Sensor-based User Authentication

He Wang[1], Dimitrios Lymberopoulos[2], and Jie Liu[2]

[1] University of Illinois at Urbana-Champaign, Champaign, IL, USA
hewang5@illinois.edu
[2] Microsoft Research, Redmond, WA, USA
{dlymper,liuj}@microsoft.com

Abstract. We study the feasibility of leveraging the sensors embedded on mobile devices to enable a user authentication mechanism that is easy for users to perform, but hard for attackers to bypass. The proposed approach lies on the fact that users perform gestures in a unique way that depends on how they hold the phone, and on their hand's geometry, size, and flexibility. Based on this observation, we introduce two new unlock gestures that have been designed to enable the phone's embedded sensors to properly capture the geometry and biokinetics of the user's hand during the gesture. The touch sensor extracts the geometry and timing of the user hand, while the accelerometer and gyro sensors record the displacement and rotation of the mobile device during the gesture. When combined, a sensor fingerprint for the user is generated. In this approach, potential attackers need to simultaneously reproduce the touch, accelerometer, and gyro sensor signatures to falsely authenticate. Using 5000 gestures recorded over two user studies involving a total of 70 subjects, our results indicate that sensor fingerprints can accurately differentiate users while achieving less than 2.5% false accept and false reject rates. Attackers that directly observe the true user authenticating on a device, can successfully bypass authentication only 3% of the time.

1 Introduction

As sensitive information, in the form of messages, photos, bank accounts, and more, finds its place on mobile devices, the need to properly secure them becomes a necessity. Traditional user authentication mechanisms, such as lengthy passwords that include combinations of letters, numbers and symbols, are not suited for mobile devices due to the small on-screen keyboards. Given that users need to authenticate on their mobile devices tens or even hundreds of times throughout the day, the traditional password authentication technique becomes a real bottleneck.

To simplify the authentication process, users tend to leave their mobile devices completely unprotected, or they leverage simple authentication techniques such as 4-digit pins, picture passwords (Windows 8), or unlock gestures (Android). Even though these techniques allow easy and intuitive user authentication, they compromise the security of the device, as they are susceptible to simple shoulder-surfing attacks [14]. Pins, picture passwords, and unlock gestures can be easily retrieved by simply observing a user authenticating on his/her device once.

Ideally, the user authentication process should be easy and fast for users to perform, and at the same time difficult for an attacker to accurately reproduce even by directly observing the user authenticating on the device.

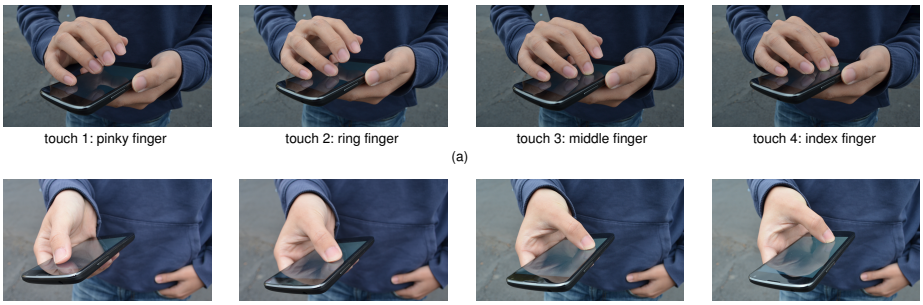


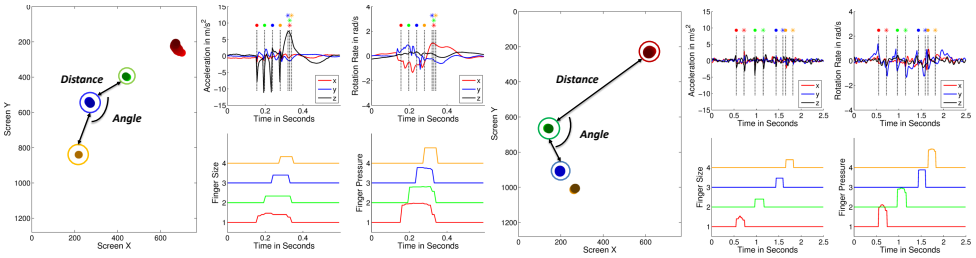
Fig. 1. Proposed unlock gestures for capturing the biokinetics of the user's hand. Users can perform the gesture anywhere on the screen, and at the speed they feel comfortable with. (a) 2-hand gesture: the user sequentially taps his four fingers on the touch screen starting from the pinky finger, and ending with the index finger. (b) 1-hand gesture: the user uses his/her thumb to touch each of the rest four fingertips through the touch screen starting with the index finger, and ending with the pinky finger. The 1-hand gesture was designed to avoid the need to use both hands at the expense of more noisy sensor data. **A video demonstrating both gestures can be seen in [1, 2]**

Towards this goal, Android devices recently brought face recognition to the masses by enabling user authentication through the front-facing camera. Even though intuitive and fast, this type of authentication suffers from typical computer vision limitations. The face recognition performance degrades under poor or different lighting conditions than the ones used during training. Given that mobile devices constantly follow their users, such fluctuations on the environmental conditions are common.

More recently, iPhone enabled users to easily and securely unlock their devices by embedding a fingerprint sensor in the home button. Even though this approach addresses both the usability and security requirements of the authentication process, it is fundamentally limited to devices with large physical buttons on the front, such as the home button on iPhone, where such a sensor can be fitted. However, as phone manufacturers push for devices with large edge-to-edge displays, physical buttons are quickly replaced by capacitive buttons that can be easily embedded into the touch screen, eliminating the real-estate required by a fingerprint sensor. Embedding fingerprint sensors into touch screens behind gorilla glass is challenging, and has not been demonstrated.

In this paper, we study the feasibility of enabling user authentication based solely on generic sensor data. The main idea behind our approach is that different users perform the same gesture differently depending on the way they hold the phone, and on their hand's geometry, size, and flexibility. These subtle differences can be picked up by the device's embedded sensors (i.e., touch, accelerometer, and gyro), enabling user authentication based on sensor fingerprints. With this in mind, we introduce two new unlock gestures, shown in Figure 1, that have been designed to maximize the unique user information we can extract through the device's embedded sensors.

While the user performs the gesture, we leverage the touch screen sensor to extract rich information about the geometry and the size of the user's hand (size, pressure, timing and distance of finger taps). We also leverage the embedded accelerometer and gyro sensors to record the phone's displacement and rotation during the gesture. To avoid the impact of gravity, we use linear acceleration provided by Android API.



(a) 2-hand gesture 1-hand gesture

Fig. 2. Raw data from the touch, accelerometer, and gyro sensors. Dots and asterisks on the sensor plots correspond to the moments of pressing and releasing.Touch screen data enables the extraction of: distances between every pair of fingertips, angles defined by any combination of 3 fingertips, and the exact timing of each fingertip. Acceleration and gyro data capture the displacement of the device in user's hand during the gesture.

When combined, the information from touch, accelerometer, and gyro sensors provides a detailed view into *how* the individual user performs the gesture, and, as we show in this work, it can be used as a sensor fingerprint to authenticate the user. Attackers willing to bypass this authentication mechanism, face a much harder task as they have to *simultaneously* reproduce the timing, placement, size, and pressure of each finger tap, as well as the accelerometer and gyro sensor signatures. Even though faking each of this information individually might be easy, *simultaneously* reproducing all this information is quite challenging even when the attacker has the opportunity to closely observe the actual user performing the unlock gesture.

In summary, this work makes three contributions. First, we propose two new unlock gestures that were designed to enable a device's sensors to extract as much information as possible about the user's hand biokinetics. Second, we collect 3000 sensor fingerprints across 50 users, and show that different users indeed perform the same gestures differently, and in a way that embedded sensor's can accurately capture and differentiate. In particular, we demonstrate false accept and false reject rates lower than 2.5%, when only a small number of training gestures per user is used. Third, we simulate realistic attack scenarios, by showing videos of real users authenticating on their devices to attackers, and then asking the attackers to reproduce the unlock gestures. Experimental results from 2000 attacks from 20 different attackers show that the proposed approach can achieve success attack rates that are lower than 3%.

2 Motivation and Challenges

To better illustrate how the biokinetics of the user's hand are captured by the proposed gestures shown in Figure 1, Figure 2 shows the raw data recorded by the touch, accelerometer, and gyro sensors when a user performs each of the gestures.

In both cases, four finger taps are recorded through the touch screen in the form of pixel coordinates. Since each of the recorded touch points directly (2-hand gesture) or indirectly (1-hand gesture) corresponds to a fingertip, the touch screen captures the geometry of the user's hand. In particular, the distance between every pair of fingertips, and the angles defined by any combination of 3 fingertips, can be used to characterize the size and geometry of the user's hand. At the same time, the timestamps of the finger taps highlight the speed at which the user is able to flex his fingers to perform the

required gesture. The duration of each finger tap, as well as the timing between pairs of finger taps varies across users depending on the size and flexibility of the user's hand.

The touch screen on most smartphones is also able to record the pressure and size of each finger tap. Both of these values depend on the size and weight of the user's hand, on how much pressure the user applies on the display, as well as on the angle at which the user holds the device while performing the gesture.

The accelerometer and gyro sensors complement the touch sensor by *indirectly* capturing additional information about user's hand biokinetics. Every time a user performs one of the unlock gestures, the device is slightly displaced and rotated. As shown in Figure 2, the displacement and rotation of the device is clearly reflected in the accelerometer and gyro sensor data.

When combined, the information from touch, accelerometer, and gyro sensors forms a sensor fingerprint that captures the geometry and biokinetics of the user's hand.

2.1 Challenges and Contributions

The use of sensor data for user authentication poses several challenges. First, the recorded sensor data can vary across different gesture instances depending on how the actual user performs the gesture or holds the device. Even worse, this variability can be user-specific. For instance, some users can be very accurate in reproducing the exact timing or distance between the finger taps, but fail to accurately reproduce other parts of the sensor data, such as the pressure or angle signatures, and vice versa. An authentication mechanism should be automatically tailored to the capabilities of each user.

To enable direct comparison of the sensor fingerprints across users and gesture instances, we introduce personalized dissimilarity metrics for quantifying the difference of any pair of sensor fingerprints in both the touch and sensor domain. The personalized dissimilarity metrics are designed to emphasize more on those features of the sensor data that exhibit the least variability across gesture instances, and thus are the most descriptive of user's gesture input behavior.

Second, mobile devices support high sensor sampling rates (up to $200Hz$). At this sampling rate large amounts of data is generated creating a processing bottleneck that can slow down the device unlock process, and render the proposed technique unusable. To address this problem, we exploit the tradeoff between sensor downsampling and overall accuracy, and show that by properly downsampling sensor data, we can achieve device unlock times of $200ms$ without sacrificing recognition accuracy.

3 Architecture

Figure 3 provides an overview of the sensor-based authentication system. During the user enrollment phase, the true user repeatedly performs the unlock gesture on the touch-enabled device. For each gesture, the touch sensor is used to record finger taps and extract information about the timing, distance, angle, pressure, and size of finger taps. At the same time, the accelerometer and gyro sensors are continuously sampled to capture the displacement and rotation of the device during the unlock gesture. The data extracted from the finger taps, along with the raw accelerometer, and gyro data becomes the actual sensor fingerprint for the user. In that way, multiple sensor fingerprints across different gesture instances are collected. This collection of fingerprints represents the identity of the user in the sensor domain.

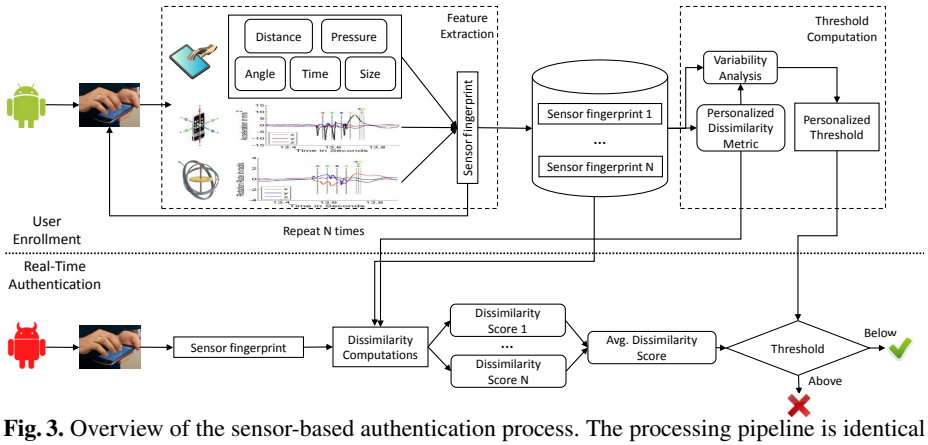


Fig. 3. Overview of the sensor-based authentication process. The processing pipeline is identical for the 2-hand and 1-hand gestures: 4 finger taps are recorded and processed in the same way.

To determine if a random sensor fingerprint belongs to the true user or not, a way to quantify the difference of two sensor fingerprints is required. We introduce a dissimilarity metric that takes into account the unique gestural behavior of the user to quantify how close two sensor fingerprints are. Given this dissimilarity metric, we analyze the variability of the recorded sensor fingerprints for a given user, and based on this variability we derive a threshold for admitting or rejecting an unknown sensor fingerprint. For those users with low variability, a stricter threshold should be enforced, while for users with high variability, a more lenient threshold should be adopted to properly balance false positives and false negatives.

At run time, when a user performs the unlock gesture, a new sensor fingerprint is recorded. The distance of this fingerprint to the true user is computed as the average dissimilarity between the recorded fingerprint and every single fingerprint recorded in the user enrollment phase. If the average dissimilarity is below the personalization threshold, the user is successfully authenticated, otherwise the device remains locked.

The next sections describe in detail the composition of sensor fingerprints, the dissimilarity metric, and the personalized threshold computation.

3.1 Sensor Fingerprints

Touch, accelerometer, and gyro sensor data are combined to form the sensor fingerprint. In the case of accelerometer and gyro sensors, the process is straightforward as the raw sensor data is directly used as part of the sensor fingerprint.

The touch sensor reports three distinct types of information for each finger tap: pixel location, pressure, and size. As shown in Figure 2, both pressure and size are continuously reported for as long as the finger touches the screen. Given that the variation of pressure and size is quite small for each finger tap, we average all the reported pressure and size values, and use them as two distinct features. Given the four finger taps, 4 pressure and 4 size values are generated (Table 1).

The majority of the touch-based features are extracted directly from the pixel locations of the 4 finger taps. First, the distances in the pixel location space are computed for every pair of finger taps. In that way, 6 feature values are computed (Table 1). At

Feature Type	Features	Num. of Features
Distance	$D_{1,2}, D_{1,3}, D_{1,4}, D_{2,3}, D_{2,4}, D_{3,4}$	6
Angle	$A_{1,2,3}, A_{1,2,4}, A_{1,3,4}, A_{2,3,4}$	4
Size	S_1, S_2, S_3, S_4	4
Pressure	P_1, P_2, P_3, P_4	4
Duration	$Dur_1, Dur_2, Dur_3, Dur_4$	4
Start Time Difference	$STD_{1,2}, STD_{1,3}, STD_{1,4}, STD_{2,3}, STD_{2,4}, STD_{3,4}$	6
End Time Difference	$ETD_{1,2}, ETD_{1,3}, ETD_{1,4}, ETD_{2,3}, ETD_{2,4}, ETD_{3,4}$	6
Distance Ratio	$\frac{D_{1,2}}{D_{2,3}}, \frac{D_{1,2}}{D_{3,4}}, \frac{D_{2,3}}{D_{3,4}}$	3
Size Ratio	$\frac{S_1}{S_2}, \frac{S_1}{S_3}, \frac{S_1}{S_4}, \frac{S_2}{S_3}, \frac{S_2}{S_4}, \frac{S_3}{S_4}$	6
Pressure Ratio	$\frac{P_1}{P_2}, \frac{P_1}{P_3}, \frac{P_1}{P_4}, \frac{P_2}{P_3}, \frac{P_2}{P_4}, \frac{P_3}{P_4}$	6
Duration Ratio	$\frac{Dur_1}{Dur_2}, \frac{Dur_1}{Dur_3}, \frac{Dur_1}{Dur_4}, \frac{Dur_2}{Dur_3}, \frac{Dur_2}{Dur_4}, \frac{Dur_3}{Dur_4}$	6
Total number of touch features		55

Table 1. Features extracted from the 4 finger taps' touch information. All features depend on the relative, and not absolute, locations of the finger taps. This enables users to perform the gesture anywhere on the screen. Indices 1, 2, 3, and 4 correspond to each finger tap as shown in Figure 1.

the same time, every combination of 3 finger taps uniquely defines an angle (Figure 2). We consider all possible angles defined by a set of three finger taps, and generate an additional 4 features (Table 1).

The touch sensor also reports a start and end timestamp for every finger tap, indicating the time the finger initially touched the screen and the time it lost contact. Using these timestamps, we compute the total duration of each finger tap, as well as as the time that elapses between the start and end time between every pair of fingerprints. In that way, the timing of each finger tap, as well as the timing across finger taps is captured. As shown in Table 1, 16 temporal features are computed.

To better capture the spatial and temporal signature of the user's hand during the gesture, we compute an additional set of meta-features that focus on capturing the dynamics across the individual features described above. In particular, we compute the ratio of every pair of distance, pressure, size, and duration features. As shown in Table 1, 21 additional features are computed.

Overall, 55 features are computed based on the touch screen data (Table 1).

3.2 Comparing Sensor Fingerprints

When comparing sensor fingerprints across gestures, different techniques are used to quantify the difference of the touch features and that of the sensor patterns.

Touch Features Let F^1 and F^2 be the set of the 55 touch features recorded across two gesture instances. We quantify the difference D_{touch} between these feature sets as the weighted average difference across all features:

$$D_{touch} = \sum_{i=1}^{55} W_i D_{F^1(i),F^2(i)} \tag{1}$$

where W_i is the weight for feature i, and $D_{F^1(i),F^2(i)}$ is the difference between the values recorded for feature i at the two gesture instances.

The distance between feature values $F^1(i)$ and $F^2(i)$ is defined by their normalized numerical difference:

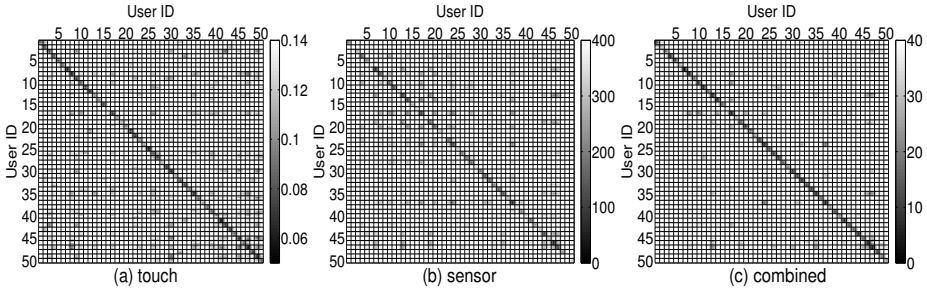


Fig. 4. Difference scores computed across 50 users. Each user performed the 2-hand gesture 30 times, for a total of 1500 gestures. Each small block corresponds to a pair of a test user and a true user, and contains the score between 30 test user gesture instances and the true user's gesture instances. Ideally, all the scores across the diagonal should be much lower (darker color) compared to the rest, indicating that gesture instances from the same user differ significantly less than gesture instances across users. True users are on the x-axis, and test users are on the y-axis.

$$D_{F^1(i),F^2(i)} = min\{\frac{|F^1(i) - F^2(i)|}{|F^1(i)|}, 2\} \tag{2}$$

When the two feature values are identical, the difference score becomes 0. In general, the higher the difference of the feature values across the gesture instances, the higher the distance for that feature will be. However, to prevent a single feature from biasing the result of Equation 1, we limit the maximum value of the distance to 2. This can be particularly useful when most feature values across two gesture instances match closely, but one of them is significantly off (i.e., outlier or faulty measurement). Even though the two gesture instances are almost identical, when an upper bound is not used, this feature can significantly bias the distance score computed in Equation 1.

The weight W_i of the feature i represents the importance of the feature for a given user. In general when users repeat gestures, they can accurately repeat feature values with varying degrees of success. The role of the weight is to emphasize on those features that a specific user can accurately reproduce across gesture instances. Given a set of enrolled gestures from a user, the weight for feature i is defined as:

$$W_i = exp\{-\frac{\sigma_{F(i)}}{\mu_{F(i)}}\} \tag{3}$$

where σ_{F_i} and μ_{F_i} is the variance and mean of the values for feature i across all the enrolled gestures from the true user. When the deviation σ_{F_i} for feature i is 0, the weight takes the maximum value of 1, indicating that this feature is accurately repeatable across gesture instances. Otherwise, a positive weight less than 1 is assigned to the feature that is determined by the ratio of σ_{F_i} and μ_{F_i}.

Figure 4(a) shows the distance scores computed by Equation 1 between every pair of 2-hand gestures recorded from 50 different subjects. Note that the scores recorded along the diagonal are much lower than the rest. This means that gestures from the same user differ less than gestures across users, indicating that touch features have enough discriminating power to differentiate users.

Sensor Patterns Each sensor fingerprint is comprised of 6 time series signals, each representing the acceleration and rotation of the device across the x, y, and z dimensions (S_{accel_x}, S_{accel_y}, S_{accel_z}, S_{gyro_x}, S_{gyro_y}, S_{gyro_z}). Even though a straightforward

approach to comparing these signals across gestures would be to simply compute the distance between them, such a method fails due to the noise in the sensor data. For instance, the total time to perform a gesture and the exact timing between finger taps inherently varies across gesture instances even for the same user. These variations can artificially increase the distance between the recorded traces.

Instead, we quantify the difference of these signals across gestures by combining two well known techniques for comparing time series data: dynamic time warping and cross-correlation. Instead of comparing each corresponding sample between the recorded signals, the two signals are slightly shifted to enable the best possible match. This allows us to take into account variations across gesture instances.

Before comparing two signals, each signal is normalized to zero mean and one energy to avoid favoring low energy over high energy signal pairs. Then, each signal is further normalized by its length to avoid favoring short signals over long signals. In particular, each time-series data $S(i)$ in the sensor fingerprint is normalized as follows:

$$S(i) = \frac{S(i) - \mu_S}{\sum_{i=1}^{L}(S(i) - \mu_S)^2 L} \tag{4}$$

where L is the length of the signal, and μ_S is the mean value of all signal samples.

Dynamic Time Warping

Let $S^1_{accel_x}$ and $S^2_{accel_x}$ be the normalized accelerometer signals over the x axis that were recorded across two different gesture instances. Since they are recorded at different times, they might have different lengths, say $L^1_{accel_x}$ and $L^2_{accel_x}$. To compare these two signals, we first compute the direct distance between every pair of samples in $S^1_{accel_x}$ and $S^2_{accel_x}$. In that way, a distance matrix D_{accel_x} with $L^1_{accel_x}$ rows and $L^2_{accel_x}$ columns is computed, where each element takes the following values:

$$D^{ij}_{accel_x} = |S^1_{accel_x}(i) - S^2_{accel_x}(j)|, 1 \leq i \leq L^1_{accel_x}, 1 \leq j \leq L^2_{accel_x} \tag{5}$$

In a similar way, distance matrices D_{accel_y} and D_{accel_z} are computed and then added together to form a single distance matrix D_{accel}.

Note that even though the range of acceleration values across different axis might differ, this addition is meaningful given the normalization of all sensor signals according to Equation 4. The exact same process is applied to the gyro data to generate a single distance matrix D_{gyro} that encodes the difference in the gyro sensor data across the x, y, and z dimensions. At the end, accelerometer and gyro distance matrices are added to form a single distance matrix $D = D_{accel} + D_{gyro}$:

Note that the number of samples in the accelerometer and gyro streams might be different depending on the sampling rates the hardware supports for these sensors. As a result, matrices D_{accel} and D_{gyro} might have different dimensions. In this case, we up-sample the lower frequency signal to ensure that both D_{accel} and D_{gyro} have the same dimensions and can be properly added.

Simply adding up the diagonal elements in matrix D, corresponds to the direct distance between the sensor fingerprints across the two gestures. In order to address the variability in the way users perform the gesture (slightly different timing etc.), we define a search space across the diagonal defined by C_{DTW}:

$$D_{ij} = \infty \quad (|i - j| \geq C_{DTW}) \tag{6}$$

where C_{DTW} is the Dynamic Time Warping constraint.By setting distances to infinity, we limit the search space along the diagonal, therefore limiting how much each signal is shifted. The distance between the two signals is now defined as the shortest warping path between the two diagonal points in matrix D:

$$D_{DTW} = \underset{p}{\operatorname{argmin}} \sum_{(i,j)\in p} D_{ij} \tag{7}$$

where p is a warping path between the two diagonal points in the matrix.

When C_{DTW} is equal to 1, the direct distance is calculated as the sum of all the diagonal elements in matrix D. As the value of C_{DTW} increases, more shifting of the two signals is allowed. In Section 4, we study the effect of the C_{DTW} value.

Cross-correlation

Similarly to Dynamic Time Warping, we combine the accelerometer and gyro data across the x, y, and z dimensions to compute a single cross-correlation value as:

$$Corr = \underset{n\in[-C_{Corr},C_{Corr}]}{\operatorname{argmax}} \sum_{k=1}^{P} \sum_{m=max\{-n+1,1\}}^{min\{L_{1k}-n,L_{2k}\}} S_{1k}(m+n)S_{2k}(m) \tag{8}$$

where C_{Corr} is a constraint on the permitted shift amount of the signals.

The scores produced by the Dynamic Time Warping and Cross-correlation techniques are combined together to quantify the overall distance between gestures in the sensor pattern domain:

$$D_{sensor} = D_{DTW} * (1 - Corr) \tag{9}$$

Figure 4(b) shows the score computed by Equation 9 between every pair of gestures recorded from 50 different subjects. Sensor pattern information appears to be stable across different gesture instances from a given user. All scores across the diagonal (gestures corresponding to the same users) have consistently low distance scores. When compared to Figure 4(a), sensor pattern information appears to have more discriminative power with respect to the touch features.

Combining Touch Features and Sensor Patterns We combine touch features and sensor patterns by multiplying the corresponding difference scores:

$$D_{combined} = D_{touch} * D_{sensor} \tag{10}$$

Figure 4(c) shows the score computed by Equation 10 between every pair of gestures recorded from 50 different subjects. When compared to Figure 4(a), and Figure 4(b), it is clear that the combination of sensor and touch data helps to better distinguish users. The distance score matrix contains low values (black lines in Figure 4(c)) primarily for gesture instances that belong to the same user.

3.3 Personalized Threshold

Equation 10 quantifies the difference between any pair of gesture instances, but it is not enough to make a decision whether or not a gesture belongs to the same user. Some users can very accurately reproduce the touch and sensor fingerprints across gesture instances, while others might exhibit higher variability. As a result, a low or high score from Equation 10 can be interpreted differently across users.

We deal with this variability by defining a personalized threshold P_{Th} for deciding when the difference between gestures is low enough to assume they belong to the same user. Given N enrolled gestures from a user, we define P_{Th} for this user as:

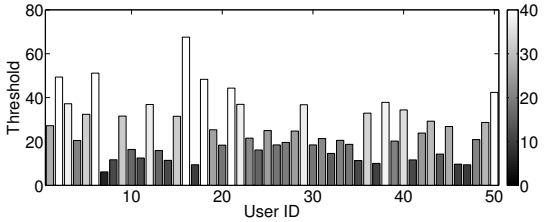


Fig. 5. The computed threshold values for 50 users (2-hand gesture). Values can differ by an order of magnitude indicating the need for a personalized threshold.

$$P_{Th} = \mu_{D^{ij}_{combined}} + 3\sigma_{D^{ij}_{combined}}, 1 \leq i, j \leq N, i \neq j \tag{11}$$

where the first term represents the median distance (Equation 10) of all pairs of gestures that belong to the user, and the second term represents the standard deviation of these distances. These two values quantify the variability in the sensor fingerprints across gesture instances for a user. The threshold value for users that accurately reproduce sensor fingerprints across gesture instances will have a low P_{Th} value, and vice versa.

Note that the personalized threshold value P_{Th} (Equation 11) is computed based on positive only data from the true user. This is highly desirable given the lack of negative data on each user's device. As we show in Section 4.1, even a small number of gestures ($\approx$10) from the true user is sufficient to generate a reliable P_{Th} value.

Figure 5 shows the P_{Th} values for 50 different users. The range of threshold values is quite large. Even though there are several users that can accurately reproduce their gestures across multiple instances and hence have low threshold values (i.e., value 5 for User 8), there are also many users for which the threshold values are an order of magnitude higher (i.e., value 70 for User 16). This indicates the need for properly computing different thresholds across users.

4 Evaluation

To evaluate the proposed approach we conducted two separate user studies. First, we asked 50 users (12 females and 38 males) to perform each of the proposed unlock gestures 30 times. All users were volunteers and were not compensated for this study. We first explained and demonstrated the proposed gestures to the users, and then allowed them to perform the gesture several times until they became comfortable with it. Each user then repeated each of the two gestures 30 times.

During data collection, several measures were taken to avoid biasing the dataset and artificially increasing the accuracy results. First, all users performed the gesture while standing up. In that way repeatability across gesture instances was not biased by the users' having their arms supported by a desk or a chair. Second, each user had to "reset" the position of his arms in between gesture instances, and pause for several seconds. In that way, data collection was able to capture the variations of how the user holds the device and taps the finger across gesture instances. In this experiment, a total of 3000 gesture instances were collected across all users and gestures. We leverage this dataset to study how different the sensor fingerprints across users are, and what parts of the sensor fingerprints have the most discriminative power.

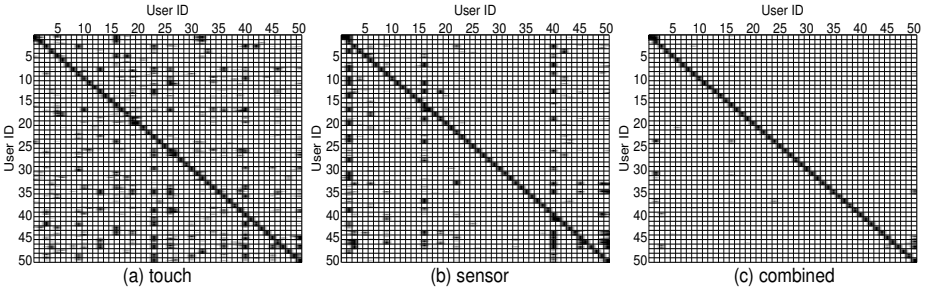


Fig. 6. User classification accuracy for the 50-subject user study when the 2-hand gesture is used. Each block corresponds to a pair of a true user and a test user, containing the classification result for 30 gesture instances from the test user. The black color indicates that the gesture instance is classified as the true user, and the white color the opposite. Ideally only the diagonal boxes should be black. The true users are on the x-axis, and the test users are on the y-axis.

The second user study focused on simulating an actual attack scenario. A separate set of 20 users (5 females, 15 males) posed as attackers aiming to falsely authenticate as the true user. For each attacker we randomly chose 5 users from the initial 50-subject user study, and gave the opportunity to the attacker to attack each of the 5 users 10 times. Overall, 2000 attack gestures were collected, 1000 for each of the proposed gestures. Right before the attackers attempted to falsely authenticate, they were shown a closeup video of the true user they were attacking. In this video, the attackers could observe, for as much time as they wanted, the true user repeatedly authenticating on the device. In addition, we allowed the attackers to spend as much time as they wanted to perfectly recreate the exact holding position of the device from the true user. Note that in practice, an attacker will rarely, if ever, be able to closely observe all this information, and then try to *immediately* attack the authentication mechanism.

In all cases, we use False Accept Rate (FAR) and False Reject Rate (FRR) to quantify the effectiveness of the proposed approach. The former represents the percentage of gesture instances that belong to users other than the true user, but are erroneously classified as belonging to the true user. The latter represents the percentage of gesture instances that belong to the true user, but are erroneously classified as belonging to a different user.

During both user studies, a single mobile device was used by all users. Specifically, a Google Nexus 4 device running Android 4.3 and a custom data collection application we built, was used to collect the touch, accelerometer and gyro data.

4.1 Differentiating Users

In this section, we leverage the data collected from the 50-subject user study to understand the discriminative power of the proposed unlock gestures in differentiating users. Using the 30 gesture instances from each user, we calculated the personalized threshold for each user. We then used this threshold to classify every single gesture instance recorded across all users as belonging or not to the user. The classification results for the 2-hand gesture are shown in Figure 6.

Ideally, only the diagonal of the classification matrices in Figure 6 should be black, indicating that only the true gesture instances are classified as belonging to the user. When touch data is only used, the classification matrix appears to be noisy. Even though

Mode	2-hand Gesture		1-hand Gesture		Pin
	FRR	FAR	FRR	FAR	FRR
Touch	2.40%	5.28%	1.8%	8.93%	10%
Sensor	2.48%	3.49%	2.61%	18.85	10%
Both	2.48%	0.41%	2.34%	2.40%	10%

Table 2. False accept and reject rates for the 2-hand and 1-hand gestures when different sensor data is used. We also report the FRR of the 4-digit pin as measured in [7].

the true user's gesture instances are always classified correctly, there are specific users that are hard to differentiate solely based on the touch fingerprints. When sensor patterns are only used for classification, the classification matrix is noticeably cleaner (only a few users are now hard to differentiate), indicating that the discriminative power of the sensor patterns is superior to that of touch sensor data. However, the combination of touch, accelerometer, and gyro data provides almost perfect classification accuracy, indicating the complementary nature of the different sensors in the classification process.

Table 2 shows the FAR and FRR values achieved by the 2-hand gesture. Overall, approximately 2.5% of the gesture instances that belong to the true user are falsely rejected. Note that even in the case of traditional 4-digit pins, FRR values as high as 10% have been reported [7]. As users try to quickly enter their 4-digit pin, they accidentally mistype it 1 in 10 times [7]. As a result, the achieved FRR rate of 2.5% is on par with the current pin-based authentication techniques. Depending on the data used in the sensor fingerprint, FAR rates are anywhere between 0.41% and 5.28%.

In the case of the 1-hand gesture, the classification accuracy degrades when touch or sensor data is only used. This is expected as the 1-hand gesture was designed to allow single-hand handling of the mobile device at the expense of quality in the data recorded. However, when touch data and sensor data is combined, the classification accuracy increases, indicating that the 1-hand gesture can be a viable unlock gesture.

Feature Sensitivity Analysis To understand the importance of individual features in the user authentication process we performed an exhaustive analysis by recomputing the classification matrices shown in Figure 6 for every possible combination of features. In addition to the 57 features available (55 touch features and 2 sensor patterns), we also experimented with two important parameters: the feature weight introduced in Equation 3, and the permitted shift amount of the raw sensor patterns as described in Equations 6, and 8. Specifically, we examined permitted shift amounts of the raw sensor patterns ranging from 0% all the way to 100% at increments of 10%. In the case of feature weights, we exploited the case where feature weights are computed using Equation 3, and when no weights are used (all the weights for all features are set to 1).

Table 3 shows the feature combinations that achieve the best results for both gestures. Consistently, across all combinations and gestures, the feature sets that achieve the best FAR and FRR results leverage feature weights. This verifies our initial intuition that individual users can accurately reproduce different parts of the sensor fingerprint across gesture instances. Feature weights are able to account for the user's variability across gesture instances, and improve the overall accuracy.

In the case of the 2-hand gesture, both accelerometer and gyro sensor patterns appear to be important for ensuring successful authentication. However, for the 1-hand gesture, the value of acceleration data seems to be less important.

Mode	2-hand Gesture			1-hand Gesture		
	Features	FRR	FAR	Features	FRR	FAR
Touch	Distance, Angle, Size, Pressure, Duration,Distance/Pressure Ratio, Feature Weights: Yes	2.40%	5.28%	Distance, Angle, Size, Pressure, Duration Feature Weights: Yes	1.8%	8.93%
Sensor	$gyro_{xyz}$, $accel_{xyz}$ Shift: 40%	2.48%	3.49%	$gyro_{xyz}$ Shift: 30%	2.61%	18.85%
Both	Distance, Angle, Size, Pressure Feature Weights: Yes $gyro_{xyz}$, $accel_{xyz}$, Shift: 40%	2.48%	0.41%	Distance, Angle, Size, Pressure Feature Weights: Yes $gyro_{xyz}$, Shift: 50%	2.34%	2.40%

Table 3. Feature combinations and parameter values achieving the best FAR and FRR values.

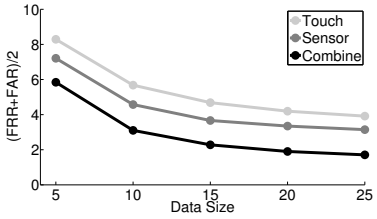


Fig. 7. Accuracy as a function of the number of available gestures per user in the case of the 2-hand gesture. Trends are similar for the 1-hand gesture.

For both gestures, though, sensor patterns need to be properly shifted to enable accurate comparison across gesture instances. According to Table 3, accelerometer and gyro patterns provide the best results when shifted anywhere between 30% and 50% depending on the gesture used.

Size of Training Data So far, all 30 gesture instances for each user were used in the authentication process. Figure 7 shows the impact of the number of gesture instances used on both the false accept, and false reject rates achieved. Intuitively, FAR and FRR are reduced as the number of gesture instances increases, but they quickly saturate, eliminating the need for 30 gestures. Anywhere between 10 and 15 gesture instances are enough to achieve FAR and FRR values that are within 0.5% of the values achieved when all 30 gesture instances are used.

4.2 Resilience to Attacks

In this section, we leverage the dataset collected by 20 subjects posing as attackers to study the resilience of the proposed authentication mechanism to an actual attack. To study the resilience of the sensor fingerprints to attacks, we compared all of attacker's sensor fingerprints to the ones of the true users and classified them as belonging to the true user or not in the same way as before. During this process, we leveraged the feature set that achieved the best FAR and FRR values in the previous section.

Table 4 shows the FAR and FRR values for the attacker sensor fingerprints. When compared to the results in Table 3, FAR values are significantly higher when touch or sensor patterns are only used as the sensor fingerprint. This is expected as the attacker was able to directly observe the true user authenticating on the mobile device, and attempted to closely resemble the process. However, when touch and sensor patterns are combined into a single sensor fingerprint, the false accept and reject rates only slightly increase and remain well below 3%. This is surprisingly low given that the attacker was able to closely monitor the true user authentication process right before the attack. In

Mode	2-hand Gesture		1-hand Gesture	
	FAR	(FRR+FAR)/2	FAR	(FRR+FAR)/2
Touch	12.99%	7.69%	15.9%	8.85%
Sensor	11.2%	6.87%	20.8%	11.71%
Both	2.86%	2.67%	5.9%	4.12%

Table 4. FAR and FRR values for the attack scenarios and for both 2-hand and 1-hand gestures.

contrast, an attacker that was able to closely observe the true user entering a 4-digit pin, would be able to get 100% false accept rates.

In the case of the single hand gesture, the trends are similar, but now the FAR value increases to reach 6% when both touch and sensor patterns are combined. However, even in this case, the FAR and FRR values remain well below 6% indicating that the 1-hand gesture can still provide reasonable protection from attackers.

4.3 Computation Overhead

On a Google Nexus 4 device running Android 4.3, processing the touch data takes only $6.7ms$. However, processing the accelerometer and gyro data on the same device takes 3.1 seconds. Such a delay is prohibitive for any realistic use of the proposed approach.

This 3 second delay is mainly caused by two factors. First, every candidate sensor fingerprint is currently compared to all 30 enrolled gestures from the true user. Second, for each comparison between a candidate sensor fingerprint and an enrolled sensor fingerprint, the cross-correlation and dynamic time wrap is computed for both the accelerometer and gyro data. This operation is time consuming when the sensors are sampled at very high data rates such as 200Hz.

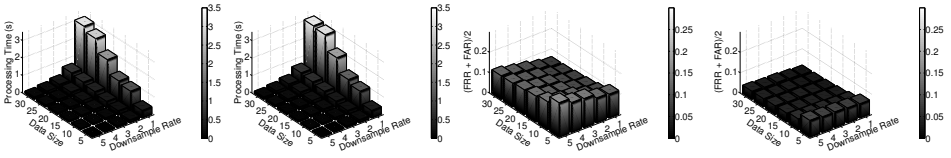
Figure 8(a) and Figure 8(b) show the processing time as a function of the number of enrolled gestures per user, and the sensor down-sampling rate. Simply down-sampling accelerometer and gyro data by a factor of 2, reduces the processing time to approximately half a second. In addition, when only 15 enrolled gestures are used per user, the overall processing time becomes approximately 200ms. This delay is practically unnoticeable by the user, resulting into an instant authentication user experience. The small processing time also implies a low energy overhead, preventing our method from draining the battery.

As Figure 8(c) and Figure 8(d) show, when sensor data is down-sampled by a factor of 2, and the number of enrolled gestures is 15, the mean FAR and FRR values remain practically unchanged. As a result, the proposed technique can provide an instant authentication experience without sacrificing accuracy and robustness to attacks.

5 Related Work

To address the susceptibility of current authentication techniques to shoulder surfing attacks [14], researchers have already proposed to understand *how* a user performs the gesture, and to leverage this information to strengthen the authentication process while maintaining its simplicity [9, 3–5, 10, 12, 8, 13].

Specifically, the work in [3] expanded the typical gesture unlock techniques employed by Android devices, to incorporate the timing of the user's gesture. The work in [9] expanded on this idea by incorporating additional information such as pressure, and size of the finger taps during the gesture. In contrast, our work focuses on designing new unlock gestures with the goal of capturing the geometry of the user hand through



(a) Sensor Only (b) Touch+Sensor (c) Sensor Only (d) Touch+Sensor

Fig. 8. Processing time ((a),(b)) and accuracy ((c),(d)) as a function of the number of enrolled gestures per user, and the sensor down-sampling rate.

the touch screen, and the embedded accelerometer and gyro sensors. Even though valuable, timing, size and pressure information does not provide enough discriminating power to accurately differentiate users, resulting into 2-3 times higher false accept and false reject values compared to the approach presented in this paper.

More recently, Shahzad et al. [12] studied various touch screen gestures to understand the feasibility of combining touch screen data with accelerometer signatures to authenticate users. Even though the same sensing modalities were used, the gestures proposed and analyzed in [12] do not focus on, and were not designed to, capture the geometry of the user's hand. Instead, they mainly focus on capturing the velocity at which finger taps take place. However, capturing the geometry of the user's hand through the unlock gesture is a key parameter in terms of accuracy. Evident of this, is the fact that the work in [12] achieves the same FAR and FRR values as the 2-hand gesture proposed in this paper, only when the user performs 3 different 2-hand gestures sequentially. Asking users to perform 3 different gestures in a row increases the cognitive overhead for the user and the time it takes to unlock the device, raising usability concerns. The work in [13] proposed to design user-generated free-form gestures for authentication. However, it was only evaluated on tablets and the effectiveness of the method on devices with smaller screens such as smartphones was not demonstrated.

The closest to our work is the one proposed by Sae-Bae et al. [10] where new multi-touch gestures were proposed to capture the geometry of the user's hand to enable reliable user authentication. In particular, multiple 5-finger gestures were proposed targeting devices with large screens such as tablets. In their approach, only touch sensor data were used to differentiate users. Even though 5-finger gestures can provide even richer information about the user's hand geometry, they can only be applied on tablet-like devices. Not only do smaller devices, such as phones, lack the physical space required by these gestures, but they can only support up to 4 simultaneous touch points.

User authentication techniques have also been proposed outside the context of touch screens, accelerometer and gyro sensors. For instance, Jain et al. [6] proposed to extract a detailed description of the user's hand geometry by taking a picture of the user's hand. Even though this is a more accurate way to capture the hand geometry, asking users to properly place their hands in front of the phone's camera can be awkward, time-consuming, and also susceptible to environmental lighting conditions. Sato et al. [11] proposed a capacitive fingerprinting approach where a small electric current is injected into the user's body through the touch screen, enabling the measurement of user's bio-impedance. However, bio-impedance measurements are inherently noisy due to grounding issues, and variability in the body's fat and water throughout the day.

6 Discussion and Limitations

Our experimental evaluation shows that carefully designed gestures can enable sensor fingerprints to accurately differentiate users and protect against attackers. Note that the goal of this work is not to achieve recognition rates that are similar to fingerprint sensors, nor to replace them. Instead, our goal is to propose an alternative authentication mechanism for mobile devices that is both intuitive and easy for users to perform, and at the same time hard for attackers to bypass. Sensor fingerprints can be significantly more secure compared to pins, picture passwords, and simple unlock gestures, but definitely not as accurate as fingerprint sensors. However, as physical buttons on mobile devices are eliminated in favor of edge-to-edge displays, and given the lack of technology to properly embed fingerprint sensors into touch screen displays, the use of fingerprint sensors becomes challenging. With this in mind, we believe that sensor fingerprints can be a viable alternative to user authentication on mobile devices.

In practice, the use of sensor fingerprints can be rather tricky. When the user is actively moving (i.e., walking, driving, etc.), the accelerometer and gyro recordings will capture the user's motion rather than the displacement of the phone due to the gesture. However, mobile devices already enable continuous sampling of sensors to recognize higher level activities such as sitting, walking, and driving. When these activities are detected, the acceleration and gyro data could be removed from the sensor fingerprint (or the device could fall back to the 4-digit pin). As Table 2 shows, even when only touch data is used, the FAR achieved is still reasonable.

References

1. 1-hand gesture. https://dl.dropboxusercontent.com/u/64756732/gestures/1-hand-gesture.avi
2. 2-hand gesture. https://dl.dropboxusercontent.com/u/64756732/gestures/2-hand-gesture.avi
3. Angulo, J., Wastlund, E.: Exploring touch-screen biometrics for user identifcation on smart phones. In: IFIP Advances in Information and Communication Technology (2013)
4. Feng, T., Liu, Z., Kwon, K., Shi, W., Carbunar, B., Jiang, Y., Nguyen, N.: Continuous mobile authentication using touchscreen gestures. In: HST (2012)
5. Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. In: IEEE Transactions on Information Forensics and Security (2013)
6. Jain, A., Ross, A., Pankanti, S.: A prototype hand geometry-based verification system. In: AVBPA (1999)
7. Jakobsson, M., Shi, E., Golle, P., Chow, R.: Implicit authentication for mobile devices. In: HotSec'09 (2009)
8. Kolly, S.M., Wattenhofer, R., Welten, S.: A personal touch: Recognizing users based on touch screen behavior. In: PhoneSense'12 (2012)
9. Luca, A.D., Hang, A., Brudy, F., Lindner, C., Hussmann, H.: Implicit authentication based on touch screen patterns. In: CHI (2012)
10. Sae-Bae, N., Ahmed, K., Isbister, K., Memon, N.: Biometric-rich gestures: A novel approach to authentication on multi-touch devices. In: CHI'12 (2012)
11. Sato, M., Poupyrev, I., Harrison, C.: Touche: Enhancing touch interaction on humans, screens, liquids, and everyday objects. In: CHI'12 (2012)
12. Shahzad, M., Liu, A.X., Samuel, A.: Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it. In: MobiCom (2013)

13. Sherman, M., Clark, G., Yang, Y., Sugrim, S., Modig, A., Lindqvist, J., Oulasvirta, A.: User-generated free-form gestures for authentication: security and memorability. In: MobiSys'14 (2014)
14. Wiedenbeck, S., Waters, J., Sobrado, L., Birget, J.: Design and evaluation of a shoulder-surfing resistant graphical password scheme. In: ACM AVI (2006)