ShakeUnlock: Securely Transfer Authentication States Between Mobile Devices

Rainhard Dieter Findling, Muhammad Muaaz, Daniel Hintze, René Mayrhofer

Abstract—As users start carrying multiple mobile devices, we propose a novel, token based mobile device unlocking approach. Mobile devices are conjointly shaken to transfer the authentication state from an unlocked token device to another device to unlock it. A common use case features a wrist watch as token device, which remains unlocked as long as it is strapped to the user's wrist, and a locked mobile phone, which is unlocked if both devices are shaken conjointly. Shaking can be done single-handedly, requires little user attention (users don't have to look at the device for unlocking it) and does not cause additional cognitive load on users. In case attackers gain control over the locked phone, forging shaking is difficult, which impedes malicious unlocks. We evaluate our approach using acceleration records from our 29 people sized ShakeUnlock database and discuss influence of its constituent parts on the system performance. We further present a performance study using an Android implementation and live data, which shows the true negative rate of observational attacks to be in the range of 0.8 – if an attacker manages to gain control over the locked device and shake it in parallel to the device owner shaking the token device.

Index Terms—Mobile Computing, Human factors, Measurement techniques, Authentication

1 Introduction

Many people already carry multiple mobile devices such as mobile phones, tablets, and smart watches. Other wearable computing gadgets (e.g. activity or fitness trackers) are on the rise as well. Most of these devices have access to, process and/or store sensitive information [2]. Well-known examples include, but are not limited to, communications (email, SMS, instant messaging), context information (location), access to non-public networks (WiFi, VPN), access to payment or identity management applications, photos, documents, and even health related information (e.g. heart rate). In addition, with the "Bring your own device" trend, employees start to store and process company data on private devices (cf. [3], [4]). To prevent attackers from gaining access to data stored on these devices, locking and unlocking mechanisms have been developed. Those lock devices while not being used (e.g. after a short idle timeout) and users have to unlock them before usage. While authentication conceptually is divided into knowledge-, biometrics-, and token based-authentication [5], [6], so far approaches for mobile devices mostly utilize either knowledge- or biometrics-based authentication.

- *R. Findling, M. Muaaz and R. Mayrhofer are with u'smile, the Josef Ressel Center for Secure Mobile Environments, University of Applied Sciences Upper Austria, and the Institute of Networks and Security, Johannes Kepler University Linz, Austria.*
  E-mail: rainhard.findling@fh-hagenberg.at
- *D. Hintze is with the FHDW University of Applied Sciences Paderborn, Germany, and the Institute of Networks and Security, Johannes Kepler University Linz, Austria.*
- *A preliminary version of this work was published in MoMM 2014 [1], which is extended by systematic performance improvements and analysis of their cause, an extended threat model, and a real world performance study using an implementation.*

The most widely used mobile knowledge-based unlocking mechanisms are PIN, password and graphical pattern [7]. All of them increase the cognitive load on the user [8], [9] and require a certain time to enter the secret knowledge – which might be cumbersome due to small user interfaces on mobile devices [10]. In addition, mobile devices are unlocked more frequently than, e.g., desktop computers, but used for shorter periods of time (cf. [11], [12], [13], [14]), which deteriorates the unlock-to-usage-time ratio. Therefore the impact of increased cognitive load and effort required to perform the unlock is higher on mobile devices than on desktop computers. Furthermore, knowledge based unlocking approaches are vulnerable to shoulder surfing attacks (attackers watching the authentication process, thereby observing the unlocking secret, cf. [15], [16]) and smudge attacks (attackers screening the display after the user authenticated using a graphic pattern to observe the residual smudge that might remain on the display, thereby observing the unlocking secret, cf. [17], [18], [19]).

Biometrics-based approaches most commonly used on mobile devices include fingerprint (e.g. Apple TouchID), face, or voice (cf. [5], [20]). While those are easy to use and do not increase the cognitive load, the drawback lies with securing biometrics of users. Unlike with knowledge- or token-based authentication, biometric features cannot be changed – which increases the impact of leakage or theft of biometrics. In addition, hardware used to capture and process biometrics on mobile devices is often proprietary, which makes identifying and analyzing potential security issues difficult.

In contrast to knowledge-based authentication and biometrics, token-based unlocking mechanisms are rarely used on mobile devices. Most approaches proposed so far are based on proximity of token and device

to perform the unlock. Examples include transmitting a secret from token to device via bluetooth, NFC, or IEEE 802.11 (WiFi) [21], [22], [23], using the device magnetometer to determine proximity to the token, or mobile speakers and microphones to transmit/receive a secret [24].

Proximity based approaches have the drawback of attackers possibly being able to unlock the mobile device they got under their control if they are close to the user. For example, with WiFi- or Bluetooth-based approaches it might be sufficient to be in the same room with the legitimate user to successfully unlock the device. As attackers are likely to be close to the user when obtaining control over the mobile device, an immediate unlock would be possible before leaving the scene. When using token-based authentication, the token needs to be brought by users everywhere they potentially want to use their mobile device. Depending on where the token is kept, it could be possible to obtain control over both token and device at once and then use the token to unlock the device. If the token itself is locked to prevent illegitimate usage in case of theft, the whole problem is transfered from the mobile device to the token – as unlocking the token itself again could be done using knowledge-, biometrics- or token-based authentication.

To address these issues we propose a novel token-based mobile device unlocking approach: transferring the authentication state between two devices by briefly shaking them conjointly. The key idea is that personal mobile devices can remain unlocked for different periods of time, one could act as a token, allowing to transfer authentication state between devices. For example, a mobile phone should lock itself as soon as it is put aside while a smart watch could remain unlocked as long as it is strapped to the wrist and automatically lock itself when detached. The smart watch could e.g. be unlocked once in the morning when attached to the wrist and automatically lock itself when detached, utilizing e.g. heart-rate measurements like with the Apple Watch[1] or a simple connection in the strap that is triggered by opening it. Using this setup, the authentication state from the unlocked watch can be transferred to the locked phone to unlock it – hence the unlocked device can serve as token for unlocking other devices. Shaking both devices simultaneously with the same hand serves as a fast, easy and secure trigger for authentication state transfer. The authentication state transfer is only triggered after an analysis of sensor time series recorded on both devices concludes that a) both devices have been shaken simultaneously and b) both devices have been shaken by the same person. For simplicity, from now on we will refer to the device from which the authentication state is transferred as *token device* where applicable.

Unlocking mobile devices by shaking them conjointly has noteworthy advantages over other unlocking approaches. Required user attention is assumed to be lower compared to current unlocking approaches, as users only need one hand and are not required to look at the devices to unlock them. In terms of speed, studies show that unlocking duration ranges from 1.5 s (PIN) to 3 s (unlock pattern) [14], [25]. We assume that these 1–3 s are considered an acceptable unlocking delay in terms of usability vs. security. To be comparable to other unlocking mechanisms, we aim for 2 s of shaking to transfer authentication states between devices while requiring less user explicit attention.

Shaking devices can be utilized on a broad range of mobile devices nowadays as accelerometers are a common feature of mobile phones, tablets and smart watches as well as activity trackers and other wearable computing gadgets. Previous research on pairing mobile devices by shaking them conjointly has stated shaking to be secure, as acceleration records are difficult to forge by shaking devices bare handed [26], making it a suitable choice for security critical applications[2]. We base our approach on these findings but focus on a different use case: transferring authentication states from a token device to another device to unlock it. Consequently, the scenario presented here implies different approaches towards security and usability with analyzing acceleration sensed on both devices. This article focuses on the technical aspect and security implications of ShakeUnlock – and leaves a thorough evaluation of usability and acceptance for future work, as such a study would need to consider longitudinal effects of muscle memory/muscle learning (users being able to perform movements without explicitly thinking about them, like 10-finger-typing on a keyboard). Summarizing, our contributions are:

- In contrast to previous research on shaking mobile devices conjointly to establish a secure channel between them, we focus on shaking as a secure trigger mechanism to transfer authentication states from a token device to another device over a pre-established secure channel.
- Our approach processes data from mobile devices situated 10-15 cm apart from each other (mobile phone held in the hand, smart watch strapped to the wrist) with the wrist as a non-static joint in between, which implies differences in sensed acceleration on both devices.
- Using this setup we record the ShakeUnlock database containing 3D acceleration and 3D gyroscope time series recordings of mobile devices being shaken conjointly. We use this data to parameterize and evaluate our approach.
- We give detailed insight into our approach to pairwise shaking time series similarity data analysis. We state in which way and how much constituent parts contribute to the overall system performance.

1. Apple Watch heart rate measurements: https://support.apple.com/en-us/HT204666

2. Hypothetical attacks could involve e.g. high speed cameras and an apparatus to precisely recreate visually observed shaking behaviors but are beyond the scope of this work.

We believe that future approaches can benefit from these detailed insights and findings.
- We implement our approach on Android and present a performance study which evaluates three different attack scenarios.

At first we give an overview of related work on shaking devices (Sec. 2) and outline the threat model for our approach (Sec. 3). We then present our approach (Sec. 4), state details about analysis concepts and evaluate their influence on system performance (Sec. 5). Finally, we present our implementation and an evaluation of different attack scenarios using live data (Sec. 6).

2 Related Work

2.1 Shaking Mobile Devices Conjointly

Analyzing movement and acceleration records for determining if mobile devices were shaken together by the same body movement has been subject of a significant body of research over the last 10 years. Research ranges from analysis of simple movements with accelerometer recordings (cf. [27], [28]) to deriving secret keys from acceleration data (cf. [26], [29], [30], [31], [32]).

With "Smart-Its Friends", Holmquist et al. [33] have been amongst the first to associate devices by shaking them together. Their devices sense acceleration and broadcast it, so that other devices may decide on pairing with them. Their approach purely focuses on pairing without taking security aspects like Man-in-the-middle (MITM) or replay attacks into account. In "Are You with Me?", Lester et al. [34] have built upon this work but use frequency domain based magnitude squared coherence instead of time domain based analysis to pair devices. Their approach has further been extended by Mayrhofer and Gellersen in "Shake Well Before Use" [26] which additionally covers security aspects of pairing devices by shaking them conjointly.

"Shake Them Up" by Catelluccia and Mutaf [35] utilizes a related idea, although it does not involve sensing acceleration. They monitor WiFi received signal strength indication (RSSI) which is likely to change when devices are moved/rotated. As devices are moved together they experience similar changes in RSSI over time on the basis of which devices decide if they have been moved together. This approach is designed with MITM protection in mind. However, it depends on wireless signals and wireless signal strength sensing capabilities to be available on both devices.

The special aspect of shaking devices conjointly which are apart from each other and have a non-static joint (e.g. the wrist) in between was addressed by Fujinami and Pirttikangas [36] for associating objects with users. Amongst other things they consider toothbrushing with sensors attached to the users hands and toothbrushes. Similarly, Bao and Intille [37] have investigated activity recognition including tooth brushing from 2D acceleration sensors and time domain features. We deal with the same complicating issues for robust acceleration time series comparison due to having a non-static joint between devices, which will cause devices to sense slightly different acceleration during shaking. Additionally, we have to consider security implications of attackers trying to forge acceleration patterns to get access to obtained devices.

In terms of data analysis, shared movement and shaking has been analyzed in both time and frequency domain. For in depth comparison we refer to [38], [39] as well as related research from the field of activity recognition (cf. [40], [41], [42]). Although analysis in time domain seems to be capable of yielding higher entropy [29], analysis in frequency domain seems more resistant to synchronization issues [34]. In our approach, devices independently record acceleration and decide if they are currently shaken. Devices will sense slightly different acceleration due to the non-static joint in between them, hence detect active shaking at slightly different points in time. As we cannot assume exact synchronization between devices we use frequency based analysis. So far the most successful analysis approach is using frequency-domain based magnitude squared coherence [43], which has been used in various previous studies (cf. [26], [34], [44], [45], [46], [47]) and which is utilized in our approach as well.

2.2 Implications of Shaking on Security

In 2011, Studer et al. [48] proved the well known and by now discontinued mobile phone application "Bump"[3] to be insecure. With "Bump" and similar approaches such as simultaneously pressing a button on both devices (cf. [28], [49], [50]) correct timing is the only critical aspect to establish a channel between devices. As timing cannot be assumed secret, attackers can easily perform MITM attacks by forging required information and communicating them with correct timing. Instead of using timing constraints we utilize shaking to trigger the transfer of authentication state from the token device to other devices. Consequently, resistance against forged shaking patterns is required to prevent attackers from triggering an authentication state transfer without being in control of both devices at the same time.

Most previous research on shaking mobile devices conjointly in the scope of security aim to establish a secure channel between devices [26], [29], [30], [51], [52] (also known as bootstrapping or human verifiable authentication problem [53]). In contrast to these approaches we study shaking as trigger mechanism to transfer an authentication states from the token device to other devices over an pre-established secure channel.

3 Threat Model

We want to emphasize that a) a user in control of the unlocked token device and the locked phone is intentionally able to trigger the authentication state transfer to

3. See http://bu.mp

unlock the phone, as no biometric authentication is performed. b) the authentication state transfer is triggered if – and only if – the token device is unlocked and the phone is locked when both devices are shaken conjointly, which renders being in control of the locked token device and phone insufficient for attacks. Consequently, access protection for the token device is required. As discussed before, when assuming that users attach their locked token device to their wrist once a day, then unlock it (e.g. in the morning), the token device can stay unlocked until users lock it manually or it is detached from the wrist. Compared to access to an unlocked phone or regular authentication token not featuring a locking mechanism, we argue that this brings an increased level of access protection to the unlocked token device:

- It is more difficult for the token device to be lost or stolen, as it is attached to the users wrist.
- For attackers it is more difficult to obtain/access to the unlocked token device, as it automatically locks itself when detached from the wrist and accessing it in an unlocked state therefore would require accessing it before detaching it from users wrist, which is unlikely to go unnoticed.

For our scenario we therefore assume the token device to be secure and restrict addressed attack scenarios to the locked phone being under control of an attacker. We further assume that the token device is unlocked, as otherwise no authentication state transfer can be triggered.

3.1 Attack Scenarios

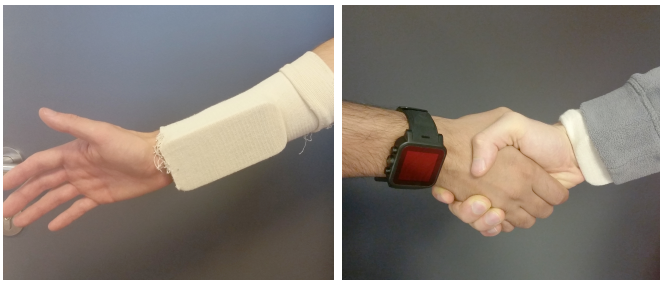
For all attack scenarios, the locked mobile phone is considered to be under physical control of an attacker trying to unlock it unnoticed by legitimate users who controls the token device. To trigger an authentication state transfer from the unlocked token device to the phone, simultaneous shaking of both devices is required. This implies the legitimate user also has to shake the token device, which is why an attacker must synchronize any attack attempts with the user's shaking of the token device. We address four such attack scenarios with different attacker capabilities:

Minimal effort attacks assume that users have been tricked into accepting a proxy device as their own and subsequently trie to unlock it by shaking it conjointly with the token device. Attackers simultaneously shake the target device they control but without trying to mimic the shaking pattern of users. Note that we use the term "minimal effort" because attackers does not take additional effort such as imitating users' shaking behavior. Sophisticated preparation, e.g. obtaining control over the device beforehand and tricking users into taking a different device for their own, is still required for this kind of attack. Being resistant against minimal effort attacks means being resistant against two people separately shaking both devices at the same time to trigger an authentication state transfer.

Observatory attacks use the same setup as minimal effort attacks, but attackers are observing the legitimate users and attempt to synchronously mimic the users' shaking patter to unlock the device, without the legitimate users noticing.

Cooperative attacks allow any cooperation between user and attacker except touching each other or the other's device in order to achieve high similarity in shaking patterns. This attack is supposed to break the approach and serve as measure of upper boundary to the security achieved, as in terms of authentication it is both unrealistic and harder than both previous attacks.

Handshake attacks assume attackers strap the mobile phone to their wrist using a bandage (see figure 1). Then users and attackers shake hands hard to achieve synchronized acceleration records on both devices. This requires the hand to which wrist the token is attached to be used for the handshake. As with cooperative attacks, handshake attacks are supposed to break the approach. In a real life scenario, attackers shaking users' hands as hard as required to trigger recording of continuous 2 s shaking would be unrealistic, as it is far from natural and would make users suspicious.



(a) (b)

Fig. 1. Possible handshake attack setup with a) the mobile phone being strapped to the attacker's wrist and b) attacker shaking the user's hand hard.

3.2 Attack Evaluation

From security perspective, evaluating these attacks scenarios could be done with a one-to-one matching of data aggregated from devices both shaken and not shaken conjointly. These can be used to state a) success rates of legitimately triggering authentication state transfer (true positive rates) and b) attack success rates (false positive rates). From a system parametrization perspective, a larger number of samples is required to obtain suitable distinguishing capabilities. We therefore use m-to-n matching of uncorrelated shaking samples in our data set to simulate minimal effort attacks which we use in turn to parameterize our approach (see section 5). To evaluate the remaining three attack scenarios we use an implementation of the proposed concept on off-the-shelf Android devices with one-to-one matching of live data (see section 6).

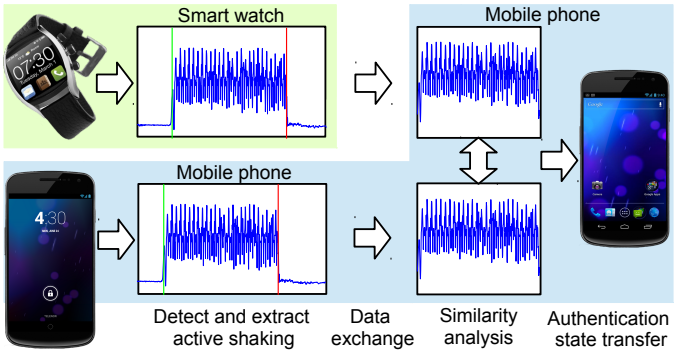


Fig. 2. Data processing chain used in the ShakeUnlock approach.

4 Our Approach

Our approach is split into two major steps: separately sampling acceleration on both devices and deciding upon triggering an authentication transfer between devices on one device (Fig. 2). The first step consists of monitoring acceleration, deciding if the device is shaken, and extracting an active shaking acceleration segment (*active segment*) independently on both devices. If active segments have been detected, both are aggregated on one device. In the second step the similarity of active segments is determined to decide if devices have been shaken conjointly and thus an authentication state transfer should be triggered. Note that in contrast to related approaches, no data is stored on the devices – not even in the form of cryptographic keys or hashes.

4.1 Active Segment Detection

In our approach devices continuously and separately monitor acceleration, which can be done without excessive draining of battery power by utilizing hardware dedicated to acceleration recording. Such hardware is already becoming available in off-the-shelf mobile devices, such as for background step counting in the Apple iPhone5, iPhone6 and Apple Watch, Samsung Galaxy S5 or Sony Xperia Z1(c)-Z3(c) devices. As shaking is detected, the power efficient hardware can e.g. power on the main CPU which then performs the computationally more expensive networking and time series comparisons tasks.

Our approach looks out for the start of an active segment by monitoring the variance of the acceleration magnitude of the 3D acceleration sensor in a sliding window as described in [26]. If the variance of acceleration within this window rises above a certain threshold, this marks the start of an active segment from which acceleration on 3 axes is recorded for a short duration, capturing the shaking of the device. For our evaluation and implementation we use an acceleration monitoring sliding window of 2 s, an acceleration variance threshold of $6 \cdot 10^{-4} \frac{m}{s^2}$ and record active segments of 2 s length after shaking is detected. If users prematurely stop shaking (i.e. active segment < 2 s), no authentication state transfer will be triggered.

After active segments have been detected and recorded separately on both devices, we aggregate them on one device. Data aggregation could be done on each of the devices, as both are assumed secure and connected via a secured channel. However, when transferring the authentication state from the watch (token) to the phone, data aggregation on the phone has the following advantages: a) Usually, mobile phones have higher computational power than smart watches, hence the decision on performing the authentication state transfer will be obtained faster. b) If we conclude to perform the authentication state transfer from watch to phone based on recorded active segments, no further data transfer between devices is required, as the decision is done on the phone already.

4.2 Authentication Transfer Decision

After active segments have been recorded on both devices individually and aggregated on one device, we analyze those active segments to determine if devices have actually been shaken conjointly. If so, we perform an authentication state transfer between devices to unlock the device still locked. Before performing the actual similarity analysis, we preprocess the two active segments. We compensate for gravity recorded within the active segments by subtracting the mean acceleration per axis throughout the active segment.

Our similarity analysis takes a pair of active segments as input and yields a scalar metric value as output. If this metric value is above a reference threshold, we conclude that active segments represent devices shaken conjointly, therefore trigger the authentication state transfer and unlock the locked device. If the metric value is below the predefined threshold, we conclude that active segments represent devices not shaken conjointly, therefore refuse the authentication state transfer and do not unlock the device. Our similarity analysis consists of different constituent parts, which we present and discuss in the next section.

5 Active Segment Similarity Analysis

Previously Mayrhofer and Gellersen [26] showed that it is feasible to detect if devices – which are pressed against each other – have been shaken conjointly using magnitude squared coherence on acceleration time series magnitudes. In previous research [1] we applied this method with adapted parameters and preprocessing to acceleration time series magnitudes of devices somewhat apart and with non-static joint in between during shaking. The presented extended approach additionally incorporates derotation of 3D time series before performing the similarity analysis, bandpass filtering, a different collapsing function, and optimal weighting of individual frequencies. In this section we discuss and evaluate

each constituent part and its influence on overall performance. Obtained performance comparisons are stated in section 5.9.

5.1 Parametrization and Evaluation Data

We parametrize and evaluate our approach using data from our publicly available ShakeUnlock database[4] [1] on the basis of devices shaken conjointly and simulated minimal effort attacks. Other attack scenarios are not based on this data and are separately covered in section 6. The ShakeUnlock database contains acceleration and gyroscope recordings of 29 participants shaking a wrist watch (strapped to their wrist) and mobile phone (held in the same hand). For each participant, 20 samples of shaking both devices for 10 s have been recorded – which results in 580 pairs of shaking samples and a total of 1 160 recordings in the database. In previous research we have evaluated the influence of shaking duration on unlocking performance [1]. Our findings support the intuition that increasing the shaking duration improves accuracy when assessing whether devices have been shaking conjointly, but obviously impair usability as the effort increases. We found a shaking duration of 2 s to constitute a reasonable trade-off between usability and security. Consequently, for this work we restrict ourselves to a shaking duration of 2 s, therefore extract one active segment of 2 s duration per time series recording. Active segments shorter than 2 s are excluded from further analysis, as this simulates users not shaking their devices long enough.

We use all 580 time series pairs of devices shaken conjointly as legitimate tries to trigger authentication state transfer between devices. Therefore, our positive class P is of size 580. To simulate minimal effort attacks we use all $580 \cdot 579 = 335\,820$ combinations of time series obtained from not shaking devices conjointly as our negative class N. Note that we exclude pairs of same type of devices (two mobile phones as well as two smart watches) as these scenarios are not realistic in real life[5].

5.2 Performance Measures

As the sizes of our P and N class differ notably, some performance measures like accuracy are not significant [54]. We therefore rely on a number of well known and more significant metrics in our evaluation. The true match rate (TMR) represents the ratio of correctly identified cases of users trying to trigger an authentication state transfer with devices being shaken conjointly (P class samples). Likewise, the true non match rate (TNMR) represents the ratio of correctly identified cases of minimal effort attacks, with devices not being shaken conjointly (N class samples). We obtain the TMR and TNMR for all possible metric thresholds, from which we construct the receiver operating characteristics (ROC) and the area under the ROC curve (AUC). Both ROC and AUC capture the overall performance instead of stating the performance at a specific metric threshold. The equal error rate (EER) states the error for TMR = TNMR, representing the intersection between the ROC curve and the diagonal from TMR = TNMR = 1 to TMR = TNMR = 0.

4. The ShakeUnlock database is available online at http://usmile.at/downloads.

5. This is different to our previous research [1] in which we included same device types being shaken conjointly. We consequently obtain slightly different performance rates with this evaluation.

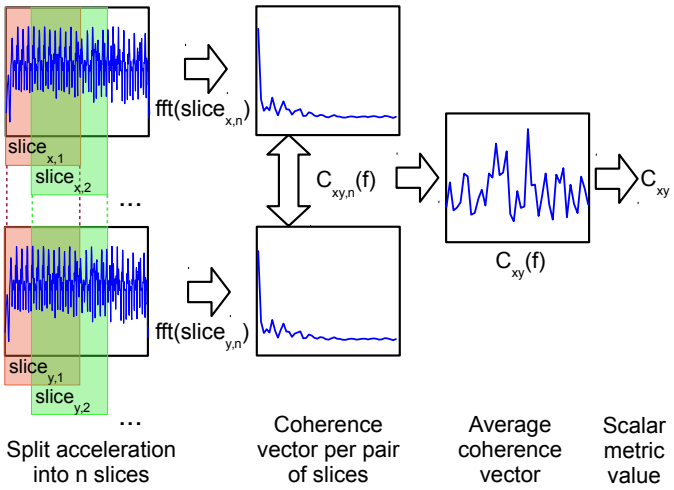


Fig. 3. Active segment similarity analysis in ShakeUnlock.

5.3 Magnitude Squared Coherence with Acceleration Time Series Magnitudes

With magnitude squared coherence [43] the time series x and y are divided into n overlapping slices (Fig. 3). Each slice is multiplied with a weighting window (such as a Hann or Hamming window). We use slices of $\frac{7}{8}$ overlap and 1 s duration (with 100 Hz sampling rate this corresponds to slice and window lengths of 100 samples), and a Hann weighting window as proposed in [26]. Next, all slices are transformed into the frequency domain by applying a standard fast Fourier transformation (FFT) with 1 s window size. For each pair of corresponding slices from x and y, the coherence vector $C_{xy,n}(f)$ is calculated from the power spectral densities $S_{xx,n}$ and $S_{yy,n}$ and the cross spectral density $S_{xy,n}$ (Eq. 1). Then, all n coherence vectors $C_{xy,n}(f)$ are averaged to the single coherence vector $C_{xy}(f)$ (Eq. 2).

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

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

Finally, a scalar metric value C_{xy} is obtained from $C_{xy}(f)$ using a collapsing function (Eq. 3).

$$C_{xy} = \mathrm{Col}(C_{xy}(f)) \tag{3}$$

This metric value C_{xy} is interpreted as confidence that devices have actually been shaken conjointly while recording x and y. Hence, if $C_{xy} \geq T$, with T being

a predefined metric threshold, we transfer the authentication state and unlock the device. If $C_{xy} < T$ we refuse to transfer the authentication state, leaving the device locked. We apply the method as summarized above on the time series magnitudes of the two active segments x and y. Using the magnitude acceleration time series is done frequently to compensate for unknown spatial alignment of accelerometers. Thereby, time series magnitudes are calculated from the L^2-norm of the active segment 3D acceleration time series. As collapsing function Col we average the coherence vector $C_{xy}(f)$ up to a cutoff frequency of 40 Hz (Eq. 4).

$$C_{xy} = \frac{1}{41} \cdot \sum_{f=0\,\mathrm{Hz}}^{40\,\mathrm{Hz}} C_{xy}(f) \tag{4}$$

Using only magnitude squared coherence with acceleration time series magnitudes, we obtain an AUC of 0.8990 and an EER of 0.1777.

5.4 Optimal Timeseries Derotation

Relying only on magnitudes for comparing acceleration time series between different devices implies losing some potentially important information in the form of rotational components during the movement. Most previous research has focused on magnitudes because in general orientation of devices potentially moved together and of the accelerometers within those devices is unknown.

Comparing movement in three dimensions instead of only the aggregated magnitude therefore requires: a) the assumption that the devices retain their relative orientation with regards to each other during the shared movement, and b) rotating one of the coordinate systems into the reference frame of the other. We refer to this process as *derotation* and it can be considered an optimization problem to find the rotation matrix that minimized the distances between two 3D vectors. Recent results show that a quaternion based approach can be used to solve this optimization problem analytically and that it improves the EER with various distance metrics on the same data set used before [55]. We apply this approach to derotation as one step for improving classification accuracy.

Fig. 4 states the coherence density over frequency for the P and N class for applying coherence on magnitudes as well as on all axes of previously derotated time series. Brighter areas represent lower coherence, darker areas represent higher coherence. Coherence is observably more dense for the P class when derotating time series before computing coherence instead of computing the series magnitudes (Fig. 4a and 4b). In contrast, the density for the N class is only marginally influenced by derotating time series before computing coherence by being slightly higher on average (Fig. 4c and 4d). This is to be expected, as correlated time series initially are rotated arbitrarily but intentionally contain similarity – which causes derotated time series to show noticeably higher similarity. In contrast, initially not correlated time series only have little coincidental similarity. Optimally rotating them therefore only causes an insignificant raise in similarity. This data suggests that for frequencies showing condensed coherence values, derotation of time series will improve class separation performance – which is supported by evaluation results stated below as well.

In contrast to comparing time series magnitudes we instead compute coherence for each pair of axes (which have been aligned through derotation). Therefore, coherence computation yields three separate coherence vectors, one per (aligned) pair of axes. Each coherence vector represents the frequency range 0-50 Hz for 100 Hz sampling in data recording. Hence, all successive operations (e.g. filtering frequencies by applying a 0-20Hz bandpass) have to be applied to these three coherence vectors individually. We apply the previously used 40 Hz cutoff to the coherence vectors, then average them to obtain a final, scalar coherence value. By adding initial time series derotation to our evaluation setup, we obtain an AUC of 0.9214 and an EER of 0.1562.

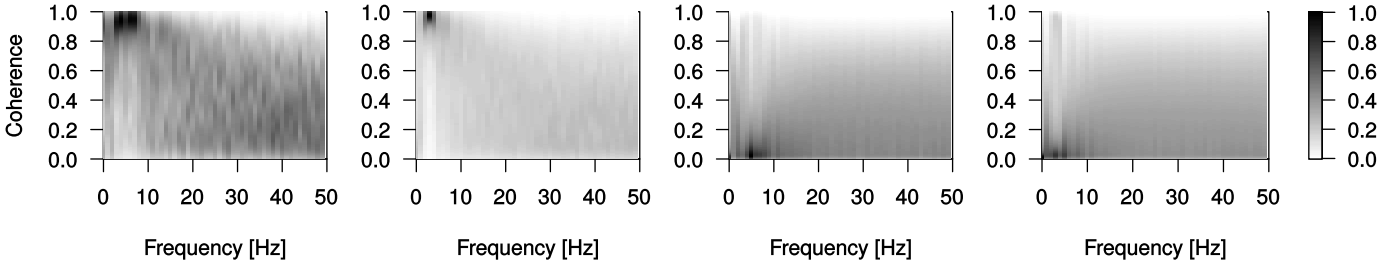
5.5 Coherence Frequency Bandpass

Overall, research on human body motion states quite different motion frequencies to usefully represent motion information. For example, in Biomechanics and Motor Control of Human Movement, Winter [56] states human body motion is in general represented by a frequency range of about 0-10 Hz. In contrast, e.g. Bouten et al. [57] find frequencies up to 20 Hz being useful to represent human movement during everyday activities. They further state that body movement of e.g. limbs is usually faster, compared to movement of torso and hip, whereas shaking mobile devices with the hand corresponds to the mentioned faster movements.

In their research on shaking devices conjointly, Lester et al. [34] pick up the frequency range of 0-10 Hz stated by Winter [56]. They average coherence in the range of 0-10 Hz to come up with a scalar similarity value. In contrast, Mayrhofer and Gellersen [26] average coherence in the range of 0-40 Hz to determine if devices were shaken conjointly without stating details on how this cutoff frequency was determined. It can be assumed that results from using a coherence range of 0-40 Hz were superior to results from using a range of only 0-10 Hz for their approach, for which the wider frequency range was used. To determine the optimal coherence frequency range we explicitly study the influence of different bandpass filters to classification performance.

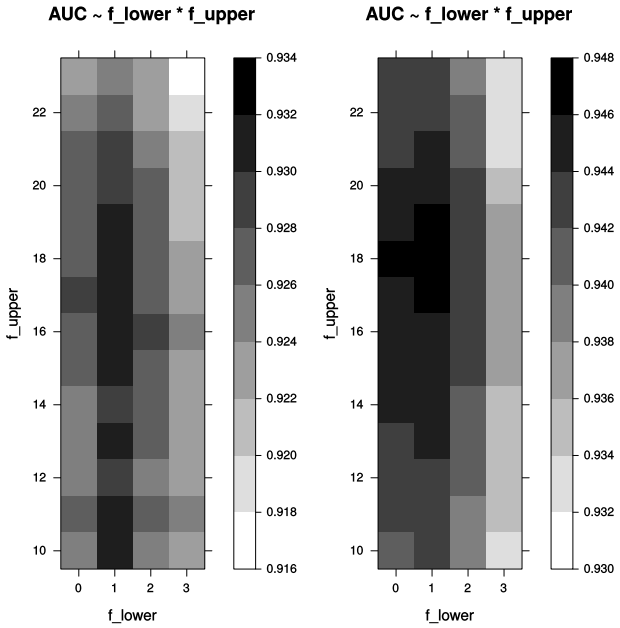
As shown in the coherence distribution over frequency (Fig. 4), coherence is unequally distributed over frequency in the ShakeUnlock database. Overall, coherence is less dense as well as less diverse across P and N class for higher frequencies, compared to lower frequencies, although the lowest frequencies in the range of 0-2 Hz are less dense and less diverse across classes as well.

In order to utilize the best performing coherence frequency range in our approach, we apply a bandpass to



(a) P, no derotation (b) P, derotation (c) N, no derotation (d) N, derotation

Fig. 4. Coherence densities per frequnecy of P and N class without and with time series derotation.



(a) On magnitude data (b) On derotated data

Fig. 5. Coherence bandpass performance (AUC per bandpass filter setting) when applied without (a) and with derotation of time series (b). Note that left and right brightness scaling is differently to increase distinguishability.

coherence frequencies before successively computing a scalar similarity value from the coherence vector. For real world applications and from an implementation point of view, using a bandpass has several advantages over more complex approaches of restricting the frequency range. Using a bandpass is intuitive and easy to understand. Further, it is fast and easy to implement and of small complexity. In our bandpass evaluation, f_L represents the lower frequency threshold, hence the lowest coherence frequency included during successive processing. Likewise, f_H represents the upper frequency threshold. The frequency bandpass performance (Fig. 5) states AUC over pairs of f_L and f_H, with darker areas representing higher AUC values, therefore better performance.

Note that with our setup, performance decreases notably when increasing f_L, while changes of f_H seem to have significantly less influence on performance. On the one hand, this indicates that the most important portion of information is contained in lower frequencies, and that higher frequency information is less reliable – which is in support of findings from previous research. If these lower frequencies are excluded, performance decreases significantly. On the other hand, including frequencies up to about 20 Hz can improve performance, which is different to what previous research would suggest [56].

With applying a bandpass to coherence frequencies from magnitudes of acceleration time series, performance peaks at $f_L = 1\,\text{Hz}$ (skipping the 0 Hz constant component) and $f_H = 16\,\text{Hz}$, with an AUC of 0.9315 and an EER of 0.1418. When combining the bandpass with initially derotating time series, peak performance is reached with consistent $f_L = 1\,\text{Hz}$ and a slightly higher $f_H = 18\,\text{Hz}$, with an AUC of 0.9469 and an EER of 0.1293. These results point out that coherence frequency range noticeably influences overall performance – and therefore should be selected carefully. In comparison to other constituent parts of our approach, using a coherence frequency bandpass turns out to hold the highest performance gain – while being amongst those easiest to implement.

5.6 Coherence Frequency Collapsing Function

In previous research on shaking devices conjointly, collapsing a coherence vector to a scalar coherence value has only been done by averaging coherence. To collapse a coherence vector, other functions are possible as well, with some of them being frequently used in other disciplines. We evaluate the following collapsing functions for obtaining a scalar similarity value from coherence vectors: sum (average), median, max, euclidean distance d_e, and square root distance d_s. Square root distance (Eq. 5) is the counterpart to euclidean distance (Eq. 6), by inversing the order of squaring and taking the square root. Additional functions such as min turned out to cause significantly worse performance in preliminary tests and therefore were disregarded in this evaluation.

$$d_s(v) = \left(\sum_i \sqrt{v_i} \right)^2 \quad (5)$$

$$d_e(v) = \|v\| = \sqrt{\sum_i {v_i}^2} \quad (6)$$

Performance comparisons (Fig. 6) show euclidean distance slightly outperforms averaging as well as all other tested functions when used to collapse coherence vectors to a scalar similarity value'for both time series magnitudes as well as initially derotated timeseries.

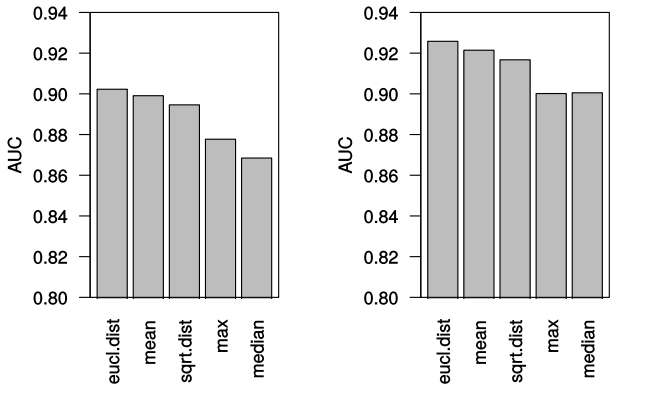
When applying euclidean distance as the best performing collapsing function to coherence obtained from time series magnitudes, we obtain an AUC of 0.9023 and an EER of 0.1670. In contrast, when applying euclidean distance as collapsing function conjointly with initially derotating timeseries and using a coherence frequency bandpass filter we obtain slightly reduced performance, with an AUC of 0.9464 and an EER of 0.1293.

On the one hand, these findings indicate that obtaining a scalar coherence value from a coherence vector might be improved by considering not only the mean, but alternative collapsing functions such as euclidean distance. On the other hand, when used with other constituent parts of our approach, the performance gain is minor (or as in our case, performance even decreased slightly).

5.7 Optimal Coherence Threshold per Frequency

5.7.1 Determining optimal coherence thresholds

After deriving a scalar similarity value from a coherence vector (obtained from two acceleration time series of devices shaken conjointly) usually one fixed threshold is used to separate the P and N class, as reported by Lester et al. [34] and Mayrhofer and Gellersen [26]. Using a single coherence threshold has a significant drawback: all frequencies are combined within one scalar value, therefore the threshold can only address all frequencies at once. Another approach is to use an individual and independent threshold for each coherence frequency. Each such threshold represents the optimal separation between P and N class for that coherence frequency – hence provides better class separation on individual frequency level. Optimal thresholds differ when derived from either time series magnitudes or from initially derotated time series as derotation changes coherence values (see example in Fig. 7). Fig. 8 states the optimal coherence threshold per frequency for using time series magnitudes as well as for incorporating initial time series derotation.



(a) Time series magnitudes (b) Derotated time series

Fig. 6. Influence of coherence vector collapsing functions on overall performance using a) time series magnitudes and b) initially derotated time series.

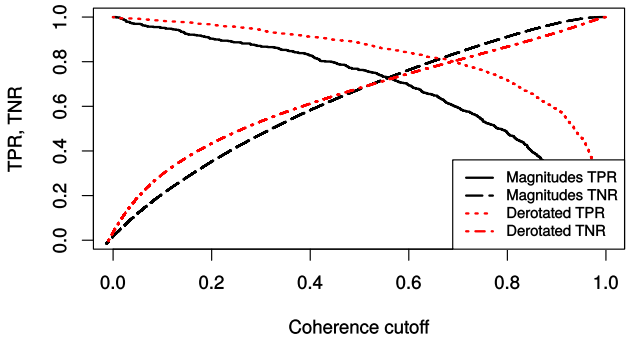


Fig. 7. True positive and true negative rate over coherence threshold for 3 Hz. Match rates as well as coherence values themselves for 3 Hz are higher with derotation than with time series magnitudes.

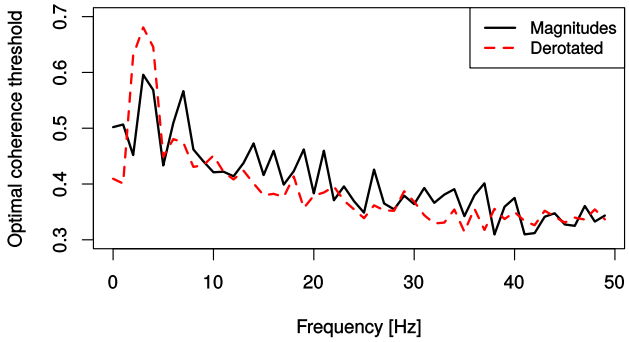


Fig. 8. Optimal coherence thresholds per frequency.

5.7.2 Using optimal coherence thresholds

Next, we determine if a coherence vector $C_{xy}(f)$ obtained by shaking device x and y corresponds to the P or N class using the optimal coherence thresholds $C_o(f)$. We have explored two ways of doing so: using a) a majority vote and b) the distances from the optimal thresholds. With the majority vote, we utilize the amount of frequencies being above their corresponding optimal threshold. If that amount is above another predefined threshold, the sample is classified as positive (shaken conjointly). If it is below the threshold, it is classified as negative (not shaken conjointly). In preliminary tests, the majority vote turned out to perform slightly worse than averaging the coherence vector.

We therefore incorporate the distance $d_{xy}(f)$ from optimal coherence thresholds $C_o(f)$ to coherence vector $C_{xy}(f)$ as well (Eg. 7). Its fundamental idea is that certainty rises with the distance to the corresponding optimal threshold. The larger the distance of a coherence value to its corresponding threshold, the higher the certainty that it belongs to the P respectively N class. To obtain a scalar similarity value from $d_{xy}(f)$, a collapsing function is required again. As with the previous collapsing functions evaluation (Sec. 5.6), once more euclidean distance slightly outperformed averaging the vector as well as all other collapsing functions (Eq. 8). Note that standard euclidean distance is not applicable anymore as it eliminates the sign for individual distances. We therefore use a signed euclidean distance $d_{es}(v)$ which preserves the sign of its components (Eq. 9 and 10).

$$d_{xy}(f) = C_{xy}(f) - C_o(f) \tag{7}$$

$$d_{xy} = d_{es}(d_{xy}(f)) \tag{8}$$

$$d_{es}(v) = a(v)^0 \cdot \sqrt{\text{abs}(a(v))} \tag{9}$$

$$a(v) = \sum_i v_i \cdot \text{abs}(v_i) \tag{10}$$

When incorporating the distance to the optimal coherence thresholds and signed euclidean distance collapsing with coherence obtained from time series magnitudes, we obtain an AUC of 0.9056 and an EER of 0.1724. When instead using it with initially derotated timeseries and using a coherence frequency bandpass filter, we obtain an AUC of 0.9495 and an EER of 0.1257.

5.8 Coherence Frequency Weighting

5.8.1 Weighting frequencies individually

The coherence density over frequency (Fig. 4) shows that coherence is denser for lower frequencies, with P and N class being visually more separated than with higher frequencies. Consequently, lower frequencies will yield better class separation performance than higher frequencies. Performances measures from classifiers using only a single coherence frequency to separate P and N class support this intuition with lower frequencies in general yielding better results than higher frequencies (Fig. 9).

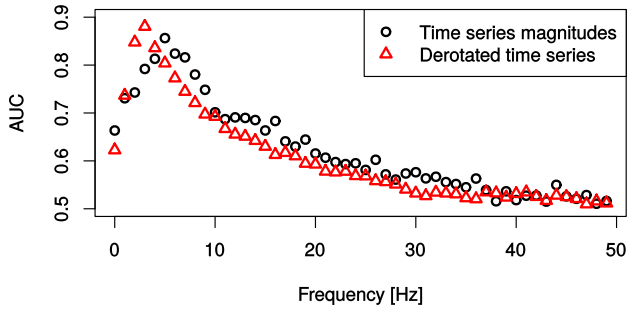


Fig. 9. AUC of classifiers using a single frequency with and without derotation.

Note that without derotation (using time series magnitudes), the best performing frequency is 5 Hz. With derotating time series, the best performing frequency is shifted to 3 Hz. This is a side effect of derotation, which uses the largest eigenvector of the quaternion rotation matrix (obtained from the time series correlation matrix). Obviously, derotation favors 3 Hz alignment which indicates that optimal derotation can be achieved when aligning time series around that frequency. The dominant frequency seems to be 3 Hz when derotation shaking acceleration time series. Although the majority of AUC values is lower with using time series derotation, overall performance is better with using derotation (Sec. 5.4). This indicates that the performance gain through best aligning lower frequencies (increasing their corresponding performance) is higher than the performance loss through concurrently decreasing higher frequency performance. This underlines the importance of lower frequencies for separating P and N class (note the strong performance gain for 2 and 3 Hz). Moreover, this is in line with our previous finding of the best performing bandpass covering a narrower range of 1-18 Hz respectively 1-16 Hz, discarding higher frequencies.

From these insights it can be concluded that individually weighting coherence frequencies (e.g. based on their class separation power) when obtaining a scalar similarity value should improve results. The coherence frequency bandpass – as a less powerful, special case of such weighting – already showed to improve performance. With the bandpass, blocked frequencies are assigned a weight of 0, whereas passing frequencies are assigned a weight of 1.

5.8.2 Obtaining coherence frequency weights

With our setup we weight 51 coherence frequencies in the range $[0, 1]$. Assuming a coarse granularity of 0.1 (11 steps of size 0.1 in the range $[0, 1]$) results in a grid search space size of 11^{51} – which is too large for a simple parameter grid search. We instead utilize an evolution strategy (ES) [58] to find an heuristic estimate of the optimal coherence frequency weights. We use a $(1 + \lambda)$-ES with $\lambda = 10$ mutants, randomly initialized starting weights, an initial maximum mutation rate of 1 per generation and a maximum mutation rate reduction of 0.005 per generation. With each generation, all parameters are mutated, and we run 919 generations in total (corresponds to a final maximum-mutation of 0.01). To obtain reliable results we repeat the ES 100 times (for both using time series magnitudes as well as initially derotating time series) and use the best obtained weights. The heuristic estimate of optimal coherence frequency weights shows that there is a decline of weights with increasing frequency (Fig. 10) – however, the decline is throughout unsteady.

It is important to understand that these estimated weights represent a highly problem-adapted optimum of weights (overfitted to our problem) and therefore cannot be derived from discrimination power metrics

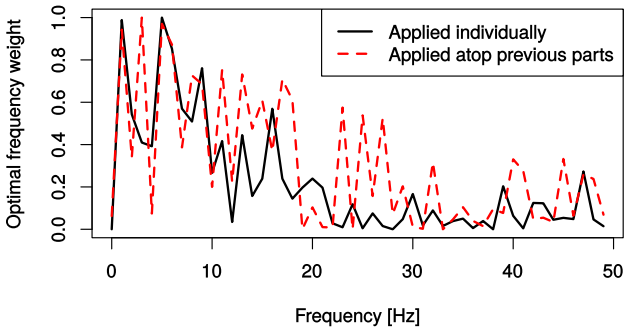


Fig. 10. Heuristic estimation of problem specific, optimal coherence frequency weights.

like AUC or directly reused for problems without re-estimating the weights. Consequently, these weights just serve as a prospect of possible performance gain using frequency weighting and would have to be re-estimated if applied to other problems. Using the heuristic estimate of optimal coherence frequency weights on top of using time series magnitudes we are able to increase AUC to 0.9420 and decrease the EER to 0.1329. When instead applying it with initially derotating timeseries, using the distance to optimal coherence thresholds and euclidean distance as coherence collapsing function while replacing the coherence frequency bandpass filter, we are able to increase the AUC to 0.9551 and decrease the EER to 0.1258. These gains do not seem to outweigh the added complexity and risk of overfitting.

5.9 Discussion of Performance Gain

Note that the order of combining constituent parts influences the associated difficulty of achieving a performance gain (Fig. 11, Tab. 1). For constituent parts applied earlier more room remains to increase performance.

The highest performance gain is achieved by including coherence frequency weighting or its special case, the coherence frequency bandpass. This emphasizes the importance of carefully selecting coherence frequencies for human body motion analysis tasks. With frequency weighting, implementation complexity is worth mentioning: we use heuristically obtained estimates of optimal weights and these weights have to be re-estimated when applied to different problems. In contrast, the coherence frequency bandpass provides an easier to implement alternative to frequency weighting. It achieves optimal performance by including acceleration frequencies of up to about 20 Hz. This supports findings from previous research which suggest – against common assumptions – that human body movement includes useful information up to or even beyond a frequency of 20 Hz.

The second highest performance gain is achieved using optimally derotated 3D acceleration time series in consecutive analysis instead of using acceleration time series magnitudes. Computing time series magnitudes strips out rotation information contained in original 3D time series. In contrast, with optimally derotated time series, parts of rotation information remain (namely changes in rotation over time), which is supported by improved performances. Consequently, derotation of 3D acceleration time series should be considered before doing consecutive analysis.

Including distance to optimal threshold and modified coherence vector collapsing functions achieve minor performance gains. With the first, the coherence threshold for separating classes is chosen optimally for each frequency. With the latter, euclidean distance turned out to slightly outperform the frequently used averaging of coherence on overall performance. When applied individually, both achieve small performance gains. When applied in combination with derotated time series and a coherence frequency bandpass, their performance gain is negligible, hence – depending on the problem – they can be excluded from implementation in favor of frequency bandpass and optimal derotation of time series.

6 Implementation and User Study

Based on findings from our evaluation we implemented our approach on Android for mobile phones and wrist watches.[6] In the implementation the link is established as soon one devices starts recording an active segment and acceleration recordings are aggregated on the mobile phone afterwards. In case one device did not detect an active segment, unlocking is aborted and the user is notified. Further, the user is notified about all successful or failed ShakeUnlock attempts on both mobile phone and smart watch. This ensures the user is informed in case case of the mobile phone being under control of an attacker. Based on our finding, for active segment similarity analysis we chose to include optimal derotation of 3D acceleration time series, applying a coherence bandpass filter and collapsing the remaining coherence vector to a single scalar value using euclidean distance.

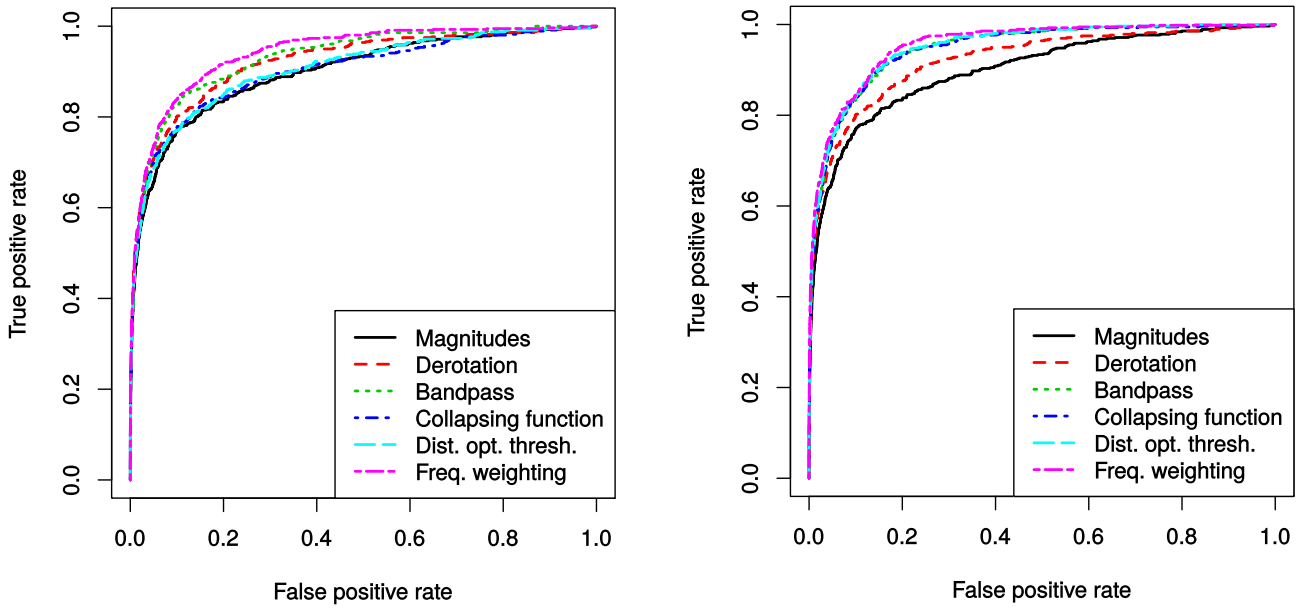
Using our implementation we conduct a user study to quantify the impact of attacks on our approach, as summarized in section 3, and to measure upper boundaries (which are expected to break unlock security). The study featured a total of 15 pairs of participants pairwise attacking each other 20 times per attack scenario (which results in a total of 600 attacks per scenario). For cooperative attacks, participants were told to utilize any cooperative strategy or tool at hand except for touching the other device or participant. This lead to participants using verbal communication, music, or even a metronome as help for synchronization.

From study results, we found observatory attacks to be successful on average with a rate of 0.20, cooperative attacks with 0.35, and handshaking attacks with 0.90 (all with a threshold of 0.522, which corresponds to a TPR of 0.82 computed from ShakeUnlock database data

6. After review, the code will be publicly available at http://www.usmile.at/downloads.

TABLE 1
Contribution of constituent parts of our approach to overall performance, applied individually and atop previous parts.

Constituent part	Implementation complexity	Individual AUC	Individual EER	Atop previous parts AUC	Atop previous parts EER
Time series magnitudes	low	0.8990	0.1777	—	—
Derotated timeseries	medium	0.9214	0.1562	—	—
Coherence frequency bandpass	low	0.9315	0.1418	0.9469	0.1293
Coherence vector collapsing function	low	0.9023	0.1670	0.9464	0.1293
Distance to optimal coherence threshold	medium	0.9056	0.1724	0.9495	0.1257
Coherence frequency weighting	high	0.9420	0.1329	0.9551	0.1258



(a) Individual contribution per constituent part (b) Combined contribution of constituent parts

Fig. 11. ROC curve stating the contribution of constituent parts of our approach to overall performance.

only[7]). On the one hand – in contrast to [26] – in our setup, forging the second shaking pattern seems feasible with a rate of about 0.2. We infer that this is caused by the wrist as joint in between devices (instead of devices being pressed against each other) – which causes sensed acceleration to be different on devices when shaking them, consequently lowering the required similarity of acceleration records for unlocks as well as attackers. On the other hand, although this is a realistic attack, it is connected to a certain effort, as attackers are required to a) acquire an identical looking device and b) replace the user's phone with the proxy device. From study results, we further consider both cooperative and handshake attacks to break our approach in terms of unlock security. We argue that this is acceptable, as we also consider them unrealistic/easily detectable in real life unlock situations.

7. The EER composed from one-vs-all comparisons using positive samples of the ShakeUnlock database and negative samples only from the observably attack study is slightly lower with 0.19; using cooperative attack data instead it is 0.23 and with handshake attack data it is 0.45.

7 Conclusion

In this article we proposed to conjointly shake an unlocked, mobile token device and another mobile device still locked to transfer the authentication state from the token device to the other device and unlock it. A common use case features a wrist watch as token device strapped to the wrist and a mobile phone held in the same hand. Both are pre-paired and can communicate over a secure channel. While devices are shaken, we record 3D acceleration time series on both devices. These are analyzed for similarity to decide if both devices have actually been shaken conjointly. Therefore, shaking devices serves as secure trigger mechanism to transfer the authentication state. Our approach has the advantage of requiring only acceleration sensors, which are commonly integrated in mobile devices. Further, acceleration recording can be done power efficiently using dedicated hardware – similar to background step counting, which is already available in several off-the-shelf mobile devices from various OEMs.

The evaluation of our approach includes the contribution of constituent parts to the system performance. We found coherence frequency filtering and optimal

derotation of 3D acceleration time series to be most effective in improving the distinguishability of legitimate unlocks and potential attacks. We further implemented our approach on off-the-shelf Android devices. Using live data from our implementation, 15 pairs of participants tried to attack each other and trigger unlocks in different attack scenarios. Results indicate that observational attacks have a success rate in the range of 0.2. This is higher than anticipated, but seems acceptable, as for this, attackers at first need to a) replace users' devices in secret with mock devices and b) need to shake the obtained device at the same time as users (with users being informed about unlock attempts), creating significant barriers for a successful attack. We conclude that ShakeUnlock is a mobile device unlock approach complementary to existing unlocking approaches (e.g. using PIN, password, unlock pattern, or fingerprint) – similar to these it solves not all, but parts of the problem of unlocking mobile devices during everyday usage.

Future work should investigate long term acceptance of ShakeUnlock with an extensive usability study. Such a study needs to consider e.g. muscle memory effects, its learning rate, and effect on usability over time. A short study would likely only give limited insights and possibly be biased towards negative feedback, as it might not be able to account for learning a muscle memory or related effects. Hence, this study should be performed longitudinally, spanning several weeks or months.

ACKNOWLEDGMENTS

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

REFERENCES

[1] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, "ShakeUnlock: Securely unlock mobile devices by shaking them together," in *Proc. MoMM 2014*. New York, NY, USA: ACM Press, December 2014, pp. 165–174, *Awarded MoMM 2014 best paper*.

[2] M. Swan, "Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0," *Journal of Sensor and Actuator Networks*, vol. 1, no. 3, pp. 217–253, Nov. 2012.

[3] G. Thomson, "BYOD: enabling the chaos," *Network Security*, vol. 2012, no. 2, pp. 5 – 8, 2012.

[4] B. Morrow, "BYOD security challenges: control and protect your most sensitive data," *Network Security*, vol. 2012, no. 12, pp. 5–8, 2012.

[5] S. N. Abdulkader, A. Atia, and M.-S. M. Mostafa, "Authentication systems: Principles and threats," *Computer and Information Science*, vol. 8, no. 3, 2015.

[6] L. Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proc. IEEE*, vol. 91, no. 12, pp. 2021–2040, Dec. 2003.

[7] D. Van Bruggen, S. Liu, M. Kajzer, A. Striegel, C. R. Crowell, and J. D'Arcy, "Modifying smartphone user locking behavior," in *Proc. SOUPS 2013*. New York, NY, USA: ACM, 2013, pp. 10:1–10:14.

[8] A. Adams and M. A. Sasse, "Users are not the enemy," *Commun. ACM*, vol. 42, no. 12, pp. 40–46, Dec. 1999.

[9] L. F. Cranor and S. Garfinkel, *Security and Usability*. O'Reilly Media, May 2008.

[10] P. Bao, J. Pierce, S. Whittaker, and S. Zhai, "Smart phone use by non-mobile business users," in *Proc. MobileHCI 2011*. New York, NY, USA: ACM, 2011, pp. 445–454.

[11] Intel computer use research: Usage tracking data. People and Practices Research, Intel Corporation. Aggregated results from multiple studies from 2007–2009.

[12] T. Beauvisage, "Computer usage in daily life," in *Proc. SIGCHI 2009*. New York, NY, USA: ACM, 2009, pp. 575–584.

[13] E. Hayashi and J. Hong, "A diary study of password usage in daily life," in *Proc. SIGCHI 2011*. New York, NY, USA: ACM, 2011, pp. 2627–2630.

[14] 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*. New York, NY, USA: ACM Press, December 2014, pp. 105–114.

[15] F. Schaub, R. Deyhle, and M. Weber, "Password entry usability and shoulder surfing susceptibility on different smartphone platforms," in *Proc. MUM 2012*. New York, NY, USA: ACM, 2012, pp. 13:1–13:10.

[16] F. Tari, A. A. Ozok, and S. H. Holden, "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords," in *Proc. SOUPS 2006*. New York, NY, USA: ACM, 2006, pp. 56–66.

[17] E. von Zezschwitz, A. Koslow, A. De Luca, and H. Hussmann, "Making graphic-based authentication secure against smudge attacks," in *Proc. 2013 international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2013, pp. 277–286.

[18] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proc. 4th USENIX conference on offensive technologies*, Berkeley, CA, USA, 2010, pp. 1–7.

[19] S. Schneegass, F. Steimle, A. Bulling, F. Alt, and A. Schmidt, "Smudgesafe: Geometric image transformations for smudge-resistant user authentication," in *Proc. UbiComp 2014*. New York, NY, USA: ACM, 2014, pp. 775–786.

[20] P. Tresadern, T. Cootes, N. Poh, P. Matejka, A. Hadid, C. Levy, C. McCool, and S. Marcel, "Mobile biometrics: Combined face and voice verification for a mobile platform," *IEEE Pervasive Computing*, vol. 12, no. 1, pp. 79–87, 2013.

[21] Y. Chen and M. Sinclair, "Tangible security for mobile devices," in *Proc. MOBIQUITOUS 2016*. Brussels, Belgium: ICST, 2008, pp. 19:1–19:4.

[22] M. Koschuch, M. Hudler, H. Eigner, and Z. Saffer, "Token-based authentication for smartphones," in *Proc. DCNET 2013*, Jul. 2013, pp. 1–6.

[23] S. Flgge, H. Scharf, S. Fahl, and M. Smith, "Poster: Preliminary investigation of an NFC-unlock mechanism for android," in *Proc. SOUPS 2013*. Newcastle, United Kingdom: ACM, 2013.

[24] H. Bojinov and D. Boneh, "Mobile token-based authentication on a budget," in *Proc. 12th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '11. NY, USA: ACM, 2011, pp. 14–19.

[25] 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 *Proc. MobileHCI 2013*. NY, USA: ACM, 2013, pp. 261–270.

[26] R. Mayrhofer and H. Gellersen, "Shake well before use: Intuitive and secure pairing of mobile devices," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 792–806, 2009.

[27] S. Antifakos, B. Schiele, and L. E. Holmquist, "Grouping mechanisms for smart objects based on implicit interaction and and stavros antifakos and bernt schiele," in *Proc. UbiComp 2003 Interactive Posters*, 2003, pp. 207–208.

[28] K. Hinckley, "Synchronous gestures for multiple persons and computers," in *Proc. UIST 2013*. New York, NY, USA: ACM, 2003, pp. 149–158.

[29] B. Groza and R. Mayrhofer, "SAPHE: Simple accelerometer based wireless pairing with heuristic trees," in *Proc. MoMM 2012*. New York, NY, USA: ACM, 2012, pp. 161–168.

[30] D. Bichler, G. Stromberg, M. Huemer, and M. Löw, "Key generation based on acceleration data of shaking processes," in *Proc. UbiComp 2007*. Berlin: Springer, 2007, pp. 304–317.

[31] D. Kirovski, M. Sinclair, and D. Wilson, "The Martini Synch," Microsoft Research, Tech. Rep. MSR-TR-2007-123, September 2007.

[32] I. Ahmed, Y. Ye, S. Bhattacharya, N. Asokan, G. Jacucci, P. Nurmi, and S. Tarkoma, "Checksum gestures: Continuous gestures as an out-of-band channel for secure pairing," in *Proc. Ubicomp 2015*. New York, NY, USA: ACM, 2015, pp. 391–401.
[33] 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 *Proc. UbiComp 2001*. London, UK, UK: Springer-Verlag, 2001, pp. 116–122.
[34] 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 *Pervasive*, 2004, pp. 33–50.
[35] C. Castelluccia and P. Mutaf, "Shake them up!: A movement-based pairing protocol for CPU-constrained devices," in *Proc. MobiSys 2005*. New York, NY, USA: ACM, 2005, pp. 51–64.
[36] K. Fujinami and S. Pirttikangas, "A study on a correlation coefficient to associate an object with its user," in *3rd IET International Conference on Intelligent Environments (IE '07)*, 2007, pp. 288–295.
[37] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds. Springer Berlin Heidelberg, 2004, vol. 3001, pp. 1–17.
[38] W. Dargie, "Analysis of time and frequency domain features of accelerometer measurements," in *Proc. ICCCN 2009*, 2009, pp. 1–6.
[39] W. Dargie and M. Denko, "Analysis of error-agnostic time- and frequency-domain features extracted from measurements of 3D accelerometer sensors," *Systems Journal, IEEE*, vol. 4, no. 1, pp. 26–33, 2010.
[40] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, 2010.
[41] M. Engin, S. Demirağ, E. Z. Engin, G. Çelebi, F. Ersan, E. Asena, and Z. Çolakoğlu, "The classification of human tremor signals using artificial neural network," *Expert Systems with Applications*, vol. 33, no. 3, pp. 754–761, 2007.
[42] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proc. Soc-EUSAI 2005*. ACM, 2005, pp. 159–163.
[43] P. D. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.
[44] H. Ben-Pazi, H. Bergman, J. A. Goldberg, N. Giladi, D. Hansel, A. Reches, and E. S. Simon, "Synchrony of rest tremor in multiple limbs in parkinson's disease: evidence for multiple oscillators," *Journal of Neural Transmission*, vol. 108, no. 3, pp. 287–296, 2001.
[45] R. Marin-Perianu, M. Marin-Perianu, P. Havinga, and H. Scholten, "Movement-based group awareness with wireless sensor networks," in *Proc. Pervasive 2007*. Springer, 2007, pp. 298–315.
[46] C. T. Cornelius and D. F. Kotz, "Recognizing whether sensors are on the same body," *Pervasive and Mobile Computing*, vol. 8, no. 6, pp. 822–836, 2012, special Issue on Pervasive Healthcare.
[47] M. Hacker, M. Crovella, and L. Reyzin, "Secure pairing of mobile devices," Master's thesis, Boston University, May 2012.
[48] 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 *Proc. ACSAC 2011*. NY, USA: ACM, 2011, pp. 333–342.
[49] J. Rekimoto, "Synctap: Synchronous user operation for spontaneous network connection," *Personal and Ubiquitous Computing*, vol. 8, no. 2, pp. 126–134, May 2004.
[50] C. Soriente, G. Tsudik, and E. Uzun, "BEDA: Button-enabled device pairing." *IACR Cryptology ePrint Archive*, vol. 2007, p. 246, 2007.
[51] R. Mayrhofer, "The candidate key protocol for generating secret shared keys from similar sensor data streams," in *Proc. ESAS 2007*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1–15.
[52] R. Mayrhofer, J. Fuss, and I. Ion, "UACAP: A unified auxiliary channel authentication protocol," *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 710–721, Apr. 2013.
[53] M. K. Chong, R. Mayrhofer, and H. Gellersen, "A survey on usability of spontaneous device association," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 8:1–8:40, May 2013.
[54] C. Ling and V. Sheng, "Class imbalance problem," in *Encyclopedia of Machine Learning*, C. Sammut and G. Webb, Eds. Springer US, 2010, pp. 171–171.
[55] R. Mayrhofer, H. Hlavacs, and R. D. Findling, "Optimal derotation of shared acceleration time series by determining relative spatial alignment," *Pervasive Computing and Communications*, 2015.
[56] D. Winter, *Biomechanics and Motor Control of Human Movement*. Wiley, 2004.
[57] C. Bouten, K. Koekkoek, M. Verduin, R. Kodde, and J. Janssen, "A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 3, pp. 136–147, 1997.
[58] H.-G. Beyer, *The Theory of Evolution Strategies*. New York, NY, USA: Springer-Verlag New York, Inc., 2001.

Rainhard Dieter Findling received his BSc and MSc degree in Mobile Computing from the University of Applied Sciences Upper Austria in 2011 and 2013 with distinction. Currently, he is researcher with u'smile, the Josef Ressel Centre for User-Friendly Secure Mobile Environments, at the University of Applied Sciences Upper Austria, and working towards his PhD with the Institute of Networks and Security, at the Johannes Kepler University Linz, Austria. His research interests include machine intelligence and security in the context of mobile environments and ubiquitous computing.

Muhammad Muaaz received his B.E degree in Computer and Information Systems from N.E.D University of Engineering and Technology, Pakistan in 2007. In 2012 he received his MSc degree in Information and Communication Systems Security from KTH Royal Institute of Technology, Sweden. Currently he is working towards his PhD with the Institute of Network Security, at Johannes Kepler University Linz, Austria. His research interests include, information security, biometrics and machine learning.

Daniel Hintze received a BSc in Business informatics and a MSc in IT-Management and Information Systems from FHDW University of Applied Sciences, Paderborn, Germany. Since 2013 he is enrolled in the PhD program at the Institute of Networks and Security, Johannes Kepler University Linz, Austria. His main research interests include authentication on mobile devices, mobile device usage and UI design. His supervisors are René Mayrhofer from JKU Linz, as well as Eckhard Koch from FHDW Paderborn.

René Mayrhofer heads the Institute of Networks and Security (INS) at Johannes Kepler University Linz (JKU), Austria, and the Josef Ressel Center on User-friendly Secure Mobile Environments (u'smile). Previously, he held a full professorship for Mobile Computing at Upper Austria University of Applied Sciences, Campus Hagenberg, a guest professorship for Mobile Computing at University of Vienna, and a Marie Curie Fellowship at Lancaster University, UK. His research interests include computer security, mobile devices, network communication, and machine learning, which he brings together in his research on securing spontaneous, mobile interaction. René has contributed to over 60 peer-reviewed publications and is a reviewer for numerous journals and conferences. He received Dipl.-Ing. (MSc) and Dr. techn. (PhD) degrees from Johannes Kepler University Linz, Austria and his Venia Docendi for Applied Computer Science from University of Vienna, Austria.