that BoD Shapes can be very fast (up to 1.5 seconds in the user study) and that learning effects have an influence on its performance. This indicates that speed improvements can be expected in long-term use.

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

General Terms
Human Factors; Performance; Security

INTRODUCTION
Today's smartphones provide a whole new level of user experience. Even though their main use case is still communication [6], they now hold large amounts of potentially sensitive information that go well beyond call logs: emails, photos, chat messages, and both private and professional documents. Smartphones are also increasingly used as security tokens (e.g., e-banking, Google 2-way authentication). Many users are concerned about others accessing this wealth of information, should they ever lose their phone [20].

CHI 2013, April 27–May 2, 2013, Paris, France.

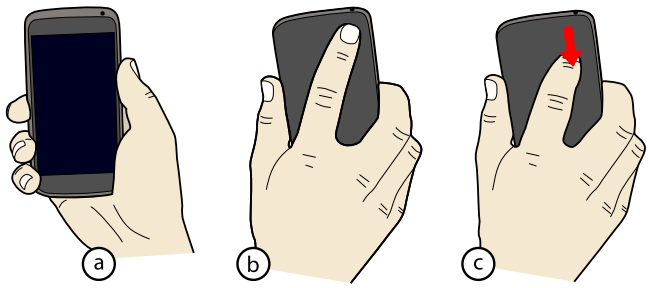


Figure 1. BoD (Back-of-Device) Shapes authentication concept. a) Typical hand posture when using one-handed input for authentication. b) The user authenticates by performing a row of simple shapes on the back. c) Example of a user performing a single-stroke shape ("Down").

Current standard authentication systems for smartphones include passwords, PINs, and Android's grid unlock (similar to draw-a-secret [19]). Research has shown that they are not safe and easily susceptible to simple attacks like shoulder surfing [13, 26] or so-called "smudge attacks" [1], which use smudge stains on the display to infer the password pattern. Clark et al. have found that PIN and password users are, to a certain degree, aware of these weaknesses and are highly concerned of others getting in possession of their PINs or passwords [10].

Preventing shoulder surfing attacks is difficult as long as the authentication process takes place at the front of the device – which is the area that is most easy to look at. By shifting the authentication step to the *back* of the device, this confidential input is moved out of the "normal" view of possible observers, rendering attacks much more difficult.

Back-of-device interaction has been proposed to address the so-called "fat finger" problem [24] where the user's finger occludes information during touchscreen interaction, in particular on small devices [3]. Back-of-device interaction is already commercially available in the form of the Playstation Vita, since early 2012, which comes with so-called "rear touch". In July 2012, NTT Docomo showed a smartphone prototype with support for back-of-device interaction [21]. It is thus very likely that commercial smartphones will soon feature touch sensitive backsides as well.

We developed several concepts for supporting back-of-device authentication on smartphones. Based on theoretical analyses and a formal user study, we identified the best candidate, "BoD Shapes". This paper describes BoD Shapes in detail and presents a user study that we conducted to evaluate its

usability and security problems. Our results show that BoD Shapes offers enhanced security properties when compared to standard front-facing authentication methods, while remaining reasonably fast to use.

RELATED WORK

There is no shortage of alternative smartphone authentication methods. The most prominent category is ad-hoc (i.e., immediate) authentication using a secret known to the user, such as a PIN. Alternatively, biometric data or a combination of a secret and biometric data can be used.

Ad-Hoc Authentication

The main goal of most ad-hoc authentication methods is securing the phone from unwanted access. In many cases, additional hardware-capabilities of the mobile device are used. In Vibrapass, De Luca et al. [13] proposed to enrich the input of a PIN or password with "lies": whenever the phone vibrates, the user should input a wrong digit or character. As a casual observer cannot detect the vibrations, differentiating between the "real" secret and false input becomes difficult for the attacker. However, due to the randomness of this approach, memorizing passwords or PINs over time becomes much more difficult for the user. In a similar way, Phonelock [4] and Spinlock [5] by Bianchi et al. use body-worn tactile actuators as an invisible communication channel to the user, thus suffering from the same memorability issues. In addition, they significantly slow down authentication speed. Instead of a hidden secondary channel, we move the interaction to the back of the device, thus hiding it from a casual attacker.

Another approach to securing authentication is to constantly vary the security question, so that observing a single entry will not allow an observer to answer subsequent challenges. While memorizing many different PINs or password is impractical, graphical memory is well suited for this. The Awase system by Takada et al. [25] requires a user to identify one's favorite pictures from a (changing) set of images in order to authenticate. Dunphy et al. [14] conducted a large study on two such image-based authentication systems and their applicability to mobile devices. While they could attest good usability properties, the systems' security properties were not satisfying. Our approach addresses memorability issues by using a shape-based authentication system that exploits the users' motor memory [15, 23].

Chong et al. [8] propose the use of discrete gestures performed with the mobile device. Azenkot et al. [2] present an approach that uses multi-touch taps on a touchscreen device to enable authentication for blind or visually impaired users. Both systems allow for a very large password space while still being fast. Their disadvantage is that authentication can be easily observed. As a solution to this problem, Kirschnick et al. [22] propose to use biometric properties in the process. Similarly, De Luca et al. [11] propose to overlay the standard Android grid unlock with biometric authentication. This way, they try to ensure that even if attackers are in possession of the device and know the password, they will still be rejected as they perform it differently. The main problems of biometric systems are high false rejection rates.

Other biometric approaches require more time to make a decision over acceptance or rejection. These include systems analyzing the users' keystrokes on a virtual keyboard [9], recognize users based on their gait [16], their file system activities [29], or by using behavioral patterns [18]. The disadvantage is that they leave a large window for attacks. Thus, they need to be combined with an ad-hoc authentication system.

Implications for Back-of-Device Authentication

Our review of prior work leads to three important implications that we wanted to fulfill in the design of our authentication method:

1. While back-of-device authentication requires additional hardware to make the input more secure, we chose hardware that is very likely to hit the mobile phone market soon, is relatively cheap (e.g. capacitive technology), and that allows for other useful applications as well [3]. This increases the chances of the system actually being realized.

2. Besides being vulnerable to direct observational attacks, pattern-based authentication mechanisms have manifold advantages. We wanted to exploit these advantages like motor memory effects [23] and thus opted for a pattern-based system design.

3. A common approach to make authentication more secure is to add overhead to the input. This is something that we wanted to avoid. By performing the input on the back side of the device, we argue that it provides high resistance to shoulder surfing without burdening the user with a complex input system.

THREAT MODEL

We assume a "shoulder surfer" that is close to the user while authentication takes place. This includes standing close or sitting in the vicinity of the user. In most cases, such an attack will take place in a public or semi-public setting, an environment in which the user lacks full control [7]. Additionally, the attacker has the ability to gain possession of the user's device.

A back-of-device authentication method should in principle be more secure against such attacks, since input takes place in a position less visible to bystanders. In order to avoid suspicion, an attacker needs to be in a "natural" position with respect to the victim. Having an attacker duck or kneel down in front of the victim in order to get a better view would certainly raise suspicion. Camera attacks would need to be located close to the floor facing upwards, which should make it equally harder to correctly align the shot.

BACK-OF-DEVICE AUTHENTICATION CONCEPTS

The basic idea of using a back-of-device authentication method was to provide a mechanism that is fast to use and easy to memorize, while being significantly more secure against shoulder surfing than PIN, password, and grid unlock. During informal brainstorming sessions, we first developed several ideas for such an authentication method, then settled on two final candidates: "BoD Pattern Unlock" and "BoD Shapes", outlined below.

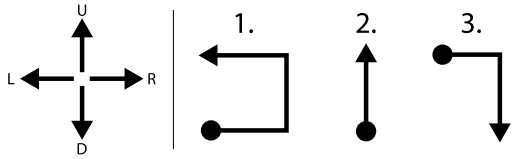


Figure 2. Left: A BoD Shapes shape can consist of an arbitrary combination of horizontal and vertical strokes. The letters indicate the internal representation. Right: An example for a BoD Shapes "password" consisting of three consecutive shapes.

Figure 3. Hardware Prototype. Left: Two devices connected back-to-back using bumpers. Right: The same device with two sponge rubber bands to avoid accidental interaction with the touchscreen.

BoD Pattern Unlock

The first candidate is a back-of-device version of the Android grid unlock. To authenticate, the user has to draw a shape between nine different dots located on the screen. To be shoulder surfing resistant, no feedback such as lines between the connected points must be shown. This means that the user has to correctly hit the (secret) start dot among the nine dots shown on the front of the device, using the touch sensitive back, then draw out the right pattern by dragging between the dots. The system thus requires the user to perform absolute movements between specific points on the back of the device.

BoD Shapes

The second candidate, BoD Shapes, uses relative rather than absolute movements. The secret input (called "password" in this paper) consists of the consecutive drawing of three shapes. Each shape is an arbitrary combination of up to three horizontal and vertical strokes (see figure 2, left). A "Down" stroke with one hand is shown in figure 1. The decision to use only horizontal and vertical strokes was made when pre-tests showed that diagonal movements were hard to perform on the back.

In contrast to BoD Pattern Unlock, this method does not require the user to precisely hit any target. In addition, the strokes drawn by the user have no predefined length, leaving it up to the user how to draw them. This supports the individual physical properties of the users' hand(s).

HARDWARE SETUP

In order to evaluate our concepts, we built a fully functional prototype that allowed us to simulate a smartphone with a touch-sensitive rear. The prototype uses two HTC One S smartphones mounted back-to-back and rotated by $180°$ in order to accommodate the protrusion of the camera lens. To allow for an easy replacement of a defective device, we used two hardcover protective shells that we glued to each other.

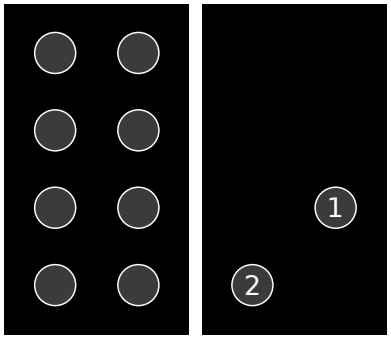


Figure 4. Accuracy study interface. Left: The screen was divided into eight areas, represented by the circles. Right: For each task, the user had to point at circle 1 and drag to circle 2.

Including the shells, the prototype was 1.5 cm thick. Figure 3 left, shows an assembled prototype. During the experiments, we used rubber band covers (shown in Figure 3, right) to prevent test subjects from accidentally hitting the standard capacitive buttons (e.g., Home, Back) on any of the devices.

We installed a custom application on each device that used Wi-Fi Direct to connect the two smartphones to each other, randomly assigning one of them to be the "rear" device. The application then collected touch events on the back and sent them to the front, translating each touch into the local coordinate system of the front device. While this approach allowed us to create a high-resolution prototype quickly, it also added considerable weight: with 269 grams, our prototype weighed around twice as much as current high-end smartphones (e.g. iPhone 4s: 140 grams; Samsung Galaxy S3: 133 grams).

CONCEPT FEASIBILITY: ACCURACY STUDY

To evaluate the feasibility of feedback-less pointing and dragging on the back of the device, we performed an accuracy study. We report the results and discuss their implications with respect to selecting the final concept.

Study Description

The study featured a simple pointing and dragging task. Given the 16:9 aspect ratio of the screen, we subdivided the area into $4x2 = 8$ evenly sized regions as shown in figure 4, left. Each task showed two circles, simply labeled "1" and "2" (see figure 4, right). The subject had to point at target 1, drag to target 2 and then lift off the finger, all on the back of the device. If one of the targets was not correctly hit, the task could be repeated up to three times. Testing all possible directions two times resulted in a minimum of $8 * 7 * 2 = 112$ input tasks, the order of which was randomized. The whole procedure was performed twice, once forcing the participants to use one hand only ("forced" mode) and the other time allowing them to use two hands (called "freestyle"). The order of forced and freestyle interaction was counterbalanced. The interface was displayed on the front device, interaction took place on the back. No virtual pointer was provided, since visual feedback would minimize the security of an authentication system based on such input. Before starting the actual task, participants were trained using random occurrences of the touch points.

Figure 5. Examples of how the device was held by participants in freestyle mode. This shows that preferences were quite different. In addition, this posture can theoretically influence the security of the authentication system.

Examples of different hand settings used in freestyle mode can be found in figure 5. Theoretically, this position can influence the security of the system. For instance, the position in figure 5, left, is harder to attack than 5, middle.

Participants and Results

We recruited 20 participants with an average age of 26 years (range: 19-38), seven female. Thus, the analysis is based on 4480 pointing and dragging tasks (without repetitions). Error rates are the most important indicators whether it is easy to hit a target on the back of the device. Errors in this work are defined by the users' inability to either correctly hit target 1, or drag to target 2. Overall, error rates for dragging were even higher.

Even though the prototype allowed for a certain threshold, that is, the user did not have to exactly hit the target but 50 pixels within its vicinity, target hit accuracy was low. To get a better overview of the results, we subsequently group the results into "top" and "bottom" targets. Figure 6 shows the amount of errors over all participants for hitting a start point at either the bottom or the top with one hand (forced) and in freestyle mode. The best case result – freestyle mode in the top area – saw 9% or 102 errors. The worst case – hitting a bottom target in forced (one-handed) mode – had 39% or 440 inputs that were not accurate enough. The overall worst performance was by a user failing at 40 of 112 inputs (36%).

A 2 x 2 (*Area* x *Hand Setting*) within-participants analysis of variance of error rate revealed a highly significant main effect for *Hand* ($F_{1.0,19.0} = 26.019, p < .001$, Greenhouse-Geisser corrected). Interaction in freestyle mode (M=5.4 errors) outperformed interaction in forced mode (M=18.7 errors). No main effect for *Area* ($p = .08$) and no interaction effect ($p = .102$) were found.

Overall, our feasibility study showed that it is hard to precisely hit a target and drag to another target using the back of the device, without providing a visible pointer. Of particular importance are the results for one-handed input, as the second hand is often occupied with a primary or secondary task [17].

Even with the relatively high threshold, one-handed back-of-device pointing seems not to be usable in practice.

CANDIDATE CONCEPT: BOD SHAPES

Based on the results of the accuracy study, we decided to abandon the BoD Pattern Unlock concept, as it would require users to accurately hit and drag without having a visible pointer. We thus chose BoD Shapes as the more promising candidate, given that it uses relative movements only. To

Target	Target 1		Target 2	
Top	409 (18%)		516 (23%)	
Bottom	554 (25%)		633 (28%)	
	Freestyle	**Forced**	**Freestyle**	**Forced**
Top	102 (9%)	307 (27%)	133 (11%)	383 (34%)
Bottom	114 (10%)	440 (39%)	137 (12%)	496 (44%)

Figure 6. Results from the accuracy study: Absolute numbers (percentage) for missing the start and end targets using the back of the device.

successfully use the system, it does neither matter where the interaction starts nor how long the strokes are. This should make it much easier to use without visible pointing feedback.

Authentication Process

BoD Shapes uses three consecutive shapes to authenticate. Each shape consists of a maximum of three strokes. A stroke can be one of Up, Down, Left, or Right. An example for a password is shown in figure 2, right.

In order to support single-handed operation, authentication takes place in the upper 40% of the back area. To make sure that the remaining area was not touchable in our prototype, we used sponge rubber as shown in figure 3, right. In several experiments with different materials, we found that sponge rubber offers a good trade-off between avoiding touch and acceptable weight. A typical hand posture and interaction example of this setting is shown in figure 1.

Strokes are extracted using the ShortStraw algorithm [28]. Every time the finger is lifted from the back, the preceding touch points are analyzed. In informal pre-tests we realized that the algorithm tends to duplicate simple strokes (e.g., detecting "Down Down" when the user instead entered "Down"). Since a single shape in BoD Shapes must always be a set of consecutive strokes, such duplicates are not possible. We can thus simply delete repeating strokes from the results. The final set of strokes is stored as a shape and the input is acknowledged by displaying a dot on the front device's screen (see figure 7). After three shapes are input, the system compares them to the user's stored secret and determines whether the authentication session was successful.

Theoretical Security Analysis

Given four strokes to choose from (Up, Down, Left, and Right) and the constraint of not directly repeating any stroke (e.g., "Down Down" would not be allowed), each stroke can be followed by one of three other strokes. There are thus 4 single-stroke shapes, $4 * 3$ double-stroke shapes, and $4 * 3 * 3$ triple-stroke shapes. A shape can have 1, 2, or 3 strokes, resulting in $4 + 12 + 36 = 52$ possibilities per shape. With three shapes, the theoretical password space of BoD Shapes is thus $52^3 = 140,608$. This is around 14 times bigger than for a four-digit PIN $(10,000)$, and more than three times bigger than a grid unlock pattern with up to 8 strokes $(49,536$ if each grid point is only allowed once). For comparison: the password space for three shapes with only up to two strokes each is only $(4 + 12)^3 = 4,096$.

To store the shape-based password, strokes are translated into characters as shown in figure 2, left. For example, the internal representation for "Up, Down, Left" is "UDL". This way, the

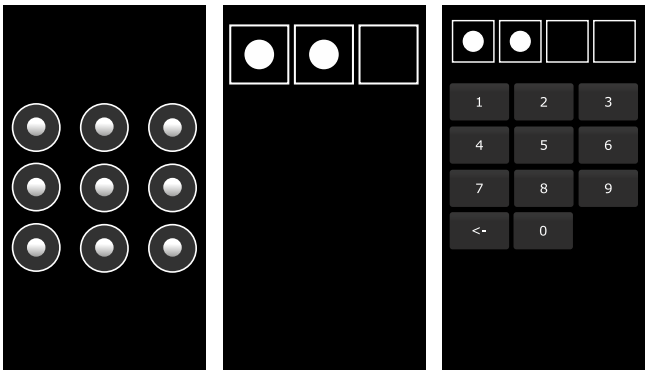


Figure 7. User interfaces of the four main study systems. Grid unlock, (BoD) Shapes and PIN (from left to right). Progress for PIN and Shapes (both front and back) is displayed with dots.

same secure storage methods that are used for character-based passwords and PINs can be used, i.e., password-shapes are not stored in plain text but only as a hash for comparison.

As opposed to grid unlock, BoD Shapes is resistant to smudge attacks (cf. [1]). This is because users tend to draw strokes at the same location, as observed in our study, which means that the smudges of different strokes are overlapping each other, making it hard or impossible to reconstruct the actual input from them. Finally, as with any back-of-device interaction, performing the input on the rear of the smartphone makes it harder to spy on (resistance to "shoulder surfing").

MAIN STUDY

The main goal of the study was to evaluate the usability and security properties of BoD Shapes in comparison to PIN and grid unlock. We also evaluated a front version of the Shapes authentication method ("Front Shapes") to directly measure performance and security differences of the back-of-device interaction. Note that Front Shapes also offered no visual feedback besides the confirmation dots, just like BoD Shapes.

User Study Design

We used a repeated measures factorial design with three independent variables: *System* (PIN, grid unlock, BoD Shapes, Front Shapes), *Password* (given, self-selected) and *Difficulty* (easy, hard). Difficulty refers to how difficult it is to enter the password. For instance, a 3-stroke shape takes longer to input and is harder to remember than a 2-stroke shape; a PIN with two identical digits in a row is faster to input and easier to remember than one without any repetition. PINs consisted of four digits and grid unlock shapes of five strokes. Both BoD Shapes and Front Shapes always used three shapes (cf. our security analysis above). The difficulty rules for the given as well as the user-defined passwords, were as follows (examples are depicted in figure 8):

- **PIN Easy**: Two identical digits in a row.
- **PIN Hard**: All digits different from each other.
- **Grid Easy**: All strokes to direct neighbors.
- **Grid Hard**: One stroke skipping a neighboring grid point.

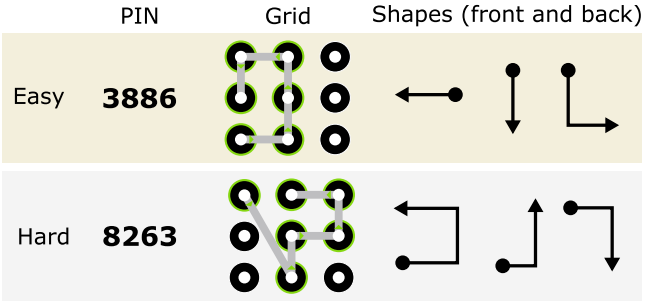


Figure 8. Examples for easy and hard PINs, patterns and shapes as used in the main user study.

- **Shapes Easy**: One shape 2 strokes, two shapes 1 stroke.
- **Shapes hard**: One shape 3 strokes, two shapes 2 strokes.

The study took place in an office with only the experimenter and the participant present and sitting opposite from each other. Two cameras recorded the interaction for performing a later security analysis. One was mounted at approximately eye height of an imaginary attacker standing behind the user (left side), one at approximately eye height of an equally imaginary observer seated across from the user. The front camera represented the "best case" for inconspicuously spying on the back-of-device input, the back camera was ideally placed for observing regular front-of-device input.

We implemented an application supporting the four authentication systems. While the back device was only required for BoD Shapes, it was not removed for the other tasks in order to have the same physical setup (and weight) for all authentication sessions. Screenshots of the systems are depicted in figure 7. All inputs (numbers, strokes) were logged. To minimize learning effects, *System* was counterbalanced, requiring $4! = 24$ participants for one iteration. The order of *Password* and *Difficulty* was randomized within each of the four authentication systems.

Procedure

We first told participants that the goal of the study was "to evaluate a new authentication system that uses the back of the smartphone for input". The two cameras were introduced as being part of the usability analysis, but we did not mention their use as an "attacker's view" for performing a security analysis later. Instead, the participants were simply told not to perform the phone interaction outside a specific area on the table, in order to avoid leaving the camera's field of view. The experimenter continuously checked the cameras to make sure that the interactions were correctly filmed at all times.

For each authentication system, four passwords were tested in random order: given easy, given hard, self-selected easy, and self-selected hard. Each password had to be used successfully in three authentication sessions. An authentication session ended when the password was input correctly, or if it was entered wrongly for three consecutive tries. That is, per system and password, the participants performed at least three and at most nine inputs. After this, the experiment moved to the next password or system.

System	PIN		Grid		Shapes (front)		BoD Shapes	
Basic	1 (0.3%)		33 (11.5%)		45 (15.6%)		66 (22.9%)	
Critical	0 (0%)		3 (1%)		3 (1%)		14 (4.8%)	
Token	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Basic	0 (0%)	1 (0.7%)	5 (3.5%)	28 (19.5%)	20 (13.8%)	25 (17.4%)	28 (19.5%)	38 (26.4%)
Critical	0 (0%)	0 (0%)	1 (0.7%)	2 (1.4%)	0 (0%)	3 (2%)	3 (2%)	11 (7.6%)

Figure 9. Absolute numbers (percentage) of basic and critical errors, summed up and subdivided into easy and hard passwords. Each system was used in $24*4*3 = 288$ authentication sessions (participants x passwords x 3). For each system, $288/2 = 144$ sessions used hard passwords and 144 used easy passwords.

With four passwords per system, and each password requiring three successful authentications, each system was used in $4*3 = 12$ authentication sessions per participant. For 24 participants, this makes $12 * 24 = 288$ individual authentication sessions per system. Each session could either be error-free, feature a basic error, or a critical error (see error chapter).

Before participants used a system, the experimenter explained it in detail, followed by a training session with predefined passwords. These passwords were different from the ones used in the actual study. During training, participants were allowed to try authenticating as long as they wanted, until they felt familiar enough with the system to start the actual task. The study ended with a questionnaire to collect demographic data and to assess the experience of participants in terms of problems, preferences and thoughts with respect to the usability and security of the four authentication systems. Participants could also ask any question that they had about the goal of the study or the study in general. Overall, the study took around 30 minutes per participant.

Participants

With the help of mailing lists and word-of-mouth we recruited 24 participants for the experiment, with an average age of 27 (range: 21-33). Eight participants were female. All had at least a university-entrance diploma and 17 had a graduate or undergraduate degree (mostly Bachelor of Science). They all owned at least one frequently used device with a touchscreen, most of them being smartphones (22) or music players (e.g. an iPod). Nine participants used more than one touchscreen device on a daily basis. On average, they were familiar with touchscreen interaction for around three years (M=3 years; SD=1.7 years). As an incentive, a 5 Euro online shop voucher was handed out to each participant after finishing the study.

In general, participants were concerned about their smartphone's data and thus, most of them used some form of protection. 18 used at least one kind of access control: 9 used PIN, 9 grid unlock. 8 of the 9 PINs were 4-digit, one was 5-digit. The average stroke length of the grid unlock was 3.9 (range: 2-5). Four participants stated that they had experienced someone "shoulder surfing" them while authenticating. In addition, ten participants mentioned that they use specific approaches to protect their input from bystanders, such as covering the input or changing the angle of the smartphone.

Results

Below, we report the results of a usability analysis focused on errors and authentication speed, and a security analysis based on the videos recorded during the study.

Error Rates

For the error analysis, we distinguish between basic and critical errors. A critical error means that the user was not able to authenticate at all during an authentication session, i.e., the password was entered incorrectly for three consecutive inputs. A basic error means that the user failed to authenticate once or twice in a session but was ultimately able to successfully enter the correct password, i.e., either on the second or the third try. This separation into basic and critical errors is particularly useful in usability studies involving mobile phones or ATMs (e.g., [13, 12]), as such systems typically block access after three failed tries. Note that when a critical error occurred, it was not additionally counted as a basic error. Thus, for each password/system combination, a maximum of three errors – critical or basic – was possible.

Overall, few critical errors occurred. Figure 9 shows the results grouped by system and subdivided into easy and hard passwords. PIN is the only system that created no critical errors. With 11 occurrences, BoD Shapes using hard passwords performed worst. However, this still means that only 7.6% of authentication sessions using hard passwords failed completely when being entered on the back of the device.

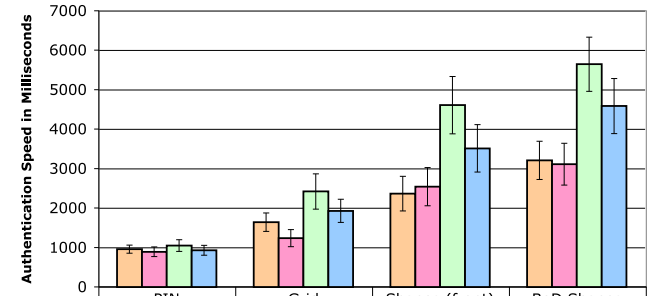
A 4 x 2 x 2 (*System* x *Password* x *Difficulty*) within participants analysis of variance of critical errors revealed a significant main effect for *System* ($F_{1.412,54.138} = 32.474, p < .05$) and *Difficulty* ($F_{1.0,23.0} = 7.667, p < .05$). No interaction effects were found. Overall, the numbers of critical errors are too low to reveal meaningful statistical results. All results are Greenhouse-Geisser corrected.

For basic errors, the picture looks similar (see figure 9). While PIN performed best with only 1 basic error, BoD Shapes and grid unlock with hard patterns created basic error rates of 19.5% and 26.4%, respectively. That is, each 5^{th} or 4^{th} input had to be repeated at least once for these methods.

A 4 x 2 x 2 (*System* x *Password* x *Difficulty*) within participants analysis of variance of basic errors revealed highly significant main effects for *System* ($F_{2.354,54.138} = 19.568, p < .001$), *Password* ($F_{1.0,23.0} = 18.526, p < .001$) and a significant main effect for *Difficulty* ($F_{1.0,23.0} = 10.292, p < .05$). In addition, there were significant interaction effects between *System x Difficulty* ($F_{2.329,53.565} = 3.363, p < .05$) and *System x Password x Difficulty* ($F_{2.348,54.008} = 5.097, p < .05$). All results are Greenhouse-Geisser corrected.

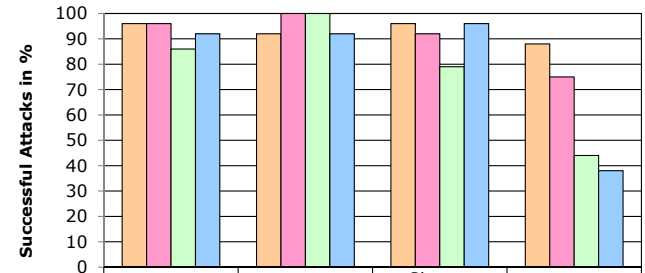
Post-hoc tests confirmed that PIN performed best and BoD Shapes was the most error-prone (highly significant differences to all other systems, all $p < .001$). Furthermore, easy passwords were less error-prone than hard passwords ($p < .05$) and self-selected passwords caused less errors than given passwords (highly significant with $p < .001$).

BoD Shapes Error Categories: The previous results showed that around each 4^{th} (easy) or 5^{th} (hard) input using BoD Shapes had at least one basic error, i.e., it had to be repeated to successfully authenticate. Thus, we performed an error analysis to find out whether and how they could be fixed. There were three main categories. All critical and 81% of basic errors could be attributed to one of those:



	PIN	Grid	Shapes (front)	BoD Shapes
Easy	963	1649	2372	3213
Easy Self	896	1243	2549	3118
Hard	1056	2426	4615	5654
Hard Self	935	1935	3519	4592

Figure 10. Average authentication speed in milliseconds for the four systems and the four different passwords.



	PIN	Grid	Shapes (front)	BoD Shapes
Easy	96	92	96	88
Easy Self	96	100	92	75
Hard	86	100	79	44
Hard Self	92	92	96	38

Figure 11. Successful shoulder surfing attacks of the four systems and the four different authentication passwords in percentage.

1. *Unintentional strokes* at the beginning or the end of a shape caused 29% of all critical and 41% of all basic errors. This happened in cases when the participants accidentally touched the back screen before actually meaning to start and when they did not properly lift the finger after the last stroke.

2. Another common error was *mixing up left and right* which caused 50% of critical and 17% of basic errors. For instance, instead of entering "Left, Right, Up", the user entered "Right, Left, Up".

3. The third category of errors is related to *slips*. We could attribute 21% of critical and 23% of basic errors to this category. Those were instances when a shape stopped too early since the participant accidentally left the touch area or slipped in another way. For instance, the user entered "Down, Up" instead of "Down, Up, Down".

Authentication Speed

The time to authenticate was measured from the first to the last touch (or lift-off). PIN was measured from touching the first digit (or button) to touching the last. Grid unlock was measured from touching the first dot until lifting the finger. Shapes (both front and back) was measured from the first touch until the lift-off of the third shape.

Only successful authentication attempts were counted. That is, for each participant, at most $4 * 4 * 3 = 48$ (system x password x authentication sessions) valid authentication sessions were considered for the analysis. One outlier that took over 25 seconds to authenticate was removed. The video material revealed that this participant had performed a long break after the first input, which lead to this delay.

The times required to authenticate can be found in figure 10 sorted by system and password difficulty. PIN was the fastest system with around 1 second for each password. BoD Shapes with hard given passwords was the slowest system (M=5.7s). This is almost twice as slow as BoD Shapes with an easy, self-selected password (M=3.1s). However, a closer look at the results reveals that BoD Shapes can be used significantly faster. For instance, the fastest user for BoD Shapes self-selected hard only needed 2.9 seconds on average, which is around 2 seconds faster than the average for this system. For BoD Shapes self-selected easy, the fastest user even managed to authenticate in 1.5 seconds on average.

A 4 x 2 x 2 (*System* x *Password* x *Difficulty*) within participants analysis of variance of speed revealed highly significant main effects for *System* ($F_{1.933,40.590} = 180.912, p < .001$), *Password* ($F_{1.0,21.0} = 16.648, p < .001$) and *Difficulty* ($F_{1.0,21.0} = 152.785, p < .001$). We found highly significant interaction effects for *System x Difficulty* ($F_{2.209,46.397} = 35.350, p < .001$) and *Password x Difficulty* ($F_{1.0,21.0} = 14.980, p < .001$) and a significant interaction effect for *System x Password x Difficulty* ($F_{2.155,45.251} = 5.136, p < .05$). All results are Greenhouse-Geisser corrected.

Post-hoc tests revealed highly significant differences between all instances of the independent variables *System*, *Password* and *Difficulty*. Interestingly, in almost all cases with the exception of Front Shapes easy, the self-selected passwords (M=2.4s over all systems) performed significantly faster than the given passwords (M=2.8s). In the case of both BoD and Front Shapes hard, this difference is larger than 1 second.

Security

For the security analysis, we had a member of the research team look at the recorded video afterwards and simulate an attacker, i.e., try to guess the entered password for each successful authentication session. While this "attacker" had naturally a high familiarity with all four systems, including BoD Shapes, the person was not involved in creating the list of given passwords (PINs, patterns, shapes) that were used in the study. For PIN, grid unlock and Front Shapes, our attacker looked at the over-the-shoulder camera material. For BoD Shapes, he used the camera that was "seated" opposite. Two examples of these views are shown in figure 12.

We simulated two different levels of attack: a one-off shoulder surfing attack, and a more determined "video review" attack. In the shoulder surfing attack, the video of a successful authentication session was shown to the attacker only once. Then he was allowed to guess three times. If the attacker was unable to guess the password correctly, he was allowed a subsequent "video review" attack, which meant that he could

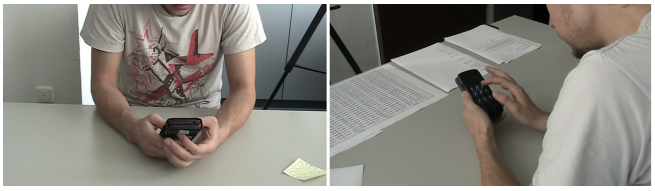


Figure 12. Left: The camera located opposite to the participant was used to analyze the security of BoD Shapes. Right: The back camera was used for PIN, grid unlock and Front Shapes.

now directly control video playback as needed (e.g., play, pause, rewind). During this, he could make up to three further guesses. That is, for all combinations of *System*, *Password* and *Difficulty*, there was either a successful shoulder surfing attack, a successful video review attack, or no successful attempt. The attacker was allowed to take notes after the first failed guess. Sounds and any other hints were removed from the video material. The whole attack procedure took three full working days.

While video review attacks were highly successful, they constitute an attack that is less likely in a real-world setting as it requires careful and well-executed (e.g., non-blurry) video recordings. All passwords could be identified in video review attacks, with the exception of two passwords in the BoD Shapes condition. Shoulder surfing attacks, however, revealed interesting differences between the four systems, as shown in figure 11. BoD Shapes was by far the most secure system with hard passwords being the hardest to observe. In the best case, a hard self-selected shape, only 9 of 24 passwords (38%) could be identified by the attacker. In contrast, all other systems performed weak with respect to security, no matter what kind of password (weak, hard, self-selected, or given) was used. The main reason for the few instances when our attacker was unable to guess the password for a front system during a shoulder surfing attack was extremely fast input by the user. The main reason in turn for making BoD Shapes difficult to differentiate in these situations was that our sample attacker often ended up confusing angled movements like "Left Up" with linear movements like "Left Right" or "Down Up".

Qualitative Data

In the questionnaire, participants were asked to rank the four systems with respect to security and ease-of-use. 21 out of 24 participants rated BoD Shapes to be the most secure. The other three each picked one of the remaining systems. For ease-of-use, BoD Shapes was rated worst by 21 participants. All rated either PIN (14) or Grid Unlock (8) as being the easiest to use. Both qualitative ratings support the quantitative results (i.e., error rate, speed, attack success rate) of the study.

We also asked participants whether they would use the system if it was available for their device. 13 of them gave a definite "yes" as an answer. Among the remaining eleven, there were two "yes under the condition that" replies. Both stated that they would use it if shapes with less strokes were allowed (i.e., the easy passwords). One of the nine "No" statements was a user that did not use any protection for her smartphone in general, thus not seeing the need to use BoD Shapes. The final eight users that did not want to use the system were users that rated BoD Shapes either hard or very hard to use. Most of them encountered at least one of the three error categories that we defined earlier in this chapter: mix-ups, unintended strokes, and slips.

DISCUSSION

Password Classification

The results of error rates and authentication speed indicate that our classification of difficulties held true for performance properties. Harder passwords significantly decreased speed and increased error rate. However, PIN is rather robust to this effect.

Performance and Improvements

In the study, all participants were highly trained to PIN, while about one third were frequent grid unlock users. BoD Shapes on the other hand was new to all of them. Still, the performance was quite good while being much more secure than the other systems. For instance, users could reach a speed of around 1.5 seconds for self-selected easy and 2.9 seconds for self-selected hard passwords (the most secure input). Self-selection is the standard approach for smartphones and thus these results can be considered most representative.

One explanation for the higher speed and lower error rate of self-selected BoD Shapes passwords could be that participants invested sufficient thought into their creation before actually using them. This made them more memorable and thus faster to use, which in turn indicates that a learning effect (of the password), even in such a short time, is likely. We argue that performance will further improve when the system is used for a longer time. However, this has to be evaluated in a long-term study.

Performance and Security

We found that the way the input was performed influenced the security of the respective system. For instance, the participants for which PIN could not be shoulder-surfed were either extremely fast (one of these users performed the input in around 500 ms on average) or used some special input method. One participant used two-handed PIN-entry. This was unexpected for the attacker, and following two hands at the same time proved too hard for a successful one-time shoulder surfing attack.

The way that input speed influenced the security of BoD Shapes was that for a quickly executed row of shapes, it was not clear to the attacker which strokes belonged to which password since finger lift-off "*is extremely hard to see and the breaks are a good indication of when a new shape starts*".

Rear Equals Higher Security

The Shapes system in itself is not secure – it is its input on the back that makes it more secure than others. However, BoD Shapes is much better suited for this kind of input than systems that require absolute positioning, such as PIN-entry. The security differences between Front Shapes and BoD Shapes illustrates well the security gain of moving input from the front to the back.

Reducing Error Rates

As described in the error section, most of the basic and critical errors occurred due to three reasons: unintentional strokes, mixing up left and right, and slips. This shows that there is a lot of potential in reducing the error rates. Mixing up left and right can be attributed to the fact that the participants were not familiar with the system and, more importantly, were not used to the passwords. It can be assumed that after using the system for a longer period, motor memory effects will reduce such errors [23].

Unintentional strokes, the second biggest error group, can be reduced (if not avoided) with clever programming. One could, e.g., discard strokes that are significantly shorter than the rest of a shape.

Finally, slips like accidentally leaving the touch-sensitive area cannot be avoided (similar to slips in the grid unlock system). This might further be influenced by form factors of the device and the touch-sensitive surface. Even though the algorithm copes with smaller strokes, some areas might just be too small for the users to perform the input. This issue, including a minimum space requirement and its influence on error rates, are still to be evaluated.

Error Recovery

The error rates of both BoD Shapes and grid unlock are significantly higher than for PIN. This is interesting since grid unlock is a widely used system. One of the reasons is that both do not provide an undo functionality. If an error occurs, input has to be started all over. Users do not mind this approach since it is considered an easy way of error recovery.

However, the number of available trials in a running system has to be increased since slips could otherwise easily lead to the device being blocked. The current approach used for grid unlock in commercial systems is to block the device for x seconds after y failed attempts. The variable x grows with the number of failed authentication sessions. A similar approach would thus be appropriate for BoD Shapes.

Angle Matters (but Much More on the Back)

The security analysis showed that the angle, in which the device was held in relation to the camera, did not influence the results for front-entry. This is because a shoulder surfer typically sees the display at a similar angle as the user. However, the security of the back-of-device interaction was indeed heavily influenced by how the device was held. The lower the angle, the harder it was to distinguish directions (e.g. "Left" and "Up"). For the user, however, the angle did not influence the performance. In addition, it is much harder for an attacker to inconspicuously take up a position that makes it easy to spy on the input. Thus, we argue that back-of-device authentication is more secure, even more in a real-world setting.

Feeling Secure

The participants' answers indicate that BoD Shapes is able to translate externally measured security into user-perceived security. For instance, one participant stated that "*I would use* [BoD Shapes] *in case that I need a really secure authentication system.*". Another participant mentioned that "*I would like to use BoD Shapes since it is very secure. It is also easy to remember.*". We argue that one of the main advantages of BoD Shapes is that it provides enhanced security for a realistic use case (shoulder surfing of mobile authentication) without introducing detours, indirect input, or forms of randomization.

Swapping Sides

The study results indicate that in most situations and contexts, using BoD Shapes makes the authentication procedure more secure. However, there are instances in which using it might add unnecessary overhead (e.g. when the user is alone at home) or actually decrease security (e.g. when the user is standing in the metro and the people nearby are sitting). To cope with this, the concept can easily be adapted to support authentication on both sides of the device simultaneously. This way, the user can make an ad-hoc decision about which side to use depending on the context and current security requirements. For instance, BoD Shapes could be combined with Front Shapes (or PIN, grid unlock or any other system the user prefers).

LIMITATIONS

Especially for one-handed interaction, the form factor of the prototype proved to be a burden for the participants. As a result, all of them used two hands to interact with the system in the main study. This means that we cannot infer any generalizable results on one-handed interaction. However, this issue was the same for all systems and arguably influenced them all in the same way. Still, in the future, we plan to test a thinner and lighter prototype to compare the results.

The security analysis was more adversarial than most real-world scenarios, in which camera attacks would not be feasible. Additionally, none of the participants tried to protect their input – some even positioned the back of the device so the camera had a better view on it as they thought it would help with the analysis. Furthermore, the attacker was aware of the strict password rules which allowed for disambiguating many observations that would have failed otherwise. For instance, even if the attacker saw one stroke only but knew there had to be a second one, there was a 1 out of 3 chance to correctly guess the missing stroke. This strategy worked several times. In a real world implementation, the password rules would be much more general (e.g. no fixed stroke length but a minimum stroke length).

In addition, we had only one attacker which might slightly bias the results. More attackers were not feasible as the task took three full working days and it was simply impossible to get more than one expert to do this. However, we argue that opting for an expert attack represents a worst-case-scenario that provides a good estimate of the security of an authentication mechanism.

Memorability is a crucial issue when it comes to the usability of authentication mechanisms. Since BoD Shapes is pattern-based, we argue that it has similarly high memorability properties as related systems [27]. However, we did not evaluate this yet as this requires a long-term study (in the real world if possible).

CONCLUSION AND FUTURE WORK

In this paper, we introduced a new authentication system for smartphones using the back of the device, called BoD Shapes. To authenticate, a user performs a row of shapes on the rear of the device. In general, a shape can consist of an arbitrary number of horizontal and vertical strokes. Performing the input on the back makes this approach much more resilient against shoulder surfing attacks. Even though performance is decreased, it is still sufficient for everyday use with high potential for improvement in the long-term. In the most secure case, hard self-selected passwords, the best user reached an average authentication speed of 2.9 seconds.

The main contributions of this work are: (1) the BoD shapes concept and its theoretical analysis (2) a user study that attested the system high security and good performance properties, (3) general findings of security and performance issues when applying back-of-device authentication to smartphones.

As we informally observed in the study, smudge attack resistance of BoD Shapes is very high since the consecutive shapes are performed on top of each other. In the future, we will perform experiments to provide empirical proof for this claim. At the same time, a practical evaluation of the memorability properties of BoD Shapes will be required.

In order to decide these two questions and therefore find out whether BoD Shapes is appropriate for real-world use, a long-term deployment is needed. To do so, however, we will have to wait until commercial devices with back-of-device interaction support are available, or until the form factor of the prototype can be highly improved.

ACKNOWLEDGMENTS

We would like to thank Emmanuel Pietriga for his helpful feedback on the BoD Shapes concept.

This work was partially funded by a Google Research Award.

REFERENCES

1. 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.
2. Azenkot, S., Rector, K., Ladner, R., and Wobbrock, J. Passchords: secure multi-touch authentication for blind people. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '12, ACM (2012), 159–166.
3. Baudisch, P., and Chu, G. Back-of-device interaction allows creating very small touch devices. In *Proc. CHI 2009*, ACM (2009), 1923–1932.
4. 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.
5. Bianchi, A., Oakley, I., and Kwon, D. Spinlock: A single-cue haptic and audio PIN input technique for authentication. In *Proc. Haptic and Audio Interaction Design*, Springer Berlin / Heidelberg (2011), 81–90.
6. Böhmer, M., Hecht, B., Schöning, J., Krüger, A., and Bauer, G. Falling asleep with Angry Birds, Facebook and Kindle: a large scale study on mobile application usage. In *Proc. MobileHCI 2011*, ACM (2011), 47–56.
7. Carr, S. *Public Space*. Cambridge Univ Press, 1992.
8. Chong, M., and Marsden, G. Exploring the use of discrete gestures for authentication. In *Proc. INTERACT 2009*, Springer Berlin / Heidelberg (2009), 205–213.
9. Clarke, N., and Furnell, S. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security 6*, 1 (2007), 1–14.
10. Clarke, N., Furnell, S., Rodwell, P., and Reynolds, P. Acceptance of subscriber authentication methods for mobile telephony devices. *Computers & Security 21*, 3 (2002), 220–228.
11. 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.
12. De Luca, A., Langheinrich, M., and Hussmann, H. Towards understanding ATM security: a field study of real world ATM use. In *Proc. SOUPS 2010*, ACM (2010), 16:1–16:10.
13. De Luca, A., von Zezschwitz, E., and Hussmann, H. Vibrapass: secure authentication based on shared lies. In *Proc. CHI 2009*, ACM (2009), 913–916.
14. Dunphy, P., Heiner, A. P., and Asokan, N. A closer look at recognition-based graphical passwords on mobile devices. In *Proc. SOUPS 2010*, ACM (2010), 3:1–3:12.
15. Fleishman, E., and Parker Jr, J. Factors in the retention and relearning of perceptual-motor skill. *Journal of Experimental Psychology 64*, 3 (1962), 215.
16. Gafurov, D., Helkala, K., and Søndrol, T. Biometric gait authentication using accelerometer sensor. *Journal of computers 1*, 7 (2006), 51–59.
17. Hirotaka, N. Reassessing current cell phone designs: using thumb input effectively. In *CHI 2003 EA*, ACM (2003), 938–939.
18. Jakobsson, M., Shi, E., Golle, P., and Chow, R. Implicit authentication for mobile devices. In *Proc. HotSec 2009*, USENIX Association (Berkeley, CA, USA, 2009), 9–9.
19. Jermyn, I., Mayer, A., Monrose, F., Reiter, M., and Rubin, A. The design and analysis of graphical passwords. In *Proc. Usenix 1999* (1999), 1–14.
20. 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.
21. 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. Last accessed: January 7, 2013.
22. Kirschnick, N., Kratz, S., and Möller, S. An improved approach to gesture-based authentication for mobile devices. In *SOUPS 2010* (2010).
23. Shadmehr, R., and Brashers-Krug, T. Functional stages in the formation of human long-term motor memory. *The Journal of Neuroscience 17* (1997), 409–419.
24. Siek, K., Rogers, Y., and Connelly, K. Fat finger worries: How older and younger users physically interact with PDAs. In *Proc. INTERACT 2005*, Springer Berlin / Heidelberg (2005), 267–280.
25. Takada, T., and Koike, H. Awase-E: Image-based authentication for mobile phones using user's favorite images. In *Proc. Human-Computer Interaction with Mobile Devices and Services*, Springer Berlin / Heidelberg (2003), 347–351.
26. Tari, F., Ozok, A. A., and Holden, S. H. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proc. SOUPS 2006*, ACM (2006), 56–66.
27. Weiss, R., and De Luca, A. Passshapes - utilizing stroke based authentication to increase password memorability. In *NordiCHI 2008: Proceedings of the 5th Nordic Conference on Human-Computer Interaction*, ACM (2008), 383–392.
28. Wolin, A., Eoff, B., and Hammond, T. Shortstraw: A simple and effective corner finder for polylines. In *Proc. Eurographics 2008* (2008), 3340.
29. Yazji, S., Chen, X., Dick, R., and Scheuermann, P. Implicit user re-authentication for mobile devices? In *Proc. UIC 2009*, vol. 5585, Springer-Verlag New York Inc (2009), 325.