A Usable Authentication System using Wrist-worn Photoplethysmography Sensors on Smartwatches

Jiacheng Shang and Jie Wu
Center for Network Computing, Temple University, Philadelphia, PA 19121
Email: {jiacheng.shang, jiewu}@temple.edu

***Abstract*—Smartwatches are expected to become the world's best-selling electronic product after smartphones. Various smartwatches have been released to the private consumer market, but the data on smartwatches is not well protected. In this paper, we show for the first time that photoplethysmography (PPG) signals influenced by hand gestures can be used to authenticate users on smartwatches. The insight is that muscle and tendon movements caused by hand gestures compress the arterial geometry with different degrees, which has a significant impact on the blood flow. Based on this insight, novel approaches are proposed to detect the starting point and ending point of the hand gesture from raw PPG signals and determine if these PPG signals are from a normal user or an attacker. Different from existing solutions, our approach leverages the PPG sensors that are available on most smartwatches and does not need to collect training data from attackers. Also, our system can be used in more general scenarios wherever users can perform hand gestures and is robust against shoulder surfing attacks. We conduct various experiments to evaluate the performance of our system and show that our system achieves an average authentication accuracy of 96.31% and an average true rejection rate of at least 91.64% against two types of attacks.**

***Index Terms*—Authentication, mobile sensing, data security.**

I. Introduction

Smartwatches are expected to become the world's best-selling electronic product after smartphones. Various smartwatches have been released to the private consumer market, such as Apple Watch [1] and Samsung Gear [2]. Those devices usually have capabilities of collecting data via smart sensors, processing sensor data, and exchanging data through wireless links. Smartwatches collect various forms of personal information, such as name, address, date of birth, messages, emails, and other health information whose exposure would be a cause for concern. However, the data on smartwatches is not well protected. According to a recent study on the top 10 popular smartwatches in the market, only 50% of tested devices offer the ability to implement screen lock using either PIN or pattern [3]. For example, Fitbit devices lack the authentication mechanism. Samsung smartwatches provide a PIN-based authentication mechanism, but it is not convenient to use due to limited screen size and is vulnerable to shoulder surfing attacks. Based on a recent report by Shaw, 56% of wearable owners access business data via applications, and 42% of them rank identity theft as their top security concern [4]. Therefore, it is essential to propose an authentication system for smartwatches that is convenient to use and provides high data security.

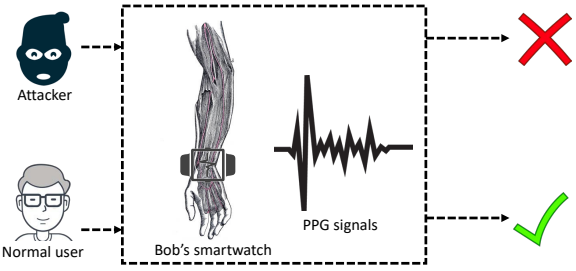


Fig. 1. System authenticates a normal user and defends against an attacker.

On the other hand, we cannot easily migrate existing security technologies from smartphones such as fingerprint and face recognition, due to the small size and battery constraints. Existing solutions on commercial smartwatches are mainly based on PIN-based passwords, pattern-based passwords, and audio. However, PIN and pattern-based methods suffer from shoulder surfing attacks, and users tend to disable PIN-based and pattern-based passwords on smartwatches since it is not convenient to input on a small touchscreen. Audio-based authentication can also be broken by replay attacks. To address this problem, researchers proposed various authentication systems on smartwatches by leveraging the characteristics in electrocardiogram (ECG) signals, fingertip photoplethysmography (PPG) signals, and hand gestures. However, ECG sensors are unavailable in smartwatches. The approaches using fingertips PPG signals rely on low-noise PPG measurements on fingertips, which is hard to ensure on smartwatches. Although motion sensors in wrist-worn devices show their potential in both hand gesture recognition and authentication, they are sensitive to body movement and cannot be used for users who are moving. Also, most existing authentication systems for wrist-worn devices is to identify one user from a group of users. Therefore, they need to collect training data from all users, which is not practical for personal wrist-worn devices. Moreover, all existing systems rely on computation-intensive classification methods (e.g. Gradient Boosting Tree and Neural Network)

Considering the limitations of existing solutions, we propose a usable authentication system using PPG signals influenced by hand gestures to protect the private data and operations on smartwatches, as shown in Fig. 1. Our solutions are designed based on the following facts: 1). Human hand gestures introduce blood volume changes in the blood flow; 2). The changes are different for different people even if they perform

the same hand gesture. Different from existing solutions, our approach leverages the PPG sensors that are available on most smartwatches and does not need to collect training data from attackers. Also, our system can be used in more general scenarios wherever users can perform hand gestures and is robust against look over the shoulder. Moreover, our classification method is low-cost and only collects the training data from the owner of the smartwatch. In order to successfully authenticate the normal user and defend against attacks, we leverage the unique features in PPG signals influenced by users' hand gestures. While using our system, the user is asked to perform a pre-selected hand gesture. Our system detects the hand gesture from raw PPG signals and matches it with those collected in the enrollment phase. If the similarity between the newly detected hand gesture and training data exceeds a threshold, the user is authenticated and allowed to access protected data.

We summarize our contributions as follows:

- Our solution is software-only and can be integrated into any smartwatch with PPG sensors for multi-factor authentication.
- Our research results serve as a feasibility assessment of using PPG signals influenced by hand gestures for authentication.
- Our classification model is trained only based on the data collected from the normal user to make the authentication decision, which makes our system more practical.
- We develop a prototype on Samsung gear smartwatches and conduct comprehensive evaluations. Experimental results show that our system achieves average authentication accuracy of 96.31% and an average true rejection rate of at least 91.64% against two types of attacks.

II. RELATED WORK

A. Authentication systems on smartwatches

PIN-based and voice-based. Existing commercial smartwatches (e,g, Apple watch and Samsung Gear) adopt the same solutions based on PINs and patterns that are used on the smartphone. However, these solutions suffer from brute force attacks [5], [6]. For example, Nguyen et al. [5] found that the error rate of the regular PIN on smartwatches is more than twice as high as that of smartphones. Inspired by the success of voiceprint-based authentication systems on smartphones, researchers have leveraged audio signals to authenticate users on smart wearables [7], [8], [9], [10]. However, audio-based systems suffer from replay attacks, and existing liveness detection systems are hard to be deployed on smartwatches.

Activity and gesture-based. To address the limitations of existing approaches, researchers proposed various systems to authenticate users on smart wearables by leveraging biometrics in hand gestures [11], [12], [13] and daily activities [14], [15], [16], [17], which are believed to be unique. For instance, Shrestha et al. [14] proposed an authentication system using the built-in motion sensors and magnetic sensor of smart wearables. Similarly, Johnston et al. showed the feasibility of using gait-based biometrics to identify and authenticate each user [15]. However, users cannot successfully authenticate themselves if they are not doing predefined activities, which greatly limits the usage scenarios of activity-based authentication systems. Researchers have also shown that it is possible to distinguish between users who perform the same gesture [12], [13]. For example, Lewis et al. [12] presented a gesture-based authentication system for smartwatches using motion sensors. However, the motion signals are noisy and can be easily influenced by other human activities, which limits their abilities to improve the system performance.

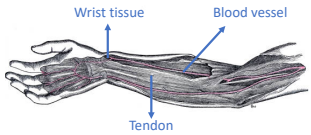


Fig. 2. Illustration of the biological structure of arm and wrist.

ECG-based and PPG-based. In [18], Chun et al. proposed an ECG-based authentication system using short-time Fourier transform. However, the ECG sensors are not widely deployed on current smartwatches. Some authentication systems are designed using hand movement information from wrist-worn PPG sensors [19], [20]. However, their performances rely on low-noise PPG measurements on fingertips, which is hard to be ensured on wrist-worn smart wearables. Kamoi et al. [13] presented a new authentication system using hand gesture information from wrist-worn PPG sensors. However, their system needs to collect training data from all users in order to identify a user from a group of users, which is not practical for personal wrist-worn devices. Moreover, they did not consider strong attackers, and their equal error rate is as high as 11.6%.

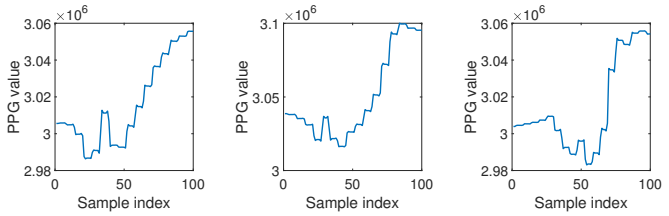
B. Hand gesture detection

Both gesture recognition and gesture-based authentication systems rely on accurate hand gesture detection from raw signals. Existing works for detecting and recognizing hand gestures mainly use microphones [21], cameras [22], radio frequency (RF) [23], ECG sensors [24], and motion sensors [25], [26]. However, camera-based, ECG-based, RF-based, and microphone-based approaches require extra sensors, which makes them hard to be implemented on smartwatches. Motion sensors show great potential on hand gesture recognition on wrists, but they are sensitive to body movement, which greatly limits their usage scenarios. Recently, Zhao et al. [27] presented the potential of PPG sensors for fine-grained hand gesture recognition.

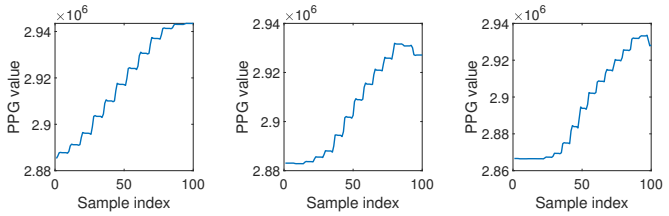
III. PRELIMINARY

A. Wearable photoplethysmography (PPG) sensor

Most smartwatches are equipped with PPG sensors for health monitoring. PPG signals are often obtained by using a pulse oximeter which illuminates the skin and measures changes in light absorption. Therefore, A PPG sensor consists of at least one light-emitting diode (LED) and a photodiode. The light-emitting LED is used to illuminate the skin with the light, and the photodiode is used to measure the amount of light



(a) The 1^{st} trail of u_1 (b) The 2^{nd} trail of u_1 (c) The 3^{rd} trail of u_1



(d) The 1^{st} trail of u_2 (e) The 2^{nd} trail of u_2 (f) The 3^{rd} trail of u_2

Fig. 3. The raw PPG signals of different users and gestures.

that is either transmitted or reflected. In current smartwatches, PPG sensors are mostly used to measure the blood flow changes in the wrist area tissue and calculate the heart rate. Recently, researcher study how to reuse PPG signals for both gesture recognition and authentication. For example, Zhao et al. first showed that PPG signals can be used to detect and recognize finger-level hand gestures [27]. A new authentication system using hand gesture information from wrist-worn PPG sensors was designed in [13]. Although their performances can be further improved, these findings show the feasibility and potential of PPG signals for hand gesture detection and authentication.

B. Key insights

Currently, PPG signals are mainly used to measure the heart rate and monitor blood pressure. Those applications only study the influence of heartbeat and treat the influences of hand gestures as noise. In this work, we show that the PPG signals influenced by hand gestures can be used as a biometric identifier to accurately authenticate users. The key insight is that muscle and tendon movements introduced by hand gestures compress the wrist artery to different degrees, which leads to various changes in blood flow. Since the light is more strongly absorbed by blood than the surrounding tissues and the skin, the changes in blood flow can be detected by PPG sensors as the intensity of light changes. Current PPG sensors in smartwatches use two green LEDs to make sure the PPG sensor can accurately measure the changes in blood flow. To study the feasibility of using PPG signals to authenticate users, we ask 2 users (u_1 and u_2) to perform the same gesture 3 times, and their raw PPG signals are shown in Fig. 3. Since the behavior of the same user is stable, the PPG signals collected from the same user share the same pattern. Also, we can see that the patterns of PPG signals differ a lot for different users in terms of the shape and the volume of light intensity. These facts imply that the PPG signals contain rich information of the identities of different users. Therefore, we can authenticate users using PPG signals while they are performing some hand gesture.

IV. Problem Formulation

A. Attack model

In this paper, we consider a capability-restricted attacker that aims at spoofing the authentication system on the victim's device. The attacker's capabilities are restricted in the sense that: 1). It only has access to the victim's device for a short period (e.g. during lunchtime); 2). It cannot have access to the storage of victim's devices in any other way, so it cannot know the PPG pattern of the victim; 3). The attacker can only rely on the following attack models to imitate the victim's identity:

Random guess attack. In this attack model, the attacker knows the details of the authentication algorithm on the victim's device. To break our system, the attacker performs random hand gestures without knowing how the victim performs the gesture and what the hand gesture the victim picks. This attack method is similar to the brute force attack, but the number of possible hand gestures cannot be enumerated.

Shoulder surfing attack. In this attack model, the attacker has all abilities it has in the random guess attack. Also, the attacker can roughly know what the gesture is by observing the victim over the shoulder and imitate the victim after observation. Ideally, the PPG pattern generated by the shoulder surfers should be more similar to the victim's pattern than that of the random guess attacker.

B. Usage case

Different from continuous authentication, our system is designed for single authentication scenarios. To use our authentication system, the user must wear a smartwatch with the PPG sensor on the left wrist or right wrist. The user needs to ensure that the PPG sensor is closely attached to the wrist during the authentication. The process of our authentication system can be divided into two phases: the enrollment phase and the authentication phase. In the enrollment phase, the user is asked to select a hand gesture and repeat it several times while wearing the wearable devices. The PPG data collected during the enrollment phase is used to build a classifier for future authentication. After enrollment, our system is triggered only when the user wants to authenticate himself/herself. In the authentication phase, the user needs to repeat the hand gesture again. The new PPG signal will be compared with the training data. Once the similarity between the new data and training data exceeds a threshold, the user is claimed to be a normal user. Otherwise, the user will be regarded as an attacker and get rejected by our system.

C. Challenges

To build a highly accurate authentication system using PPG signals influenced by hand gestures on smartwatches, several challenges need to be addressed:

Hand motion detection with noisy PPG signals. In order to extract the identity information from hand gesture, our system needs to detect both the starting point and ending point of the

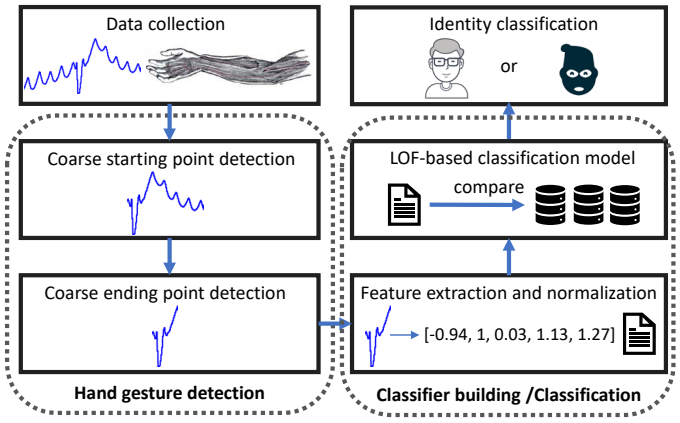


Fig. 4. System pipeline.

hand gestures from raw PPG signals. However, the raw PPG signals are influenced by not only hand gestures but also the heartbeat. Therefore, we need to propose an intelligent hand gesture detection scheme that can robustly work for different users and eliminate the influence of heartbeats. To address this challenge, approaches are proposed to detect the starting point by analyzing the short-time energy of filtered PPG signals and locate the ending point by finding the last peak/valley caused by hand gestures.

Efficient feature extraction scheme. In order to accept the normal user and reject attackers with high accuracy, our system should have the ability to extract efficient features that can uniquely represent the identity of a user. This is challenging since the PPG signals contain much noise. Also, even the same user cannot ensure that they will perform exactly the same hand gesture every time, which introduces small variances to extracted PPG signals. The feature extraction scheme should select proper features that exist over time and uniquely represent the identity of the user. In our system, we study how hand gestures influence the PPG signals and select 5 features in the time domain to describe users' identity information.

Accurate classifier with limited knowledge. In practice, we can only acquire data from the normal user. Therefore, it is challenging to build an accurate classifier since we cannot cover every possible behavior of attackers for all hand gestures. To address this issue, we first normalize the testing data so that the data of attackers act as outliers relative to the training data on the feature hyperplane. Then, we leverage local outlier factor (LOF) [28] algorithm to detect attackers accurately without collecting data from them.

V. Solutions

A. System pipeline

The goal of our system is to leverage the blood flow changes influenced by hand gestures for authentication. To achieve this, we build a system that mainly contains two major phases: the enrollment phase and the authentication phase. The processes of both phases follow the pipeline shown in Fig. 4.

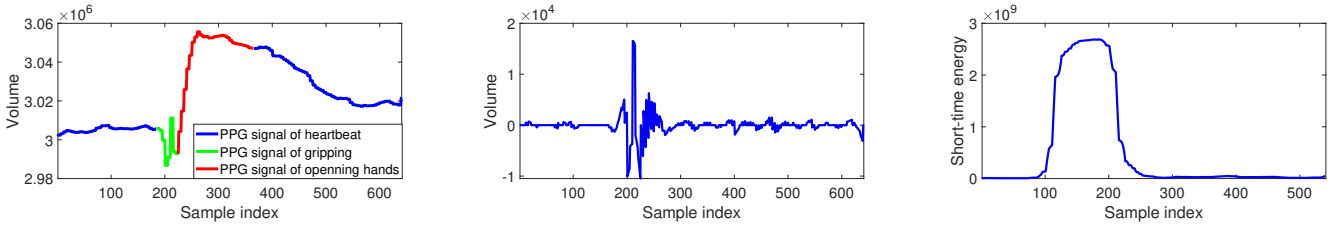
Enrollment phase. In the enrollment phase, the user is asked to select a hand gesture and repeat it several times. Since the user is not able to give the accurate starting time and ending time of the gesture, we first remove the influence of heartbeats by filtering the PPG signals through a high-pass filter and detect the starting point of the hand gesture by studying the short-time energy of filtered PPG signals. After that, we detect the ending point by searching the last peak/valley generated by the hand gesture. The extracted hand gestures are used to extract features that can represent the identity of the user. The extracted features are stored locally on the device and used to verify the identity in the authentication phase.

Authentication phase. After collecting enough training data from the user, the system is ready to be used for authentication. The system can be used by the normal user or an attacker. For each authentication attempt, we first detect and extract the hand gesture in the same way as in the enrollment phase. After that, we extract the same 5 features of the new gesture and normalize it together with all training data. The normalized testing data and training data are sent to the LOF-based model. An attacker is detected if the calculated value of LOF is larger than a threshold.

B. Starting point detection

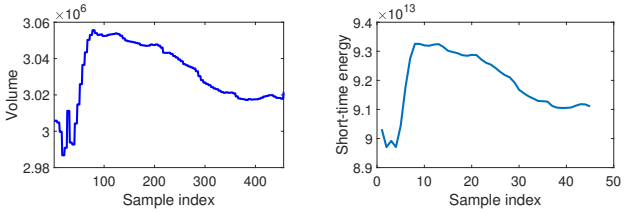
To authenticate users using their behaviors of hand gestures, we need to detect and extract the PPG signals that are influenced by users' hand gestures. However, the PPG signals are constantly under the influence of heartbeat and noise, which makes it difficult to locate the starting point and ending point of each hand gesture on the raw PPG signals. To address this issue, a robust hand gesture detection scheme should be proposed. Fig. 5(a) shows the raw PPG signal when a user performs a gripping-and-opening hand gesture. It is clear that hand gestures will introduce stronger fluctuations compared with the heartbeat. Based on this observation, we propose schemes to filter out the influence of heartbeat on raw PPG signals and detect the starting point of the hand gesture by analyzing the short-time energy of filtered PPG signals.

In order to remove the pulses caused by the heartbeat, we apply a 3-order high-pass butter filter on the raw signals with a cut-off frequency of 2 Hz since human heart rates are mostly under 120 bits per minute. Fig. 5(b) shows the filtered PPG signal. We can see that pulses introduced by hand gestures are much more significant in the filtered PPG signal. To further remove the pulses caused by the heartbeat from filtered PPG signals, we apply a threshold filter on the output of the high-pass filter. The threshold is set as the mean of the high-pass filter's output, excluding the highest 40% and the lowest 40% of the measurements. All measurements whose values are lower than the threshold will be 0 after passing through the threshold filter. However, it is still difficult to detect the starting point of the hand gesture on filtered signals since filtered PPG signals still contain high-frequency noise and we do not have knowledge of the frequency range of the hand gesture. To solve this problem, we apply a moving window to the filtered PPG signals and compute the short-time energy within each window.



(a) The raw PPG signal (b) The output of the high-pass/threshold filter (c) The short-time energy of filtered PPG signal

Fig. 5. Example of detecting the starting point of the hand gesture.



(a) PPG signal from starting point (b) Short-time energy of PPG signal

Fig. 6. Example of detecting the ending point of the hand gesture.

The window size is set to 1 second in our system because it is the average time to perform a hand gesture. If the PPG signal of the hand gesture appears perfectly in the window, the short-time energy of the filtered PPG signal should reach the highest value. Therefore, the starting point of the hand gesture is detected by solving:

$$\arg\max_{s} \quad ([y_s, y_{s+1}, \ldots, y_{s+w}])([y_s, y_{s+1}, \ldots, y_{s+w}])^T \quad (1)$$

where s is the detected starting point of the hand gesture, $Y = [y_1, y_2, \ldots, y_n]$ is the filtered PPG signal, n is the length of filtered PPG signal Y, w is the size of the moving window, and $([y_s, y_{s+1}, \ldots, y_{s+w}])([y_s, y_{s+1}, \ldots, y_{s+w}])^T$ computes the short-time energy of the window starting from the s^{th} sample to the $(s+w)^{th}$ sample. Fig. 5(c) shows the short-time energy of windows starting from different samples. When $s = 381$, the short-time energy reaches the highest value. We use an extra camera to record the whole gesture as the ground truth, and the 381^{th} sample in Fig. 5(a) is closely around the real starting point we get from the video.

C. Ending point detection

Although we can achieve coarse-grained starting point detection, it is still challenging to detect the ending point of the hand gesture based on the short-time energy of each moving window. Since the time to perform the hand gesture is inconsistent even for the same user, it is difficult to know the accurate duration of the hand gesture in advance, and thus we cannot set the window size accurately or detect the ending point by finding the maximal short-time energy. In order to accurately detect the ending point of each hand gesture, we analyze users' behaviors of hand gestures and their influences on PPG signals. For example, Fig. 5(a) shows the raw PPG signals when users perform a gripping-and-opening gesture. A straightforward idea to detect the ending point is to regard the colored part in Fig. 5(a) as a pattern and match the PPG signal from the detected starting point to any possible ending point with the pattern using the dynamic time wrapping (DTW) algorithm. By using DTW-based approach, an ending point is detected if the distance between two sequences is minimal. However, this idea cannot work since different users perform the same gesture in different ways, which leads to different patterns. For example, some users will not perform the gripping gesture at all. In this case, the distance computed by DTW can be quite large, which produces wrong ending point detection.

To solve this problem, we propose a new scheme based on the nature of hand gestures. Hand gestures involve blood volume changes in the microvascular bed of tissue and produce significant peaks or valleys on the PPG signals. Also, the amplitudes of pulses generated by heartbeat are much lower than those generated by hand gestures, which enables us to detect the ending point by finding the last significant pulse or valley. Here, significant pulses and valleys are those pulses and valleys whose heights or depths are higher than the average height of pulses produced by heartbeat. However, we cannot directly apply the peak finding algorithm on the raw PPG signals since high-frequency noise will also produce peaks on the raw PPG signal and influence the peak finding results. In order to remove the influences of high-frequency noise while still maintaining the trend and shape of raw PPG signals, we cut the raw PPG signal from the starting point into several non-overlapped segments with the equal size of λ. The λ must be a positive integer and is set to 10 in our implementation. In our system, we do not adopt the moving-average filter to smooth out the raw PPG signals because the moving-average filter does not ensure that all spikes are removed even if we carefully set the value of window size. Within each segment, we compute the short-time energy of the raw PPG signal.

Fig. 6 shows an example of detecting the ending point. We can see the line chart of short-time energy in 6(b) reserves the trend and shape of the raw PPG signal, and the influences of high-frequency noise are removed. Then, we apply the peak finding algorithm on the waveform of short-time energy to find the last significant peak or valley. Since the duration for a user to perform the defined gesture is no more than 1.5 seconds in most cases, we let l donate the location of detected peak or valley within 1.5 seconds from the detected starting point, and the ending point of the gesture is denoted as $l \times \lambda$.

D. Feature extraction and normalization

After detecting the starting and ending points of the hand gesture, we need to extract features that can be used to uniquely identify a normal user in order to defend against attacks. Since we have no knowledge of other users and attackers, we want to only use strong features that can describe the shape of the

PPG signals. Otherwise, noisy or irrelevant features will greatly hurt the performance of the classification model. Based on our preliminary study, we select the following 5 features to describe a raw PPG signal $X = [x_1, x_2, \ldots, x_n]$:

1. The mean of X, excluding the highest and lowest 20% data values. Since the PPG values vary in a large range, we scale the mean value by $1/10,000$ to reduce the impact of the amplitude.
2. The location of the valley with the lowest amplitude.
3. Peak to peak distance.
4. Number of peaks within 0.2 seconds from the valley with the lowest amplitude.
5. The minimal dynamic time warping distance between a new PPG signal and those in the training dataset. Each signal is normalized to a range from 0 to 1 in order to remove the influence of the amplitude that we already considered in the first feature.

Suppose we extract feature matrix $F = [F_1, F_2, \ldots, F_5]$ of above 5 features from both training and testing data. The feature matrix F is defined as:

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{15} \\ f_{21} & f_{22} & \cdots & f_{25} \\ \vdots & \vdots & \vdots & \vdots \\ f_{d1} & f_{d2} & \cdots & f_{d5} \end{bmatrix} \quad (2)$$

where $f_{i,j}$ is the feature value of the i^{th} PPG signal and the j^{th} feature, $F_j = [f_{1j}, f_{2j}, \ldots, f_{dj}]$ is the vector that contains the j^{th} feature of all training and testing data, and d is the number of feature vectors in F. In order to achieve good classification performance and balance the influence of each feature, we normalize each feature using the z-score. Technically, z-score is a measure of how many standard deviations below or above the population mean a feature value is. For each feature vector $[f_{i1}, f_{i2}, \ldots, f_{i5}]$ of the testing data, its normalized feature value z_{ij} of the j^{th} feature is:

$$z_{ij} = (f_{ij} - mean(F_j))/std(F_j) \quad (3)$$

where $mean(F_j)$ and $std(F_j)$ are the mean and standard deviation of the j^{th} feature, respectively. After normalization, the values of the j^{th} feature are centered to 0. Also, smartwatches are typical single-user systems, where we only have the user's training data while lacking attackers training data in the enrollment phase. Based on this observation, The first $d-1$ rows in F are from the training data of the normal user and the d^{th} row is the feature vector of the testing data. If the testing data is from an attacker, the normalized feature values should be much larger than those of the normal user on at least one feature dimension.

E. User authentication

It is common that the authentication system can only have the user's training data while lacking knowledge of attackers' feature distribution. Therefore, we need to build a strong classifier based on only the data of the normal user. Since the features of the attackers differ from those of the normal user on at least one feature dimension, the feature vector of the attacker can be regarded as the outlier compared to those of the normal user. Therefore, we use the training data of the normal user to build a local outlier factor (LOF) model [28]. Compared with other one-class classification models, LOF model has the following advantages: 1). LOF model has fewer requirements on parameter adjustment compared with support vector data description (SVDD)-based approach, which enables us to quickly build a new classifier for a new user; 2). Due to the local approach, LOF is able to identify outliers in a data set that would not be outliers in another area of the data set. For example, a point at a "small" distance to a very dense cluster is an outlier, while a point within a sparse cluster might exhibit similar distances to its neighbors.

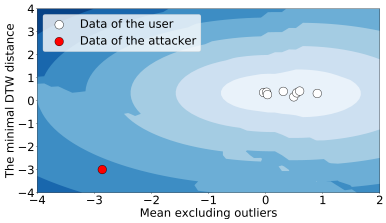


Fig. 7. An example shows how our LOF-based classification works.

Given a normalized feature vector $z = [z_{d1}, z_{d2}, \ldots, z_{d5}]$ of the testing data, LOF determines whether the signal is from an attacker based on comparing the local densities of z and its k-nearest neighbors. The local reachability density (LRD) of a feature vector z is defined as:

$$lrd(z) = 1/(\frac{\sum_{r \in N_k(z)} max\{k - dis(r), d(z,r)\}}{|N_k(z)|}) \quad (4)$$

where $N_k(z)$ are the k nearest neighbors, $k - dis(r)$ is the distance of the object r to the k^{th} nearest neighbor, and $d(z,r)$ is the euclidean distance between feature vectors z and r on the feature hyperplane. The local reachability density of r is then compared with those of the neighbors using

$$LOF_k(z) = \frac{\sum_{r \in N_k(z)} \frac{lrd(r)}{lrd(z)}}{|N_k(z)|} \quad (5)$$

For each authentication attempt on the smartwatches, we only involve the M training instances collected from the normal user during the enrollment phase and a new instance of the current attempt. By selecting a proper k, the value of $LOF_k(z)$ of an attacker should be much larger than those of the normal user. Moreover, if $LOF_k(z)$ is approximately equal to 1, it indicates that the object z is located in the same cluster of the normal user's data so that the new PPG signal is from the normal user. If $LOF_k(z)$ is much larger than 1, it means the new object should not be classified in the same cluster of the normal user's data and the PPG signal is generated by an attacker. Our system determines whether the signal is generated by the normal user by setting a threshold τ. If the value of $LOF_k(z)$ is larger than τ, an attacker is claimed to be detected. Otherwise, the user is authenticated to access protected data.

Fig. 7 shows an example of how our authentication system detects the attacker on a 2-D feature hyperplane. The white nodes represent the data collected from the normal user, and

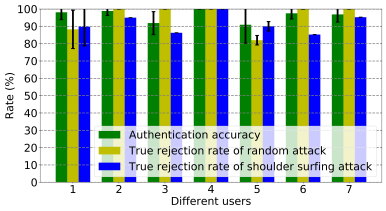


Fig. 8. Overall performance.

the red node is the data produced by the attacker. Also, all white nodes appear around 0 for each feature, while the node of the attacker is away from 0 on at least one feature dimension. The darkness of the blue background represents the level of LOF values relative to the nearest 4 white nodes. The darker the background is, the higher the LOF values are. It is clear that the node of the attacker has a much higher LOF value than any node of the normal user. If we use the counter of the lightest blue area as the decision boundary (or threshold τ), the attacker is successfully detected.

VI. EVALUATION

A. Experimental Methodology

To evaluate and validate the effectiveness of our system, we built a prototype implemented on the Samsung Gear 3 smartwatch running Tizen OS 3.0. The Samsung Gear 3 smartwatches that are equipped with two PPG sensors can record the raw PPG signals with a sampling rate of 100 Hz. In order to help volunteers to collect data, we built a simple graphical user interface (GUI) on the smartwatches. The raw data was stored locally on the smartwatches. For data analysis and processing, the data was then transmitted to a desktop computer with Intel(R) Core(TM) Devils Canyon Quad-Core i7-8700K @ 4.00 GHz CPU and 16 GB of RAM. Our experiments involved 12 volunteers where 7 of them act as normal users. Each normal user was asked to select a hand gesture they prefer and repeat it 30 times. For each normal user, we asked 5 volunteers of the remaining 11 volunteers to act as the shoulder surfers and the other 4 volunteers perform random guess attacks. For data collection, each volunteer picked a preferred posture and room in the same building.

B. Evaluation Metrics

To evaluate the performance of our system, we define four metrics. 1). Authentication accuracy: The probability that a user who is correctly authenticated. 2). False acceptance rate: The probability that a non-registered user is authenticated as a registered user. 3) True rejection rate: The probability that an attacker is rejected. 4). False rejection rate: The probability that a user is authenticated as an attacker. 5). Equal error rate: The rate at which the acceptance and rejection errors are identical. To measure the EER for each test round, we vary the value of the decision threshold of each verification component.

C. Authentication performance for normal users

We first evaluate the authentication accuracy of our system. Fig. 8 shows the authentication accuracies of 7 different volunteers in our system. For all volunteers, the decision threshold τ was set to 1.5. The average authentication accuracy is 96.31% with an average standard derivation of 7.09%. Moreover, we find that the stability of behaviors is different for volunteers in our experiment. For some users (e.g. user 4), their authentication accuracies were stable since their behaviors did not change too much during the data collection. For the other users (e.g. user 2), their behaviors changed a little during the data collection, which led to lower authentication accuracies than those of users who had more stable behaviors. Even so, our system can authenticate the normal user with an accuracy of about 90% as long as there are enough trials in the training dataset that can reflect changes in behavior.

D. Performance against two types of attacks

A good authentication system should provide not only a high authentication accuracy for normal users but also a high true rejection rate for various attackers. Therefore, we also evaluate our system against two kinds of attacks.

Resistance to random guess attacks. We first evaluate the performance of our system against random guess attacks. For each user, we asked four attackers to randomly guess the victim's hand gesture and use the gesture to break our system. For each random guess attacker, we collected ten trials. We evaluated the true rejection rate of each user for 10 iterations. For each iteration, we randomly selected 12 trials as training data. Fig. 8 shows the true rejection rate of our system against random guess attacks. We can see that our system can provide a mean true rejection rate of 95.89%, which means that our system can accurately detect a random guess attacker.

Resistance to shoulder surfing attacks. Then, we evaluate the performance of our system against random guess attacks. Different from random guess attacks, the attacker can observe how the victim performs the gesture over the shoulder and imitate the victim after observation. Similarly, we evaluate the true rejection rate of our system for ten iterations with randomly picked training data. Fig. 8 shows the true rejection rate of our system against shoulder surfing attack. We can see that our system can provide a mean true rejection rate of 91.64% with a mean standard derivation of 10.64%.

We also notice that the system performance of the fifth volunteer is slightly lower than those of others. The reason is that the behavior of the fifth volunteer changed more significantly than others. If the behavior changes significantly, the training data will be spread out in a larger area and have a low density, which causes that the local reachability density of the attacker's data to be smaller. If we use the default threshold for the user who may change his/her behavior, some data of the attacker can be regarded as coming from the victim. But even so, our system can still provide a true rejection rate of at least 83%, which means the attacker needs at least 5 attempts to break the system, on average. By setting the maximal number of attempts to a small value (e.g. 3), the security level of users whose behavior is less stable can be further improved.

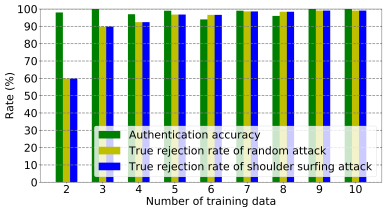


Fig. 9. System performance with different number of training data.

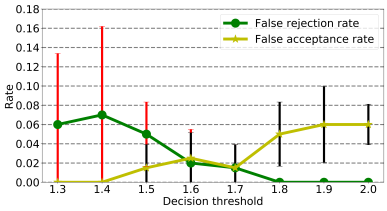


Fig. 10. System performance with different decision thresholds.

E. Impact of training set size

The size of the initial training set is also important. For authentication systems on personal wearables, we expect the size of the initial training set to be as small as possible so that we can reduce the time and cost of the enrollment phase. At the same time, the training data should be sufficient enough to reflect the behavior of the user and train an accurate classifier. To evaluate the impact of training set size, we evaluate the average authentication accuracies and true rejection rates of 3 volunteers we randomly picked, where the decision threshold τ was set to 1.5 and the number of neighbors was set to half of the training set size. For each user, we randomly selected trials as training data and repeated the experiment 5 times, and the results are shown in Fig. 9. We can see that the size of the training dataset does not influence the authentication accuracy a lot since users' behaviors are stable and a small amount of data is sufficient enough to build a robust classifier. Moreover, an attacker is more easily to be detected with more knowledge of the user. We can see that the true rejection rates against two types of attacks rise with the increase in the size of the training dataset. When the size of the training data is 5, our system can provide an authentication accuracy of 99% and a true rejection rate of at least 96.8%, which shows that our system can quickly train an accurate classifier on the new user's device with only a small amount of training data.

F. Impact of decision threshold

There is a trade-off in determining the decision threshold for LOF classifiers. If the decision threshold τ is too large, the user can be accurately authenticated, but more attackers will be wrongly recognized as the user. If the decision threshold is too small, the system can reject almost all attackers, but the normal user may also be rejected, which is unacceptable. To understand how the decision threshold influences the overall performance, we adjusted the value of τ and studied its impact on the system performance. For each value of τ, we randomly selected 9 trials from three volunteers as training data and repeated the experiment for 10 times, and the k was set to 4 for the LOF classifier. As shown in Fig. 10, with τ increasing, the false rejection rate drops, and the false acceptance rate rises. When τ is between 1.5 and 1.6, the EER is about 2.3%.

G. Authentication time

In our experiments, the authentication time is defined as how many times a user has to perform the gesture before being authenticated by our system. In general, we want the authentication time to be a small number (≤ 3) for normal users and as large as possible for attackers, so that we can defend against attackers by setting a maximal number of attempts. Therefore, we conducted experiments to evaluate the average authentication time for normal users and two types of attackers. For each role of the user (normal users and attackers), we asked them to use our authentication system 40 times with randomly selected training data and log down the number of attempts. Across all experiments, the number of training trials is 5 and the decision threshold is 1.5. Fig. 11 shows the distribution of authentication time for users and two types of attackers. It is clear that normal users need much fewer attempts to successfully log in the system. More specifically, about 100% of normal users can successfully authenticate themselves within three attempts, and only about 10% of attackers can log into our system with no more than 3 attempts.

H. Summary

Based on our experiments and collected data, we show that PPG signal can be used to authenticate users with a mean accuracy of 96.31%. Also, our system can reject attackers with high accuracy of 91.64% even if the attacker observes the victim's hand gestures over the shoulder. Moreover, we evaluate the impacts of the decision threshold and the size of the training dataset. Experimental results show that our system can provide good authentication accuracy and defend normal users against two types of attacks with only 5 trials of training data. The average number of attempts is 1.08 for normal users, while attackers need an average of 18.65 attempts to break our system. Therefore, we can use the same scheme on smartphones and set the maximal number of attempts as 3.

VII. Discussion

Availability of PPG sensors on smartwatches. Most of the current smartwatches are equipped with a pair of PPG sensors, such as Apple Watch and Samsung Gear. Also, the newest operating system of smartwatches provides APIs or functionalities for developers to fetch raw PPG readings. For example, iOS developers can access HealthKit (where the Apple Watch stores its measurements) and queries for the last 600 heart rate measurements, which is used in an open-project of B. Larso [29]. Tizen OS 3.0 also provides an API to get the raw PPG readings of two green LEDs in real time.

Repeatability on hand gestures in different scenarios. The raw PPG signals may be influenced by various factors. To show the repeatability of hand gestures, we evaluated our system in various scenarios including indoor and outdoor environments. We randomly selected one user and studied

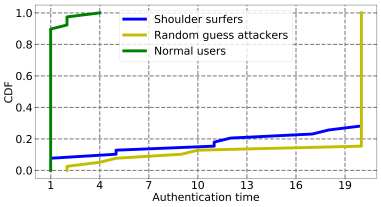


Fig. 11. Authentication times for normal users and attackers.

whether a classifier trained in indoor environment will also work in outdoor scenarios. Evaluation results show that the average authentication accuracy is above 90%, which means that our system is robust to scenario change.

In addition, we studied if the PPG profile of the same user will change on different devices of the same brand. We built the classifier for a random user on one smartwatch and tested the system performance on the other one. Evaluation results show that different devices will influence the final results, which may be caused by minor differences in hardware. This can be solved by re-training a classifier for the user when he/she switches to a new device. Since our system can provide good system performance with only 5 training trials, the cost is limited and acceptable. Moreover, we find the tightness of the watch band will influence the system performance a lot. If the watch band is too loose, there will be a significant gap between the PPG sensors and the wrist, which leads to inaccurate PPG measurements. This will not be a problem in practice since users need to make sure that the PPG sensors touch their skin in order to get accurate health data. Otherwise, the smartwatches cannot provide accurate health data either.

VIII. CONCLUSIONS

This paper presents a novel authentication system on smartwatches. Our solutions leverage the fact that raw PPG signals contain rich information about hand gestures and the user's identity. To defend against attackers, our system first detects the starting and ending point of each new hand gesture and determines if there is an attacker by comparing extracted features from detected hand gestures with those collected from the normal user. The prototype of our authentication solutions achieves high accuracies in both accepting a normal user and rejecting attackers. The experiment results show that our solution is capable of authenticating normal users accurately and defending against two types of attacks on smartwatches. Furthermore, our usable authentication system significantly raises the level of security for existing smartwatches.

ACKNOWLEDGEMENT

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

REFERENCES

[1] Apple watch. [Online]. Available: https://www.apple.com/watch/
[2] Samsung gear. [Online]. Available: https://www.samsung.com/us/explore/gear-s3/
[3] K. Rawlinson, "HP study reveals smartwatches vulnerable to attack," *HP News*, 2015.
[4] R. Shaw. Wearables invade enterprise security. [Online]. Available: https://ssm-nc.com/news/wearables-invade-enterprise-security/
[5] T. Nguyen and N. Memon, "Smartwatches locking methods: A comparative study," in *Proc. of SOUPS*, 2017.
[6] Y. Zhao, Z. Qiu, Y. Yang, W. Li, and M. Fan, "An empirical study of touch-based authentication methods on smartwatches," in *Proc. of ISWC*. ACM, 2017, pp. 122–125.
[7] L. Zhang, S. Tan, and J. Yang, "Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication," in *Proc. of CCS*. ACM, 2017, pp. 57–71.
[8] J. Shang, S. Chen, and J. Wut, "Srvoice: A robust sparse representation-based liveness detection system," in *Proc. of ICPADS*. IEEE, 2018, pp. 291–298.
[9] J. Shang, S. Chen, and J. Wu, "Defending against voice spoofing: A robust software-based liveness detection system," in *Proc. of MASS*. IEEE, 2018, pp. 28–36.
[10] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, Y. Liu, and M. Li, "Lipp ass: Lip reading-based user authentication on smartphones leveraging acoustic signals," in *Proc. of INFOCOM*. IEEE, 2018.
[11] F. Hong, M. Wei, S. You, Y. Feng, and Z. Guo, "Waving authentication: your smartphone authenticate you on motion gesture," in *Proc. CHI*. ACM, 2015, pp. 263–266.
[12] A. Lewis, Y. Li, and M. Xie, "Real time motion-based authentication for smartwatch," in *Proc. of CNS*. IEEE, 2016, pp. 380–381.
[13] T. Ohtsuki and H. Kamoi, "Biometrie authentication using hand movement information from wrist-worn PPG sensors," in *Proc. of PIMRC*. IEEE, 2016, pp. 1–5.
[14] B. Shrestha, M. Mohamed, and N. Saxena, "Walk-unlock: Zero-interaction authentication protected with multi-modal gait biometrics," *arXiv preprint arXiv:1605.00766*, 2016.
[15] A. H. Johnston and G. M. Weiss, "Smartwatch-based biometric gait recognition," in *Proc. of BTAS*. IEEE, 2015, pp. 1–6.
[16] W.-H. Lee and R. Lee, "Implicit sensor-based authentication of smartphone users with smartwatch," in *Proc. of HASP*. ACM, 2016, p. 9.
[17] S. Davidson, D. Smith, C. Yang, and S. Cheah, "Smartwatch user identification as a means of authentication," *Department of Computer Science and Engineering Std*, 2016.
[18] S. Y. Chun, J.-H. Kang, H. Kim, C. Lee, I. Oakley, and S.-P. Kim, "Ecg based user authentication for wearable devices using short time fourier transform," in *Proc. of TSP*. IEEE, 2016, pp. 656–659.
[19] A. Bonissi, R. D. Labati, L. Perico, R. Sassi, F. Scotti, and L. Sparagino, "A preliminary study on continuous authentication methods for photoplethysmographic biometrics," in *Proc. of BIOMS*. IEEE, 2013, pp. 28–33.
[20] P. Spachos, J. Gao, and D. Hatzinakos, "Feasibility study of photoplethysmographic signals for biometric identification," in *Proc. of DSP*. IEEE, 2011, pp. 1–5.
[21] W. Mao, J. He, and L. Qiu, "Cat: high-precision acoustic motion tracking," in *Proc. of MobiCom*. ACM, 2016, pp. 69–81.
[22] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE transactions on multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.
[23] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "Widraw: Enabling hands-free drawing in the air on commodity wifi devices," in *Proc. of MobiCom*. ACM, 2015, pp. 77–89.
[24] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and emg sensors," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.
[25] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, "Mole: Motion leaks through smartwatch sensors," in *Proc. of MobiCom*. ACM, 2015, pp. 155–166.
[26] A. Ferrari, V. Galli, D. Puccinelli, and S. Giordano, "On the usage of smart devices to augment the user interaction with multimedia applications," in *Proc. of WoWMoM*. IEEE, 2017, pp. 1–9.
[27] T. Zhao, J. Liu, Y. Wang, H. Liu, and Y. Chen, "PPG-based finger-level gesture recognition leveraging wearables," in *Proc. of INFOCOM*. IEEE, 2018.
[28] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.
[29] Healthkit. [Online]. Available: https://github.com/BradLarson/HealthKitHeartRateExporter/