User Verification Leveraging Gait Recognition for Smartphone Enabled Mobile Healthcare Systems

Yanzhi Ren, *Student Member, IEEE*, Yingying Chen, *Senior Member, IEEE*,
Mooi Choo Chuah, *Senior Member, IEEE*, and Jie Yang, *Member, IEEE*

Abstract—The rapid deployment of sensing technology in smartphones and the explosion of their usage in people's daily lives provide users with the ability to collectively sense the world. This leads to a growing trend of mobile healthcare systems utilizing sensing data collected from smartphones with/without additional external sensors to analyze and understand people's physical and mental states. However, such healthcare systems are vulnerable to user spoofing, in which an adversary distributes his registered device to other users such that data collected from these users can be claimed as his own to obtain more healthcare benefits and undermine the successful operation of mobile healthcare systems. Existing mitigation approaches either only rely on a secret PIN number (which can not deal with colluded attacks) or require an explicit user action for verification. In this paper, we propose a user verification system leveraging unique gait patterns derived from acceleration readings to detect possible user spoofing in mobile healthcare systems. Our framework exploits the readily available accelerometers embedded within smartphones for user verification. Specifically, our user spoofing mitigation framework (which consists of three components, namely Step Cycle Identification, Step Cycle Interpolation, and Similarity Comparison) is used to extract gait patterns from run-time accelerometer measurements to perform robust user verification under various walking speeds. We show that our framework can be implemented in two ways: user-centric and server-centric, and it is robust to not only random but also mimic attacks. Our extensive experiments using over 3,000 smartphone-based traces with mobile phones placed on different body positions confirm the effectiveness of the proposed framework with users walking at various speeds. This strongly indicates the feasibility of using smartphone based low grade accelerometer to conduct gait recognition and facilitate effective user verification without active user cooperation.

Index Terms—User verification, smartphone, mobile healthcare systems, gait recognition

1 Introduction

Smart devices (e.g., Smart phone, PDAs, tablets, etc.) have become increasingly popular and are playing significant roles in our daily lives. In particular, with sensors that can be easily attached to smartphones and the plurality of sensors embedded within smartphones, the collected sensing data can be mined for the understanding of people's physical and mental health states. For example, barometer sensor can be attached to smartphones equipped with accelerometer and microphone to collect sensing data, which can be mined to uncover people's daily life activities [1]. Information about users' daily life activities and behaviors can further assist in the development of various emerging applications in the healthcare domain. For instance, walking activities and conversations extracted from collected sensor data can be used to predict users' physical and mental conditions [1].

However, such healthcare systems are vulnerable to user spoofing, in which an adversary can distribute his registered device to other users such that data collected from these users can be claimed to be his own. By doing so, the adversary can claim potential health benefits that are allocated to people with certain illnesses even though he may not have any illnesses. For instance, in the social community-based mobile healthcare systems for facilitating epidemiology research [2] and disease propagation control [3], an adversary can attract additional vaccine allocation by performing user spoofing and thus undermine the regular operations of such mobile healthcare systems.

Mitigating user spoofing is not an easy task. Most smartphones only offer user verification methods which rely on explicit manual entry of a secret PIN number. This is insufficient as many users only go through such a verification process once when a smartphone is switched on [4]. In addition, verification based on PIN numbers are not applicable to the cases when an adversary collude with other users. Recently, new techniques utilizing biometric characteristics such as fingerprints have been proposed for user verifications. However, fingerprint readers are not available on most smartphones, making it less suitable for mobile healthcare systems. Further, this technique also requires an explicit user action for verification, e.g., putting a finger on the fingerprint reader.

In this work, we exploit users' unique physical traits, which are hard to forge, to mitigate user spoofing in mobile

- *Y. Ren and Y. Chen are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07307. E-mail: {yren2, yingying.chen}@stevens.edu.*
- *M. Chuah is with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015. E-mail: chuah@cse.lehigh.edu.*
- *J. Yang is with the Department of Computer Science, Florida State University, Tallahassee, FL 32306. E-mail: jyang5@fsu.edu.*

Manuscript received 2 Dec. 2013; revised 15 Oct. 2014; accepted 17 Oct. 2014. Date of publication 26 Oct. 2014; date of current version 3 Aug. 2015.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2014.2365185

healthcare systems. Our design goal is to enable user spoofing detection without relying on explicit user cooperation or additional hardware such as a fingerprint reader. The basic idea is to utilize a user's gait pattern because a person's gait is often unique and can serve as a useful discriminator. The presence of user spoofing causes the newly identified gait patterns to be dramatically different from a user's normal gait patterns and hence such attacks can be detected. To the best of knowledge, our work is the first that utilizes gait information to detect user spoofing in mobile healthcare systems.

Specifically, we design a user verification system leveraging gait patterns derived from accelerometer readings. Our framework employs readily available accelerometers embedded within smartphones instead of deploying additional hardware for user verification. While gait recognition via accelerometer sensors have been studied using sensors with high sampling rates (e.g., larger than 100 Hz in [5], [6]), we focus on addressing several unique challenges that one faces when using low grade accelerometers in mobile healthcare systems. First, low grade accelerometers (e.g., those in smartphones) has a lower sampling rate (e.g., lower or equal to 50 Hz), posing possible difficulty in capturing each user's unique gait patterns. Second, users' walking speeds usually vary during the verification process, making it hard to identify step cycles accurately. Third, the user verification process should be able to complete with small number of measurements. Last but not least, an attacker is able to mimic a legitimate user's walking patterns in an attempt to fool the system. Our proposed user verification system should be able to reject all such attempts.

To cope with these challenges, our gait pattern based user verification system consists of three components: *Step Cycle Identification*, *Step Cycle Interpolation*, and *Similarity Comparison*. During Step Cycle Identification, we utilize the fact that a user's gait patterns should be repeatable, and hence walking traces collected from a user should be highly correlated. We thus construct a template for each user's unique gait pattern by identifying the first distinguishable step cycle, and then utilize the high correlation between a user's step cycles to identify other step cycles within a trace. This approach can derive step cycles more accurately than other methods used in previous studies [5], [6], which identify step cycles by identifying local minimas repeatedly within a trace. Our Step Cycle Identification method has the adaptive learning capability to update a user's step cycle template using real-time feedback.

A user's walking speed varies and is determined by many factors such as his/her health conditions, age, gender, environment, and so on. Our goal is to design a scheme that works well irrespective of what speed a user walks at when the accelerometer readings are collected. Our Step Cycle Interpolation component helps to align identified step cycles of different lengths into normalized cycles of fixed length. This interpolation step allows our scheme to perform gait recognition robust to various walking speeds. In our system, a user's walking profile is constructed during a training process. And the system only needs the user to upload one accelerometer trace under any speed at his convenience for user profile construction, without requiring extensive uploading of multiple traces to cover different walking speeds.

Our Similarity Comparison component makes the related mobile healthcare system robust by using the fact that several sub-events embedded within gait patterns can uniquely characterize a user and are hard to imitate. A user may change his/her walking speeds, but the uniqueness embedded in each gait pattern remains unchanged. Our user verification system can be deployed in two ways: user-centric or server-centric. In the user-centric approach, each user stores his user profile in a mobile device (e.g., smartphone), which runs the verification software. To work with the limited computational resources on a mobile device, our user-centric verification approach utilizes a weighted Pearson correlation coefficient (PCC) based method with low computation overhead during the similarity score computation. Whereas in the server-centric approach, the server (e.g. deployed by a healthcare insurance company) conducts user verificattion based on the gait features extracted from accelerometer readings collected within a user's mobile device. At the server side, our similarity comparison scheme is based on the Support Vector Machine (SVM) algorithm [7]. The machine learning based SVM approach yields a higher accuracy compared to the basic correlation coefficient based scheme.

Our techniques can be useful to many healthcare systems utilizing human sensing data. We summarize our contributions as following:

- We design a user verification system by extracting the unique gait patterns of users for mobile healthcare systems.
- We exploit readings from the low-grade accelerometer embedded in a user's smartphone to derive the correlation relationship inherent in the user's walking traces. Our scheme can achieve more robust step cycle identification compared to previous studies even when the user's walking speeds vary.
- We develop several techniques including automatic template update and step cycle interpolation to cope with varying walking speeds, and preserve the unique characteristics in the user's gait pattern for accurate user verification.
- We develop our framework in two ways: user-centric and server-centric. We show that both approaches are robust to attacks including the attacker walks using his/her own walking style, and the attacker observes and mimics a legitimate user's walking patterns.
- We collect 3,048 accelerometer traces from multiple users over a period of 6 months. The results show that our system can effectively verify honest users when they walk at various speeds with the phone placed on different body positions.

The rest of the paper is organized as follows: We first put our work into the broader background of the related research in Section 2. We then describe the system model and the attack model used in this paper in Section 3. Next, we present our user verification system in Section 4. In Section 5, we validate the feasibility and effectiveness of our user verification system through extensive experiments conducted using smartphones. Finally, we conclude our work in Section 6.

2 Related Work

In this work, we consider user spoofing in mobile healthcare systems, which is different from the device-identity spoofing attacks [8], [9] considered in wireless networks. It may appear that cryptographic authentication schemes [10] are effective for thwarting user spoofing. However, the adversarial user may let the person (who collects the sensor data for him) know about the security information (e.g., passwords) stored in mobile devices to pass the security checks easily. Schemes utilizing the user's unique physical or physiological characteristics such as fingerprint [11], [12], [13] are attractive. These methods rely on additional hardware or require users to take explicit actions, and may not be suitable for mobile healthcare systems that continuously monitor users' behaviors.

Along this direction, there is existing work employing users' behavioral traits such as gaits for user verification. In [14], [15], [16], a vision-based gait recognition scheme has been proposed. The system uses several cameras to record a user when walking. Some background segmentation techniques are then used to extract features from recorded images to verify the user. In floor sensor-based approaches [17], [18], the sensors are placed on the floor and when people walk on the floor, the identity of a user can be authenticated by the exerted force measured by the sensor. However, additional hardware such as cameras and floor sensors is also needed for these schemes to work but such hardware may not be always available.

There is existing work [5], [19] using dedicated wearable accelerometers for gait recognition. The main advantage of using a wearable accelerometer sensor for gait recognition is that it provides unobtrusive verification of a user's identity without requiring his explicit actions. Gafurov et al. [20] further studies a hostile scenario on the accelerometer based gait recognition system. In this scenario, the attackers tend to mimic the targeted person's gait patterns. The authors show that their proposed system appears to be robust against such hostile scenario. However, expensive dedicated accelerometers with high sampling rates are used in these works. Bajrami et al. [21] focuses on using accelerometer sensors on smartphones to detect physical activities including walking, running, sitting and standing. It points out the possibility to perform gait recognition by using the results from activity recognition. In [6], the uniqueness of the gait in terms of foot motion with respect to the shoe attribute and axis of the motion is analyzed. It is not clear how their methods can deal with variable walking speeds. Our work is different in that we aim to employ gait information to perform user verification and detect the presence of user spoofing in mobile healthcare systems. Our approach targets to extract the unique characteristics of a user's gait pattern from sensor data collected from low grade accelerometers embedded within smartphones, and is robust with varying walking speeds.

3 Framework Overview

In this section, we first describe the mobile healthcare system model that is used in this paper. We then present the adversary model and provide an overview of our user verification framework.

3.1 System Model

We consider a mobile healthcare monitoring system in which a monitoring application runs on a user's smartphone and each user registering for its service is given a unique user identifier. This monitoring application can collect readings from embedded sensors within smartphones or external sensors attached to smartphones. Such sensor data will be analyzed to assess that user's physical activity levels or physiological conditions. For instance, a user's physical activity level can be assessed by monitoring his conversational activities, while measurements of heartbeats and blood pressure can be used to predict his psychological conditions [22]. Such sensing data collected by the monitoring device (e.g. smartphones) is sent to a system server. The server can then derive users' physical and mental well beings based on the rich information embedded in the sensing data. The system server then takes relevant follow-up actions based on such analysis, e.g. rewards those users who have weight problems for increasing their physical activity level. We envision that this type of mobile healthcare system will become very useful as it utilizes the information derived from users' daily lives, instead of requesting manual reporting from a user which could be inaccurate and error-prone [23]. Emerging applications enabled by such mobile healthcare systems include:

- The medical professionals from healthcare companies can monitor the health conditions of patients with heart diseases by monitoring their heartbeats. Based on patients' health conditions, the healthcare company can determine the frequency at which such patients should visit the doctors [24].
- Users' behavioral patterns and physical activity levels can be tracked by healthcare companies to facilitate early detection of signs of health problems (e.g., depression) [1].
- Companies that sell healthcare related applications e.g."I Do Move" [25] can convince healthy food companies to provide discount coupons for users who use their healthcare applications by sharing some statistics, e.g. total number of walking steps collected by their applications, with these food companies.

3.2 Adversary Model

We consider user spoofing in such mobile healthcare systems. The user spoofing could be conducted by an adversary user who passes his monitoring device, e.g. his smartphone, to a colluding person for a short period of time, and uploads the data collected by the other person instead in an attempt to gain more health benefits. For example, users who registered at "I Do Move" periodically upload their total number of walking steps to their account. Once their total walking steps reach a certain milestone, they will be rewarded with healthy food discount coupons. Adversarial users can ask others to walk with their devices and hence reach the qualifying milestone faster. Furthermore, an adversarial user can distribute his monitoring device to other people (e.g., a colluded user) who may suffer certain illnesses to collectively fool a mobile healthcare system which allocates health benefits to certain patients. The data collected from these spoofers will be mistakenly regarded as being obtained from the adversarial user. Thus, the related healthcare system may mistakenly classify this adversarial user as someone who has certain illness and hence qualified to enjoy certain healthcare treatments or benefits. Such spoofing attacks will significantly reduce the effectiveness of the healthcare management system and undermine the successful deployment of mobile healthcare applications since healthcare benefits will be given to the adversarial users.

In this work, we explore utilizing users' unique physical traits (i.e., gait patterns) which are hard to forge to perform unobtrusive user verification in the mobile healthcare systems. Nevertheless, a spoofer may attempt to mimic adversarial user's walking patterns to fool the system. Thus, we evaluate the robustness of our user verification scheme by studying two representative attacks:

- *Random attack.* A spoofer walks using his/her own walking style and has no information other than the knowledge of the user verification system, e.g., gait-based patterns are used for user verification.
- *Mimic attack.* Besides the knowledge of the user verification system, a spoofer has the knowledge of a legitimate user's (i.e., the adversary user who passes his device to the spoofer) walking styles and hence mimics his walking patterns in an attempt to fool the system.

3.3 User Verification Framework

We build a framework that utilizes user gait patterns extracted from accelerometer readings using readily available accelerometers embedded within smartphones. To prevent an adversary from launching replay attacks using a user's accelerometer readings, similar to [26], an encrypted time-stamped identifier can be generated for each block of the accelerometer measurements. The system will then examine these time-stamped identifiers to ascertain that the measurements are originally collected. The full discussion of this security issue is out of the scope of this paper and will be included in our future work. Our framework can be implemented in two ways: *server-centric* and *user-centric*.

In the *user-centric approach*, user verification is performed on the smartphone. An initial user's profile will be constructed and stored in the smartphone. The details of the user profile construction are described in Section 4. The user verification is then performed on smartphones with low computational complexity. Specially, we use a correlation coefficient based approach to help computing similarity scores between the user profile and gait features extracted from user's smartphone. If the user verification fails, i.e., the user spoofing is detected, the sensing data collected from this user's mobile device will not be reported back to the server.

In the *server-centric* approach, gait features extracted from accelerometer readings are sent to a centralized server for user verification. We use a data mining based technique to perform user verification by comparing the stored user profiles with gait features extracted from user's smartphone. In particularly, we train a SVM-based classifier using collected traces from users. The trained SVM-based classifier is then utilized to test real time measurements submitted for user verification. The server will decide whether to accept the sensing data collected from this user based on the user verification result.

4 User Verification Based on Gait Patterns

In this section, we detail our user verification system. The system can be deployed in both user-centric as well as server-centric framework.

4.1 Challenge and Design Goals

The goal of our smartphone enabled gait-based user verification system is to conduct user verification without relying on additional infrastructures or explicit user actions. This allows a pure software solution. Such a system should be robust against user spoofing attacks. To fulfill such a goal in mobile healthcare systems, we need to deal with the following challenges.

Robust to various walking speeds. Users' walking speeds vary under different scenarios and environments. The gait recognition process should be robust to various walking speeds in order to facilitate an effective user verification.

Reasonable accuracy. Our framework leverages the accelerometers on smartphones with a lower sampling rate (e.g., 50 Hz), which is about half the sampling rate of the regular accelerometer sensors. Our technique needs to achieve reasonable attack detection accuracy using readings collected from accelerometers embedded within off-the-shelf smartphones.

Low detection latency. Our user verification system should be able to detect the presence of user spoofing with small number of measurements. In this way, the framework can avoid wasting computational cost spent on processing the sensor data reported from a user's mobile device for the corresponding healthcare needs.

4.2 Scheme Overview

The basic idea underlying our user verification system is based on the observation that the gait pattern is unique for each person and differs between different people. This is inline with the observation made by researchers in [27] who conduct experiments with a sufficiently large gait database with over 700 users and found that their gait patterns are unique. When a user spoofing is present, the extracted gait pattern from the run-time accelerometer measurements from smartphones may differ significantly, and hence such observation can be used to detect the attack and perform user verification.

Our scheme, as shown in Fig. 1, consists of three main sub-tasks: *Step Cycle Identification*, *Step Cycle Interpolation* and *Similarity Comparison*. When the verification procedure starts, step cycle sequence needs to be first identified from the run-time accelerometer measurements. A step cycle template based technique is proposed to accurately capture the uniqueness embedded in each person's gait. This template can be dynamically updated when a user's physical/medical situation changes. The identified step cycle sequence is further interpolated to deal with various walking speeds when a user is at different environments. A user's initial profile contains the user's gait pattern and is constructed when a user first submits its accelerometer

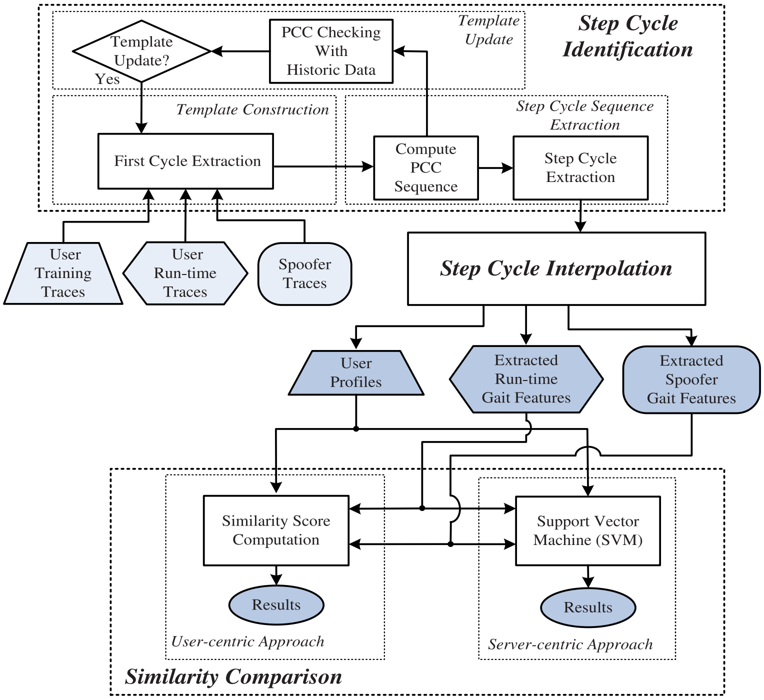


Fig. 1. Overview of our user verification system using gait patterns.

measurements. The profile is obtained by utilizing the Step Cycle Identification and Step Cycle Interpolation. The Step Cycle Interpolation component allows robust user verification even when the user's walking speed during an run-time measurement is different from that in a user's profile. A user profile can also be updated when the initial profile significantly deviates from a user's current gait patterns caused by physical/medical situation changes: the user submits a new sequence of accelerometer measurements and the user profile will be re-generated. This updating process happens infrequently and an in-person verification of that user is required to confirm that the accelerometer measurements sent by a user match his/her true identity. The user verification is then performed by conducting similarity comparison between the initial user profile and the extracted gait features from the run-time traces. If a user distributes his device to another person who acts as a spoofer, a lower similarity value will be obtained after the computation because the gait patterns between two people differ dramatically, and consequently the user spoofing is detected. We next describe the sub-tasks in detail.

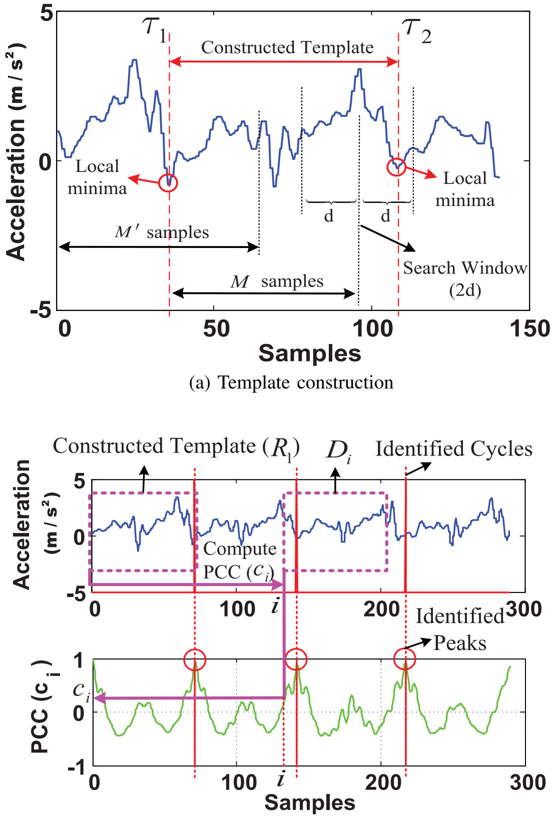
4.3 Step Cycle Identification

Human gait follows a cyclic pattern. In this work, the event that we use to mark the beginning of the step cycle is the heel strike of the swing leg [14]. At that moment, the person's feet are both on the ground and they are farthest from each other. The vertical acceleration of the impact can be observed as a local minima in the accelerometer readings. Thus, the step cycle can be identified by extracting the timestamps of the heel strikes. However, identifying the step cycle is challenging because the accelerometer readings can be distorted due to the irregular movement of the user's body or the change of walking speeds. The commonly used step cycle identification techniques [5], [6], [20] identify the gait cycles by conducting the typical cycle identification repeatedly in the traces. The problem is that if the cycle identification for one period is not accurate, the detection of the following periods will be affected. The detection errors are propagated and compounded throughout the whole cycle identification. For these reasons, we utilize the fact that the same user's gait patterns are unique and the consecutive step cycles should present a high correlation in a collected walking trace. We thus extract a person's gait pattern as a template by identifying the first distinguishable step cycle. We then utilize the correlation relationship inherent in the same user's walking trace to search for the maximum correlation between the first distinguishable cycle and the rest of trace to derive the step cycle sequence.

4.3.1 Template Construction

Let $\{r(1), \ldots, r(N)\}$ be a sequence of N accelerometer measurements in the vertical axis from a smartphone and we assume the τ_kth measurement is the first sample of the kth step cycle. We do not consider the acceleration in other two axes because one axis points to the direction the user is moving and another one points to the direction of the sideways movements, which cannot be used to identify the uniqueness of a user's gait patterns. To construct the step cycle template, we need to find the first two consecutive local minimas $r(\tau_1)$ and $r(\tau_2)$ in the accelerometer readings which represent the beginning and the end of the first distinguishable step cycle $R_1 = \{r(l), \tau_1 \leq l < \tau_2\}$. To identify R_1, we assume the user's maximum and regular step cycles have approximate M' and M samples respectively according to the sampling frequency of the accelerometer. Thus, the beginning of the first step cycle τ_1 can be found by searching the minimum value from the first M' observations

$$\tau_1 = \arg\min_{l}(r(l)), 1 \leq l \leq M'. \qquad (1)$$



(a) Template construction

(b) Step cycle sequence extraction by utilizing the template

Fig. 2. Illustration of step cycle identification.

We then search for the end of the first step cycle τ_2 by extending M samples from τ_1. Because the user's walking speed is unknown, the τ_2 can be determined by relaxing the searching range by d samples before and after the M samples:

$$\tau_2 = \arg\min_{l}(r(l)), \tau_1 + M - d \leq l \leq \tau_1 + M + d. \quad (2)$$

Thus, $r(\tau_1)$ and $r(\tau_2)$ are the first two consecutive local minimas in the sequence of recorded accelerometer readings. The $L = \tau_2 - \tau_1$ consecutive samples of $R_1 = \{r(l), \tau_1 \leq l < \tau_2\}$ in sequence $\{r(1), \ldots, r(N)\}$ will then be used as a template to identify the rest of step cycles in the collected trace. We illustrate template construction in Fig. 2a.

Provided with the knowledge about human walking patterns, we can then determine suitable values for M', M and d: The natural cadence of the human walking, irrespective of what speed he/she walks, is usually in the range of $[45, 65]$ step cycles/min [28] and we assume the sampling frequency of accelerometer is 50 samples per second. Thus, the number of samples in one step cycle is in the range $[46, 67]$ samples/step and each regular cycle contains about $(46 + 67)/2 \approx 56$ samples. With the aid of such clues, in this work, we empirically set the M' as the number of samples that a maximum step cycle has with $M' = 67$ samples and M as the number of samples a regular cycle has with $M = 56$ samples, respectively. The search range d is then set as the half of the difference between the maximum and minimum number of samples the step cycle has (i.e., $d = (67 - 46)/2 \approx 11$ samples).

4.3.2 Step Cycle Sequence Extraction

The template $R_1 = \{r(l), \tau_1 \leq l < \tau_2\}$ contains the samples of user's first step cycle. We assume there are S step cycles in the collected trace. To identify the subsequent step cycles R_k, $k = 2, 3, \ldots, S$, in a collected trace, we utilize the step cycle correlation inherent in a user's walking trace. The correlation among step cycles of the same person allows us to extract a user's step cycles accurately due to the fact that the correlation coefficient between two step cycles of the same person is robust to distorted readings caused by irregular movement of the user's body. Further, after examining the correlation coefficient between the template and subsequent step cycles, we can update the template dynamically based on the changes in user's walking speeds. This is because the step cycles should be highly correlated if the speed of the template step and subsequent steps are similar. Thus, a significant decrease in correlation coefficient between two step cycles indicates a large speed change. The template, consequently, should be updated based on the new speed (shown in the next step).

Fig. 2b illustrates the step cycle sequence extraction using Pearson correlation method [29]. To identify the subsequent step cycles, the template R_1 is slid across the recorded accelerometer readings and the Pearson correlation coefficients between the template R_1 and the consecutive L samples in recorded accelerometer readings are calculated. The Pearson correlation coefficient is a statistical method that measures the degree of the linear relationship between two given vectors. The Pearson correlation coefficient value ranges from -1 to 1. Correlation 1 and -1 means that there is a perfect positive/negative linear relationship between the two vectors. Specifically, given the template R_1 with length $L = \tau_2 - \tau_1$ and consecutive L samples $D_i = \{r(l), i \leq l < i + L\}, i = 1, \ldots, N - L$, from the recorded accelerometer readings $\{r(1), \ldots, r(N)\}$, the Pearson correlation coefficient is defined as

$$c_i = corr(R_1, D_i) = \frac{\sum_{l=0}^{L-1}\left(\frac{r(\tau_1+l)-\overline{R}_1}{\sigma(R_1)}\right)\left(\frac{r(i+l)-\overline{D}_i}{\sigma(D_i)}\right)}{L-1}, \quad (3)$$

where $\bar{R}_1$ ($\bar{D}_i$, resp.) and $\sigma(R_1)$ ($\sigma(D_i)$, resp) are the mean and standard deviation of R_1 and D_i. The values in Pearson correlation coefficient sequence $C = \{c_i, i = 1, \ldots, N - L\}$ increase and decrease successively, indicating similarity between the template R_1 and the segment D_i. The peaks arise periodically in PCC sequence C indicating good matches between the template and the subsequent D_is. Thus, these periodical peaks can be used to identify the subsequent step cycles. The local maximas in C are detected and marked as beginning points of each walking step, which occur at the heel strikes of a swing leg. The algorithm of step cycle sequence extraction is provided in Algorithm 1.

In Fig. 2b, the blue line in the upper plot represents the accelerometer readings on smartphones. The green line in the lower plot represents the correlation coefficient sequence C computed between the step cycle template and each data segment D_i. The step cycles are identified by searching periodical local peaks in sequence C. The identified step cycle sequence R_k is

$$R_k = \{r(l), \tau_k \leq l < \tau_{k+1}\}, k = 1, \ldots, S. \quad (4)$$

Algorithm 1. Step Cycle Identification

```
INPUT:
  $Data = \{r(1), \ldots, r(N)\}$;  A sequence of accelerometer readings
  $R_1 = \{r(l), \tau_1 \leq l < \tau_2\}$;  The constructed template
  $L = \tau_2 - \tau_1$;  Number of samples in extracted template
  $counter = 0$;  Number of peaks in PCC sequence
PROCEDURES:
for All $i \in [1, N - L]$ do
  $D_i = \{r(l), i \leq l < i + L\}$;
  $c_i = corr(R_1, D_i)$;
end for
for All $i \in [1, N - L - 1]$ do
  if $c_i > c_{i-1} \& c_i > c_{i+1} \& c_i > threshold$ then
    $counter = counter + 1$;
    $\tau_{counter} = i$;
  end if
end for
Return number of step cycles $S = counter - 1$
Return step cycle sequence $R_k = \{r(l), \tau_k \leq l < \tau_{k+1}\}, k = 1, \ldots, S$
```

4.3.3 *Template Update*

The length of the step cycle changes as the user's walking speed varies. With the help of the correlation coefficient between the template and the subsequent step cycles, we are able to tell when the user's speed changes, and update the template timely once the speed change is detected. The update decision may be system triggered. Particularly, the system automatically searches the peaks in the Pearson correlation coefficient sequence: if most of the peaks (e.g., 80 percent) in a past time period (e.g., a few minutes) are lower than a threshold (e.g., 0.8), the template update is triggered. A new template R_1 will be generated using the Template Construction scheme on newly collected accelerometer readings.

4.4 Step Cycle Interpolation

A user usually walks at different speeds in different scenarios such as taking a leisure walk after dinner or walking rapidly to catch a commuter train after work. Furthermore, the walking speed of a user during the runtime data collection process is most likely different from the speed when the user profile is constructed. The number of samples in step cycles varies as the user's walking speed changes. To deal with variable walking speeds, our framework preforms step cycle interpolation. This interpolation step allows us to perform robust user verification by directly measuring the similarity between the step cycle sequence in the user profile and the interpolated sequences obtained from run-time measurements under different walking speeds. More importantly, by using Step Cycle Interpolation, a user only needs to upload one accelerometer trace under any speed at its convenience for user profile construction without requiring extensive uploading of multiple traces to cover different speeds.

To perform step cycle interpolation, we align the extracted step cycle sequence to a reference step cycle with length P by using cubic spline interpolation [30], a fast, efficient and stable method of function interpolation. Further, we choose a large P (e.g., $P = 300$ samples) so that it is larger than any user's longest one step cycle irrespective of what speed the user walks. The step cycle sequence after interpolation is represented as

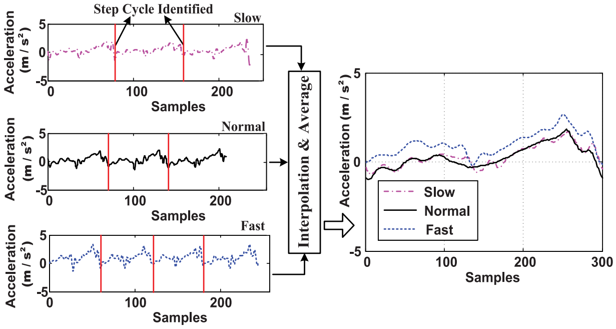


Fig. 3. Illustration of step cycle interpolation for a user under three typical walking speeds: slow, normal and fast.

$$N_r = \{r(1,k), \ldots, r(P,k)\}, k = 1, \ldots, S. \tag{5}$$

To capture the pattern of all the step cycles, we average over the interpolated step cycles. Thus, the final interpolated step cycle can be represented as $I = \{\overline{r}(1), \ldots, \overline{r}(P)\}$ with

$$\overline{r}(j) = \sum_{k=1}^{S} \frac{r(j,k)}{S}, j \in [1, P]. \tag{6}$$

The algorithm Step Cycle Interpolation is provided in Algorithm 2. Fig. 3 shows an example on how the final interpolated step cycle is extracted under different walking speeds for a specific user. In Fig. 3, the collected acceleration readings under three representative speeds (i.e., *slow, normal, and fast*) are depicted in the left side of the figure. The detailed description of these three speeds are presented in Section 5.1. The final interpolated step cycles corresponding to these three different speeds are shown in the right side figure. Before interpolation, it is hard to directly compare the step cycles under different speeds due to different lengths of the step cycles. After Step Cycle Interpolation, we find that the final interpolated step cycles under three different speeds are highly correlated regardless of the walking speeds. This result is encouraging as it indicates a particular user's gait pattern is unique and not sensitive to a user's walking speeds.

Algorithm 2. Step Cycle Interpolation

```
INPUT:
  $R_k = \{r(l), \tau_k \leq l < \tau_{k+1}\}, k = 1, \ldots, S$;  Identified step cycles
  $P = 300$;  Number of samples
PROCEDURES:
for All $k \in [1, S]$ do
  $\{r(1,k), \ldots, r(P,k)\} = Interpolation(\{r(\tau_k), \ldots, r(\tau_{k+1})\})$;
end for
for All $j \in [1, P]$ do
  $\overline{r}(j) = \sum_{k=1}^{S} \frac{r(j,k)}{S}$
end for
Return interpolated step cycle $I = \{\overline{r}(1), \ldots, \overline{r}(P)\}$
```

4.5 Similarity Comparison

Our Similarity Comparison component can be implemented in two ways: user-centric and server-centric. In the user-centric approach, our scheme uses an approach which utilizes weighted Pearson correlation coefficient with low computation complexity in similarity score computation to verify honest users. However, in the server-centric approach, our scheme uses the Support Vector Machine classifiers for user verification to achieve a higher accuracy.

4.5.1 *User-Centric Approach*

The interpolated step cycle represents a user' gait pattern. Based on the foot motion, a step cycle can be further decomposed into several sub-events such as initial contact, loading response, and midstance [31]. The user's gait pattern in certain sub-events may remain constant while others change. The sub-events within a gait pattern remain constant should be treated more significantly since they can better represent the uniqueness of the user's gait pattern. Thus, to capture this observation in a quantitative way in our user-centric approach, we propose to use *weighted* Pearson correlation coefficients when computing the similarity between the extracted gait patterns and the user profile.

We next calculate the weights from sub-events in a user's step cycle. Based on the interpolated step cycle sequence N_r, we first equally divide P samples in the interpolated cycle into B (e.g., $B = 6$) blocks: $\{P_n, \ldots, P_{n+1}\}, n = 0, \ldots, B-1$ with $P_0 = 1$ and $P_B = P$. Thus, the average sample distance over these blocks can be represented as: $Dist = \{\overline{d}_n, n = 0, \ldots, B-1\}$, where each $\overline{d}_n$ is defined as

$$\overline{d}_n = \frac{\sum_{c=P_n}^{P_{n+1}} \sum_{\substack{k,l\in[1,S]\\ k\neq l}} |r(c,k) - r(c,l)|}{(S-1)\times S\times(P_{n+1}-P_n+1)}. \tag{7}$$

Each $\overline{d}_n$ in $Dist$ measures the average sample distance in the nth block between each pair of S interpolated step cycles. Based on the sample distance, we define *weights* over these blocks as $\{w_n, n = 0, \ldots, B-1\}$, where each w_n is defined as: $w_n = 1/\overline{d}_n$.

We then define the similarity score between the final interpolated step cycle obtained from run-time measurement $I_g = \{\overline{r}^g(1), \ldots, \overline{r}^g(P)\}$ and the user profile $I_h = \{\overline{r}^h(1), \ldots, \overline{r}^h(P)\}$ by computing weighted Pearson correlation coefficient with the weight as $\{w_n, n = 0, \ldots, B-1\}$

$$\mathbf{C}(I_h, I_g) = \frac{\sum_{n=0}^{B-1} corr\big(\{\overline{r}^h(P_n), \ldots, \overline{r}^h(P_{n+1})\}, \{\overline{r}^g(P_n), \ldots, \overline{r}^g(P_{n+1})\}\big) w_n}{\sum_{n=0}^{B-1} w_n}. \tag{8}$$

If the similarity scores are lower than a pre-defined threshold, the framework will declare the presence of the user spoofing for this particular user ID.

Feasibility study. We study how the similarity scores change when acceleration readings are collected from different users under three typical walking speeds (i.e., *slow*, *normal*, and *fast*). We collect 6 traces per user with two traces per walking speed. Fig. 4 plots the final interpolated step cycles generated for these two users and the cumulative distributed function (CDF) of the similarity score. From the left subfigure in Fig. 4, we observe that the final interpolated step cycles of a particular user are very similar even at three different walking speeds, while the final interpolated step cycles between two users differ significantly. Furthermore, from the right subfigure, the similarity scores are high (larger than 0.8) for the same user regardless of walking speeds. Whereas the similarity scores reduce to $[-0.2, 0.3]$ between two users. These observations strongly confirm the feasibility of using our user-centric approach to detect user spoofing.

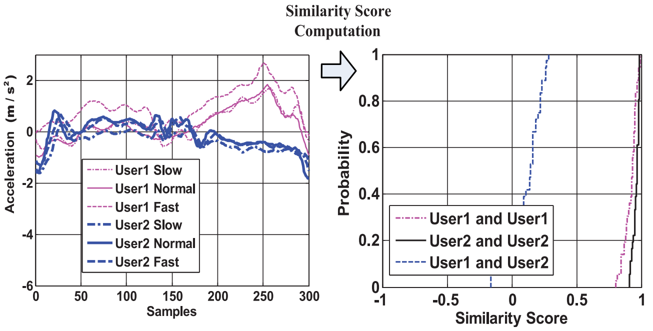


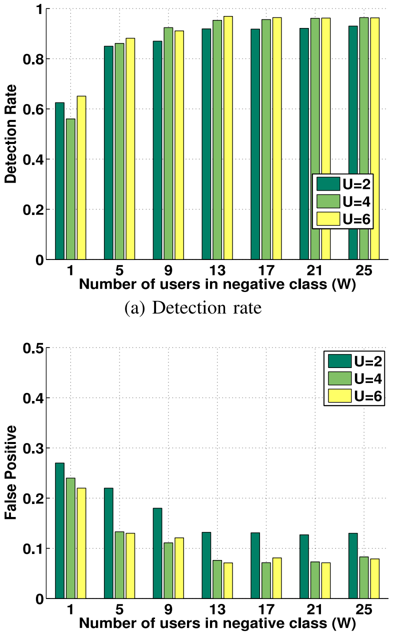
Fig. 4. An illustration of interpolated step cycles of user 1 and 2 under different walking speeds and CDF of their similarity scores.

4.5.2 *Server-Centric Approach*

In our server-centric approach, we view the user verification problem as a two-class classification problem and we choose Support Vector Machine as our classifier. SVM has been successfully used for a number of classification problems especially if only little sample data is available for training [32].

As shown in Section 4.5.1, the final interpolated step cycle can characterize a user's unique gait pattern and is not sensitive to the walking speeds. The final interpolated step cycle can be used to distinguish different users. Thus, we consider each sample value within an final interpolated step cycle as a feature (e.g., we have $P = 300$ features). In our classification model, we label the original user's data as the positive class and all other users' data as the negative class. In particular, to train the SVM classifier for each user, we select U traces from this user. For each trace, we average all interpolated step cycles to get the final interpolated step cycle. Thus, we have U final interpolated step cycles, labeled as positive instances. We then choose U traces from each of the rest users (e.g., the rest is W users) and similarly calculate U final interpolated step cycles for this user, labeled as negative instances. Thus, we have U positive instances and $U \times W$ negative instances as our training set for each specific user. Training instances (including positive and negative ones) are put together in the training data set to train the SVM classifier. In the user verification phase, the extracted features (i.e., interpolated step cycles) obtained from run-time measurement is input to the user's classification model and then SVM classifier outputs a predictive label. If the label is positive, the user verification is a success. Otherwise, the label is negative, indicating the presence of user spoofing for this particular user under verification.

Minimum training size study. We conduct experiments to determine the minimum values of U and W so that we can



(a) Detection rate

(b) False positive rate

Fig. 5. Server-centric approach: study of training size to obtain a stable SVM classifier.

create a stable SVM classifier for each new user when joining the system. We keep both user profile trace and runtime measurement trace length as 20 seconds under normal walking speed. We then incrementally increase the values of U and W and check which point can generate a stable classifier. Figs. 5a and 5b depict the detection rate and false positive rate of our server-centric approach when W is changed from 1 to 25 users at the interval of 4 users under different values of U.

We find that the detection rate and false positive rate stabilize when W (i.e., the number of users in the negative class) exceeds 13 users and U (i.e., number of training traces from each user) exceeds four instances: over 90 percent detection rate and less than 10 percent false positive rate can be achieved. It indicates that we need to select at least 13 users in the negative class and choose at least four traces from each user to create a stable SVM classifier. Thus, unless otherwise specified, we choose $W = 13$ users in this paper.

5 Performance Evaluation

In this section, we conduct experiments using the readily available accelerometers embedded within smartphones to evaluate the effectiveness of our user verification system under the presence of user spoofing. The following sections detail our experimental methodology and results.

5.1 Experimental Methodology

We use HTC EVO smartphones equipped with accelerometer that supports 50 Hz sampling rate to collect data from volunteered users. Each HTC EVO smartphone runs Android operating system with 192 MB RAM and a 528 MHz MSM7200A processor. The accelerometer readings are collected when the users are walking and then written into a log file on a smartphone. Of the three dimensional accelerometer signals retrieved from the smartphone, only the acceleration in vertical direction is used as it is sufficient to capture the user's unique gait patterns. Additionally, we experiment with three representative user walking speeds, namely *slow* (slower than 0.7 m/s), *normal* (about 0.7 m/s-1.1 m/s), and *fast* (faster than 1.1 m/s). During the experiments, we let each user keep the phone in three commonly used positions including *hip pouch*, *waist pouch* and *pant pocket*. Unless otherwise specified, the accelerometer readings collected from hip pouch position are used to present evaluation results.

Trace collection. We collect accelerometer traces from 26 volunteers for our evaluation. A size of 26 users is quite typical in user monitoring and verification studies [6], [33]. To test the robustness of our system, each volunteer can walk with different shoes in our six-month trace collection period. Unless otherwise stated, each trace represents accelerometer readings of a user walking for a period of about 10 minutes. In the user-centric approach, user profiles are constructed from each user based on the traces collected. In the server-centric approach, a per-user SVM-based classifier is created for each smartphone placement position. To evaluate the robustness of our system, we study two representative attack scenarios.

Robustness testing. We first test the scenarios where a spoofer does not have the knowledge of an adversary user's gait patterns (i.e. random attacks). Thus, we choose each user as an adversary user, and use the rest of the users as spoofers to launch user spoofing attacks. Spoofers are told to walk using their own walking styles during trace collection. For each spoofer, we collect traces of his regular walking speed as well as traces with varying walking speeds where he modifies his walking speed every 10 seconds.

We then study the scenarios where the spoofer has the knowledge of the walking styles to perform the mimic attack. Thus, we choose eight users as adversary users. We then select another 10 users as spoofers whose physical characteristics are similar to the selected adversary users to launch the mimic attack. For each smartphone placement, a spoofer observes the adversary user and then mimic his/her walking styles to produce traces from each walking speed.

For each user, we construct a user profile in the user-centric approach and train a SVM classifier in the server-centric approach. In total, we collect $3{,}048$ accelerometer traces over 6 months to evaluate the robustness of our system.

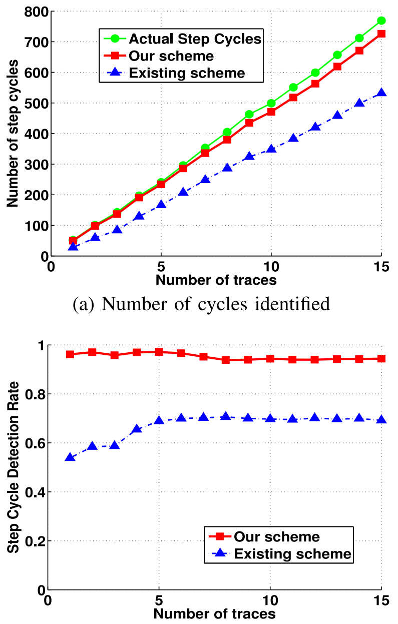
Metrics. We use the following metrics to evaluate the effectiveness of our user verification system:

- *Detection rate*. It is defined as the percentage of attack instances that are correctly identified by our system;
- *False positive rate*. It is defined as the percentage of non-attack instances that are mistakenly detected as attack instances.

We first present the evaluation of our step identification scheme in Section 5.2. We next test the robustness of our system under random attacks from Sections 5.3 to 5.5. The robustness of our system under mimic attacks is then evaluated in Section 5.6.

5.2 Comparison of Step Cycle Identification

In the first set of experiments, we evaluate the effectiveness of our proposed step cycle identification scheme by comparing it with an existing method that identifies



(a) Number of cycles identified

(b) Detection rate

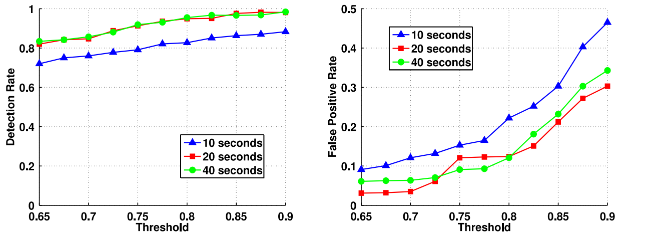
Fig. 6. Comparison of step cycles identification by using different schemes with 15 traces from 5 users under different walking speeds.

cycles based on local minimum searching [5], [6], [20]. For comparison, we use 15 walking traces from five users with one minute length for each trace. Thus, there are three traces from each user under three different walking speeds. We compare the *step cycle detection rate*, which is the percentage of step cycles that are accurately identified, of our proposed method to the existing method.

Fig. 6 depicts the cumulative number of identified step cycles and the corresponding detection rate with increasing number of walking traces for both our proposed method and the existing method. First, we observe that the number of the step cycles identified by our method stays very close to that of the actual number of step cycles present in each trace (reported by each user), whereas the gap between the curve of using the existing method and that from the actual step cycles is significantly larger than that of our proposed method. Therefore, the step cycle detection rate of our scheme is significantly higher than that of using the existing method: our method can achieve a step cycle detection rate over 90 percent with different number of walking traces, while the step cycle detection rate ranges from 50 to 70 percent for the existing scheme. These observations indicate that our proposed step cycle identification scheme can derive step cycles much more accurately than the existing schemes [5], [6], [20]. This is because the existing schemes only rely on local minima searching which is easily affected by the noise caused by irregular movement of a user. Further, the detection errors also propagate and affect the accuracy of subsequent cycle detections. Whereas our method exploits the high correlation inherent within a user's step cycles and is more robust to such noise.

5.3 Detection Latency Study

The detection latency study tests the robustness of our system when run-time measurements of different durations are used for attack detection. Specifically, we evaluate the detection performance when the run-time measurement trace length equals to "10 seconds", "20 seconds" and "40 seconds" respectively, under normal walking speed. The same corresponding length of the user profile trace is used. The time length of "10 seconds", "20 seconds", and "40 seconds" corresponds to about 9, 18 and 36 step cycles respectively under the normal speed.

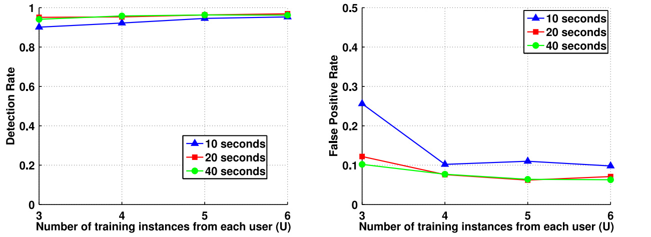


(a) Detection rate (b) False positive rate

Fig. 7. User centric approach: detection latency of random attacks by varying the duration of run-time measurement trace and user profile trace.

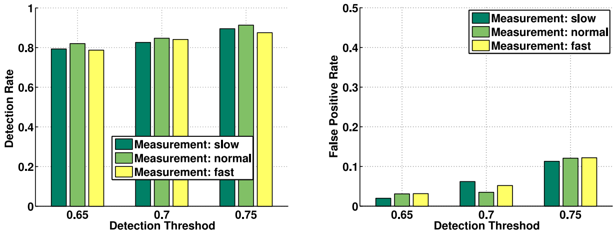
User-centric approach. Fig. 7 presents the detection rate and false positive rate of random attacks under different detection thresholds. We first observe that the longer traces result in better detection performance. In particular, our scheme can achieve over 80 percent detection rate with less than 10 percent false positive rate when the trace length is longer than 20 seconds. This is because more step cycles can be identified in a longer trace which captures of a user's unique gait pattern better. The encouraging observation is that a trace length of 20 seconds is sufficient for our scheme to achieve a reasonable detection rate and a low false positive rate.

Server-centric approach. Fig. 8 presents the detection rate and false positive rate of random attacks using different number of training traces from each user (i.e., U). Similarly, better performance can be achieved when the trace length is longer than 20 seconds: our server-centric approach can achieve over 90 percent detection rate with less than 10 percent false positive rate. It indicates that a trace length of 20 seconds is sufficient to achieve a stable detection rate and false positive rate for server-centric approach. Further, we also observe that the detection rate and false positive rate stabilize when U exceeds four traces, which demonstrates that having four training traces from each user is sufficient to create a stable SVM classifier. Moreover, when comparing these results with those obtained using the user-centric approach in Fig. 7, we observe that our server-centric approach can yield higher accuracy. This is because the server has all the users' gait information and uses a more powerful data mining technique for the verification.



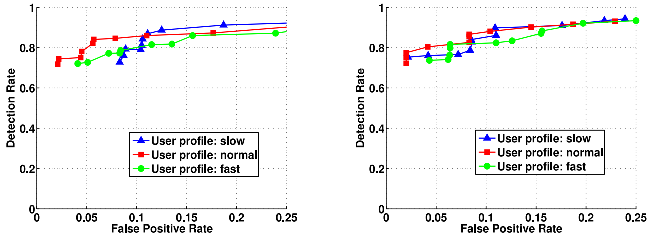
(a) Detection rate (b) False positive rate

Fig. 8. Server centric approach: detection latency of random attacks by varying the duration of run-time measurement trace and user profile trace.



(a) Detection rate (b) False positive rate

Fig. 9. User centric approach: robustness study of random attacks under different walking speeds with user profile/run-time measurement traces of 20 seconds.



(a) Run-time measurements: 20 seconds (b) Run-time measurements: 40 seconds

Fig. 11. User centric approach: varying speed study of random attacks using user profile traces of different walking speeds.

5.4 Walking Speed Study

We next present our study using the run-time measurements with different walking speeds or varying walking speeds.

5.4.1 Robustness against Different Walking Speeds Study

We first study the robustness of our method under the scenarios where the run-time measurement traces are of different walking speeds from that used for constructing a user profile. Specifically, the run-time measurement traces are in slow, normal, and fast walking speeds, respectively. The duration of both user profile traces and run-time measurement traces is set as 20 seconds and a user profile is constructed from traces collected with a normal walking speed.

User-centric approach. Fig. 9 shows the detection rate and false positive rate under random attacks using different detection thresholds. We observe that the detection rate increases as the detection threshold increases. This is because with higher detection threshold, it is easier for our scheme to detect traces which are from different users. Further, we find that the overall detection rate remains around 80 percent and the false positive rate is lower than 10 percent. Moreover, the figure clearly shows similar detection rate and false positive rate are achieved even if the traces of run-time measurements are collected using different walking speeds from that used to construct a user profile. This demonstrates that our system operating the user-centric approach is robust even if the user is walking under different speeds.

Server-centric approach. Fig. 10 shows the detection rate and false positive rate under random attacks using different number of training traces from each user (i.e., U). We observe that the overall detection rate remains over 90 percent and the false positive rate is lower than 10 percent when U exceeds four traces. Further, the similar detection performance is achieved even if the traces of run-time measurements are collected using different walking speeds. This indicates that our system operating server-centric approach is also robust against different walking speeds.

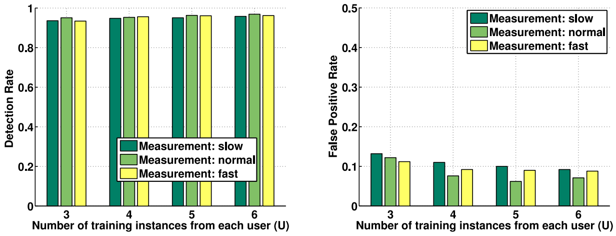
5.4.2 Varying Walking Speed Study

We then evaluate the effectiveness of our method using the run-time measurement traces with varying speeds. We keep the user profile trace as 20 seconds while varying the duration of the run-time measurement traces.

User-centric approach. Fig. 11 plots the Receiver Operating Curve (ROC) under random attacks when the detection threshold changes from 0.65 to 0.9. We show the results using the run-time measurement traces of 20-second (Fig. 11a) and 40-second (Fig. 11b) long. The legends "*User profile: slow*", "*User profile: normal*" and "*User profile: fast*" denote the traces for constructing a user's profiles are chosen from constant speed traces with slow, normal and fast walking speeds, respectively.

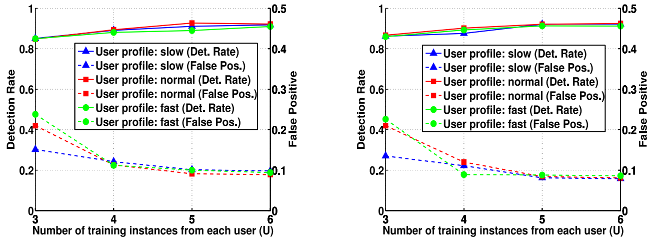
Similarly, the overall performance of our method can achieve over 80 percent detection rate with less than 10 percent false positive rate. This shows that our system operating user-centric approach is robust to the dynamic walking speed. Further, we observe that the detection rates under user profiles constructed from traces of different speeds are comparable when the false positive rate is around 10 percent, indicating our system is not sensitive to the walking speeds of training traces. With a user profile constructed from traces with normal speed, the detection performance is slightly better than that of with other speeds when the false positive rate is below 10 percent.

Server-centric approach. Fig. 12 depicts the detection rate and false positive rate under random attacks when the number of training traces (i.e., U) changes from 3 to 6. We show the results using the run-time measurement traces of 20-second (Fig. 12a) and 40-second (Fig. 12b) long. The legends



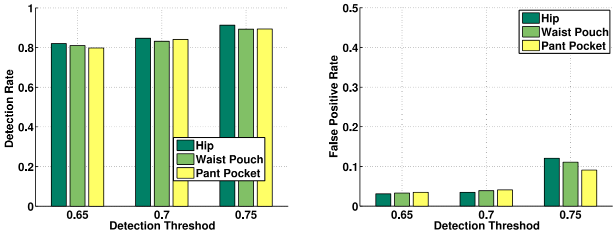
(a) Detection rate (b) False positive rate

Fig. 10. Server centric approach: robustness study of random attacks under different walking speeds with user profile/run-time measurement traces of 20 seconds.



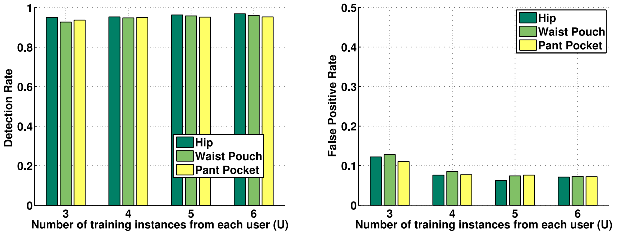
(a) Run-time measurements: 20 seconds (b) Run-time measurements: 40 seconds

Fig. 12. Server centric approach: varying speed study of random attacks using user profile traces of different walking speeds.



(a) Detection rate (b) False positive rate

Fig. 13. User centric approach: performance comparison of random attacks under different smartphone placements.



(a) Detection rate (b) False positive rate

Fig. 14. Server centric approach: performance comparison of random attacks under different smartphone placements.

"Det. Rate" and *"False Pos."* denote the detection rate and false positive rate, respectively.

We observe that our method can achieve over 90 percent detection rate with less than 10 percent false positive rate. This shows that our system operating server-centric approach is also robust to the dynamic walking speed. Further, similar performances can be achieved under the user profiles constructed from different speeds. This indicates that our server-centric approach is not sensitive to the walking speed of training traces. Again, as we have observed previously, the detection rate and false positive rate also stabilize when U exceeds four traces, which further confirms that having four traces is sufficient for our server-centric scheme.

5.5 Smartphone Placement Study

We further evaluate our system when the phone is placed in other two body positions: the waist pouch position and pant pocket position. The legends *"Hip"*, *"Waist Pouch"* and *"Pant Pocket"* denote the traces are collected from the hip pouch position, the waist pouch position and the pant pocket position, respectively. For each position, the duration of both user profile traces and run-time measurement traces is set as 20 seconds under normal walking speed.

User-centric approach. Fig. 13 presents the detection rate and false positive rate under random attacks using different detection thresholds when smartphones are placed in different body positions. We observe that the overall detection rate remains over 80 percent and the false positive rate is lower than 10 percent. Further, the similar detection rate and false positive rate are achieved even if the traces are collected from different body positions. This demonstrates that our system operating user-centric approach is robust even if the smartphones are placed on different body positions.

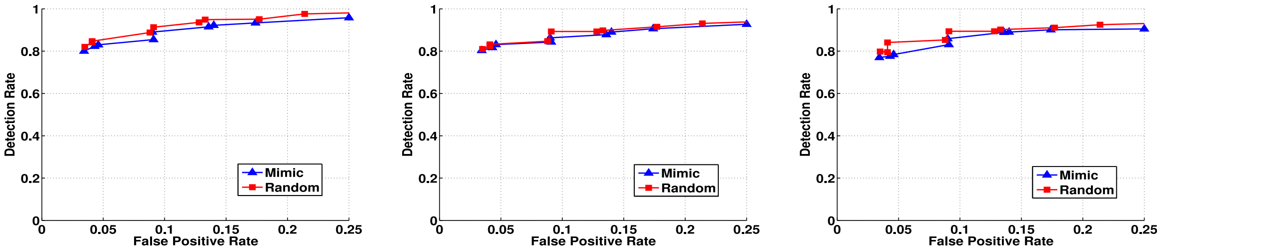
Server-centric approach. Fig. 14 depicts the results from different body positions under random attacks using different number of training traces from each user (i.e., U). Similarly, the overall performance of our server-centric approach can achieve over 90 percent detection rate with lower than 10 percent false positive rate on each body position. This verifies that our system operating server-centric approach is also not sensitive to the smartphone placement.

5.6 Robustness to Mimic Attacks

Finally, we evaluate the effectiveness of our user verification system under mimic attacks. The duration of both user profile traces and run-time measurement traces is also set as 20 seconds under normal walking speed. The legends *"Mimic"* and *"Random"* denote the run-time measurement traces are collected from the mimic and random attack scenarios respectively. The results from different smartphone placements are also presented.

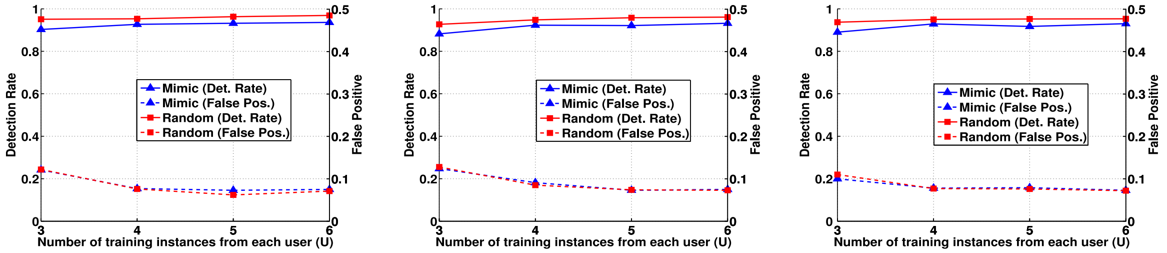
User-centric approach. Fig. 15 plots the ROC under mimic attacks and random attacks when the detection threshold changes from 0.65 to 0.9. The overall performance of our user-centric approach can achieve over 80 percent detection rate with less than 10 percent false positive rate under mimic attacks in three smartphone placements. Further, we observe that the performances under random attacks and mimic attacks are comparable, indicating that it is hard for a spoofer to mimic other users' gait patterns well and fool our user verification system operating the user-centric approach.

Server-centric approach. Fig. 16 plots the detection rate and false positive rate under mimic and random attacks when the number of training traces (i.e., U) is changed from 3 to 6. We also observe that similar performances can be achieved under random attacks and mimic attacks: the overall performance of our server-centric approach remains over 90 percent detection rate with less than 10 percent false positive rate in both scenarios under three smartphone placements. This further indicates our system operating server-centric approach is also robust to mimic attacks under different smartphone placements.



(a) Hip Position (b) Waist Pouch Position (c) Pocket Position

Fig. 15. User centric approach: robustness study of mimic attacks by keeping the duration of both run-time measurement trace and user profile trace as 20 seconds with normal walking speed.



(a) Hip Position

(b) Waist Pouch Position

(c) Pocket Position

Fig. 16. Server centric approach: robustness study of mimic attacks by keeping the duration of both run-time measurement trace and user profile trace as 20 seconds with normal walking speed.

6 Conclusion

In this paper, we propose a user verification system leveraging unique gait patterns derived from acceleration readings, which has the capability to mitigate user spoofing in emerging mobile healthcare systems. Our system employs readily available accelerometers embeded within smartphones instead of requiring additional hardware for user verification. It exploits the correlation relationship inherited from a user's walking traces and extracts the step cycle template that can uniquely identify each user's gait patterns. We show that our step cycle extraction technique is more accurate than existing studies. Furthermore, our step cycle interpolation component can perform robust user verification under various walking speeds. Our framework can be implemented either in smartphones (i.e., the user-centric approach) or at the back-end server (i.e., the server-centric approach). To evaluate the robustness of our system, real experiments are conducted when mobile phones are placed on different body positions (including hip pouch, waist pouch and pant pocket) with users walking at various speeds. There are total 3,048 traces collected from users' smarpthones over a period of 6 months. The extensive experimental results show that our user verification system can effectively cope with different phone placements under various walking speeds. It is also robust to both random attacks (when the spoofer does not have the knowledge of the user's walking styles) and mimic attacks (when the spoofer possesses the knowledge of the user's walking styles).

Acknowledgments

Preliminary results of this paper have been presented in part in IEEE SECON 2013 [34]. The work was supported in part by U.S. National Science Foundation Grants CNS1016303, CNS1217387, CCF1018270, CNS1016296, CNS1217379 and CNS1409767.

References

[1] M. Rabbi, S. Ali, T. Choudhury, and E. Berke, "Passive and in-situ assessment of mental and physical well-being using mobile sensors," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 385–394.

[2] M. A. Kazandjieva, J. W. Lee, M. Salath, M. W. Feldman, J. H. Jones, and P. Levis, "Experiences in measuring a human contact network for epidemiology research," in *Proc. ACM 6th Workshop Hot Topics Embedded Netw. Sensors*, 2010, p. 7.

[3] Y. Ren, J. Yang, M. C. Chuah, and Y. Chen, "Mobile phone enabled social community extraction for controlling of disease propagation in healthcare," in *Proc. IEEE 8th Int. Conf. Mobile Ad-Hoc Sensor Syst.*, 2011, pp. 646–651.

[4] N. L. Clarke and S. Furnell, "Authentication of users on mobile telephones - A survey of attitudes and practices," *Comput. Security*, vol. 24, no. 7, pp. 519–527, 2005.

[5] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, 2005, pp. ii/973–ii/976.

[6] D. Gafurov and E. Snekkenes, "Gait recognition using wearable motion recording sensors," *EURASIP J. Adv. Signal Process*, vol. 2009, p. 7, 2009.

[7] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. New York, NY, USA: Cambridge Univ. Press, 2000.

[8] Q. Li and W. Trappe, "Relationship-based detection of spoofing-related anomalous traffic in ad hoc networks," in *Proc. 3rd Annu. Conf. Sensor, Ad Hoc Commun. Netw.*, 2006, pp. 50–59.

[9] J. Yang, Y. Chen, W. Trappe, and J. Cheng, "Determining the number of attackers and localizing multiple adversaries in wireless spoofing attacks," in *Proc. IEEE Conf. Comput. Commun.*, 2009, pp. 666–674.

[10] A. Wool, "Lightweight key management for IEEE 802.11 wireless lans with key refresh and host revocation," *Wireless Netw.*, vol. 11, no. 6, pp. 677–686, 2005.

[11] X. Chen, J. Tian, Q. Su, X. Yang, and F. Y. Wang, "A secured mobile phone based on embedded fingerprint recognition systems," in *Proc. IEEE Intell. Security Informat.*, 2005, pp. 549–553.

[12] Q. Su, J. Tian, X. Chen, and X. Yang, "A fingerprint authentication mobile phone based on sweep sensor," in *Proc. 3rd Int. Conf. Pattern Recognit. Image Anal.*, 2005, pp. 295–301.

[13] P. Gupta, S. Ravi, A. Raghunathan, and N. Jha, "Efficient fingerprint-based user authentication for embedded systems," in *Proc. Design Autom. Conf.*, 2005, pp. 244–247.

[14] T. Teixeira, D. Jung, G. Dublon, and A. Savvides, "PEM-ID: Identifying people by gait-matching using cameras and wearable accelerometers," in *Proc. 3rd Int. Conf. Distrib. Smart Cameras*, 2009, pp. 1–8.

[15] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, Feb. 2006.

[16] Z. Liu and S. Sarkar, "Improved gait recognition by gait dynamics normalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 863–876, Jun. 2006.

[17] J. Jenkins and C. Ellis, "Using ground reaction forces from gait analysis: Body mass as a weak biometric," in *Proc. 5th Int. Conf. Pervasive Comput.*, 2007, pp. 251–267.

[18] K. Nakajima, Y. Mizukami, K. Tanaka, and T. Tamura, "Footprint-based personal recognition," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 11, pp. 1534–1537, Nov. 2000.

[19] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S.-M. Makela, "Identifying people from gait pattern with accelerometers," *Proc. SPIE*, vol. 5779, pp. 7–14, 2005.

[20] D. Gafurov, E. Snekkenes, and P. Bours, "Spoof attacks on gait authentication system," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 491–502, Sep. 2007.

[21] G. Bajrami, M. O. Derawi, and P. Bours, "Towards an automatic gait recognition system using activity recognition (wearable based)," in *Proc. 3rd Int. Workshop Secur. Commun. Netw.*, 2011, pp. 23–30.

[22] J. J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A. C. Cheng, "A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 734–740, May 2010.

[23] L. Dipietro, C. J. Caspersen, A. M. Ostfeld, and E. R. Nadel, "A survey for assessing physical activity among older adults," *Med. Sci. Sports Exercise*, vol. 25, no. 5, pp. 628–642, 1993.

[24] D.-H. Shih, H.-S. Chiang, B. Lin, and S.-B. Lin, "An embedded mobile ecg reasoning system for elderly patients," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 854–865, May 2010.

[25] (2012, Sep.). I do move work out and win [Online]. Available: http://itunes.apple.com/us/app/idomove-work-out-and-win/id510602229?mt=8

[26] A. Baayer, N. Enneya, and M. Elkoutbi, "Enhanced timestamp discrepancy to limit impact of replay attacks in manets," *J. Inf. Security*, vol. 3, no. 3, pp. 224–230, 2012.

[27] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication," *Pattern Recognit.*, vol. 47, no. 1, pp. 228–237, 2014.

[28] C. BenAbdelkader, R. Cutler, and L. Davis, "Stride and cadence as a biometric in automatic person identification and verification," in *Proc. 5th IEEE Int. Conf. Automatic Face Gesture Recognit.*, 2002, pp. 372–377.

[29] G. Casella and R. L.Berger, *Statistical Inference*. Pacific Grove, CA, USA: Duxbury Press, 1990.

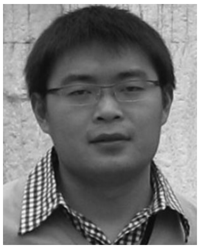
[30] R. V. Dukkipati, *Numerical Methods*. New Delhi, India: New Age International Pvt Ltd Publishers, 2010.

[31] C. Vaughan, B. Davis, and J. O'Cononor, *Dynamics of Human Gait*. Stellenbosch, South Africa: Kiboho Publishers, 1999.

[32] T. Plötz, N. Y. Hammerla, A. Rozga, A. Reavis, N. Call, and G. D. Abowd, "Automatic assessment of problem behavior in individuals with developmental disabilities," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 391–400.

[33] M. Lin, N. Lane, M. Mohammod, X. Yang, H. Lu, G. Cardone, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. T. Campbell, "A scalable approach for multidimensional wellbeing monitoring: Community and energy based adaptation of mobile sensing and feedback," presented at the Proc. Wireless Health, San Diego, CA, USA, 2012.

[34] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "Smartphone based user verification leveraging gait recognition for mobile healthcare systems," in *Proc. IEEE 10th Annu. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, 2013, pp. 149–157.



Yanzhi Ren received the BS and MS degrees from the University of Electronic Science and Technology of China, Chengdu, China, in June 2005 and June 2008, respectively. He is currently working toward the PhD degree of the Electrical and Computer Engineering Department, Stevens Institute of Technology. His research interests include mobile social networks, mobile healthcare, security and privacy in wireless networks and cloud computing security. He is currently working in the Data Analysis and Information SecuritY (DAISY) Lab with Prof. Yingying Chen. He is student member of the IEEE.

Yingying (Jennifer) Chen received the PhD degree in computer science from Rutgers University. She is an associate professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology. Her research interests include cyber security and privacy, mobile and pervasive computing, and mobile healthcare. She has published more than 80 journals and referred conference papers in these areas. Prior to joining Stevens, she was with Alcatel-Lucent. She received the US NSF CAREER Award and Google Faculty Research Award. She also received the NJ Inventors Hall of Fame Innovator Award, and the Best Paper Award from ACM International Conference on Mobile Computing and Networking (MobiCom) 2011. She also received the IEEE Outstanding Contribution Award from IEEE New Jersey Coast Section each year in 2005-2009. Her research has been reported in numerous media outlets including *MIT Technology Review*, *Wall Street Journal*, and National Public Radio. She is on the editorial boards of *IEEE Transactions on Mobile Computing (IEEE TMC)*, *IEEE Transactions on Wireless Communications (IEEE TWireless)*, and *IEEE Network Magazine*. She is senior member of the IEEE.

Mooi Choo Chuah received the PhD degree in electrical engineering from the University of California San Deigo. She is a professor in the Computer Science & Engineering Department at Lehigh University. Her research interests include designing next generation network, mobile computing, mobile healthcare, network security, and secure cyber physical systems. Prior to joining Lehigh, she was a distinguished member of Technical Staff and technical manager at Lucent Bell Laboratories, NJ. Based on her research work at Bell Laboratories, she has been awarded 62 US patents and 16 international patents related to mobility management, 3G and next generation wireless system design, etc. She has served as a technical co-chair for IEEE INFOCOM 2010, symposium co-chair for IEEE Globecom Next Generation Networking Symposium 2013 and editor of *IEEE Transaction for Mobile Computing*. She is currently the associate editor for *IEEE Transactions on Parallel & Distributed Systems*, and *Computer Networks*. She is senior member of the IEEE.

Jie Yang received the PhD degree in computer engineering from Stevens Institute of Technology in 2011. He is currently an assistant professor in the Department of Computer Science at Florida State University. His research interests include cyber security and privacy, and mobile and pervasive computing, with an emphasis on network security, smartphone security and applications, security in cognitive radio and smart grid, location systems and vehicular applications. His research is supported by the US National Science Foundation (NSF) and Army Research Office (ARO). He received the Best Paper Runner-up Award from IEEE Conference on Communications and Network Security (CNS) 2013 and the Best Paper Award from ACM MobiCom 2011. His research has received wide press coverage including *MIT Technology Review*, *The Wall Street Journal*, NPR, CNET News, and Yahoo News. He is a member of the IEEE.