ShakeUnlock: Securely Unlock Mobile Devices by Shaking them Together

Rainhard Dieter Findling
JRC u'smile, University of Applied Sciences Upper Austria
Softwarepark 11, 4232 Hagenberg, Austria
rainhard.findling@fh-hagenberg.at

Muhammad Muaaz
JRC u'smile, University of Applied Sciences Upper Austria
Softwarepark 11, 4232 Hagenberg, Austria
muhammad.muaaz@fh-hagenberg.at

Daniel Hintze
FHDW University of Applied Sciences Paderborn
Fürstenallee 3–5, 33102 Paderborn, Germany
daniel.hintze@fhdw.de

René Mayrhofer
JRC u'smile and Johannes Kepler University Linz
Altenberger Straße 69, 4040 Linz, Austria
rene.mayrhofer@jku.at

ABSTRACT

The inherent weakness of typical mobile device unlocking approaches (PIN, password, graphic pattern) is that they demand time and attention, leading a majority of end users to disable them, effectively lowering device security.

We propose a method for unlocking mobile devices by shaking them together, implicitly passing the unlocked state from one device to another. One obvious use case includes a locked mobile phone and a wrist watch, which remains unlocked as long as strapped to the user's wrist. Shaking both devices together generates a one-time unlocking event for the phone without the user interacting with the screen. We explicitly analyze the usability critical impact of shaking duration with respect to the level of security. Results indicate that unlocking is possible with a true match rate of 0.795 and true non match rate of 0.867 for a shaking duration as short as two seconds.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Authentication; I.4.6 [**Segmentation**]: Time series segmentation; I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination

General Terms

Human Factors, Measurement, Performance, Security

Keywords

Accelerometer; authentication; frequency domain; mobile devices; shaking; time series analysis; usability;

MoMM '14, December 08 - 10 2014, Kaohsiung, Taiwan

ACM 978-1-4503-3008-4/14/12 ...$15.00
http://dx.doi.org/10.1145/2684103.2684122.

1. INTRODUCTION

Current personal mobile devices are ubiquitous. They are no longer only being used for calling and texting purposes, but offer a multitude of services ranging from simple localized tourist guides to city-scale wireless network services and multiple nation-wide mobile payment and identity management applications. Under the "Bring Your Own Device" (BYOD) trend, more and more private mobile devices are integrated in company networks. As a consequence, personal as well as company data which deserves adequate protection is stored and processed by these mobile devices.

Consequently, device unlocking mechanisms have been developed that restrict access to mobile devices until being unlocked directly before usage. However studies show that a majority of users do not use any unlocking mechanism for their mobile devices [3, 6, 22] with usability drawbacks being the main reason. Usability of the currently widely deployed mobile device unlocking mechanisms (PIN, password, and graphic pattern) suffers from two drawbacks. First, they require user attention as users have to keep their unlocking secret in mind and have to look at the screen to perform the unlock. Second, they cause users to spend time to enter their secret into the mobile device. Traditional desktop/laptop computers usually features a full sized keyboard for capturing user input quickly and reliably. In contrast, entering a secret can be difficult and time consuming especially on mobile devices because of their limited user interfaces (e. g. small on screen display with small keys and haptic feedback only on keypress for mobile phones). Further, desktop/laptop computers are usually unlocked once and then used for a longer period of time — which is different for typical mobile device usage. Users tend to use mobile devices more frequently and for shorter time periods, leading to a higher amount of unlocking events [11]. Therefore, mobile users face usability issues caused by unlocking mechanisms that are even more significant than those already well-known for desktop users. There typical examples are: short passwords, using the same password over and over or even deactivating locking mechanisms in the first place – to increase usability.

Besides these usability issues the mentioned mobile device unlocking mechanisms are vulnerable to different attacks, such as the shoulder surfing [23, 25] or the smudge attack [2, 27] (attackers screening the device display after the user au-

thenticated using a graphic pattern in order to observe the residual smudge that might remain on the display, thereby observing the the unlocking secret).

1.1 Shaking devices to transfer authentication state

Many users already carry more than one mobile device (such as a mobile phone and a wrist watch). Hence unlocking a currently locked device **L** can be done by implicitly transferring the authentication state from a currently used – therefore already unlocked – device **U** of the same user. This is especially interesting for mobile devices as they can remain unlocked for different durations. For example, wrist watches may remain unlocked as long as they are strapped to their legitimate users' wrists (under the assumption that only the legitimate owners will have control of that device as long as it is strapped to their wrist), while for mobile phones it might be necessary to lock them whenever users put them aside (as they could immediately be taken by malicious users).

We propose a method for transferring the authentication state from an already unlocked device **U** to another, still locked device **L** by shaking those devices together – such as from a wrist watch strapped to the wrist to a mobile phone held in the same hand. We assume that device **U** is unlocked by observing that it is with the correct user, such as with a wrist watch locking itself as it is taken off the wrist. Our approach is intended to work even when devices are a little apart from each other, such as in the wrist watch/mobile phone scenario. Further the approach is intended to not incorporate any biometric user information, but should work amongst different users for the same devices. Assuming the devices **U** and **L** are owned by user A, unlocking should work for user A shaking the devices **U** and **L** together, but should work as well in case user B is shaking the devices **U** and **L** together.

Shaking devices together has been found intuitive as well as fast and convenient to use for e. g. establishing a secure connection between devices in previous research [5, 9, 12, 16, 4, 18]. An advantage of shaking devices over e. g. simultaneously pressing a button on both devices or bumping devices against each other (which can easily be forged by attackers [24]) is, that shaking patterns are hard to fake. It has been shown that shaking devices provides a sufficient level of assurance that both devices have been shaken together, as it is hard to accurately recreate shaking patterns on a remote device [21]. Consequently, for attackers it is hard to trick a device they have under their control into unlocking, by shaking it simultaneously with the user shaking the counterpart device. For the same reason it is unlikely to accidentally unlock a mobile device placed in a trousers pocket or carrying bag, e. g. while walking.

In this paper, we build upon previous research on pairing mobile devices – and extend them from rare, explicit authentication events to regular, implicit usage throughout a typical day. The main differences are:

- From usability point of view, the significant difference is the inversion of priorities from security to usability (see below).
- From security point of view, we do not aim to create a long-lived association between devices but trigger one-time unlocking events on remote devices.
- From practical point of view, mobile devices are farther apart (not held with the same hand as in previous published research, but with a distance of around 10–15 cm and a non-static joint in between).

Although similar data analysis methods (cf. section 3) are applicable to both the previously discussed and our new use cases, relevant parameters (cf. section 5) and protocols are noticeably different.

1.2 Secure channel vs. device unlocking

In previous research, shaking is used for pairing mobile devices to establish a secure communication channel between them (also called the bootstrapping problem [15, 19]). There are two essential differences in requirements for shaking mobile devices when establishing a secure channel – which is done once for each pair of devices – and when unlocking them – which is done frequently.

First, for establishing a secure channel, usability is important, but not the highest priority. It may take some time (within reasonable limits) and require the user's attention (having the user consciously in the loop is even beneficial for the perceived level of security). For unlocking a device, usability is critical and the highest priority when designing a solution. It must only take a small amount of the user's time and require as little of the user's attention as possible. Otherwise, users will disable the unlocking mechanism.

Second, for establishing a secure channel between mobile devices it is essential to prevent any possible on-line and off-line attacks, including brute-force dictionary attacks. If the channel is compromised, an attacker can read or manipulate all (recorded) information on that channel at a later point in time. For unlocking mobile devices, we have to focus on on-line, real-time attacks as part of the inherent user interaction of unlocking a device with a sufficiently short time window (a few seconds).

In the scope of this work, we assume the two mobile devices to already share a secure communication channel. Therefore, we will not discuss the required device authentication and secure data transmission protocols. Our analysis starts with the second step of exploiting such a channel to automatically unlock one of the devices.

1.3 Improvements in usability

Typical usage scenarios for our approach are: users pick up their mobile phone with the hand the wrist watch is strapped to and shake it briefly for unlocking. Alternatively, they take it out of their pocket while walking, then shake it instead of having to look at or interact with the display.

This is one advantage in usability of shaking over classical mobile device unlocking approaches: users don't need to look at or interact with the display of the mobile phone at all for performing the authentication. Another advantage is that shaking is not knowledge based (does not involve a secret that users need to remember). As users are going to carry more and more devices in the future, with knowledge based approaches they would either need to use different secrets for unlocking these devices (which has an impact on usability as it raises the cognitive load). Or they would use the same secret over and over (which is a well known risk in terms of security).

We argue that the most critical factor in terms of usability will be the time it takes users to perform the unlock. One important observation by Zezschitz et al. [26] is that the average time taken to enter a PIN on a mobile device is in

the range of 1.5 s and the average time for drawing a graphical pattern is in the range of 3 s. More thorough analysis of user expectations is difficult because real-life experimental conditions negate additional questions or external user observation for comparative studies. We currently assume these 1–3 s as a reasonably acceptable delay considering the short interaction times with mobile devices. Therefore we use a shaking duration of 2 s to provide comparable speed to the well-known PIN lock, but require significantly less user attention. The same argument holds in comparison to graphical patterns as the second well-known unlocking method.

Summarizing, our novel contributions are:

- In contrast to previous research we focus on a different use case with the implications on usability, security, and data analysis parameters discussed above.
- We utilize data from devices that are located apart from each other, specifically being strapped to the wrist and held in the hand. Further, we use accelerometers embedded in off-the-shelf devices, namely in a wrist watch and a mobile phone instead of custom made or research oriented sensors with better accuracy or higher sampling rates.
- Using this setup, we record the u'smile ShakeUnlock database of mobile devices being shaken concurrently, which we use for evaluating our approach.
- We analyze the impact of the usability critical shaking duration on unlocking the device as the trade-off between usability and security. Using a shaking duration of only two seconds we achieve a true match rate of 0.795 and true non match rate of 0.867.

2. RELATED WORK

Shaking mobile devices has been tackled by a significant amount of research over the past 10 years: starting with the user interaction method of moving devices together [12, 1, 10] (including the popular but insecure [24], now discontinued mobile phone application "Bump"), the associated data analysis methods for segmentation, feature extraction, and classification [16, 13, 18, 7], and different protocols for deriving shared secret keys from accelerometer data of mobile devices shaken [21, 20, 4, 9] or swayed together [14] or from the wireless signal strength readings of devices moved together [5]. The standard use case for the security conscious publications was pairing or associating devices previously unknown to each other (also referred to as the bootstrapping problem or as human verifiable device authentication). In contrast we focus on shaking as a user interaction method for unlocking of previously paired devices.

In terms of required data analysis methods, comparing accelerometer time series in both time domain [12, 4, 18, 9, 14]. and frequency domain [16, 21] has been studied. For an overview of the different features, we refer to [7]. Although comparison in time domain might yield higher rates of entropy per second of shaking data [9], we choose to do the analysis in frequency domain for the same practical reasons discussed previously. In real-life deployments of this method, two (or multiple) devices will independently record their local accelerometer time series, and we cannot realistically assume perfect synchronization of sampling intervals and start-of-sampling offsets. Additionally, sensors will physically be aligned in different coordinate systems. Reducing the three dimensions to the magnitude of acceleration and comparing the single-dimensional time series in frequency domain is more robust against jitter, drift, and start-of-sampling offset, as well as rotational components that make it difficult to filter gravity from movement acceleration. Therefore, our data analysis approach (cf. section 3) is based on the respective previous approach in frequency domain, specifically the use of the coherence metric [16] based on "active segments" sampled independently on two devices [21]. Applying one of the other published approaches in time domain [4, 9, 18, 14, 8] would require an additional synchronization step in the communication protocol, but could potentially provide higher entropy. A practical experiment on power-efficient, accurate time synchronization between mobile device or post-hoc correction of synchronization errors is a potential extension for future research.

The special aspect of devices being jointly moved but physically located slightly apart has been also studied by Fujinami and Pirttikangas [8] in the form of a user's hand holding a toothbrush. We deal with the same issue for robust comparison of time series, but additionally consider the security and usability implications.

In terms of security, only a subset of the previous publications on shaking explicitly considered malicious attacks on this user interaction method [9, 21, 4, 14, 5]. We use a different threat model because we aim to secure one-time unlocking trigger events valid for a limited time instead of long-lived shared secret keys for secure communication channels (cf. section 3.3). However, we can build upon the previous result that a dedicated attack on shaking authentication by trying to mock the movement of one device while the other is independently shaken by another person is impractical, even if an attacker is able to visually observe the device movement [21, 20].

3. APPROACH

We shake a locked mobile phone together with an unlocked wrist watch in order to transfer the authentication state from the watch to the phone and unlock it. Unlocking is done in two major steps: recording acceleration time series and extracting active acceleration periods separately on each device, then determining the similarity of active periods across devices (see figure 1). The result is a single *one-time unlocking event* on the mobile phone; in contrast to pairing devices, no data is kept after the similarity comparison, not even in the form of cryptographic key material. This distinction is important for the security analysis discussed below.

3.1 Acceleration recording and active segments extraction

In our approach, devices independently and continuously record acceleration time series. We argue that this can be achieved without necessarily draining the device batteries by including hardware specifically designated to recording acceleration. Such hardware is slowly becoming available in off-the-shelf mobile devices with the initial use case of background step counting. Similarly, power efficient hardware has successfully been used to e. g. continuously record audio [17]. Using acceleration time series, devices continuously check for the start of an active period (in which users actively shake their device). This is achieved by monitoring the variance of the magnitude of 3 axes accelerometers with

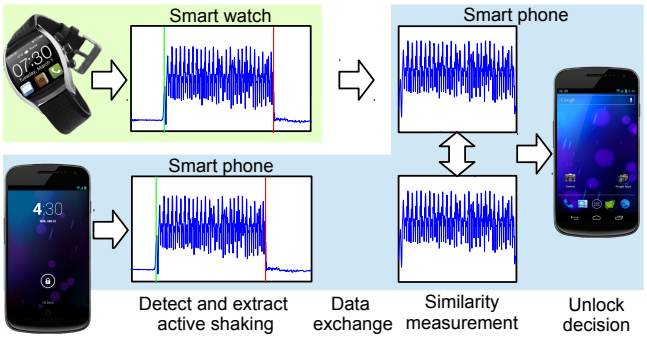
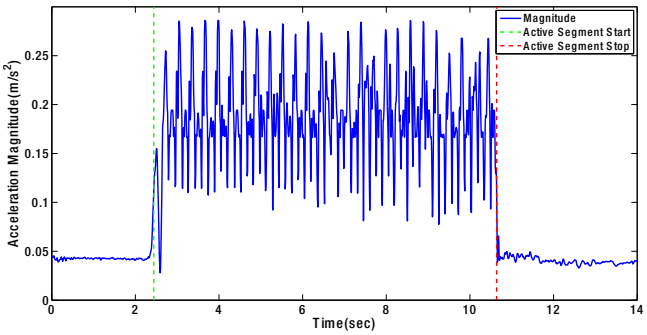


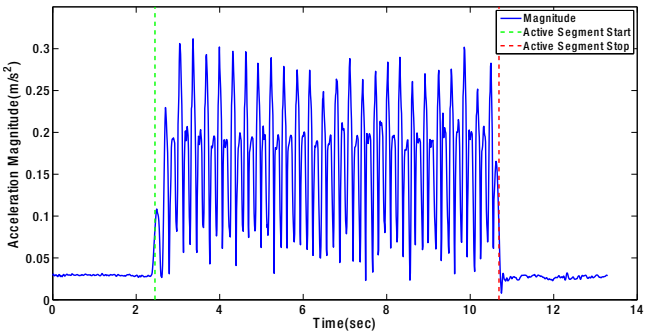
Figure 1: Data processing chain used in the Shake-Unlock approach.

a sliding window as stated in [20] (we use a sliding window of 0.5 s). If the variance within this window rises above a certain threshold this marks the start of an active period (we use a variance threshold of $6 \cdot 10^{-4} \frac{m}{s^2}$). As a device detects such a start, it records acceleration data for a certain time (e. g. 2 s) in which users shake their devices (see figure 2). This acceleration data is called "active segment" and used later for determining if devices were actually shaken together.

For usability reasons an application implementing this approach should give feedback to users about starting and stopping recording acceleration, as users are expected to shake their device throughout the fixed recording time – otherwise the unlocking will be aborted. After detecting and extracting active segments each device preprocesses its respective active segment. Preprocessing is done by removing the gravity per axis (approximated by the mean acceleration per axis throughout the active segment), calculating the magnitude and normalizing it to the range $[-1, 1]$.



(a) Active segment detected on mobile phone



(b) Active segment detected on wrist watch

Figure 2: Active segments detected independently on the mobile phone and wrist watch.

The active segment magnitudes extracted on the wrist watch and mobile phone are used as basis for determining if devices were actually shaken together. Therefore, one of these devices sends it's active segment magnitude over the secure channel to its counterpart device for further processing. Our approach does not restrict on which device further processing is done. But as both devices are assumed secure and connected by a secured channel, the natural choice is to process data on the mobile phone. This allows the mobile phone to immediately use the unlock decision without further sending data between devices. Additionally, calculations are usually faster on mobile phones than on wrist watches due to higher processing power.

3.2 Similarity of active segments

After active segments have been aggregated on the mobile phone, we determine the similarity of those segments to decide if the devices were actually shaken together. Therefore we adapt a weighted overlapped segment averaging form of magnitude squared coherence by Lester et al. [16]. This approach that has been successfully used for similarity detection in previous research [21] (see figure 3).

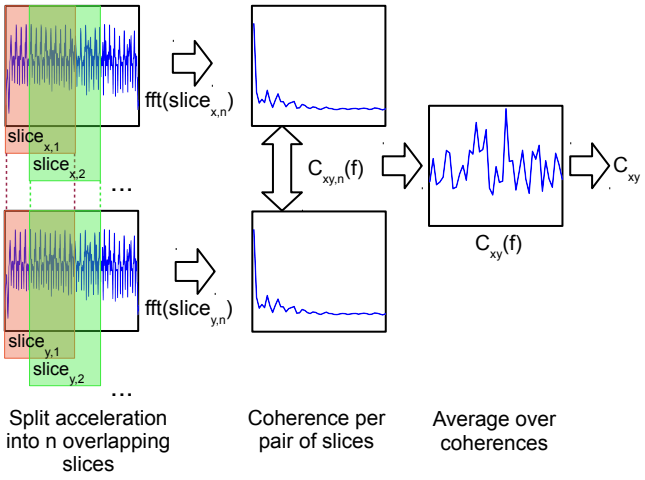


Figure 3: Overview of similarity measurements between active segments of device x (mobile phone) and y (wrist watch), computed on the mobile phone.

At first we divide the pairwise active segments of devices x (mobile phone) and y (wrist watch) into n overlapping slices by multiplying them with an according window (such as the Hamming or Hann window). Slices are transformed into the frequency domain. For pairwise slices x, n and y, n, first, the coherence $C_{xy,n}(f)$ is calculated using the cross spectral density $S_{xy,n}$ and the power spectral densities $S_{xx,n}$ and $S_{yy,n}$. Then those coherences for all n slices are averaged together to obtain an overall coherence $C_{xy}(f)$ (see equations 1 and 2).

$$C_{xy,n}(f) = \frac{|S_{xy,n}|^2}{S_{xx,n} \cdot S_{yy,n}} \tag{1}$$

$$C_{xy}(f) = \frac{1}{n} \cdot \sum_{n} C_{xy,n}(f) \quad (2)$$

In order to obtain a single, scalar similarity measure C_{xy}, the coherence $C_{xy}(f)$ is averaged up to a cutoff frequency $f_{\max}$, which is important for not incorporating the frequencies that are not correlated to users shaking their device (see equation 3). In previous research [21] a cutoff frequency of $f_{\max} = 40$Hz has proven useful, which we are adapting for our approach as well.

$$C_{xy} = \frac{1}{f_{\max}} \cdot \int_{0}^{f_{\max}} C_{xy}(f) \quad (3)$$

To finally decide if the mobile phone should unlock we apply a threshold $T = [0, 1]$ to the scalar similarity in order to obtain a binary accept/reject decision.

The main reason for our approach processing the shaking acceleration time series of both devices in the frequency domain is robust processing. Given the assumption that a constantly active link between the devices (to stream raw sensor data from one device to the other using constant radio traffic) would drain their batteries, both devices should independently estimate the start of active segments, which can be done in the background on a low-power microcontroller without involving the main CPU (cf. [17] for speaker detection or current Apple iPhone 5 devices for background step counting based on accelerometer data). However, as devices are apart from each other and they are expected to experience slightly different levels of acceleration, they will trigger the start of active segments at a slightly different time for both devices – which will further lead to the active segments being slightly shifted in time. Moreover, we have to assume additional sources of sampling errors such as imprecise (and slightly different) clock cycles for sampling the accelerometers and therefore drift and jitter in inter-step sample timing. Hence without direct synchronization, the similarity measure has to handle slightly asynchronous data. Processing such data in frequency domain works more reliably compared to processing in time domain.

One possibility to obtain time synchronization without a constantly active radio channel is to establish a connection between the devices as soon as either one or both devices detect active shaking. However, this requires a radio link capable of establishing this connection within a single sample period (e. g. 10 ms for 100 Hz sampling rate). This is inapplicable with currently deployed off-the-shelf wireless standards on mobile devices such as Bluetooth, which can take up to several seconds even after having been paired before. When using the connection creation time as synchronization reference (1-way handshake) devices are possibly not synchronized by the time the connection has been established – which consequently also causes data recording to start at a different time. Alternatively, shaking is possibly over when both devices start recording data synchronously after the connection has been established (2- or 3-way handshake).

3.3 Threat model

As devices are initially paired, we assume a secure wireless channel between devices. Therefore, traditional passive attacks (e. g. eavesdropping) are impossible when none of the devices are compromised. But assuming that the locked mobile phone is physically controlled by an attacker (after loss or theft), an active attack to unlock it could be possible. These attacks are limited to:

1. Periods of the user accidentally shaking the counterpart device (e. g. when toothbrushing), as otherwise unlocking attempts will be left unanswered.

2. Physical proximity, as both devices have to be in signal range of each other.

It has also been shown that it is impractical for attackers to shake a mobile device accordingly when observing the legitimate user shaking their device concurrently [21]. This would require a mechanism to artificially shake a mobile device based on observed shaking, which we assume impractical as it will raise the cost/benefit ratio for attackers. For example: we assume the user's wrist watch is compromised by having malware installed, which continuously records acceleration data and transmits it to the attackers in real time. Such malware could be embedded in other applications, therefore installed by unaware users themselves – without attackers requiring access to the unlocked wrist watch. Even when attackers have access to the user's acceleration time series, they would still need to concurrently (in real time) shake the mobile phone in a pattern matching the wrist watch closely enough to fall above the coherence threshold T, which seems impractical.

3.3.1 Malware accessing the secure channel

The scenario changes assuming that the wrist watch is compromised to a level on which installed malware can access the secure channels – such as to the user's mobile phone – to send data. Using such malware (which requires *root* permissions on an Android devices to manipulate active network connectivity from other apps) an attacker could continuously send forged acceleration data from the user's wrist watch to the mobile phone. We assume that this forged acceleration data is structured in a way that attackers can artificially shake an object accordingly. Hence, if attackers bring the mobile phone under their control they can unlock it if they can shake it according to their forged acceleration data. In order to prevent such attacks, the unlocking software needs to ensure it receives data from it's correct counterpart, which can be achieved by using cryptographic signatures for mutual authentication of all messages/packets.

3.3.2 Tricking users into shaking behavior

A different approach for attackers to unlock the mobile phone could be to trick the user into shaking behavior the attackers want – as they might be able to somehow attach the mobile phone to the same source of movement and acceleration. Theoretically this could be achieved by an attacker shaking hands very hard with the legitimate user, in case the user is using the arm with the wrist watch strapped on as well as the mobile phone being attached to the attackers wrist. None of the devices can recognize the ongoing attack, as they are actually shaken in a legitimate way.

Practically, for this scenario the attacker must prevent users from recognizing their mobile phone, which has to be physically close to the wrist watch in order to experience the same acceleration. As this is a conceptually easy to perform non-technical attack, we expect attackers to be inventive when it comes to hiding the presence of the users' mobile phone. Consequently this could become a potential

attack scenario to our approach – although it still requires a local and highly personalized attack.

4. U'SMILE SHAKEUNLOCK DATABASE

We recorded the *u'smile ShakeUnlock database*[1] which contains 29 participants shaking a wrist watch (strapped to their wrist) and mobile phone (held in the hand). For each participant, we recorded 5 shaking samples each for four different setups (see table 1), which results in 20 samples per participant and device, and to 1160 samples in total – which overall reflect large differences in shaking style, vigor, and frequency.

Setup	Watch	Phone	Posture
1	left wrist	left hand	sitting
2	right wrist	right hand	sitting
3	left wrist	left hand	standing
4	right wrist	right hand	standing

Table 1: The u'smile ShakeUnlock database features 5 samples for each 4 different setups per participant.

For data collection, we used an Android application recording 3 axes accelerometer time series and storing them in the form of comma separated value files locally on each device. The devices are connected over a Bluetooth channel, sending start/stop recording instructions as well as experiment metadata (e. g. subject ID) in a synchronized fashion when starting/stopping data recording. We explicitly note that this synchronization is only facilitating an easier experiment, but that it is not required for real-world use outside the recording setup.

Before data recording participants strapped the watch to their wrist and grabbed the phone with the same hand (see figure 4). Immediately before starting data recording all participants were given the same, brief instructions: "Shake the devices as you would shake them intuitively, but shake them a bit harder/a bit quicker and try to not bend your wrist while shaking.".

Each recording has a total length of 13 s: 10 s of active shaking and 3 s of neutral device movement. Participants started the recording by pressing a button on the mobile phone and started shaking. They were informed to stop shaking by audio and vibration feedback from the phone after 10 s of recording (therefore active shaking is close to 10 s for most samples) – with the devices continuing to record for 3 s after the notification.

In total we recorded data from 25 male and 4 female participants, with an average age of 27 years and from different backgrounds and professions (we do not distinguish by profession, age or gender as it does not seem important for performing a simple shaking movement). Further we used a mix of different devices running Android 4.0 or above and table 2). For 26 participants we used a Samsung Galaxy S4 mobile phone (model GT-I9500) together with a Samsung Galaxy Gear wrist watch (model SMV700). For the remaining 3 participants we used a Moto G mobile phone (XT1032) together with a Simvalley Mobile wrist watch (model AW-420.RX) to analyze how dependent various parameters of the data analysis pipeline are on the specific recording hardware.

[1]The u'smile ShakeUnlock database is publicly available for download at `http://usmile.at/downloads`.

(a) Front side (b) Rear side

Figure 4: Phone and watch placement for all setups, with the watch being strapped just as hard as necessary to prevent slipping during shaking.

The recording acceleration sensor sampling rate was fixed on operating system side to 100 Hz. Therefore any inaccuracies in sample timing are caused by the operating system itself which would also occur in other implementations of our approach.

Pair of devices	Male	Female	Total
Galaxy S4, Galaxy Gear	23	3	26
Moto G, Simvalley watch	2	1	3

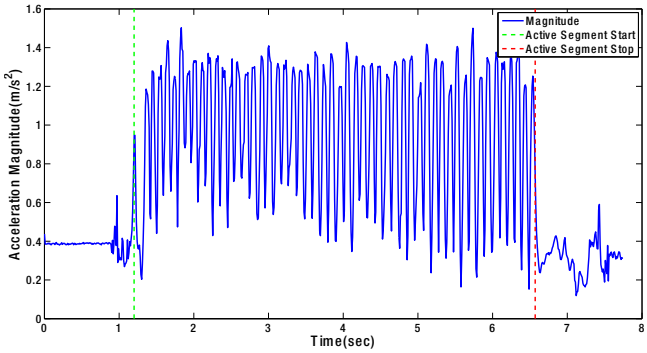
Table 2: Amount of recordings done per pair of devices and gender of participants.

5. EVALUATION

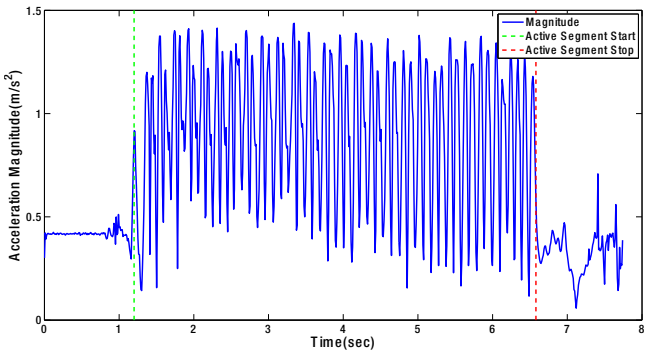
Our goal is to utilize short (therefore usable) periods of e. g. two seconds of actively shaking devices. We use the u'smile ShakeUnlock database to quantify the impact of such short shaking on determining if devices were actually shaken together. First, we extract active segments of different lengths (simulating different shaking durations) for all recordings of all participants. Then, we perform similarity measurements to determine the reliability of detecting that devices were actually shaken together. We further evaluate the impact of shaking devices with the dominant and non dominant hand as well as the impact of sitting/standing while shaking.

Additionally, we quantify the impact of devices being apart from each other during shaking (being held in the hand and being strapped to the wrist). To compare our results to results of devices being pressed against each other we apply our approach to the "shake well before use" database [21][2]. In this database devices were pressed against each other and the start of the recordings on both devices were exactly synchronized (see figure 5; note the visually observable similarity between the time series in contrast to differences in our data in figure 2).

[2]From the "shake well before use" database we use the recordings of dataset 1 in which devices were shaken both either in the right or left hand.



(a) Sensor 1



(b) Sensor 2

Figure 5: Active segments from [21] detected independently for sensor 1 and 2.

To report the error rates of unlocking we measure the true match rate (TMR) and the true non match rate (TNMR). The TMR represents true positive identifications of devices being shaken together. In contrast, the TNMR represents true negative identifications of devices being shaken together. The false match rate (FMR) typically used for receiver operating characteristic (ROC) curves is the inverse of the TNMR.

To measure the TMR we compare datasets of devices that were actually shaken together (positive class size C_P). To measure the TNMR we compare datasets with all other datasets of devices not shaken concurrently (negative class size C_N) – which simulates a possible attacker trying to unlock the device by shaking it concurrently with the user (see equation 4 and table 3).

$$\begin{aligned} C_P &= \text{Sub} \cdot \text{Sets} \\ C_N &= 2 \cdot \text{Sub} \cdot \text{Sets} \cdot (2 \cdot \text{Sub} \cdot \text{Sets} - 1) \end{aligned} \tag{4}$$

5.1 Parameterization

For our evaluation we adapt parameters from successful previous research on shaking time series comparison [21]. For determining the start of users actively shaking their device we use a 0.5 s rectangular sliding window with a variance threshold of $6 \cdot 10^{-4} \frac{m}{s^2}$ – which we found to reliably and accurately detect the start of active shaking (see figure 2)[3]. Active segments are detected and extracted for all data samples independently, then cropped to a length of 1.2 s–5 s to simulate users shaking devices that long. To calculate the coherence based similarity of active segments first we use a 1 s Hann/Hanning sliding window with $\frac{7}{8}$ overlap to split acceleration time series into slices. To average the overall coherence of all slices we use a cutoff frequency of $f_{\max} = 40\,\text{Hz}$. To finally obtain a accept/reject decision we apply a coherence threshold $T = [0, 1]$.

[3]For data from the shake well before use database we use a variance threshold of $2.25 \cdot 10^{-3} \frac{m}{s^2}$ according to their findings.

Database	Sub	Sets	C_P	C_N
u'smile ShakeUnlock	29	20	580	672.2K
Shake well before use [21]	51	20	1020	2.1M

Table 3: Amount of tests performed, resulting from different numbers of subjects (Sub) and datasets (Sets) in databases, to check for positive matches (positive class size C_P) and negative matches (negative class size C_N) in order to obtain overall match and error rates.

5.2 Impact of shaking duration and devices being apart from each other

Results show that increasing shaking durations decreases overall error rates – for devices being held in the hand and strapped to the wrist, as well as devices being pressed against each other (see figure 6).

For devices being strapped to the wrist and held in the hand per shaking duration key metrics are listed in table 4. The equal error rate (EER) states the error rate for TMR = TNMR. Listed TMR and TNMR values were selected using the square root of the minimum squared error $\sqrt{\text{MSER}}$ [4] (see equation 5) as representing the ROC curve point closest to TMR = TMR = 1.

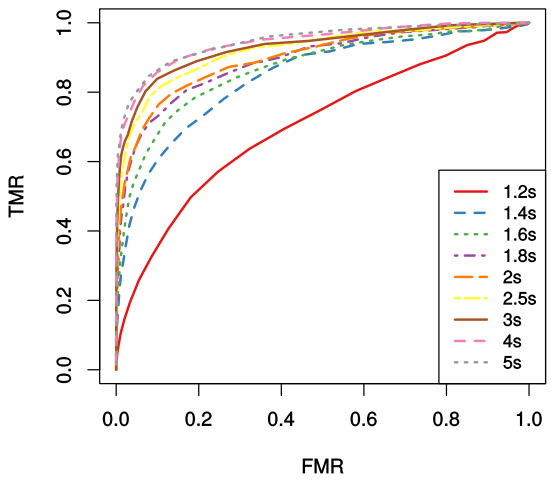
$$\sqrt{\text{MSER}} = \sqrt{\min(\text{FNMR}^2 + \text{FMR}^2)} \tag{5}$$

Shaking	EER	TMR	TNMR	$\sqrt{\text{MSER}}$
1.2 s	0.343	0.637	0.677	0.513
1.4 s	0.240	0.733	0.789	0.378
1.6 s	0.203	0.785	0.808	0.304
1.8 s	0.182	0.806	0.829	0.274
2.0 s	0.176	0.795	0.867	0.289
2.5 s	0.153	0.819	0.889	0.256
3.0 s	0.139	0.838	0.901	0.229
4.0 s	0.121	0.856	0.902	0.203
5.0 s	0.122	0.846	0.922	0.218

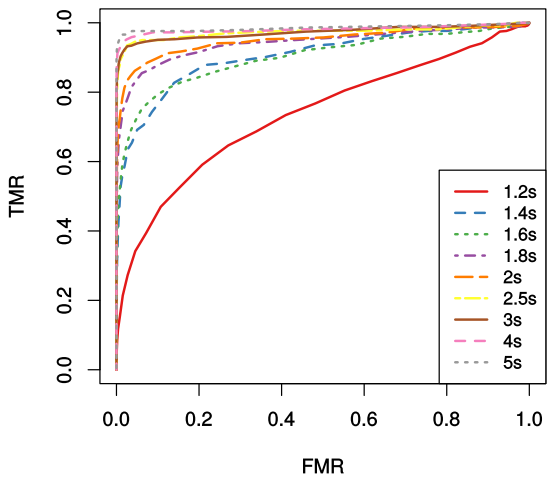
Table 4: Best true match rates (TMR) and true non match rates (TNMR) for different durations of users shaking their device.

Using a shaking duration of 2 s – which we assume is just short enough for users to consider shaking as an unlocking approach – we obtained an EER of 0.176 and a TMR/TNMR of 0.795 and 0.867, respectively. These rates assume that both devices are shaken concurrently. Consequently, attackers trying to unlock the mobile phone which they previously

[4]$\sqrt{\text{MSER}}$ represents the euclidean distance between the point TMR = TMR = 1 and the TMR/TNMR closest to this point.



(a) Devices at hand and wrist



(b) Devices pressed against each other

Figure 6: ROC curves with true match rate (TMR) and false match rate (FMR) for different durations of users shaking their device, with the devices being strapped to the wrist and held in the hand using our database (left) and being pressed against each other in one hand using the database of [21] (right).

got under their control have to perform this attack in parallel to users shaking their wrist watch accordingly. Further, the unlocking security level can easily be raised for users willing to shake their device longer (which could be chosen per user and application individually).

Using data of devices being pressed against each other for 2 s of shaking, we obtain an EER of 0.100 and a TMR/TNMR of 0.885 and 0.925, respectively (see figure 6(b)) – which is observably better over devices being apart form each other. These results support the intuition that the closer devices are together, the harder it is for an attacker to trick the approach into unlocking the mobile phone using non-correlated shaking. Furthermore, this suggests that an attacker being able to attach an acceleration sensor at the user (e. g. in clothing) will not be able to make immediate use of recorded acceleration data, except for when the acceleration sensor is very close to the wrist or hand, as the recordings will differ too much from the actual device acceleration.

Based on the shake well before use database, Mayrhofer and Gellersen [21] report a TMR and TNMR of 0.010 and 0, respectively. This differences to our current findings are caused by utilizing a different threat model and differently sized negative classes C_N (time series used to compute the TNMR). To obtain the TNMR, the earlier evaluation uses a small dataset of 177×2 time series recorded by shaking devices simultaneously, but not with the same hand. Based on the resulting 177 time series comparisons, the TNMR is computed. In contrast, in this paper we utilize the same dataset to obtain the TNMR as well as the TMR: we compare all time series not recorded by shaking devices simultaneously with the same hand to compute the TNMR. Consequently, our current evaluation uses a far larger negative class (2.1 million comparisons for the shake well before use database) to obtain the TNMR.

5.3 Impact of sitting/standing and shaking with dominant/non dominant hand

Figure 7 shows the impact of shaking devices with the dominant and the non dominant hand as well as sitting or standing while shaking the devices based on our database.

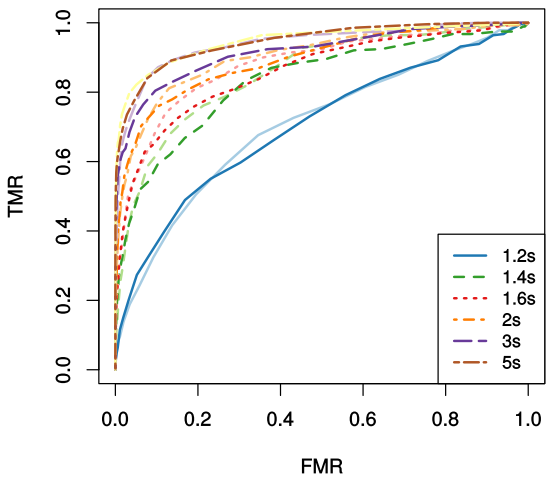
It is clearly visible that shaking devices with the dominant hand (represented by the brighter lines in the left graph) with an EER of 0.168 and a TMR/TNMR of 0.811/0.870 for 2 s of shaking consistently causes lower error rates compared to shaking devices with the non dominant hand (represented by the darker lines) with an EER of 0.184 and a TMR/TNMR of 0.779/0.863. We assume this to be the result of users shaking the devices slightly harder and/or faster as well as keeping the wrist more stiff – therefore causing more similar acceleration time series on both devices.

Similar to using the dominant or non dominant hand, sitting while shaking devices seems to cause slightly lower error rates compared to standing – with sitting (represented by the brighter lines in the right graph) causing an EER of 0.176 and a TMR/TNMR of 0.806/0.866 for 2 s of shaking compared to standing (represented by the darker lines) causing an EER of 0.177 and a TMR/TNMR of 0.818/0.828.

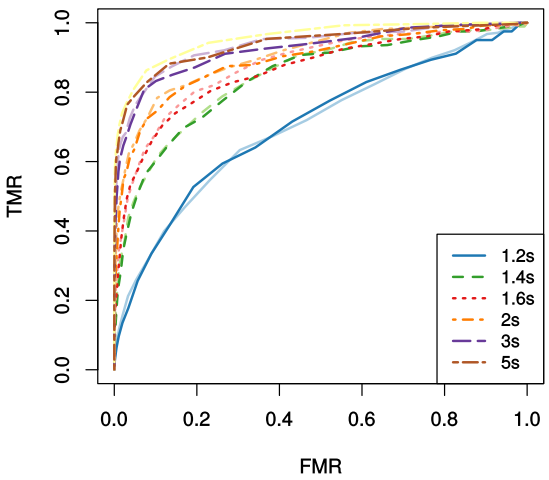
5.4 Implementation

To measure the actual time required for unlocking the mobile phone we implemented a ShakeUnlock prototype based on our approach. This way the duration of establishing a link between the wrist watch and mobile phone (using previous pairing) and transmission of an active segment can be incorporated in our measurement. The ShakeUnlock prototype records active segments of 2 s to determine if devices were shaken together – but starts to establish the link right after the start of an active segment is detected.

We tested our prototype using the Moto G and Simvalley watch also used for recording the u'smile shake database. Results show that the average time it takes to unlock the mobile phone after users start to shake their wrist watch is in the range of 2.5 s, with the additional 0.5 s over 2 s of shaking caused mainly by powering the link between devices. These results indicate that unlocking a mobile phone by shaking it together with a wrist watch is practically feasi-



(a) Dominant and non dominant hand



(b) Standing and sitting

Figure 7: Devices being shaken for different durations with the non dominant hand (left, dark) and with the dominant hand (left, bright) as well as when standing (right, bright) and sitting (right, dark).

ble, compared to the widely deployed unlocking mechanisms PIN and graphic pattern (with average unlocking times of 1.5 s and 3 s) – while requiring less user attention.

6. CONCLUSION

In this paper, we propose to use the shaking interaction that was previously proposed to pair mobile devices for unlocking one of the devices. To transfer the authentication state from one device (e.g. a wrist watch) to the other (e.g. a mobile phone), we create a one-time event by shaking them together. In comparison to previous research, we focused on a new use case – unlocking a device already paired to another trusted device – and determined the required data analysis parameters for this setting. We specifically analyzed devices being shaken apart from each other (being strapped to the wrist and held in the hand) and utilize sensors embedded in off-the-shelf devices. In terms of practical use, we do not maintain an active link between devices in order to not drain the battery.

For purpose of evaluation we recorded the u'smile Shake-Unlock database, which contains 20 acceleration time series sets of pairwise shaking a mobile phone and wrist watch each, for 29 participants, in 4 different settings. Our evaluation shows that transferring the authentication state from an unlocked wrist watch, strapped to users wrists, to a locked mobile phone, held in their hands by shaking them together for only 2 seconds is possible with a true match rate of 0.795 and a true non match rate of 0.867. Applying our approach to data of devices that were pressed against each other during shaking resulted in considerable improvements – which affirms previous intuition that the device proximity plays a major role when shaking devices together. We explicitly note that the increased false non match rate (in comparison to previous research with devices shaken in the same hand) is still acceptable for our new use case of unlocking devices because of a different threat model: we do not derive long-lived cryptographic shared secret keys from the shaking motion, but only create one-time unlocking events if – and only if – both devices are shaken at the same time and with sufficiently similar motion patterns. This use case avoids brute force off-line attacks as discussed in our threat model.

Future work might evaluate in depth the impact of a) users bending their wrists during shaking and b) users not feeling comfortable with strapping their watch too tightly to their wrist, which will cause the watch to move and which we both assume to have additional impact on error rates of our approach. Combination of time and frequent domain features might also be effective for our approach and therefore be in focus of future research.

7. ACKNOWLEDGMENTS

This work has been carried out within the scope of *u'smile*, the Josef Ressel Center for User-Friendly Secure Mobile Environments. We gratefully acknowledge funding and support by the Christian Doppler Gesellschaft, A1 Telekom Austria AG, Drei-Banken-EDV GmbH, LG Nexera Business Solutions AG, and NXP Semiconductors Austria GmbH.

8. REFERENCES

[1] S. Antifakos, B. Schiele, and L. E. Holmquist. Grouping mechanisms for smart objects based on implicit interaction. In *Proceedings of UBICOMP 2003 Interactive Posters*, pages 207–208, Seattle, Washington, USA, 2003.

[2] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX conference on offensive technologies*, pages 1–7, Berkeley, CA, USA, 2010.

[3] N. Ben-Asher, N. Kirschnick, H. Sieger, J. Meyer, A. Ben-Oved, and S. Möller. On the need for different security methods on mobile phones. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 465–473, New York, NY, USA, 2011. ACM.

[4] D. Bichler, G. Stromberg, M. Huemer, and M. Löw. Key generation based on acceleration data of shaking

processes. In *Proceedings of the 9th International Conference on Ubiquitous Computing*, UbiComp '07, pages 304–317, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] C. Castelluccia and P. Mutaf. Shake them up!: A movement-based pairing protocol for cpu-constrained devices. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, pages 51–64, New York, NY, USA, 2005. ACM.

[6] N. Clarke and S. Furnell. Authentication of users on mobile telephones – a survey of attitudes and practices. *Computers and Security*, 24(7):519–527, 2005.

[7] W. Dargie. Analysis of time and frequency domain features of accelerometer measurements. In *Proceedings of 18th Internatonal Conference on Computer Communications and Networks, 2009. ICCCN 2009.*, pages 1–6, 2009.

[8] K. Fujinami and S. Pirttikangas. A study on a correlation coefficient to associate an object with its user. In *3rd International Conference on Intelligent Environments, 2007, IE 07*, pages 288–295, 2007.

[9] B. Groza and R. Mayrhofer. Saphe: Simple accelerometer based wireless pairing with heuristic trees. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, MoMM '12, pages 161–168, New York, NY, USA, 2012. ACM.

[10] K. Hinckley. Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 149–158, New York, NY, USA, 2003. ACM.

[11] D. Hintze, R. D. Findling, S. Scholz, and R. Mayrhofer. Mobile device usage characteristics: The effect of context and form factor on locked and unlocked usage. In *Proc. MoMM 2014: 12th International Conference on Advances in Mobile Computing and Multimedia*, New York, NY, USA, Dec. 2014. ACM.

[12] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Proceedings of the 3rd International Conference on Ubiquitous Computing*, UbiComp '01, pages 116–122, London, UK, UK, 2001. Springer-Verlag.

[13] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *Proceedings of Smart Objects and Ambient Intelligence Soc-EUSAI 2005*, pages 159–163. ACM Press, October 2005.

[14] D. Kirovski, M. Sinclair, and D. Wilson. The Martini Synch. Technical Report MSR-TR-2007-123, Microsoft Research, September 2007.

[15] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun. Caveat eptor: A comparative study of secure device pairing methods. In *IEEE International Conference on Pervasive Computing and Communications, 2009. PerCom 2009.*, pages 1–10, 2009.

[16] J. Lester, B. Hannaford, and G. Borriello. "Are You with Me?" - using accelerometers to determine if two devices are carried by the same person. In *Proceedings of the 2nd International Conference on Pervasive Computing*, pages 33–50, 2004.

[17] H. Lu, A. J. B. Brush, B. Priyantha, A. K. Karlson, and J. Liu. Speakersense: energy efficient unobtrusive speaker identification on mobile phones. In *Proceedings of the 9th international conference on Pervasive computing*, Pervasive'11, pages 188–205, Berlin, Heidelberg, 2011. Springer-Verlag.

[18] R. Marin-Perianu, M. Marin-Perianu, P. Havinga, and H. Scholten. Movement-based group awareness with wireless sensor networks. In *Proceedings of the 5th International Conference on Pervasive Computing, Pervasive'07*, pages 298–315. Springer-Verlag, 2007.

[19] R. Mayrhofer, J. Fuss, and I. Ion. UACAP: A unified auxiliary channel authentication protocol. *IEEE Transactions on Mobile Computing*, 12(4):710–721, Apr. 2013.

[20] R. Mayrhofer and H. Gellersen. Shake well before use: Authentication based on accelerometer data. In *Proc. Pervasive 2007: 5th International Conference on Pervasive Computing*, volume 4480 of *LNCS*, pages 144–161. Springer-Verlag, May 2007.

[21] R. Mayrhofer and H. Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing*, 8(6):792–806, 2009.

[22] I. Muslukhov, Y. Boshmaf, C. Kuo, J. Lester, and K. Beznosov. Understanding users' requirements for data protection in smartphones. In *Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on*, pages 228–235, 2012.

[23] F. Schaub, R. Deyhle, and M. Weber. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 13:1–13:10, New York, NY, USA, 2012. ACM.

[24] A. Studer, T. Passaro, and L. Bauer. Don't bump, shake on it: The exploitation of a popular accelerometer-based smart phone exchange and its secure replacement. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 333–342, New York, NY, USA, 2011. ACM.

[25] F. Tari, A. A. Ozok, and S. H. Holden. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proceedings of the second symposium on Usable privacy and security*, SOUPS '06, pages 56–66, New York, NY, USA, 2006. ACM.

[26] E. von Zezschwitz, P. Dunphy, and A. De Luca. Patterns in the wild: a field study of the usability of pattern and pin-based authentication on mobile devices. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, MobileHCI '13, pages 261–270, New York, NY, USA, 2013. ACM.

[27] E. von Zezschwitz, A. Koslow, A. De Luca, and H. Hussmann. Making graphic-based authentication secure against smudge attacks. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 277–286, New York, NY, USA, 2013. ACM.