IEICE TRANS. INF. & SYST., VOL.E102–D, NO.10 OCTOBER 2019

PAPER

WearAuth: Wristwear-Assisted User Authentication for Smartphones Using Wavelet-Based Multi-Resolution Analysis*

Taeho KANG[†], ***Student Member***, **Sangwoo JI**[†], **Hayoung JEONG**[†], **Bin ZHU**[††], ***and*** **Jong KIM**[†a)], ***Nonmembers***

SUMMARY Zero-effort bilateral authentication was introduced recently to use a trusted wristwear to continuously authenticate a smartphone user. A user is allowed to use the smartphone if both wristwear and smartphone are determined to be held by the same person by comparing the wristwear's motion with the smartphone's input or motion, depending on the grip—which hand holds the smartphone and which hand provides the input. Unfortunately, the scheme has several shortcomings. First, it may work improperly when the user is walking since the gait can conceal the wrist's motions of making touches. Second, it continuously compares the motions of the two devices, which incurs a heavy communication burden. Third, the acceleration-based grip inference, which assumes that the smartphone is horizontal with the ground is inapplicable in practice. To address these shortcomings, we propose *WearAuth*, wristwear-assisted user authentication for smartphones in this paper. WearAuth applies wavelet-based multi-resolution analysis to extract the desired touch-specific movements regardless of whether the user is stationary or moving; uses discrete Fourier transform-based approximate correlation to reduce the communication overhead; and takes a new approach to directly compute the relative device orientation without using acceleration to infer the grip more precisely. In two experiments with 50 subjects, WearAuth produced false negative rates of 3.6% or less and false positive rates of 1.69% or less. We conclude that WearAuth operates properly under various usage cases and is robust to sophisticated attacks.

key words: *smart devices, user authentication, motion sensor, signal processing*

1. Introduction

Nowadays, smartphones have been widely used in our daily lives. People carry smartphones around and use them for various applications; some applications might be sensitive, such as executing financial transactions or paying goods and services. In general, a smartphone is used frequently in a day, and briefly each time. Unlike wearable devices such as wristbands that people wear them around, a smartphone may be temporally out of the control of its legitimate user or even lost or stolen, which may pose a risk of unwanted access to the smartphone. Protection of smartphones from any unauthorized access has become an imperative requirement. Such protection should be user-friendly, desirably transparent to their users so that there is no impedance to smartphone's frequent yet brief usage patterns. Like other commercial off-the-shelf (COTS) devices, smartphones are typically protected with user authentication based on the following three factors: knowledge (e.g., passwords), inherence (e.g., fingerprint, face, and iris), and possession (e.g., trusted devices). Possession-based user authentication is an attractive approach for smartphones since it can make the whole process of user authentication completely transparent.

A possession-based method called *Continuous Seamless Authentication using Wristbands (CSAW)* [1] has recently been proposed. CSAW extends the idea of the zero-effort bilateral authentication introduced in ZEBRA [2] to smartphones. It relies on a trusted wristwear to authenticate a smartphone user continuously by verifying whether the user is wearing the trusted wristwear. If both trusted wristwear and the smartphone are possessed by the same user, referred to as *same-ownership* in this paper, it determines that the user is a legitimate user and allows the user to access the smartphone. Otherwise, i.e., *different-ownership*, it determines that the user is unauthorized and does not allow the user to access the smartphone. CSAW determines same-ownership by comparing the wristwear's motions with the smartphone's inputs and motions. If the wristwear shows correlated motions with them, CSAW determines same-ownership, otherwise different-ownership. CSAW uses wrist movements naturally occurred in using smartphones for authentication; it does not require the user to make any specific motions or behaviors such as fingerprint inputs or to memorize any secrets such as passwords for authentication.

Despite a stride forward, CSAW suffers from the following shortcomings: First, the authentication may work improperly when the user is walking. This is because the gait can conceal the wrist's motion of making touches which are usually small and transient. This problem can be severe especially in the different-hand state—the hand holding the smartphone is not the wristwear-worn hand providing the input; it is hard to determine whether the wristwear's motion, which is concealed by the gait, is correlated with the smartphone's input. On the contrary, if both devices are in the same hand, it is relatively easy to determine that their motions are correlated regardless of the user's moving state. Second, a large amount of communication is con-

Manuscript received January 22, 2019.
Manuscript revised May 28, 2019.
Manuscript publicized June 21, 2019.

[†]The authors are with the Pohang University of Science and Technology (POSTECH), Republic of Korea.
[††]The author is with the Microsoft Research Asia, China.

*This research was supported by IITP-2018-0-01392, IITP-2018-001441 and NRF-2017R1A2B4010914 through the Institute of Information and Communication Technology Planning and Evaluation (IITP) and the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT). This work was partially done when Taeho Kang worked as an intern at Microsoft Research Asia.

a) E-mail: jkim@postech.ac.kr

DOI: 10.1587/transinf.2019EDP7024

stantly needed to compare the motions of the two devices. To compute similarity measures such as correlation and coherence, one device needs to send its motion data to the other. Third, the acceleration-based grip inference, which assumes the smartphone is horizontal with the ground, is inapplicable in practice.

In this paper, we propose *WearAuth*, wristwear-assisted user authentication for smartphones, which overcomes the aforementioned problems. Frist, to extract the desired touch-specific movements regardless of whether the user is stationary or mobile, WearAuth applies the maximal overlap discrete wavelet transform (MODWT) and its multi-resolution analysis (MRA) [3], [4]. The fundamental idea is that a touch interaction usually takes shorter time than other behaviors. Thus, the corresponding movement at the time of the touch can be separated from the other behaviors and highlighted at high-frequency components. Second, to reduce the communication overhead, WearAuth uses the discrete Fourier transform (DFT)-based approximate correlation [5]. With the approach, the motion similarity can be measured by sending a few DFT coefficients instead of sending all of the original data. Third, to infer the grip more precisely, we directly compute the relative orientation of the two devices by measuring the Z-axes of the two devices in the Earth coordinate system (ECS), where each Z-axis represents the direction perpendicular to the device's screen. This approach is irrelevant to the acceleration and applicable regardless of the user's motion.

To evaluate WearAuth, we conducted two experiments with 50 subjects. We have tested the classifier performance using multiple machine-learning algorithms. The classifier showing the best performance obtained an equal error rate of 0.69%, area under the receiver operating characteristic (ROC) curve of 0.9997, and an F1 score of 0.9901. In addition, WearAuth achieved false negative rates of 3.6% or less for legitimate usage cases, and false positive rates of 1.69% or less for attack cases. The results indicate that WearAuth works properly under various usage cases and is robust to sophisticated attacks.

Our main contributions can be summarized as follows:

- To the best of our knowledge, this is the first research which introduces a novel approach of applying signal analyses to the wristwear-assisted user authentication for smartphones. In WearAuth, the wristwear's touch-specific movement is assessed by using the high-frequency components which capture short-time transients, while the motion similarity of the two devices is measured by using the low-frequency components which represent the general shapes of signals.
- We propose a lightweight approach to measure the motion similarity of the devices by employing the DFT-based approximate correlation.
- We propose new features to infer the smartphone usage cases and to measure the likelihood of the same-ownership. We demonstrate that the proposed features are effective for their own purposes.
- We conducted an extensive evaluation of various legitimate and attack models with 50 subjects.

2. Background

In this section, we briefly address the methods of signal analysis and approximate correlation used in this paper.

2.1 MODWT and MRA

To separate the motion signals into low- and high-frequency components and analyze them separately, we use the MODWT and its MRA. The MODWT introduced by Percival [3], is a modified version of DWT. The MODWT and its MRA [4] have few important properties. First, unlike DWT, MODWT does not subsample, and is defined naturally for all sample sizes. Second, MODWT and its MRA are shift invariant—circularly shifting the time series by any amount will circularly shift the MODWT coefficients by the corresponding amount.

Consider MODWT of a time sequence $x = x_0, x_1, \ldots, x_{N-1}$ with a level J. The MODWT wavelet coefficients ($\tilde{d}_j$) and the scaling coefficients ($\tilde{c}_j$) are formulated as:

$$\tilde{d}_{j,n} = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} x_{n-l \bmod N}, \text{ and}$$
$$\tilde{c}_{j,n} = \sum_{l=0}^{L_j-1} \tilde{g}_{j,l} x_{n-l \bmod N}, \quad (1)$$

for $n = 0, \ldots, N-1$ and $j = 1, \ldots, J$, where L_j denotes the filter length, h_j and g_j are the wavelet and scaling filters, respectively, and $\tilde{h}_{j,l} = h_{j,l}/2^{j/2}$ and $\tilde{g}_{j,l} = g_{j,l}/2^{j/2}$.

The MODWT-MRA decomposes the original time sequence as follows:

$$x = \tilde{D}_1 + \tilde{D}_2 + \cdots + \tilde{D}_J + \tilde{S}_J, \quad (2)$$

where $\tilde{D}_j$ is the jth order detail (high-frequency) component and $\tilde{S}_J$ is the Jth order smooth (low-frequency) component for x. The detail and smooth can be calculated as follows:

$$\tilde{D}_{j,n} = \sum_{l=0}^{N-1} \tilde{h}^{\circ}_{j,l} \tilde{d}_{j,n+l \bmod N} \text{ and}$$
$$\tilde{S}_{J,n} = \sum_{l=0}^{N-1} \tilde{g}^{\circ}_{J,l} \tilde{c}_{J,n+l \bmod N}, \quad (3)$$

where $\tilde{h}^{\circ}_l$ and $\tilde{g}^{\circ}_l$ are the $\tilde{h}_l$ and $\tilde{g}_l$ periodized to length N, respectively [6]. Interested readers may refer [4], [7], [8] for more information.

2.2 DFT-Based Approximate Correlation

We use the Pearson's correlation coefficients to measure the similarity of motion signals. We first define the normalization of x as $\hat{x} = \hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{N-1}$, such that $\hat{x}_k = (x_k - \mu_x)/\sigma_x$,

where μ_x and σ_x are the average and standard deviation of x. For two given signals x and y of equal length N, their correlation coefficient is computed as:

$$corr(x,y) = \frac{1}{N}\sum_{k=0}^{N-1}\left(\frac{x_k-\mu_x}{\sigma_x}\right)\left(\frac{y_k-\mu_y}{\sigma_y}\right) = \frac{1}{N}\sum_{k=0}^{N-1}\hat{x}_k\hat{y}_k. \quad (4)$$

Since $\sum_{k=0}^{N-1}\hat{x}_k{}^2 = \sum_{k=0}^{N-1}\hat{y}_k{}^2 = N$, the equation can be rephrased as:

$$corr(x,y) = \frac{1}{N}\sum_{k=0}^{N-1}\hat{x}_k\hat{y}_k = 1 - \frac{1}{2N}d^2(\hat{x},\hat{y}). \quad (5)$$

We define the DFT of x as $\breve{x}$ such that $\breve{x}_f = \frac{1}{N}\sum_{m=0}^{N-1} x_m e^{i\frac{-2\pi fm}{N}}$. Since the DFT is a linear transformation, the Euclidean distance is preserved as follows: $\frac{1}{N}d^2(\hat{x},\hat{y}) = d^2(\bar{x},\bar{y})$, where $\bar{x}$ and $\bar{y}$ are the DFT of $\hat{x}$ and $\hat{y}$, respectively. Thus, the above equation can be rephrased as follows:

$$corr(x,y) = 1 - \frac{1}{2}d^2(\bar{x},\bar{y}) \quad (6)$$

Generally, the first few DFT coefficients contain the most energy of the signal and capture the raw shape of it. With the symmetry property of the DFT, $2d_p^2(\bar{x},\bar{y}) = 2\sum_{k=0}^{p}(\bar{x}_k - \bar{y}_k)^2$ is a good approximation of $d^2(\bar{x},\bar{y})$ for $p \ll N$.

As a result, we can calculate the approximate correlation with p DFT coefficients as follows:

$$corr_p(x,y) = 1 - d_p^2(\bar{x},\bar{y}) \approx 1 - d^2(\bar{x},\bar{y})/2 = corr(x,y). \quad (7)$$

The approximate correlation has the following error bound [5]:

$$corr(x,y) - \epsilon \le corr_p(x,y) \le corr(x,y) + \epsilon, \quad (8)$$

where the value of p is chosen such that

$$min(2\sum_{k=0}^{p}|\bar{x}_k|^2, 2\sum_{k=0}^{p}|\bar{y}_k|^2) \ge 1 - \frac{\epsilon}{2}. \quad (9)$$

For instance, if we want an error bound of 0.05, we need to compute as many DFT coefficients that contain normalized energy greater than 0.975.

3. Threat Model and Design Assumptions

3.1 Threat Model

3.1.1 Legitimate Usage

We regard a person who wears a wristwear coupled with a smartphone as a *legitimate user* of the smartphone. As with the previous studies [2], [9], the wristwear serves as a trusted device, and the smartphone authenticates its user by verifying the same-ownership of both devices.

In real life, legitimate users use their smartphones in different ways, leading to various patterns of motions sensed by the two devices. These motions play a crucial role in WearAuth's verification of the same-ownership of the two devices. They are affected by two main factors: *the moving state* and *the hand state*. The former can be categorized into two cases, *walking state* (Walk) and *sitting state* (Sit), depending on whether the legitimate user is moving or not. The latter can also be categorized into two cases, *same-hand state* (Same) and *different-hand state* (Diff), depending on whether the two devices are held in the same hand or in two different hands. Their combinations lead to the following four cases: Sit-Same, Sit-Diff, Walk-Same, and Walk-Diff. These four cases will be studied in this paper.

3.1.2 Attacks

We focus on attacks aiming at WearAuth. Such an attack can be defined as any attempt by any user to use the smartphone without wearing the coupled wristwear, i.e., when the same-ownership of both devices is violated. The legitimate user who wears the wristwear in an attack is referred to as a *victim* in this paper.

According to the launching methods, attacks can be classified into two types: *random attacks* and *mimic attacks*. The former is launched by an attacker, referred to as a random attacker, who is unaware of how WearAuth works and makes touch inputs on the smartphone without considering the victim's wrist movements. The latter is launched by an attacker, referred to as a mimic attacker, who knows WearAuth inside out and attempts to make his touch inputs to match the victim's wrist movements to circumvent WearAuth's detection.

3.2 Assumptions

We make the following assumptions in this work:

- **Devices.** The smartphone has a touch screen, and both smartphone and wristwear are equipped with motion sensors.
- **Inputs.** For simplicity, this work focuses on the widely used scenario that a user uses finger(s) to make inputs on the smartphone.
- **Wristwear involvement.** For the legitimate usage, we assume that the wristwear-worn hand is involved in touch interactions—either holding the smartphone or touching its screen. Otherwise, the motion sensed by the wristwear is irrelevant to the interactions on the smartphone, and our scheme cannot differentiate legitimate usage and attacks. We will further discuss this issue in Sect. 6.1.
- **Normal operations.** Both devices are operating under normal circumstances without communication disruption or attacks on their systems or communications.

4. WearAuth Architecture

Figure 1 shows the architecture of WearAuth. It consists of

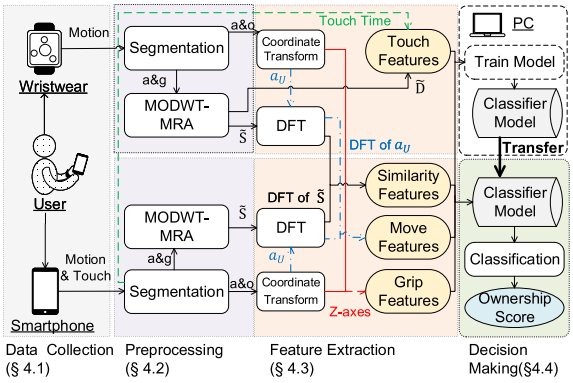


Fig. 1 WearAuth architecture. The a, g, and o denote sensor measurements of the accelerometer, gyroscope and orientation sensor. The $\tilde{S}$ and $\tilde{D}$ represent the smooth and detail components, respectively. Note that the PC in this figure is used to train the classifier model in an offline manner; it is not involved in operations of the smartphone and wristwear.

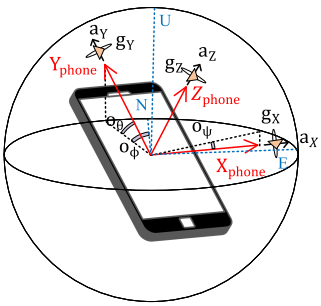


Fig. 2 The motion sensors in Android. The accelerometer measures linear acceleration in m/s^2, the gyroscope measures rotational velocity in rad/s, and the orientation sensor measures degree of rotation that a device makes with respect to the ECS. The N, E, and U are north, east, and up (the opposite direction to gravity) in the ECS.

the following four modules: *data collection*, *preprocessing*, *feature extraction*, and *decision making*, and we will address the details of them in order.

4.1 Data Collection

Whenever the user authentication is required, e.g., the smartphone's screen turns on, WearAuth collects the motion data of three sensors: accelerometer, gyroscope, and orientation sensor, from both smartphone and wristwear, and collects the touch data from the smartphone. The motion data is a set of evenly sampled motion signals with time-stamps written as: $t = \{t_n\}$, $a = [a_X, a_Y, a_Z]^T$, $g = [g_X, g_Y, g_Z]^T$, and $o = [o_\phi, o_\theta, o_\psi]^T$, where T indicates transpose, t is the sequence of time-stamps at which each sample is measured, and n is the array index of the signals. The a, g, and o denote the sensor measurements of the accelerometer, gyroscope and orientation sensor, respectively (see Fig. 2). The X, Y, and Z indicate the three axes of the device, where they extend out of the right-side, the top-side, and the front face of the device. At the same time, the smartphone records the touch data. When a touch input occurs, a sequence of touch event packets is generated, where each packet contains the time-stamp and touch event of finger down, up, and move on the screen.

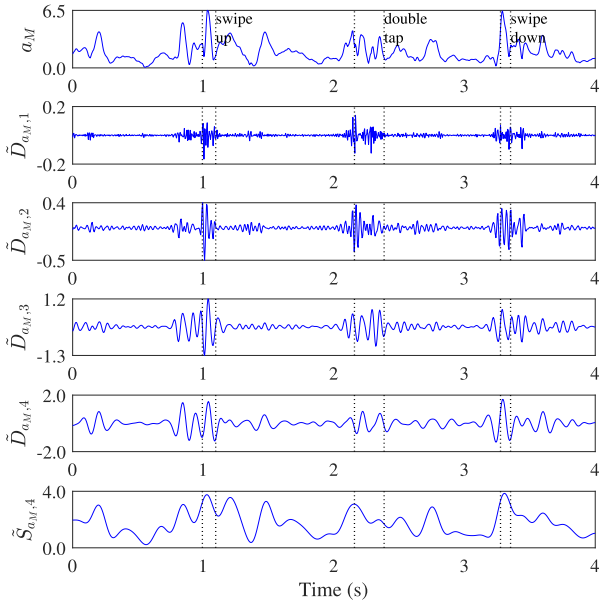


Fig. 3 An example of 4-level MODWT-MRA. The wristwear's acceleration magnitude signal (a_M) is decomposed into the four detail components ($\tilde{D}_{a_M,\{1...4\}}$) and one smooth component ($\tilde{S}_{a_M,4}$). The vertical dotted lines indicate the touch start time and touch end time of the three touch inputs.

4.2 Preprocessing

Segmentation. The collected data is divided into several chunks for further processes. Each touch input is defined by the two touch event packets, *down* and *up*, while the double-tap input contains two downs and two ups. For each touch input, the exact touch start time and touch end time are extracted. The motion data is divided into segments of the equal size N_w written as follows:

$$W_{x,i} = \{x_n\}_{w_i \le n \le w_i + N_w - 1} = [x_{w_i}, x_{w_i+1}, \ldots, x_{w_i+N_w-1}],$$

where x is any motion signal from the three sensors, and $W_{x,i}$ represents ith segment of x with the start index of w_i.

After segmenting the motion signals, WearAuth computes the two magnitude sequences to obtain axis-independent motion information. For each sample of acceleration ($[a_{X,n}, a_{Y,n}, a_{Z,n}]^T$) and rotational velocity ($[g_{X,n}, g_{Y,n}, g_{Z,n}]^T$), the corresponding magnitudes can be calculated as follows: $a_{M,n} = \sqrt{a_{X,n}^2 + a_{Y,n}^2 + a_{Z,n}^2}$ and $g_{M,n} = \sqrt{g_{X,n}^2 + g_{Y,n}^2 + g_{Z,n}^2}$

Signal Decomposition. WearAuth decomposes each motion signal into smooth and detail components using the MODWT-MRA. Figure 3 illustrates an example of 4-level MODWT-MRA of a motion signal. As can be seen, the smooth component characterizes the general shape of the original signal, while the detail components capture the short-time transitions, i.e., touch-specific movements. WearAuth uses the smooth components of both devices to

measure the motion similarity, and the detail components of the wristwear to assess the touch-specific movements. We note that the smartphone's detail components ($\tilde{D}_{\{1,\ldots,J\}}$) are not used. For each segmented motion signal x of the length N_w, the J-level MODWT-MRA decomposes the signal into a matrix of size $(J+1) \times N_w$ as follows:

$$MODWT\text{-}MRA(x) = [\tilde{D}_{x,1}, \tilde{D}_{x,2}, \ldots, \tilde{D}_{x,J}, \tilde{S}_{x,J}]^T. \quad (10)$$

4.3 Feature Extraction

For each motion segment, WearAuth extracts the following four types of features where each type has its own purpose:

- **Move features**: to determine the moving state of each device (or its holder), e.g., sitting or walking
- **Grip features**: to determine the hand state, e.g., same-hand or different-hand
- **Similarity features**: to determine whether the wristwear's motion is similar to the smartphone's motion
- **Touch features**: to determine whether the wristwear performs touch-specific movements

The first two types are used to infer the four legitimate usage cases introduced in Sect. 3.1.1, while the latter two types are used as metrics of the likelihood of the same-ownership where each targets the different usage case.

4.3.1 Move Features

WearAuth infers the moving state of each device by using the acceleration pointing up to the sky (a_U) in the ECS (see Fig. 2). The underlying idea is that when the user is walking, the periodic contacts of user's feet with the ground generate forces perpendicular to the ground. As a result, both devices repeatedly experience fluctuation of the acceleration directed to the sky (a_U) and to the ground ($-a_U$) regardless of their orientation. To compute a_U, we need coordinate transformation of the acceleration from the device coordinate system (DCS) to the ECS using the rotation matrix.

After then, the following four features are extracted from the sequence of $a_U = \{a_{U,n}\}$ of each device: energy, spectral centroid, spectral bandwidth, and spectral entropy. The reason we use these features is because we want to take advantage of spectral components induced by the gait that contains periodic movements. And these features were used in gait analysis of the previous studies [10], [11]. We first define the DFT coefficients of a_U as $\breve{a}_U = \{\breve{a}_{U,f}\}$, and $p_{\breve{a}_U}(f)$ is the normalized magnitude of the fth DFT coefficient: $p_{\breve{a}_U}(f) = |\breve{a}_U(f)| / \sum_{m=0}^{N_w-1} |\breve{a}_U(m)|$.

The energy (E) is the sum of the square of a signal:

$$E(a_U) = \sum_{k=0}^{N_w-1} |a_{U,k}|^2 = N_w \sum_{f=0}^{N_w-1} |\breve{a}_{U,f}|^2. \quad (11)$$

The spectral centroid (SC) is the balance point of the spectral power distribution:

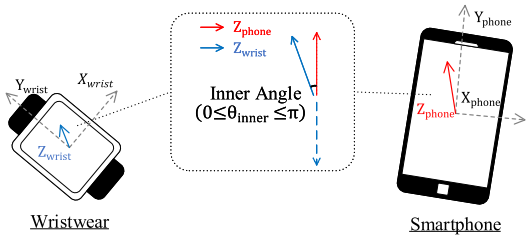


Fig. 4 Inner angle (θ_{inner}) between the Z-axes of the wristwear and smartphone. The value tends to be larger for the same-hand state and smaller for the different-hand state.

$$SC(a_U) = \frac{\sum_{f=0}^{N_w-1} f \cdot p_{\breve{a}_U}(f)^2}{\sum_{f=0}^{N_w-1} p_{\breve{a}_U}(f)^2}. \quad (12)$$

The spectral bandwidth (SB) measures the width of the range of the spectral power distribution:

$$SB(a_U) = \frac{\sum_{f=0}^{N_w-1} (f - SC(a_U))^2 \cdot p_{\breve{a}_U}(f)^2}{\sum_{f=0}^{N_w-1} p_{\breve{a}_U}(f)^2}. \quad (13)$$

The spectral entropy (SE) measures whether the DFT coefficients are concentrated or widespread; the higher the concentration, the lower the value.

$$SE(a_U) = -\sum_{f=0}^{N_w-1} p_{\breve{a}_U}(f) \cdot log(p_{\breve{a}_U}(f)). \quad (14)$$

For each motion segment, the move feature vector has the following form:

$$F_{move} =< E_{wrist}, SC_{wrist}, SB_{wrist}, SE_{wrist}, E_{phone}, SC_{phone}, SB_{phone}, SE_{phone} >,$$

where *wrist* and *phone* indicate the wristwear and smartphone, respectively.

4.3.2 Grip Features

To figure out the hand state, we compute the inner angle between the Z-axes of the two devices in the ECS (see Fig. 4). The Z-axis in the ECS is computed by multiplying the rotation matrix to the Z-axis in the DCS, i.e., $[0\ 0\ 1]^T$. Then, the inner angle (θ_{inner}) is calculated as follows:

$$\theta_{inner,n} = cos^{-1}\left(Z_{wrist,n} \cdot Z_{phone,n}\right), \quad (15)$$

where $(\cdot)$ represents the inner product, and Z_{wrist} and Z_{phone} are the Z-axis unit vectors of the wristwear and smartphone in the ECS, respectively.

Thereafter, the mean (μ_θ) and standard deviation (σ_θ) of the inner angles form the grip feature vector (F_{grip}) as follows: $F_{grip} =< \mu_\theta, \sigma_\theta >$.

4.3.3 Similarity Features

To measure how much the wristwear's motion is similar to

the smartphone's motion, WearAuth compares the smooth components ($\tilde{S}_J$) by using the DFT-based approximate correlation introduced in Sect. 2.2. This can reduce the communication overhead while preserving a proper error bound.

WearAuth computes a total of 21 approximate correlation coefficients ($corr_{p,\{1,\ldots,21\}}$) from the smooth components of the following motion signal pairs:

- $corr_{p,\{1,\ldots,9\}}$: $a_{wrist,\{X,Y,Z\}}$ with $a_{phone,\{X,Y,Z\}}$
- $corr_{p,\{10,\ldots18\}}$: $g_{wrist,\{X,Y,Z\}}$ with $g_{phone,\{X,Y,Z\}}$
- $corr_{p,\{19,20\}}$: $a_{wrist,M}$ with $a_{phone,M}$, and $g_{wrist,M}$ with $g_{phone,M}$
- $corr_{p,21}$: $a_{wrist,U}$ with $a_{phone,U}$

Unlike CSAW, WearAuth compares all possible pairs of the motion signal axes. That is because the associated axis pairs of the two devices depend on the relative device orientation. For example, if the wristwear rotates about the Z-axis by 90 degrees and the X-axis by -90 degrees, the wristwear's X, Y, and Z-axes point to the same directions with the smartphone's Y, $-Z$, and $-X$-axes, respectively, where the minus ($-$) sign indicates the opposite direction.

After then, WearAuth selects the K_s largest values out of the 21 absolute correlation coefficients to form the similarity feature vector ($F_{similarity}$) as follows:

$$F_{similarity} =< Max_1(|corr_p|), \ldots, Max_{K_s}(|corr_p|) >$$

where Max_i indicates the ith largest value. The reason we use the absolute value is that if a pair of associated axes point to the opposite directions, they produce a large but negative correlation coefficient.

4.3.4 Touch Features

In order to assess the relationship between the wristwear's motion and smartphone's input, WearAuth extracts the two features, peak height (PH) and peak location (PL), from the power spectrum of each detail component ($E_j = |\tilde{D}_j|^2$) of the wristwear's motion signals. As shown in Fig. 5, the PH is the maximum value of the normalized power spectrum of the detail component within the touch time defined as:

$$PH_{x,j} = max\{\hat{E}_{x,j,n}\} \text{ for } x \in \{a_X, a_Y, a_Z, a_M, g_X, g_Y, g_Z, g_M\} \text{ and } t_{start} \le n \le t_{end},$$

where $\hat{E}_{x,j}$ is the normalized power spectrum of the jth level detail component of the segmented signal x. The PL is the minimum distance of the PH's time index from the touch start time (t_{start}) or touch end time (t_{end}) defined as:

$$PL_{x,j} = min\{n_{max} - t_{start},\ t_{end} - n_{max}\},$$
$$\text{where } n_{max} = \underset{t_{start} \le n \le t_{end}}{\arg\max}\{\hat{E}_{x,j,n}\}.$$

Generally, in the legitimate usage cases, we have observed that the peaks occur near the touch start time or touch end time because people generally move their wrists sharply at that time, which results in the low PL values. Thus, we have considered the PL as a time-sensitive feature which helps to distinguish between the legitimate usage and attack; if an attacker fails to match his mimicking touch with the victim's movement, which easily occurs, the PL value becomes large.

The challenging problem is that even for the same type of touch interactions, the major detail component(s) that highlights the touch-specific movement, changes every time. For example, a tap is highlighted once at $\hat{E}_{a_Z,2}$, while the next tap can be highlighted at $\hat{E}_{g_X,1}$. This is due to the wristwear's orientation, and the movement type (linear or rotational) and its speed for a touch interaction vary over time.

To handle this problem, WearAuth selects the K_t largest PHs ($\{LPH_1, LPH_2, \ldots, LPH_{K_t}\}$) and its PLs ($\{LPL_1, LPL_2, \ldots, LPL_{K_t}\}$) computed from the $8J$ detail components (J-level MODWT-MRA of 8 motion signals) to make the touch feature vector as follows:

$$F_{touch} =< LPH_1, \ldots, LPH_{K_t}, LPL_1, \ldots, LPL_{K_t} > .$$

If no touch occurs, the feature vector is filled with dummy values; for example, we used -1 as the dummy value. If multiple touches occur within the time duration of a segment, the mean vector is computed. In summary, the touch feature vector forms as follows:

$$F_{touch} = \begin{cases} \underbrace{< -1, -1, \ldots, -1 >}_{2 \times K_t} & \text{, if } N_t = 0 \\ mean\left(\begin{bmatrix} F_{touch,1} \\ \ldots \\ F_{touch,N_t} \end{bmatrix}\right) & \text{, else} \end{cases}$$

where N_t is the number of touch inputs within the time duration of the motion segment, and $F_{touch,i}$ corresponds to the ith touch input of the segment.

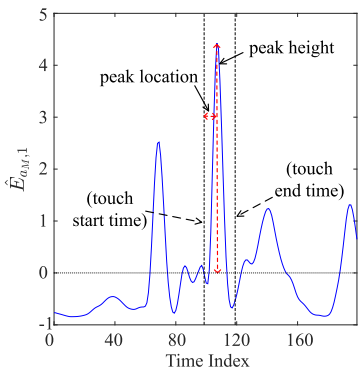


Fig. 5 Two types of the touch features. The *peak height* (vertical red dotted line) is the maximum value of the normalized power spectrum of the detail component within the touch time, and the *peak location* (horizontal red dotted line) is the minimum distance of the PH's time index from the touch start time or touch end time.

4.4 Decision Making

Finally, a pre-trained classifier model periodically takes the four feature vectors as input and produces a score as output. The output score, or *ownership score* (OS), indicates

the likelihood of the same-ownership. We chose a simple threshold-based approach to make the final decision: the same-ownership if $OS \geq threshold$, and the different-ownership otherwise.

The classifier can be trained by most commonly used machine learning algorithms. Among them, we have tested the following five algorithms for WearAuth: decision tree [12], naive Bayes [13], support vector machine [14], random forest [15], and AdaBoost [16]. Note that we build a general classifier model using the data of all participants for each algorithm; the model is not an individual model for each user which relies on the user's unique behavioral traits. The general model is not completely user-agnostic because it has contained the data of each user, but we can verify whether the model works well for all users through the 10-fold cross-validation technique. Their performance will be addressed in Sect. 5.3.2.

5. Evaluation

5.1 Implementation

We used a Samsung Galaxy Note 5 and an LG Watch Urbane to collect smartphone and wristwear data, respectively. The former ran Android Marshmallow (v6.0.1) with 64 GB storage and 4 GB RAM, and the latter ran Android Wear OS (v2.0) with 4 GB storage and 512 MB RAM. We developed applications for each device to collect smartphone's touch data and both devices' motion data. Built-in APIs of Android are used for the most parts of the modules.

At the beginning of an experiment, both devices accessed to a pre-installed stratum-3 server of Network Time Protocol (NTP) [17] to synchronize their time. We note that this explicit time synchronization is not necessary in practice. Once the timelines of the two devices are aligned, there is no need to apply it again. If the NTP is unavailable, cross-correlation can be used to achieve the same effect [18].

5.2 Experimental Settings and Data Collection

We conducted two experiments for our user study. We designed the first experiment to evaluate the overall performance of WearAuth; we captured the data of legitimate usage as well as random and naive mimic attacks. Later we designed an additional experiment to evaluate the robustness of WearAuth against more sophisticated attacks, the opportunistic attacks, introduced in [9]. We recruited a total of 50 volunteers in our experiments; 30 subjects for the first one and 20 subjects for the second one. Most of the subjects were campus students and the other were workers in our institution. Each subject took about an hour to complete assigned tasks and received a $10 reward after completion. The demographic information of the subjects of the two experiments is summarized in Table 1. Our experiments were approved by the Institutional Review Board of our institution.

Table 1 Demographics of a total 50 subjects of two experiments.

Gender		**Watch-worn hand**	
Male	Female	Left	Right
36	14	48	2
Age			
~24	25~29		30~34
33	12		5

Unlike the experiments in CSAW, we did not distinguish tasks separately. We only focused on the task of phone use, i.e., touch interactions, for the (de)authentication of users; other tasks such as smartphone pick-up or rotate evaluated in CSAW were not considered. If the smartphone is actually in use, there will be a combination of tap and swipe inputs depending on the type of an application used. For example, users may mainly use swipes to do scrolling in a document reader application while they use taps to type in a messenger application. Because of the vast amount of combinations, we instructed the subjects to perform basic touch interactions in a random order instead of using application-specific combinations. For the attacks, we assumed the cases where the attackers (subjects) took the automatically unlocked smartphone; at the same time, the victim (one researcher) was nearby and she was stationary, walking, or using another smartphone or a PC depending on the case. The details are addressed in the following sections.

5.2.1 Data of Legitimate Usage

As mentioned in Sect. 3.1.1, we have considered the four legitimate usage cases with the two moving states, i.e., walking and sitting, and two hand states, i.e., same-hand and different-hand. As in CSAW [1], the same-hand state manifests in three different hand grips, *WW*, *WN*, and *BB*, while the different-hand state has one hand grip, *NW* (see Fig. 6). Each grip shown in the figure is named by two letters: the first letter indicates the hand holding the smartphone, and the second letter indicates the hand interacting with the smartphone; *W* stands for the hand wearing the wristwear, *N* stands for the other hand without the wristwear, and *B* stands for the both hands.

We collected the legitimate usage data with 30 subjects at the first experiment. Each subject wore the wristwear to play the role of a legitimate user using the smartphone and conducted eight sessions; each session devoted to one of the two moving states and one of the four hand grip conditions. The smartphone application instructed the subject to make touch inputs by displaying the six types of basic touches, *tap, double tap, swipe up, swipe right, swipe down*, and *swipe left* in a random order with auxiliary information, one basic touch at a time. A total of 33,236 touch inputs and 19,843 motion segments were collected, with the detail given in Table 2.

5.2.2 Data of Attacks

As described in Sect. 3.1.2, attacks can be classified into two

(a) WW (b) WN (c) BB (d) NW

Fig. 6 Four hand grips. The left three are labeled as the *same-hand* state, and the remaining one is labeled as the *different-hand* state.

Table 2 The number of subjects, touch inputs and motion segments under the four states of the legitimate usage.

Legitimate usage	Hand grip	Subject number	Tap	Double tap	Swipe up	Swipe right	Swipe down	Swipe left	Total input	Segment number
Sit-Same	WW	30	624	625	625	625	619	623	3741	2408
	WN	30	709	716	710	717	706	704	4262	2503
	BB	30	728	725	725	723	721	726	4348	2508
Sit-Diff	NW	30	694	695	692	693	702	694	4170	2501
Walk-Same	WW	30	686	688	689	686	680	685	4114	2500
	WN	30	711	711	718	715	715	714	4284	2505
	BB	30	704	704	711	710	705	712	4246	2499
Walk-Diff	NW	30	677	673	683	681	677	680	4071	2419

Table 3 The number of subjects, touch inputs and motion segments of seven attack cases. The attack data consists of the researcher's wristwear data and subjects' smartphone data. The smartphone data of the two random attacks is same with the Mimic-Phone-Naive-All attack. However, there is the difference in the number of motion segments because each victim data is longer in the random attacks than in Mimic-Phone-Naive-All.

Attack case	Subject number	Tap	Double Tap	Swipe up	Swipe right	Swipe down	Swipe left	Total input	Segment number
Random-Sit	30	2511	1912	1989	2081	1933	2040	12466	9987
Random-Walk	30	2511	1912	1989	2081	1933	2040	12466	9987
Mimic-Phone-Naive-All	30	2511	1912	1989	2081	1933	2040	12466	9941
Mimic-Phone-Opp-All	20	466	357	363	397	386	414	2383	2017
Mimic-Phone-Opp-Tap	20	1952	0	0	0	0	0	1952	1878
Mimic-PC-Opp-All	20	2167	200	118	232	99	269	3085	2486
Mimic-PC-Opp-Keyboard	20	2634	0	0	0	0	0	2634	2463

types: *random* or *mimic*. A total of 47,452 touch inputs and 38,759 motion segments were collected with the detail given in Table 3. For the attack data, one of the researchers played the role of a victim. To reduce the impact of changes in the researcher's data, the researcher wore the wristwear and did some prescribed behaviors before conducting the experiment, and her wrist movements were recorded through the wristwear's motion sensors and the video camera. Thereafter in the evaluation, the attack data is built by combining the researcher's pre-recorded wristwear data with each subject's smartphone data.

The design target of attack data is twofold: to show 1) the robustness of WearAuth with various types of wristwear motions when the smartphone is used by other person, and 2) how the attacker's strategy of making touch inputs by matching them with the victim's wrist movements affects the security of WearAuth. For that, we conducted two experiments. At the first experiment with 30 subjects, we collected three types of attack data: Random-Sit, Random-Walk, and Mimic-Phone-Naive-All. Later at the second experiment with 20 subjects, we collected the data of more sophisticated attacks, the opportunistic attacks introduced in Huhta et al. [9].

The random attacks are divided into *Random-Sit* and *Random-Walk* depending on the moving state of the victim. The victim in the random attacks did not hold the smartphone, i.e., no hand grip condition, and moved her hands freely such as putting her hands on the desk, drinking water, or just keeping her hands still in Random-Sit, and putting her hands in her pocket or waving her arms while walking in Random-Walk. Note that the smartphone data used in the random attacks is same with Mimic-Phone-Naive-All. This is because we want to reduce the burden of the subjects. Although the data was generated with the subjects following the victim's touch interactions, it was also the data of subjects using a smartphone. Thus, instead of requiring additional touch interactions of subjects, which should be irrelevant to victim's wrist movements in the random attacks, we combined the same smartphone data of Mimic-Phone-Naive-All with the three different types of researcher's wristwear data to generate Random-Sit, Random-Walk, and Mimic-Phone-Naive-All.

For the mimic attacks, the subjects were instructed to mimic touch interactions to match the victim's wrist movements in the pre-recorded video. To help the subjects, the video included the hints of captions and audio. We tested WearAuth's robustness under various cases of the mimic attacks. Among them, the following two cases are considered as the most advantageous to the attackers and will be reported in this paper: *Mimic-Phone* and *Mimic-PC*. The former simulates an attacker observing and mimicking the victim's touch interactions on another smartphone, while the latter simulates an attacker exploiting the wrist movements of the victim who moves a mouse or types on a keyboard with a PC. Both cases would create many opportunities for an attacker to launch the mimic attacks since the victim's wrist motions in these two cases are either same or similar with those in the legitimate usage. Specifically for Mimic-PC, moving a mouse resembles swipes, i.e., horizontal wrist move, and typing on a keyboard resembles taps, i.e., vertical wrist move, on a smartphone. Thus, a smart attacker might exploit the opportunity to match his touch interactions on the victim's smartphone with the victim's wrist movements of using a PC.

According to the attacker's strategy, the mimic attacks are further divided into the five categories as shown in Table 3. *Naive* stands for the naive mimic attacks where the subject tried to mimic all of the victim's interactions. *Opp* stands for the opportunistic mimic attacks, introduced in Huhta et al. [9], where the subjects could choose *when* and *the type of interactions* to mimic; they were instructed to skip mimicking if they thought it was not possible to succeed. Accordingly, the subjects opportunistically mimicked the victim making all types of the touch interactions in *Mimic-Phone-Opp-All*, and the victim making tap interactions only in *Mimic-Phone-Opp-Tap*. Similarly, the subjects mimicked the victim using both keyboard and mouse in *Mimic-PC-Opp-All*, and the victim using the keyboard only in *Mimic-PC-Opp-Keyboard*.

The victim and attackers conducted four sessions with the four hand grip conditions in Mimic-Phone-Naive-All; conducted one session with one hand grip condition, NW (different-hand), in the opportunistic mimic attacks. We excluded the three same-hand conditions for the opportunistic attacks because the strategy of the attacks is not effective in matching the motions of the two devices—if an attacker chooses the same-hand condition and skips to match smartphone motions, WearAuth would detect the mismatched motions periodically with or without touch interactions. Note that the mimic attackers can choose the hand state by observing the victim's wristwear and adjusting the orientation of the victim's smartphone. In addition, if the victim uses a PC, her wristwear is likely to point up, which makes the situation similar to the different-hand state as long as the smartphone's front face points up, too. We also excluded the walking state for the mimic attacks because if the victim and/or the attacker walked, the mimic attacks might be infeasible in practice.

Table 4 Parameters used for the evaluation results.

Data Collection & Segmentation	Sampling rate of motion sensors	200 Hz
	Motion segment size	4 sec
	Segment overlapping rate	0.5
Signal Decomposition	Wavelet function	symmlet-4
	Decomposition level (J)	4
Feature Extraction	# of inner angles (N_θ) used to compute the grip features	80
	# of DFT coefficients (p) used to compute the approximate correlation	30
	# of approximate correlation coefficients (K_s) selected to form the similarity features	9
	# of LPHs and LPLs (K_t) to form the touch features	5

5.3 Experimental Results

Table 4 gives the parameter description for the evaluation results. We tuned our parameters similar to the previous work [1]. We collected the motion sensor data with the sampling rate of 200 Hz. We also segmented the motion signals into 4s-segments (800 samples) with overlapping rate of 0.5; as a consequence, the first segment contains the data of 0∼4s and the second one contains the data of 2∼6s and so on. For the signal decomposition, we chose the symmlet-4 wavelet [7], since it is simple but has lesser phase shift than the Haar wavelet which is the simplest one. In addition, the decomposition level (J) was set to 4 to extract 4 detail components and 1 smooth component. For a 4s-segment, a total of 80 inner angles were computed for the grip features; a simple down sampling technique is used, i.e., choose one for every 10 samples, to reduce the computation overhead. Since the frequency resolution of a 4s-segment is 0.25 Hz, the 30 DFT coefficients can cover the frequency range of the smooth components (< 6.25 Hz), which are used for the approximate correlation, with small information leakage. The number of the similarity features (K_s) and touch features (K_t) were empirically chosen to 9 and 5, respectively. Consequently, every two seconds, a total of 29 ($8+2+9+5\times2$) features were extracted from a 4s-segment, and an ownership score is yielded as the classification result.

5.3.1 Feature Analysis

To evaluate the effectiveness of the proposed features, we computed the cumulative distribution functions (CDFs) of each feature for each case, and compared them using the two-sample Kolmogorov-Smirnov (KS) test.

Two-sample Kolmogorov-Smirnov Test. The two-sample KS test tests the hypothesis that the samples come from the same distribution. It is a non-parametric test, and its usefulness comes from the absence of any assumption on the sample distributions [19]. For two random variables X and Y, the KS statistic ($D_{X,Y}$) quantifies the largest discrepancy between the two CDFs of the samples as follows: $D_{X,Y} = \sup_z |F_X(z) - F_Y(z)|$, where *sup* refers to the

Table 5 The KS statistics of the 29 features used in WearAuth. The larger value indicates more discrepancy of the feature.

Move Features								**Grip Features**	
E_{wrist}	SC_{wrist}	SB_{wrist}	SE_{wrist}	E_{phone}	SC_{phone}	SB_{phone}	SE_{phone}	μ_θ	σ_θ
0.9651	0.8442	0.3070	0.8106	0.9558	0.2815	0.4765	0.3522	0.9163	0.1524
Similarity Features ($Max_i(\|corr_p\|)$)									
$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$	
0.9366	0.9380	0.9414	0.9406	0.9369	0.9345	0.9316	0.9276	0.9236	
Touch Features									
LPH_1	LPH_2	LPH_3	LPH_4	LPH_5	LPL_1	LPL_2	LPL_3	LPL_4	LPL_5
0.7393	0.7608	0.7666	0.7650	0.7574	0.2012	0.2143	0.2823	0.3391	0.3669

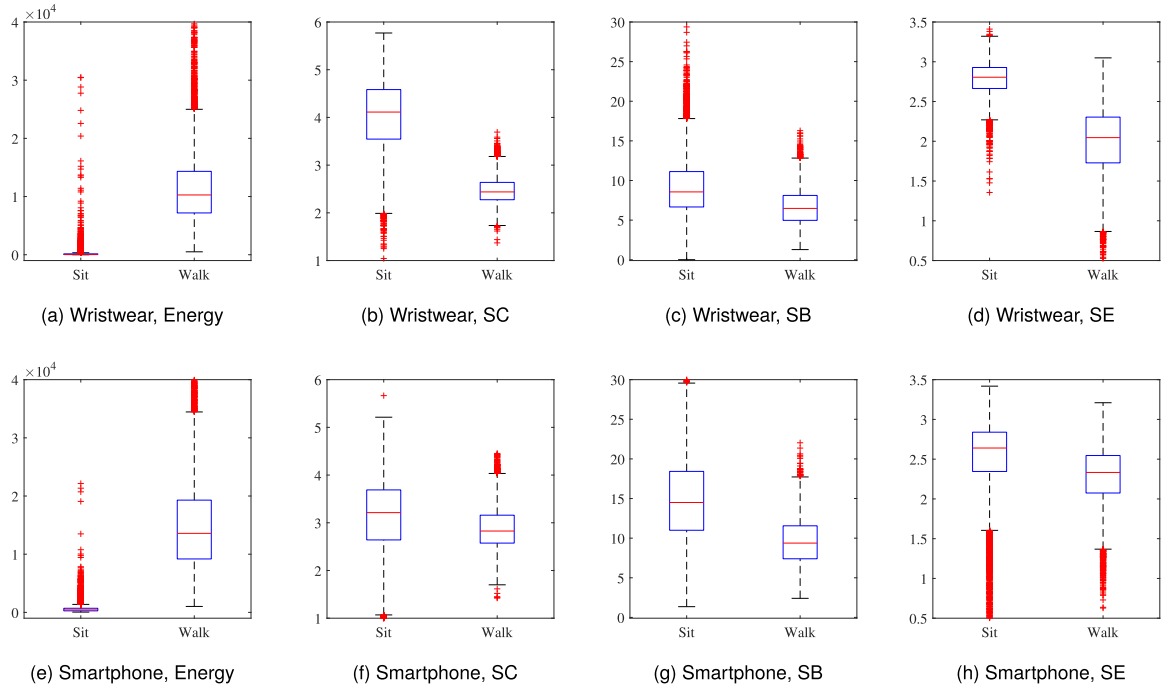


Fig. 7 The boxplots of the move features comparing between the sitting state and walking state. The four graphs above indicate the features from the wristwear, while the four graphs below indicate the features from the smartphone. Each boxplot contains the median (red line), inter-quartile range (blue box), extreme values (whiskers), and outliers ('+' symbol).

supremum function, and F_X and F_Y indicate the CDFs of X and Y, respectively. The null hypothesis that the two samples come from the same distribution is rejected at level α if

$$D_{X,Y} > c(\alpha)\sqrt{\frac{N_X+N_Y}{N_X N_Y}}, \text{ where } c(\alpha)=\sqrt{-\frac{1}{2}ln\left(\frac{\alpha}{2}\right)}, \tag{16}$$

and N_X and N_Y are the sizes of X and Y, respectively.

Table 5 shows the KS statistics of the features used in WearAuth. To compute the KS statistic of each feature, we used the different dataset depending on the own purpose of each feature described in Sect. 4.3. Specifically, the two datasets used for each feature are summarized as follows:

- **Move features:** Sit (Sit-Same+Sit-Diff) vs. Walk (Walk-Same+Walk-Diff).
- **Grip features:** Same (Sit-Same+Walk-Same) vs. Diff (Sit-Diff+Walk-Diff).
- **Similarity features:** Same (Sit-Same+Walk-Same) vs. Attacks (only Same-hand).
- **Touch features:** Diff (Sit-Diff+Walk-Diff) vs. Attacks (only Diff-hand).

Move features - Sit vs. Walk. Figure 7 depicts the boxplots of the move features. As expected, both wristwear and smartphone had higher energy when they were in the walking state (see Fig. 7 (a) and 7 (e)). Figure 7 (b) and 7 (f) depict that the sitting state had higher SCs than walking state. This is because the devices were mainly influenced by touch interactions in the sitting state and by gait in the walking state, where the touch interactions have higher frequency components than the gait. The spectral bandwidth (see Fig. 7 (c) and 7 (g)) and spectral entropy (see Fig. 7 (d) and 7 (h)) indicate that the spectral components were more concentrated in the walking state because the gait dominated both devices. The result implies that the energy feature

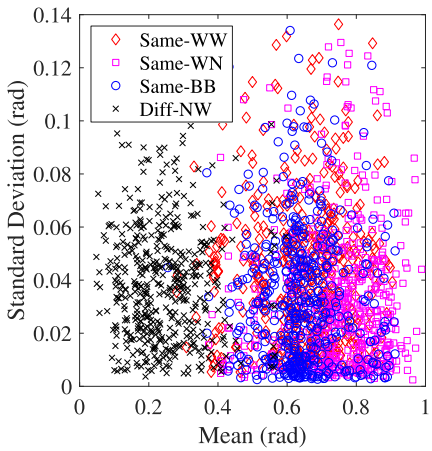


Fig. 8 The mean vs. standard deviation of inner angles for the four hand grip conditions. The graph depicts that the different-hand state (NW) has smaller inner angles than the same-hand state (WW, WN, and BB).

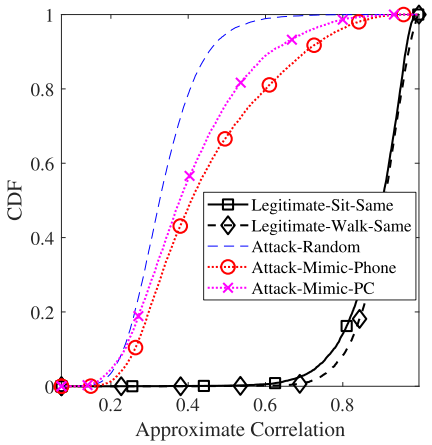
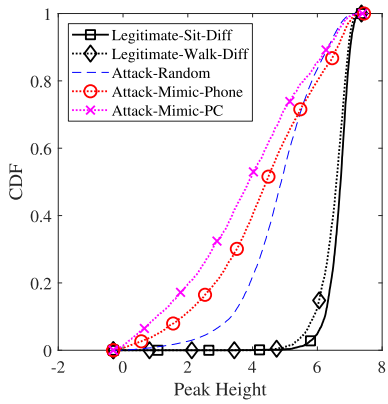


Fig. 9 The cumulative distribution function of the first largest approximate correlation used as the similarity feature. The figure depicts that the same-hand state has higher correlation than the attack cases.

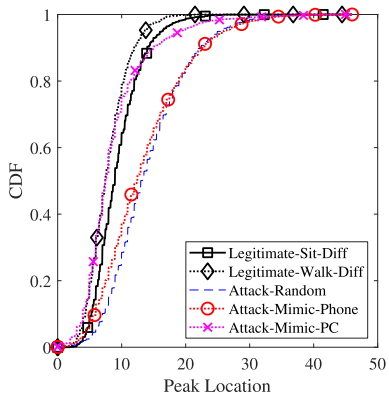
played a major role in distinguishing the moving state, and the remaining features support it.

Grip feature - Same-hand vs. Different-hand. Figure 8 shows the plot of the mean and standard deviation of the inner angles for the four hand grips. As expected, the same-hand state induced the larger inner angles than the different-hand state (0.7 rad vs. 0.2 rad). However, the standard deviation did not show much discrepancy that its KS statistic is lower than the mean (0.1524 vs. 0.9163) as shown in Table 5.

Similarity feature - Similar motion or not. Table 5 shows that the similarity features had high discrepancy with the KS statistics greater than 0.92. Figure 9 depicts the CDFs of the first largest approximate correlation for each case. The figure shows that about 90% of the same-hand state had the correlation of 0.8 or more. On the contrary, about 90% of the Random, Mimic-Phone, and Mimic-PC attacks had the correlation smaller than 0.45, 0.7, and 0.62, respectively. The result implies that the approximate



(a) The first largest peak height



(b) The peak location of the fifth largest peak

Fig. 10 The cumulative distribution functions of the peak height and peak location used as the touch features. The figure indicates that both of the two different-hand states (Sit-Diff and Walk-Diff) have higher values of the peak height and smaller values of the peak location than the attack cases.

correlation is appropriate to measure the motion similarity between the two devices.

Touch feature - Touch-specific motion or not. According to Table 5, the PHs were better to distinguish the different-hand state from the attack cases than the PLs. (0.7393∼0.7666 vs. 0.2012∼0.3669). Figure 10 depicts the CDFs of the LPH_1 and LPL_5 for each case. As can be seen, the legitimate usage case had larger peak heights and smaller peak locations except for the Mimic-PC case. The figure also implies that the touch features are effective even in the walking state. The reason for the small PL values of the Mimic-PC attack was because the subjects mainly chose the tap inputs in that case (see Table 3). Since the largest possible PL value is the half duration of the touch time ($(t_{end} - t_{start})/2$), the tap inputs, which were usually short, induced the small PL values. Nonetheless, the Mimic-PC attacks could not obtain high PHs. The result implies that the mimicking attacks were not very helpful in negating the effectiveness of the touch features.

5.3.2 Security and Usability Analysis

To evaluate the security and usability of WearAuth, we

Table 6 The performance comparison of the classifiers with different machine-learning algorithms.

Algorithm	EER	AUC	F1 Score
Decision Tree	0.0239	0.9926	0.9652
Naive Bayes	0.0360	0.9931	0.9468
Support Vector Machine	0.0332	0.9945	0.9505
Random Forest	0.0153	0.9987	0.9783
AdaBoost	0.0069	0.9997	0.9901

employed widely used k-fold cross-validation with $k = 10$; the collected data was randomly partitioned into k equal-sized sets with $k - 1$ sets used as training data and the remaining set used as testing data. This process was repeated k times, and their average was used as the result. Note that we built a general classifier model for all users as described in Sect. 4.4.

In this paper, the same-ownership (authorized access) is considered as *positive* (P) and the different-ownership (unauthorized access) is considered as *negative* (N). If the positive is classified correctly, it is called *true positive* (TP), otherwise *false negative* (FN). Similarly, if the negative is classified correctly, it is called *true negative* (TN), otherwise *false positive* (FP). WearAuth's performance is measured with the following metrics:

- **False negative rate (FNR).** The FNR is the proportion of legitimate users' segments misclassified as attackers' written as: $FNR = FN/P = FN/(TP + FN)$.
- **False positive rate (FPR).** The FPR is the proportion of attackers' segments misclassified as legitimate users' written as: $FPR = FP/N = FP/(TN + FP)$.
- **Equal error rate (EER).** The EER is the rate where the FNR and FPR are equal. The lower EER indicates the better performance of classifiers.
- **Area under the ROC curve (AUC).** An ROC curve is a two-dimensional depiction where the true positive rate is plotted on the Y axis and the FPR is plotted on the X axis. The AUC is another metric to measure the performance of a classifier which ranges between 0 and 1, where the higher value indicates the better performance.
- **F1 score.** F1 score is the harmonic mean of the precision and recall written as: $F1 = 2TP/(2TP + FP + FN)$. Its value will be between 0 and 1, where the higher value indicates the better accuracy.

Table 6 shows the performance comparison of the classifiers of the five machine-learning algorithms mentioned in Sect. 4.4. To evaluate the performance, we used MATLAB R2018a [20], where the parameters for each classifier were optimized before the evaluation. As can be seen, all of the classifiers obtained the proper EERs ranging from 0.69% to 3.60%, which implies that our features are effective regardless of the classifier algorithms. Among the algorithms, the two ensemble-based algorithms, random forest and AdaBoost, achieved better performance than the others.

Table 7 shows the error rates for each of the four legitimate usage cases and seven attack cases introduced in Sect. 5.2.1 and 5.2.2. Among the legitimate cases, the Sit-Diff case obtained the highest FNR of 3.60%. This is because the Sit-Diff case mainly depends on the touch features that have less discrepancy than the similarity features as shown in Table 5. On the other hand, the Walk-Diff case achieved 0% error rate by exploiting the benefits of both touch-specific movements as well as the strong correlation induced from the gait. For the attack cases, the Mimic attacks achieved better success rates than the Random attacks (0.49~1.69% vs. 0~0.29%). We note that all of the Random-Walk attacks, where the two devices have different moving states, were detected by WearAuth. Among the Mimic attacks, the Mimic-Phone cases showed the higher attack success rates than the Mimic-PC cases. That is because the victim using the smartphone gave better opportunities to the attacker than using the PC. Comparing the Mimic-Phone-Naive-All (1.47%) and Mimic-Phone-Opp-All (1.69%), the opportunistic strategy gave little improvement of the attack success rate to the subjects. One unexpected result was shown that the Mimic-Phone-Opp-All (1.69%) achieved the higher FPR than the Mimic-Phone-Opp-Tap (1.28%), and the Mimic-PC-Opp-All (0.84%) achieved the higher FPR than the Mimic-PC-Opp-Keyboard (0.49%). Before conducting the experiment, we had assumed that the strategy of mimicking the simple actions only, i.e., tapping and typing, would give better opportunities to attackers. However, the result led to the opposite conclusion. We believe this is because the victim's simple actions were shorter than other interactions which made the attackers hard to match their mimicry to the victim's wrist movements.

Table 7 The error rates of WearAuth for the legitimate usage and attacks cases. AdaBoost, which shows the best performance in Table 6, was used.

Case	Error Rate
Legitimate usages	
Sit-Same	0.65%
Sit-Diff	3.60%
Walk-Same	0.00%
Walk-Diff	0.00%
Attacks	
Random-Sit	0.29%
Random-Walk	0.00%
Mimic-Phone-Naive-All	1.47%
Mimic-Phone-Opp-All	1.69%
Mimic-Phone-Opp-Tap	1.28%
Mimic-PC-Opp-All	0.84%
Mimic-PC-Opp-Keyboard	0.49%

5.4 Comparative Evaluation

We compared the performance of WearAuth and CSAW. Since no implementation of CSAW was available, we followed the description of its architecture in the paper [1]. We used the same parameter values if available. For instance, we used the same values such as sampling rate, motion segment size and segment overlapping rate.

The main purpose of the evaluation is to show how much performance changes when CSAW is used with our

Table 8 The error rates of CSAW. AdaBoost which showed the best performance was used. M2MC (M) and M2IC (I) exclusively targeted at the same-hand state and different-hand state, respectively.

Case	Error Rate
Grip detection	4.95%
Legitimate usages	
Sit-Same	0.28%
Sit-Diff	16.40%
Walk-Same	0.06%
Walk-Diff	8.35%
Attacks	
Random-Sit	0.19%(M) / 2.08%(I)
Random-Walk	0.02%(M) / 19.10%(I)
Mimic-Phone-Naive-All	0.30%(M) / 14.18%(I)
Mimic-Phone-Opp-All	21.34%
Mimic-Phone-Opp-Tap	5.91%
Mimic-PC-Opp-All	15.56%
Mimic-PC-Opp-Keyboard	8.11%

collected data. For that, we focused on the performance of the following components of CSAW for continuous authentication: grip detector, motion-to-motion correlator (M2MC), and motion-to-input correlator (M2IC). We constructed and evaluated each component using the datasets for each purpose; for example, M2MC targeted at the data of the same-hand state, while M2IC targeted at the data of the different-hand state. We omitted the further evaluation such as the decision policy, confidence booster and scorer. The overall result of CSAW is summarized in Table 8.

Grip Detector. CSAW's grip detector obtained the error rate of 4.95% using the threshold of $4.949 m/s^2$ similar to the original scheme. With the same data, WearAuth had the error rate of 1.79% using the two grip features, i.e., mean and standard deviation of relative orientation. Note that WearAuth does not have an individual classifier of each component, thus we built an extra classifier using AdaBoost which classifies the hand state only.

The error rate of CSAW's grip detector does not seem so severe, but their scheme involves a fatal weakness. Figure 11 shows the difference of Z-axis acceleration used in CSAW and the relative orientation used in WearAuth for the same subject data. Each sub-figure contains the time series data of the four hand grip conditions of the subject. This subject did not hold the smartphone horizontally with the ground when in the BB grip condition. As a result, CSAW could not distinguish the BB grip from the NW grip, while WearAuth could differentiate them.

M2MC. With the results in Table 7 and Table 8, M2MC obtained slightly better performance than WearAuth. For the Sit-Same case, for example, M2MC had an FNR of 0.28%, while WearAuth had an FNR of 0.65%. The interesting part of the results was the attack case. Compared to the random attacks, the mimic attack did not seem very helpful for the attackers in obtaining motion similarity of the two devices (0.02∼0.19% vs. 0.30%). M2MC did not have results of the opportunistic attacks because we excluded the same-hand state for the attacks as mentioned in Sect. 5.2.2.

WearAuth is better than M2MC in terms of the

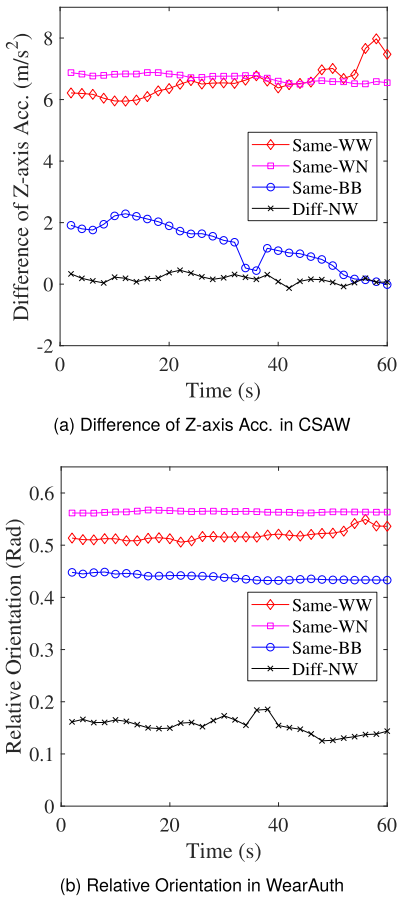


Fig. 11 Comparison between the difference of Z-axis acceleration and relative orientation for the same subject data. CSAW cannot differentiate Same-BB and Diff-NW while WearAuth can differentiate them.

overhead of communication and computation. For every 2s (segment size: 4s, overlapping rate: 0.5), the entire samples (800) of each motion signal of the wristwear should be transferred to the smartphone in CSAW, while only 30 DFT coefficients are transferred in WearAuth. In addition, M2MC computes and uses 256 features for every segment while WearAuth uses only 9 approximate correlation coefficients. The further discussion is addressed in Sect. 6.2.

M2IC. CSAW's M2IC worked improperly with our data. First, M2IC could not differentiate peaks occurred from touch interactions and from gaits. With the results of comparing Sit-Diff (16.40%) vs. Walk-Diff (8.35%), and Random-Sit (2.08%) vs. Random-Walk (19.10%), we concluded that M2IC tended to misclassify peaks from gaits as from legitimate touch inputs. On the contrary, WearAuth obtained less error rates of 0∼3.60% for the legitimate cases and 0∼0.29% for the random attacks. Second, M2IC could not detect the mimic attacks well. M2IC obtained 5.91∼21.34% of FPRs for the mimic attacks which were worse than WearAuth's results of 0.49∼1.69%.

We believed that the bad results of M2IC were due to its simple feature design. M2IC used 12 standard statistics for each of accelerometer and gyroscope magnitudes of the

wristwear. Thus, the feature vector lacks of information about axis-specific movements, and it can only determine whether the magnitudes of motion signals change in short time of touch interactions. In addition, since the M2IC's feature vector is computed from raw motion data, time-sensitive motion information highlighted at high-frequency cannot be obtained. As a result, the evaluation of M2IC with the data of gait and mimic attacks showed its weaknesses.

With the results of the mimic attacks, we address the additional findings as follows: 1) the strategy of the opportunistic attacks gave improvement of the attack success rate to the subjects (14.18% for Mimic-Phone-Naive-All vs. 21.34% for Mimic-Phone-Opp-All and 15.56% for Mimic-PC-Opp-All), 2) exploiting the victim's wrist movements of using a PC gave the subjects opportunities to defeat M2IC (15.56% for Mimic-PC-Opp-All and 8.11% for Mimic-PC-Opp-Keyboard), 3) like WearAuth, mimicking longer interactions gave better opportunities to the subjects (21.34% for Mimic-Phone-Opp-All vs. 5.91% for Mimic-Phone-Opp-Tap, and 15.56% for Mimic-PC-Opp-All vs. 8.11% for Mimic-PC-Opp-Keyboard).

6. Discussions

6.1 Inapplicable Hand Condition

WearAuth is inapplicable only when the two devices are completely irrelevant to each other. As described in Sect. 3.2, we have assumed that the wristwear-worn hand either holds the smartphone or makes touches on it, for applying WearAuth. Thus, the case where the wristwear-free hand both holds and makes touches on the smartphone is excluded. According to [21], 49% of smartphone use is performed by one-handed. Since the hand can be either wristwear-worn hand or wristwear-free hand, about 25% of the smartphone use can be inapplicable. WearAuth may ask users to either hold or touch with the wristwear-worn hand when authentication is required, or combining WearAuth with other approaches such as using motion-dependent variations of radio signal strength [22] can be applicable for the case, and we will continue to study to solve this issue.

6.2 Efficiency of the DFT-Based Approximate Correlation

The DFT-based approximate correlation helps reduce the communication overhead. In our setting shown in Table 4, we compute 21 correlation coefficients with 9 motion segments, where each segment contains 800 data samples. For the approximate correlation, we use 30 DFT coefficients. Without the approximate method, the wristwear needs to send the entire motion segments to the smartphone periodically. Naively speaking, the data transmission can be reduced from 7200 (9×800) to 270 (9×30) with the approximate method. The number of computations of the Euclidean distance is also decreased from 16800 (21×800) to 630 (21×30). However the approximate method requires additional computations of 30 DFT coefficients of both devices.

The further evaluation is needed, because 1) the power consumption for the communication and computation are different, 2) the overhead highly depends on the parameters such as the sampling rate, segment size, and number of DFT coefficients used, and 3) the above mentioned approach can be more optimized; for example, if we set a minimum threshold (ρ_{min}) for the similarity measure and the approximate correlation computed with the fewer DFT coefficients (e.g., 5) is smaller than the threshold, we do not need further DFT computations because $corr_{30} \leq corr_5 \leq \rho_{min}$.

6.3 Optimization

Since WearAuth operates in mobile devices, its optimization should be considered. For example, if we decrease sampling rate or motion segment size, the overall power consumption decreases but the performance would also decrease. Similarly, if we tune WearAuth to use less number of features, there would be the same trade-offs.

We tested the performance of our proposed features of Move, Similarity and Touch. Since it was hard to evaluate WearAuth with all possible combinations of features, we additionally trained feature-specific classifiers and evaluated them separately by adjusting the number of used features. The positive and negative datasets used for the models were same with the ones used to compute KS statistics addressed in Sect. 5.3.1. For move features, we chose K_m features from each device in order of magnitude of KS statistics shown in Table 5. We selected the K_s largest absolute values of approximate correlation coefficients for similarity features, and the K_t largest PHs and its PLs for touch features as described in Sect. 4.3.3 and 4.3.4.

Table 9 shows the error rates of the individual classifiers with the varying number of used features. As expected, the performance of each classifier degrades with the less number of features. We found that 1) the classifier using features of higher KS statistics obtained better performance; for example, the classifier of move features was better than the classifier of touch features, 2) even a single approximate correlation coefficient can work in the same-hand state with error rate of 1.97%, and 3) though the less number of touch features of WearAuth outperformed the M2IC's 24 features, the touch features need to cooperate with others for better performance in the different-hand state.

Table 9 The error rates of feature-specific classifiers with varying number of used features. All classifiers were trained using AdaBoost like the original model of WearAuth.

Classifier using Move Features						
$K_m = 1$	$K_m = 2$	$K_m = 3$	$K_m = 4$			
0.77%	0.56%	0.37%	0.29%			
Classifier using Similarity Features						
$K_s = 1$	$K_s = 3$	$K_s = 5$	$K_s = 7$	$K_s = 9$	$K_s = 11$	$K_s = 13$
1.97%	1.43%	1.36%	1.35%	1.35%	1.32%	1.31%
Classifier using Touch Features						
$K_t = 1$	$K_t = 2$	$K_t = 3$	$K_t = 4$	$K_t = 5$	$K_t = 6$	$K_t = 7$
12.67%	10.95%	9.17%	8.66%	8.52%	8.49%	8.41%

Though we empirically evaluated WearAuth in Sect. 5.3.2, we cannot simply choose the proper number of each type of features because the overall model, which uses all types of features, may work differently with different number of features. In addition, a user of WearAuth may require different degree of power consumption and performance of security and usability with different applications. Thus, we leave the further tuning of WearAuth as the future work.

6.4 Users in a Vehicle

Though we have considered the four hand grips and two moving states, we have not evaluated the usage cases of users being in a vehicle. If a user is in a car, bus, or subway, the external factors from the vehicle would affect the motion sensors of both devices. We believe that the motion incurred from the vehicle would not contain much of short-time and high-frequency components, thus MODWT-MRA can still extract the desired touch-specific movements. In addition, both devices may experience similar motions from the vehicle, and WearAuth can exploit the similarity to determine the same-ownership. We leave the further evaluation for the future work.

7. Related Work

In WearAuth, a wristwear works as a security token which provides the proximity of a user to her smartphone. Instead of using a simple network connection between the two devices, WearAuth determines the proximity by leveraging naturally occurring wrist movements during smartphone usage. In this section, we first address the prior works of conventional proximity-based approaches and their drawbacks. Thereafter, we point out the major differences of WearAuth from the following two prior approaches of wristwear-assisted authentication: one introduces the bilateral authentication—comparing different types of data, i.e., input and motion, of the two devices; the other utilizes each user's distinct motion traits recorded in both devices.

7.1 Proximity-Based Authentication

Proximity-based automatic authentication has inspired a long list of works. Unlike the biometrics that rely on person's inherent traits (e.g., fingerprint, face, voice or iris) or unique behavioral dynamics (e.g., gait [23], keystroke [24], and touch [25]), the proximity-based approach usually relies on the instant data that always changes. To verify the proximity of two or multiple devices, the following types of information have been widely used: network connection, motion, and ambience.

Recently, the network-based methods have been mounted on the COTS devices. Google's Smart Lock [26] and Apple Watch's Auto Unlock [27] provide the zero-effort authentication for smartphones and laptops based on the wireless connection(s) between a target device and a trusted device. Nonetheless, this approach implies a potential risk of not being able to directly identify users—anyone who stays within the coverage range can use automatically unlocked device.

The motion-based approach compares the motion signals simultaneously collected by multiple devices. The motion includes the gait [28], hand shaking [29], and finger movements [30]. WearAuth differs from these prior works as it does not require user intervention, specific actions, or specific positions between devices for authentication.

The ambience-based approach verifies the co-presence of devices without user intervention. Its convenience is particularly useful in situations where many of small devices such as Internet of Things are installed. The popular sources are audio [31], [32] and radio signals [33], and multiple sensor modalities [34]. Although the proximity range can be adjusted, the ambient-based approach still lacks a way to directly identify users.

7.2 Wristwear-Assisted Authentication

Mare et al. [2] introduced a zero-effort bilateral authentication scheme called ZEBRA. Instead of comparing the same type of data, ZEBRA compares terminal inputs with wrist motions that occur naturally in typing on a keyboard, scrolling a mouse, and switching between the keyboard and mouse. ZEBRA continuously verifies a user and automatically deauthenticates the user if terminal inputs are not correlated with the user's wrist motions. Nonetheless, ZEBRA has been successfully attacked by opportunistic attacks [9] that can evade detection by mimicking only a portion of easy-to-mimic interactions.

ZEBRA was later extended to CSAW [1] for smartphones. CSAW consists of three components: M2MC, M2IC, and grip detector. M2MC correlates wristwear's motions to smartphone's motions by comparing the four signals (X, Y, Z, and magnitude) from the accelerometer and gyroscope. M2IC is used when M2MC is not reliable, i.e., the different-hand state. It correlates wristwear's motions to smartphone inputs. The grip detector identifies the grip using the wristwear's orientation relative to the smartphone's orientation, which helps choose an appropriate correlator.

Compared with CSAW, WearAuth has the following differences:

- WearAuth additionally extracts features to infer the moving states of the devices. Since the motion data of each devices is highly affected by its moving state, the features are useful for determining the usage cases (Sect. 4.3.1).
- To infer the grip, CSAW assumes the smartphone is horizontal with the ground and computes the difference in Z-axis acceleration between the two devices. We refer to the smartphone's Z-axis acceleration as a_Z, of which the range of the difference can be from 0 (when the two devices face the same direction), to $2a_Z$ (when the two devices face the opposite directions). Thus,

if a_Z keeps changing, which easily happens when the user is moving, it is hard to set a threshold to determine the grip. In addition, when the wristwear's orientation is orthogonal (90 degrees) to the smartphone's orientation, the wristwear's Z-axis is irrelevant to the smartphone's Z-axis. Thus, the difference in Z-axis acceleration cannot help to infer the grip. To handle the problem, WearAuth computes the inner angle, which is the direct measure of the relative device orientation, between Z-axes of the two devices in the ECS (Sect. 4.3.2).

- WearAuth reduces the communication overhead by using the DFT-based approximate correlation. Moreover, to measure the motion similarity of the two devices, CSAW compares the motion signals of the same axis pairs only, e.g., X-to-X or Y-to-Y, while WearAuth compares all possible pairs of the axes, e.g., X-to-Y or Z-to-X. This is because the associated axis pairs of the two devices depend on the relative device orientation (Sect. 4.3.3).
- With the help of the MODWT-MRA, WearAuth can highlight the touch-specific movements regardless of the moving state. CSAW's M2IC is applicable to the case where the user is stationary, only. (Sect. 4.3.4).

Recently, implicit continuous authentication for smartphones called iAuth [35] which leverages motions recorded in the smartphone and wearable device was introduced, and was later extended to SmarterYou [36] with context-awareness. Their schemes are based on the idea that the behavioral patterns collected in both devices are distinct from each person and each moving context. Compared to iAuth and SmarterYou, WearAuth has the following differences:

- WearAuth does not depend on user-specific behavioral traits. Instead, our scheme leverages motion similarity of the two devices and touch-specific movements of the wristwear which occur naturally for most people. Accordingly, model retraining is not necessary in WearAuth.
- WearAuth additionally considers the context of relative positions of the two devices, i.e., the same-hand state or different-hand state, combined with the moving state. The wristwear exhibits different motion patterns with different contexts of the hand states and moving states.

8. Conclusion

In this paper, we propose a wristwear-assisted user authentication method for smartphones. Our method adopts MODWT-MRA to decompose the motion signals into the smooth and detail components, and analyzes them separately. To verify the same-ownership of the two devices, WearAuth extracts the four types of features. The move and grip features are used to infer the four usage cases of Sit-Same, Sit-Diff, Walk-Same, and Walk-Diff. The similarity features measure how much the wristwear's motion is similar to the smartphone's motion, where the DFT-based approximate correlation is used for reducing the communication overhead. To assess the touch specific movement, the touch features are extracted from the peaks highlighted at the high-frequency components of the wristwear's motions.

We have evaluated the effectiveness of the features with the two sample KS test, and evaluated the performance of security and usability of WearAuth by conducting two experiments with 50 subjects. The classifier showing the best performance obtained the EER, AUC, and F1 score of 0.69%, 0.9997, and 0.9901, respectively. Specifically, WearAuth shows FNRs of 3.6% or less for the four legitimate usage cases, and FPRs of 1.69% or less for the seven attack cases. The results imply that our proposed features are effective, and WearAuth properly operates regardless of the moving and hand states and is robust to the sophisticated attacks.

References

[1] S. Mare, "Seamless Authentication for Ubiquitous Devices," Tech. Rep. TR2016-793, Dartmouth College, Computer Science, Hanover, NH, May 2016.
[2] S. Mare, A.M. Markham, C. Cornelius, R. Peterson, and D. Kotz, "ZEBRA: Zero-effort bilateral recurring authentication," IEEE Symposium on Security and Privacy, pp.705–720, May 2014.
[3] D.P. Percival, "On estimation of the wavelet variance," Biometrika, vol.82, no.3, pp.619–631, 1995.
[4] D.B. Percival and H.O. Mofjeld, "Analysis of subtidal coastal sea level fluctuations using wavelets," Journal of the American Statistical Association, vol.92, no.439, pp.868–880, 1997.
[5] A. Mueen, S. Nath, and J. Liu, "Fast approximate correlation for massive time-series data," Proc. 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10, pp.171–182, 2010.
[6] D.B. Percival and A.T. Walden, Wavelet Methods for Time Series Analysis, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 2000.
[7] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, 2009.
[8] I. Daubechies, Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics, 1992.
[9] O. Huhta, P. Shrestha, S. Udar, M. Juuti, N. Saxena, and N. Asokan, "Pitfalls in designing zero-effort deauthentication: Opportunistic human observation attacks," The Network and Distributed System Security Symposium, 2016.
[10] E. Sejdić, K.A. Lowry, J. Bellanca, M.S. Redfern, and J.S. Brach, "A Comprehensive Assessment of Gait Accelerometry Signals in Time, Frequency and Time-Frequency Domains," IEEE Trans. Neural Syst. Rehabil. Eng., vol.22, no.3, pp.603–612, 2014.
[11] M. Kojima, S. Obuchi, O. Henmi, and N. Ikeda, "Comparison of Smoothness during Gait between Community Dwelling Elderly Fallers and Non-Fallers Using Power Spectrum Entropy of Acceleration Time-Series," Journal of Physical Therapy Science, vol.20, no.4, pp.243–248, 2008.
[12] L. Breiman, J. Friedman, C. Stone, and R. Olshen, Classification and Regression Trees, The Wadsworth and Brooks-Cole statistics-probability series, Taylor & Francis, 1984.
[13] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer-Verlag, New York, 2009.
[14] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods, Cambridge University Press, New York, NY, USA, 2000.
[15] L. Breiman, "Random forests," Machine Learning, vol.45, no.1, pp.5–32, Oct. 2001.

[16] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol.55, no.1, pp.119–139, 1997.
[17] NTP: The Network Time Protocol, http://www.ntp.org/.
[18] D. Bichler, G. Stromberg, M. Huemer, and M. Löw, "Key generation based on acceleration data of shaking processes," Proc. 9th International Conference of Ubiquitous Computing, pp.304–317, 2007.
[19] A. Ivanov and G. Riccardi, "Kolmogorov-smirnov test for feature selection in emotion recognition from speech," 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.5125–5128, March 2012.
[20] MathWorks, MATLAB, https://www.mathworks.com/products/matlab.html.
[21] "How Do Users Really Hold Mobile Devices," https://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php.
[22] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang, "Proximity based iot device authentication," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp.1–9, May 2017.
[23] Y. Ren, Y. Chen, M.C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," IEEE Trans. Mobile Comput., vol.14, no.9, pp.1961–1974, Sept. 2015.
[24] D. Buschek, A. De Luca, and F. Alt, "Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices," Proc. 33rd Annual ACM Conference on Human Factors in Computing Systems, pp.1393–1402, 2015.
[25] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," The Network and Distributed System Security Symposium, 2013.
[26] Google: Smart lock, https://get.google.com/smartlock/.
[27] "How to unlock your mac with your apple watch," https://support.apple.com/en-us/HT206995.
[28] W. Xu, C. Javali, G. Revadigar, C. Luo, N. Bergmann, and W. Hu, "Gait-key: A gait-based shared secret key generation protocol for wearable devices," ACM Trans. Sen. Netw., vol.13, no.1, pp.6:1–6:27, Jan. 2017.
[29] R. Mayrhofer and H. Gellersen, "Shake well before use: Intuitive and secure pairing of mobile devices," IEEE Trans. Mobile Comput., vol.8, no.6, pp.792–806, June 2009.
[30] L. Li, X. Zhao, and G. Xue, "A proximity authentication system for smartphones," IEEE Transactions on Dependable and Secure Computing, vol.13, no.6, pp.605–616, Nov. 2016.
[31] N. Karapanos, C. Marforio, C. Soriente, and S. Čapkun, "Sound-proof: Usable two-factor authentication based on ambient sound," 24th USENIX Security Symposium, pp.483–498, 2015.
[32] T. Li, Y. Chen, J. Sun, X. Jin, and Y. Zhang, "iLock: Immediate and automatic locking of mobile devices against data theft," Proc. ACM SIGSAC Conference on Computer and Communications Security, pp.933–944, 2016.
[33] A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, and A. LaMarca, "Ensemble: Cooperative proximity-based authentication," Proc. 8th International Conference on Mobile Systems, Applications, and Services, pp.331–344, 2010.
[34] M. Miettinen, N. Asokan, T.D. Nguyen, A.-R. Sadeghi, and M. Sobhani, "Context-based zero-interaction pairing and key evolution for advanced personal devices," Proc. ACM SIGSAC Conference on Computer and Communications Security, pp.880–891, 2014.
[35] W.-H. Lee and R. Lee, "Implicit sensor-based authentication of smartphone users with smartwatch," Proc. Hardware and Architectural Support for Security and Privacy 2016, pp.9:1–9:8, 2016.
[36] W.-H. Lee and R.B. Lee, "Sensor-based implicit authentication of smartphone users," 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp.309–320, June 2017.

Taeho Kang received the BS degree from Pohang University of Science and Technology (POSTECH), Korea. He is currently working toward the PhD degree at Computer Science and Engineering, POSTECH. His research interests include mobile authentication, network security, data mining and machine learning.

Sangwoo Ji received the BS degree in computer science and engineering from the Pohang University of Science and Technology (POSTECH), Korea, in 2015. He is working toward the PhD degree in computer science and engineering at POSTECH. His research interests include usable security, authentication, system security, and hardware security.

Hayoung Jeong received the BS degree in computer science and electrical engineering from Handong University, Korea (2016) and received the MS degree in computer science and engineering from Pohang University of Science and Technology (POSTECH). Her main research work is in the area of behavioral biometrics and privacy.

Bin Zhu received the B.S. degree in physics from the University of Science and Technology of China, Hefei, China, and the M.S. and Ph.D. degrees in electrical engineering from the University of Minnesota, Minneapolis, MN, in 1986, 1993, and 1998, respectively. From 1997 to 2001, he was a Lead Scientist with Cognicity, Inc., a startup company he cofounded with his dissertation Advisor and a Colleague in 1997. Since 2001, he has been a Researcher with Microsoft Research Asia, Beijing, China. His research interests include Internet and data security, privacy protection, and data and multimedia processing.

Jong Kim received the BS degree in electronic engineering from Hanyang University, Korea, in 1981, the MS degree in computer science from the Korean Advanced Institute of Science and Technology, Korea, in 1983, and the PhD degree in computer engineering from Pennsylvania State University in 1991. He is currently a professor in Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Korea. From 1991 to 1992, he was a research fellow in the Real-Time Computing Laboratory of the Department of Electrical Engineering and Computer Science, University of Michigan. His major areas of interest are fault-tolerant, parallel and distributed computing, and computer security.