Gait-Key: A Gait-Based Shared Secret Key Generation Protocol for Wearable Devices

WEITAO XU, University of Queensland
CHITRA JAVALI and GIRISH REVADIGAR, University of New South Wales
CHENGWEN LUO, Shenzhen University
NEIL BERGMANN, University of Queensland
WEN HU, University of New South Wales

Recent years have witnessed a remarkable growth in the number of smart wearable devices. For many of these devices, an important security issue is to establish an authenticated communication channel between legitimate devices to protect the subsequent communications. Due to the wireless nature of the communication and the extreme resource constraints of sensor devices, providing secure, efficient, and user-friendly device pairing is a challenging task. Traditional solutions for device pairing mostly depend on key predistribution, which is unsuitable for wearable devices in many ways. In this article, we design Gait-Key, a shared secret key generation scheme that allows two legitimate devices to establish a common cryptographic key by exploiting users' walking characteristics (gait). The intuition is that the sensors on different locations on the same body experience similar accelerometer signals when the user is walking. However, one main challenge is that the accelerometer also captures motion signals produced by other body parts (e.g., swinging arms). We address this issue by using the blind source separation technique to extract the informative signal produced by the unique gait patterns. Our experimental results show that Gait-Key can generate a common 128-bit key for two legitimate devices with 98.3% probability. To demonstrate the feasibility, the proposed key generation scheme is implemented on modern smartphones. The evaluation results show that the proposed scheme can run in real time on modern mobile devices and incurs low system overhead.

CCS Concepts: • **Security and privacy** → **Cryptography**; • **Human-centered computing** → **Ubiquitous and mobile computing**

Additional Key Words and Phrases: Device pairing, IMU sensors, secret key generation, source separation, wearable devices

ACM Reference Format:
Weitao Xu, Chitra Javali, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. 2017. Gait-Key: A gait-based shared secret key generation protocol for wearable devices. ACM Trans. Sen. Netw. 13, 1, Article 6 (January 2017), 27 pages.
DOI: http://dx.doi.org/10.1145/3023954

6

The research was partially supported by the National Natural Science Foundation of China (grant 61602319) and Natural Science Foundation of SZU (grant 2016048).
Authors' addresses: W. Xu and N. Bergmann, School of Information Technology and Electrical Engineering, University of Queensland; emails: w.xu3@uq.edu.au, n.bergmann@itee.uq.edu.au; C. Javali, G. Revadigar, and W. Hu, School of Computer Science and Engineering, University of New South Wales; emails: {chitraj, girishr, wenh}@cse.unsw.edu.au; C. Luo (corresponding author), College of Computer Science and Software Engineering, Shenzhen University; email: chengwen@szu.edu.cn.

DOI: http://dx.doi.org/10.1145/3023954

1. INTRODUCTION

During the past decade, the number of Internet of Things (IoT) devices introduced in the market has increased considerably. It is estimated that there will be 20 billion connected devices by the year 2020, the majority of which are IoT and wearable devices [Middleton et al. 2013]. With this trend, the number of connected devices per person rises dramatically. Much like the embedded systems from which they originate, on-body IoT devices are equipped with several sensors that offer a means to collect significant personal information and transmit the collected data to other personal devices. As such, secure data exchange among them becomes a significant problem. For example, smartphones need to frequently push notifications to devices such as smartwatches and read health-related sensor data from wearables or IMDs. Since these devices usually contain sensitive private information, data sharing needs to be kept strictly among devices that belong to the same user (on the same body).

The wireless nature of the communication between these devices gives rise to security problems. A malicious external device can listen to the wireless communication between legitimate on-body devices and eavesdrop private information about the user. To address this problem, conventional mechanisms rely on cryptographic keys to support the integrity and confidentiality of data communication. Specifically, two devices need to agree on a common secret key before communication, and then the established key can be used to encrypt/decrypt subsequent communications between these two parties. In dynamic mobile environments, devices need to perform peer-to-peer associations on-the-fly. However, a trusted authority for key management is not always available, making it difficult to distribute keys between legitimate devices.

In this article, we propose and implement a motion-assisted key generation technique for secure on-body device communication. The intuition of the proposed key generation approach is that the devices on the same body experience similar motion signals that are produced by the unique walking pattern of the user. Therefore, the unique gait signal can be exploited as shared information to generate secret keys for all on-body devices. Since walking is a common daily activity, human gait can be automatically detected and measured in daily life without requiring the users to perform key generation explicitly. The proposed approach enables unobtrusive establishment of secure communications between on-body devices.

1.1. Motivation

Thissection discusses the benefits offered and applications enabled by the motion-assisted key generation technique proposed in this article, such as the following:

—***On-body authentication*:** By allowing secure communication establishment only between legitimate on-body devices using the unique body motion signals, Gait-Key enables on-body device authentication without any intrusive manual assistance. Unlike state-of-the-art biometric authentication methods that use the face and fingerprints, Gait-Key reduces expensive computation and the manual user input required by conventional authentication approaches. This makes it a promising technique for lightweight continuous authentication for on-body IoT devices. This feature is desirable especially for wearable and implantable devices, which are usually small and sensor equipped, produce sensitive private data, and require frequent authentication.

—***Automatic secure pairing*:** In mobile systems, device pairing is required to agree on common encryption schemes and encryption keys before communicating data. Currently, device pairing is achieved either through explicit input (e.g., entering the key manually on the device's screen) or sophisticated peer-to-peer key-exchange algorithms.

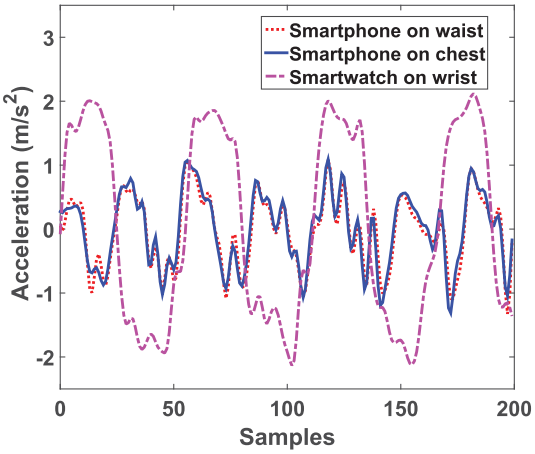


Fig. 1. Acceleration signal in the gravity direction captured by devices located at different body locations when a user is walking.

For explicit input, some common mechanisms are a personal identification number (PIN) code entry or pushing buttons on the devices to be paired. However, these manual approaches suffer from several limitations. First, the form factor of wearable devices are usually small, making it hard for users to enter the keys manually. Second, the number of pairings required is expected to grow considerably as IoT devices become increasingly pervasive. Consequently, explicit pairing places a large burden on device users, and automatic pairing improves the user experience significantly. Another approach is through a peer-to-peer key-exchange algorithm. A popular key-exchange algorithm is the Diffie-Hellman (DH) protocol [Diffie and Hellman 1976], which is used to distribute symmetric keys between two parties. However, the DH protocol requires computationally intensive operations and a public key infrastructure, and is infeasible for resource-constrained wearable devices.

—***Spontaneous key generation*:** To reduce manual input, a user can choose to store the static keys on the device locally. For example, a user can pair two devices on their first use together and use the same key afterward. However, a critical component of key management is key revocation, which is used to revoke and update the secret key. Storing static keys locally poses significant security risks, especially when devices are only authorized to communicate temporarily for short-lived data exchange. Thus, it is crucial that the keys are generated on-the-fly only when they are authorized to communicate.

1.2. Challenges and Contributions

Gait refers to an individual's unique walking pattern [Murray 1967]. The gait signal produced when a user is walking serves as a valuable signal for key generation for on-body devices, as the sensors on different body locations sense the same signal. The key idea of the proposed key generation approach is based on this observation. However, due to the complexity of body movements, devices placed on different body locations will capture different acceleration signals due to the movement of other body parts (e.g., arms), and this becomes the key challenge when exploiting the common gait signal for key generation.

Figure 1 plots the acceleration signal in the gravity direction captured by devices placed at different body locations when the user is walking. The acceleration readings on the body trunk (waist and chest) originate primarily from the walking action and

generate similar patterns. However, the sensors on the wrist capture the aggregated acceleration signal produced by both gait and arm swing. Thus, the common motion signals (caused by gait) for key generation are overwhelmed by noise (caused by the arm swing motion). This makes it infeasible to use the raw motion signals captured by the sensors to generate a common secret key directly. To address this challenge, Gait-Key uses the blind source separation (BSS) technique described in Section 4 to separate the signals produced from gait and arm swing, and the common gait signal to generate a key for secure communication for all on-body devices.

The second challenge is that the on-body devices are limited by their computational capacity and power supply. As described in Rostami et al. [2013], IMDs are long-lived devices, and battery replacement requires surgical intervention. Therefore, the pairing protocol should be lightweight and energy inexpensive. The proposed key generation scheme requires only lightweight signal processing techniques, Advanced Encryption Standard (AES) invocations, and hash computations by the on-body devices.

To the best of our knowledge, this is the first work that exploits gait signals to achieve efficient key generation and secure communication establishment for devices placed at different body locations. The main contributions of this article are threefold:

—***Source separation for body motion signal*:** By using BSS to separate motion signals generated from different body movements, such as gait and arm swing motions, the proposed key generation approach achieves robust performance in generating keys for devices on different body locations.
—***Shared key generation scheme*:** We present a novel, lightweight key generation scheme for on-body IoT devices based on body motion signals. We experimentally demonstrate that a common 128-bit key can be successfully generated by two independent wearable devices on the same body in 98.3% of cases, whereas the scheme also provides adequate security guarantees against impersonation attacks. By walking for 4.6 seconds (≈9 steps), the proposed key generation approach is able to generate a 128-bit key with entropy varying from 0.94 to 1.
—***System implementation*:** We illustrate the practicability of the proposed key generation approach by implementing the system in Bluetooth low energy (BLE) peripheral mode. We report the system computation overhead and power consumption, and demonstrate the feasibility of the proposed scheme for contemporary on-body IoT devices.

The rest of the article is organized as follows. We introduce the user model and the adversary model in Section 2. We specify the design overview in Section 3, signal processing in Section 4, and key generation in Section 5. We then evaluate the performance of the proposed scheme and analyze security issues in Section 6, and present the system implementation in Section 7. Section 8 discusses the related work, and Section 9 concludes the article.

2. MODEL

Before discussing the framework of Gait-Key, we first introduce the user model and the adversarial model.

2.1. User Model

We envision the use of Gait-Key primarily for pairing wearable and implantable devices. Figure 2 illustrates a typical user model for on-body device communication in Gait-Key. Suppose that one morning, a user wants to pair his smartwatch (Alice) with a pacemaker (Bob) to read health information. The user launches Gait-Key on the smartwatch and walks several steps, then both Alice and Bob generate a secret symmetric key by exploiting the measured gait signals during this period. The key is then used to encrypt/decrypt the messages between Alice and Bob.

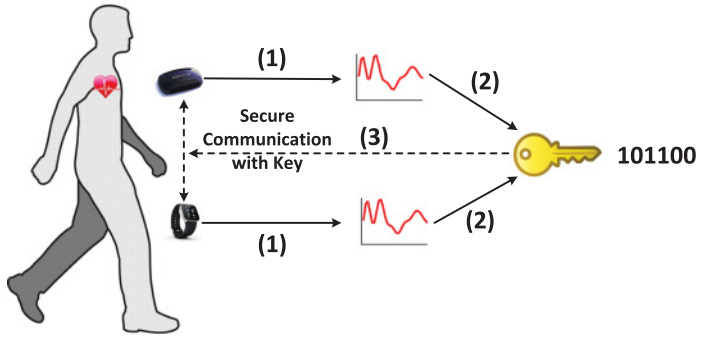


Fig. 2. A pacemaker and smartwatch measure the gait signals simultaneously and use the gait signals to generate a shared secret key. The key is then used to ensure the security of communication between two parties.

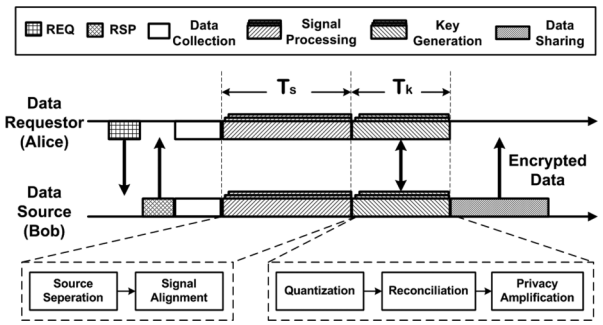


Fig. 3. Flowchart of the key generation scheme.

2.2. Adversarial Model

To achieve secure communication, a common attack that needs to be addressed is the *impersonation attack*, in which an adversary (Eve) tries to impersonate a legitimate device to steal private information. We assume the presence of two types of impersonation attack during a key generation session; a passive eavesdropping adversary and an active spoofing attack. The passive adversary knows the key generation mechanism and can eavesdrop on the messages exchanged between Alice and Bob during the key generation process. The active spoofing attacker tries to mimic the walking style of the genuine user to pair with one or both of the legitimate devices.

As discussed in Mayrhofer and Gellersen [2009], although the attacker can monitor messages exchanged between the legitimate devices, we assume that they can neither control the acceleration recorded locally by these devices nor perfectly estimate it, as otherwise the protection of legitimate devices is impossible. We also assume that all devices on the user's body are legitimate devices (i.e., an adversary cannot insert a device on the user to get the acceleration data). Further potential threats include deriving the acceleration by studying a video of the target's gait through computer vision techniques. We believe that this is a potential vulnerability of unknown severity and leave it as future work.

3. DESIGN OVERVIEW

Figure 3 shows the workflow of Gait-Key. Suppose that Alice (e.g., smartwatch) wants to read data from Bob (e.g., pacemaker). Alice first broadcasts a *REQ* request to Bob.

Table I. Summary of the Main Symbol Notations

Symbol	Meaning
$Acc(t)$	Raw linear acceleration data
A	Mixing matrix
$S(t)$	Independent components (ICs)
W	unmixing matrix
$\tilde{S}(t)$	Estimated ICs
$Acc^{'}(t)$	Reconstructed acceleration
q_+, q_-	Quantization boundaries (upper and lower)
L_{Alice}, L_{Bob}	Index list of generated bits
$\tilde{L}$	Common index list between L_{Alice} and L_{Bob}
$MAC(\cdot)$	Message authentication code algorithm
K_{Alice}, K_{Bob}	Generated key after quantization
$K^{'}_{Alice}, K^{'}_{Bob}$	Generated key after reconciliation
$K^{''}_{Alice}, K^{''}_{Bob}$	Final key after privacy amplification

After receiving the *REQ*, Bob replies with a *RSP* response. Then both Alice and Bob start to collect local motion sensor data and follow the steps shown in Figure 3 to generate a shared secret key. Finally, the key is used to encrypt/decrypt data to ensure secure communication between Alice and Bob.

The key component of Gait-Key consists of the following two steps:

—***Signal processing*:** Signal processing consists of two steps: source separation and signal alignment. Source separation is performed on the acceleration data collected from the on-body devices to extract the signals produced by gait. As Alice and Bob sample acceleration data independently, we apply signal alignment to synchronize acceleration samples at Alice and Bob and transform the acceleration to the same body coordinate system to facilitate key generation.

—***Key generation*:** The key generation component consists of three basic steps: quantization, reconciliation, and privacy amplification. In quantization, the legitimate devices, Alice and Bob, convert acceleration samples into bits if they are both on the same body. In the reconciliation stage, Alice and Bob exchange error-correcting messages over a public channel that allows them to agree on an identical string of bits. However, the publicly exchanged messages reveal a certain amount of information about the bit strings to Eve. To address this issue, Alice and Bob diminish the partial information revealed to Eve by privacy amplification.

In the following sections, we describe design details of each component. Table I summarizes the notations used in this article.

4. SIGNAL PROCESSING

4.1. Independent Component Analysis–Based Source Separation

When an individual is walking, accelerometer recordings from one body location are typically a mixture of accelerations produced from multiple body locations (e.g., leg, waist, and arm). For wearable and implantable devices, most common locations are the waist, chest, head, and wrist. As described in Section 1.2, the sensors on the body trunk measure the motion signals produced by gait primarily. Therefore, the devices on the body trunk can exploit the acceleration readings directly to generate a key. However, sensors worn on the wrists capture signals from a combination of gait and arm swing motions. To exploit the useful signal (gait) to generate a key, we need to separate signals produced from leg motions (walking) and arm swing motions.

In this article, we apply the independent component analysis (ICA) technique to separate signals from different body sources [Hyvärinen et al. 2004]. ICA is one of the most popular BSS methods, which aims to separate the mixed signals into a set of independent sources given very little information (or no prior information) about the source signals. Before applying ICA, we first justify that the on-body accelerometer satisfies the conditions for ICA. First, the acceleration from the different sources is mixed linearly at each sensor location, as we record the linear acceleration along three channels of the accelerometer sensor for each location. Second, the acceleration of arm swing is independent from that originating from heel strike. As stated in Murray [1967], the movement patterns of various parts of the body are independent, and gait is the total pattern of movement when they are integrated together. Third, time delays in signal transmission through the body are negligible. Fourth, there are fewer sources than mixtures. For each location, we attach a three-channel accelerometer sensor, and thus we have an observation of three channels and the signals are mainly from two sources: arm swing and walking. Fifth, statistical distributions of the acceleration values produced by body movement are not Gaussian [Hyvärinen 1999].

Suppose that a smartwatch is worn on one wrist of the user, and the measured linear accelerations by the built-in three-channel accelerometer are $Acc(t)$. As the accelerometer signals recorded on the wrist are a mixture of the signal from leg and arm swing, respectively, the ICA model of our problem can be written as follows:

$$Acc(t) = A \cdot S(t), \tag{1}$$

where A is the mixing matrix and $S(t)$ represents independent sources. Our aim is to find an unmixing matrix $W(W = A^{-1})$ so that we can calculate the estimated source signal $\tilde{S}(t)$ by

$$\tilde{S}(t) = W \cdot Acc(t) = W \cdot A \cdot S(t). \tag{2}$$

In this work, we use FastICA (a fast fixed-point algorithm of ICA) to solve the ICA model in Equation (1) (i.e., to estimate W). FastICA has been found to be 10 to 100 times faster than conventional gradient descent methods for ICA [Hyvärinen 1999]. Therefore, FastICA is well suited for the resource-constrained on-body devices in this work.

After obtaining W, we obtain the estimated sources $\tilde{S}(t)$ by Equation (2). In our problem, the rows of $Acc(t)$ are the linear acceleration values along three axes of the accelerometer. The acceleration signal without arm swing motion can be derived from $Acc'(t) = W\bar{S}$, where $\bar{S}$ is the matrix of derived independent components (ICs) with the row representing the arm swing set to 0. Assume that the second ICA component represents the signal from arm swing. $\bar{S}$ can then be written as follows:

$$\bar{S} = \begin{bmatrix} \tilde{S}_{11} & \tilde{S}_{12} & \cdots & \tilde{S}_{1N} \\ 0 & 0 & \cdots & 0 \\ \tilde{S}_{31} & \tilde{S}_{32} & \cdots & \tilde{S}_{3N}, \end{bmatrix}, \tag{3}$$

where $\tilde{S}_{ij}(i, j = 1, \ldots, N)$ are the elements of matrix $\tilde{S}(t)$ and N is the number of acceleration samples. In the following section, we describe how we identify different motion components.

4.2. Identifying Motion Component

From the ICA model in Equation (1), it can be seen that one cannot determine the order of the ICs, as a permutation matrix P and its inverse P^{-1} can be added in the model to yield $Acc(t) = AP^{-1}PS(t)$. The elements of $PS(t)$ are the original independent

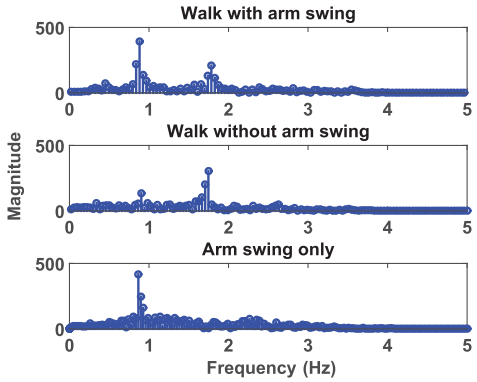


Fig. 4. Frequency of different activities.

variables, but in a different order. The matrix AP^{-1} is therefore a new unknown mixing matrix to be solved by the ICA algorithm. Furthermore, the order of components may also vary from one data segment to the next. Consequently, one has to depend on visual inspection of the ICA components for further processing, a method that is not desirable for on-body sensors.

In practice, the separated components tend to have more distinctive properties than the original signals both in time and frequency domains. Figure 4 shows the frequency of walking while swinging an arm, walking without swinging an arm, and swinging an arm only. We notice that the dominant frequency of the signal from walking only is two times that of an arm swing signal. This is easy to understand because a gait cycle is composed of two steps and one arm swing cycle. Therefore, each step (left or right) registers as a strong repetitive acceleration signal, and the signal is transmitted through the foot to the whole body. Due to the symmetry of the body, the signal produced by the left and right step can be deemed the same. However, the arm swing signal only repeats every two steps as the smartwatch is worn on one wrist of the user. We use this observation to identify the signal from the arm swing and foot. Specifically, after obtaining $\tilde{S}(t)$ by Equation (2), we perform a fast Fourier transform (FFT) on the three ICs in $\tilde{S}(t)$ (i.e., three rows of $\tilde{S}(t)$). Figure 5(d) illustrates the magnitude of the acceleration signals in three directions before ICA and after ICA. We can see that the original acceleration data contains signals from two frequencies primarily. The three separated ICs present different frequency distributions. The frequencies of IC-2 are concentrated on the fundamental frequencies. As discussed earlier, the reconstructed signal without arm swing motion can be obtained by setting the second row of the matrix $\bar{S}$ to 0 (see Equation (3)).

Figure 6 presents the acceleration in the gravity direction before and after source separation. We can see that the acceleration produced by walking is overwhelmed by arm swing in the raw acceleration signals. The acceleration after source separation is quite similar to the readings on the chest, only the magnitude of the signal is reduced, because the signal produced from leg motion is attenuated through the body to the wrist. Note that one cannot simply apply a low-pass filter to filter out the signal produced by arm swing motion, as the walking signal also contains a fundamental frequency component as shown in Figure 4.

4.3. Signal Alignment

The raw acceleration data cannot be used to generate the key directly, as the accelerometer values are sensitive to sensor orientation and location. Additionally,

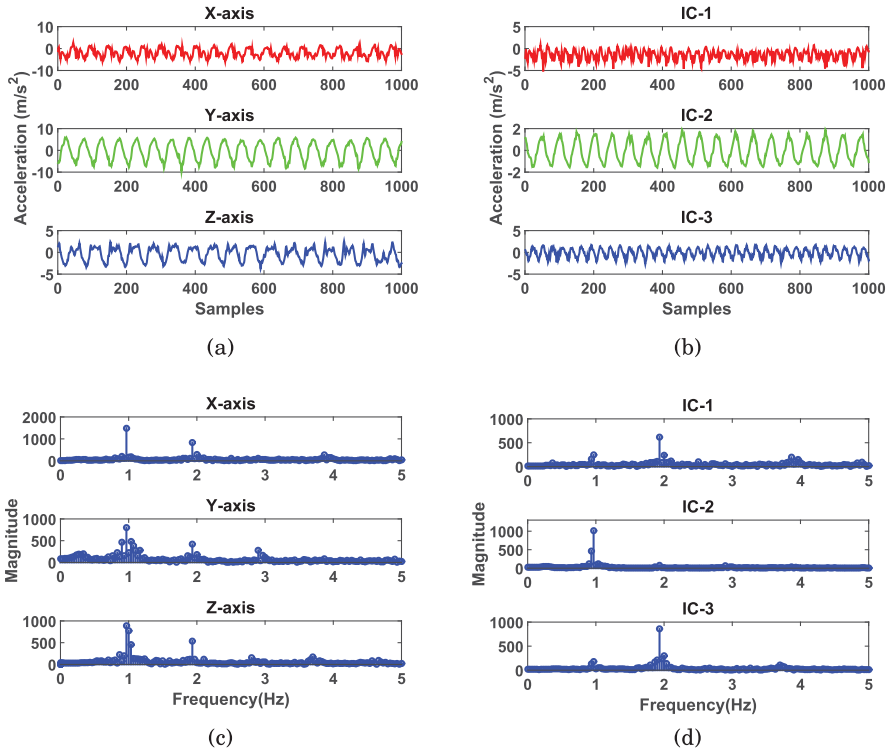


Fig. 5. ICA results. (a) Raw acceleration $Acc(t)$. (b) Estimated ICs $\tilde{S}(t)$. (c) Frequency of raw acceleration. (d) Frequency of estimated ICs.

different devices are usually not well time synchronized, which leads to the problem of signal synchronization. We address these two issues by temporal alignment and spatial alignment.

4.3.1. Temporal Alignment. As devices sample acceleration values independently, temporal synchronization is required for key generation. In this work, we use an event-based approach in which devices detect the time point of a heel-strike event, using this event as an anchor point. The intuition is that the acceleration values along gravity direction reach the peak simultaneously when the foot touches the ground, and time delays in signal transmission through the body are negligible. To detect heel strike, we first apply a low-pass filter on acceleration along the gravity direction to reduce noise. The cutoff frequency is chosen as 3Hz, as the normal step frequency lies between 1.6 and 2.8 Hz [Murray 1967]. Then the local maxima are detected to identify heel-strike events as shown in Figure 7.

Heel-strike events can be detected locally at each device without communication, which eliminates the need for explicit synchronization between devices. When Alice receives an *RSP* from Bob, both of them reach an agreement to record acceleration from the next n_{start}-th heel-strike event and end recording at the subsequent n_{end}-th heel-strike event. The acceleration samples are then transformed to the body coordinate system as described in the following section.

4.3.2. Spatial Alignment. Walking is inherently a 3D movement, and 3D acceleration data independently recorded at different locations lack spatial alignment and cannot be directly used to generate a shared secret key. We address this by transforming

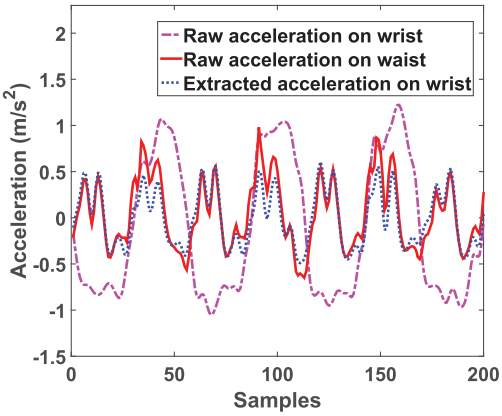


Fig. 6. Comparison of raw signal and extracted signal.

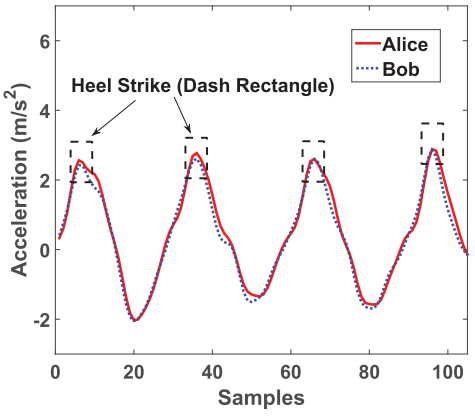


Fig. 7. Peak of acceleration along the gravity direction indicates a heel strike on the ground.

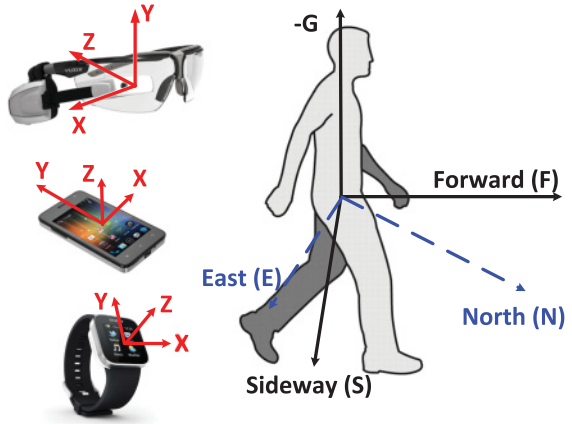
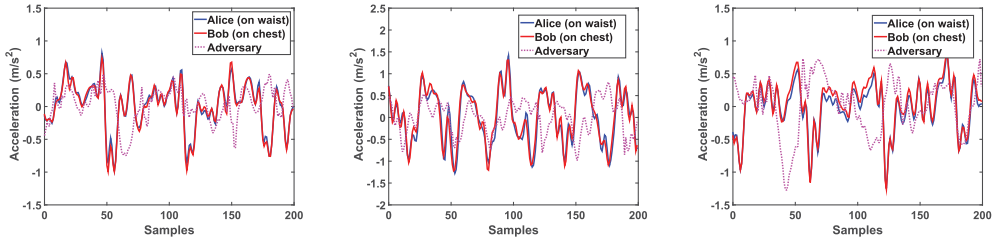


Fig. 8. Different coordinate systems.

acceleration values of different devices to a common body reference coordinate system independent of orientation and location. Figure 8 illustrates the definition of the world coordinate system, the body reference coordinate system, and the coordinate system of different devices. The world coordinate system is defined by North, East, and the Down or gravity direction (−G). We refer to the device's local coordinate system as (X, Y, Z). The user plane of motion is defined as the Forward-Sideways plane, which is perpendicular to gravity. Sideways points toward the right side of the user's forward direction.

Taking a smartphone as an example, assume that the linear acceleration signals along three orthogonal directions of a smartphone are Acc_x, Acc_y, and Acc_z, respectively. The linear acceleration in the body reference system can be computed as

$$\begin{bmatrix} Acc_G \\ Acc_F \\ Acc_S \end{bmatrix} = R_b^w \cdot R_w^d \cdot \begin{bmatrix} Acc_x \\ Acc_y \\ Acc_z, \end{bmatrix}, \tag{4}$$



(a) Acceleration in the walking direction (b) Acceleration in the gravity direction (c) Acceleration in the sideway direction

Fig. 9. Acceleration of two legitimate devices and an adversary device.

where Acc_G, Acc_F, and Acc_S are linear accelerations along the direction, forward direction, and sideways direction in the body reference system; R_b^w is the rotation matrix from the world coordinate system to the body coordinate system and can be computed by the method in Mohssen et al. [2014]; and R_w^d is the rotation matrix from the device coordinate system to the world coordinate system and can be obtained by the Android API. Note that the absolute walking direction of the user cannot be obtained accurately using a smartphone compass [Roy et al. 2014]. In Walkie-Talkie [Xu et al. 2016a], we do not have this problem because we consider the acceleration values only instead of walking direction. After obtaining the acceleration in the body coordinate system, we use Acc_G, Acc_F, and Acc_S for key generation.

5. KEY GENERATION

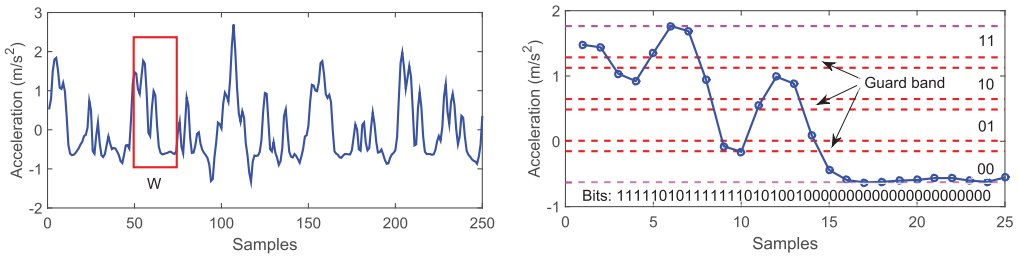
After source separation and signal alignment, we obtain acceleration values caused by gait along three directions: Acc_G, Acc_F, and Acc_S. Figure 9 plots the acceleration of two legitimate devices and an adversary device in three directions. We can see that the devices on the same body follow the same pattern; however, the acceleration signal recorded by an adversary device significantly differs. This result is promising since our goal is to generate symmetric keys only for devices on the same body. The following key generation method is applied on two legitimate devices separately.

5.1. Multilevel Quantization

We perform filtering, then quantization for the acceleration values along the three directions separately. We first apply a low-pass filter for noise reduction. The cutoff frequency is chosen as 10Hz, as the useful frequency of human motion lies below 10Hz [Lester et al. 2004]. Note that the cutoff frequency of this low-pass filter is different from that used for heel strike mentioned in Section 4.3.1. After filtering, the acceleration values are normalized to have zero mean and unit length to alleviate the influence of different body locations. Then we extract multiple bits from the accelerometer signal samples by employing a multilevel quantization technique [Zeng et al. 2010]. More specifically, we segment the acceleration values with a moving window with no overlap (window size W). Thereafter, for each window, we generate bits by the following steps.

5.1.1. Determining the Upper Bound on the Number of Bits. The first step is to determine the maximum number of bits that can be assigned per sample. For a given window, we calculate the approximate entropy of samples by using the following equation:

$$\mathcal{E} = -\sum\nolimits_a p(a) \log_2 p(a), \tag{5}$$



(a) Raw acceleration data (b) Quantization

Fig. 10. Illustration of the quantization process for $W = 25$, $m = 4$, and $\alpha = 0.2$.

where $p(a)$ is the probability of occurrence of acceleration sample a in a selected window. The upper bound of the quantization level m_{MAX} is calculated as $m_{MAX} \leq 2^{\mathcal{E}}$.

5.1.2. Determining the Quantization Intervals. In the next step, we calculate the size of each level in the quantization. In a multibit (i.e., m-ary) quantization, *guard bands* (g_i) are inserted between two consecutive levels (i.e., q_{i-1} and q_i) to increase the bit agreement ratio. The guard band samples are excluded during quantization, and the remaining samples are encoded to bits according to their levels. The notation α represents the ratio of guard band to data (i.e., the excluded acceleration values in all of the guard bands over the total number of samples). Each level in an m-ary quantization, with m being the number of levels, is represented by a number (i.e., level-0 to level-(m-1)). The individual quantization intervals are calculated by the following equations:

$$I_0 = (q_0, q_1 - g_1], I_1 = (q_1, q_2 - g_2], \ldots, I_{m-1} = (q_{m-1}, q_m], \tag{6}$$

where q_0 is the minimum and q_m is the maximum value of acceleration samples in the window. For each level, we calculate the size of the quantization interval and guard band by the following equations:

$$\int_{q_{i-1}}^{q_i - g_i} f_a da = \frac{1-\alpha}{m}, \int_{q_i - g_i}^{q_i} f_a da = \frac{\alpha}{m-1}, \tag{7}$$

where $1 \leq i \leq m-1$. Each level in the quantization is assigned an n-bit code ($n = log_2 m$). We assign the bits to each level such that their decimal value denotes the index of the level. The secret bits are then extracted from acceleration samples based on their level in the quantization.

5.1.3. Extracting the Final Key. Similar to the single bit quantization, we perform quantization for the acceleration values along the three directions separately. Three separate bit streams K_G, K_F, K_S are extracted from Acc_G, Acc_F, and Acc_S respectively, and the secret key for Alice is obtained by concatenating three bit streams together as $K_{Alice} = [K_G, K_F, K_S]$. The same quantization process is also performed by Bob independently to get K_{Bob}. Figure 10(a) plots the raw acceleration data recorded in an experiment, and Figure 10(b) illustrates the process of 2-ary quantization for a window.

5.2. Reconciliation

After quantization, each device ends up with a secret key string independently. However, there may be some bit mismatches due to noise, and we often get $K_{Alice} \approx K_{Bob}$. The purpose of reconciliation is to correct the bit mismatches between Alice and Bob. In this system, we employ the error correction code (ECC) [Clark and Cain 2013] to reduce the bit mismatch rate. The ECC method was also used in a recent secure pairing system [Lin et al. 2017].

Suppose that the mismatching bits between Alice and Bob is $\epsilon = K_{Alice} \oplus K_{Bob}$, and let $C(n, k)$ be an ECC that encodes a k-bit message into an n-bit code to resist r-bit random error. Function $f(\cdot)$ and $g(\cdot)$ denote the corresponding encoding function and decoding function. To start the reconciliation, Alice first computes the offset δ_{Alice} between K_{Alice} and its corresponding code word as follows:

$$\delta_{Alice} = K_{Alice} \oplus f(g(K_{Alice})). \tag{8}$$

Then, Alice transmits δ_{Alice} to Bob via a public channel. Upon receiving δ_{Alice}, Bob can deduce K_{Alice} as follows:

$$K'_{Alice} = \delta_{Alice} \oplus f(g(K_{Bob} \oplus \delta_{Alice})). \tag{9}$$

If the mismatching rate ϵ is lower than the error-correcting ability of C, an appropriate ECC C can be employed to ensure that $K'_{Alice} = K_{Alice}$. Therefore, both Alice and Bob agree on the same key $K'_{Alice} = K_{Alice}$, and they use the key to encrypt/decrypt the communication between them.

Since Alice and Bob do not share an authenticated channel, Eve can impersonate as Alice or Bob during the reconciliation process. Such an attack would allow Eve to insert her own fake messages, thus spoofing a legitimate device and disrupting the protocol without revealing his presence. To address this issue, we employ the message authentication code (MAC) method [Mathur et al. 2008] to verify that the message has not been modified. Specifically, the MAC method contains the following three steps:

—To ensure the message δ_{Alice} is indeed sent from Alice, Alice sends a MAC message with δ_{Alice}; the overall message sent by Alice is $L_{Alice} = \{\delta_{Alice}, MAC(K_{Alice}, \delta_{Alice})\}$. After receiving L_{Alice}, Bob computes K'_{Alice} by Equation (9) and uses it for MAC verification. If Bob obtains $MAC(K_{Alice}, \delta_{Alice}) \neq MAC(K'_{Alice}, \delta_{Alice})$, he can conclude that the message was not sent by Alice, indicating the presence of an adversary.
—If Bob does not detect the presence of an adversary, he computes δ_{Bob} and transmits the following message to Alice: $L_{Bob} = \{\delta_{Bob}, MAC(K_{Bob}, \delta_{Bob})\}$.
—Upon receiving L_{Bob}, Alice computes K'_{Bob} and uses it for MAC verification. If Alice obtains $MAC(K'_{Bob}, \delta_{Bob}) = MAC(K_{Bob}, \delta_{Bob})$, she can confirm that the message was indeed sent by Bob. Since Eve does not know the bits in K_{Bob} generated by Bob (he can just listen to the output of the $MAC(K_{Bob}, \delta_{Bob})$), modifying δ_{Bob} will fail the MAC verification at Alice.

Apart from verifying that the message has not been modified, the MAC verification also verifies whether Alice and Bob generate the same key. Because if $K'_{Alice} \neq K_{Alice}$, Bob cannot obtain $MAC(K'_{Alice}, \delta_{Alice}) = MAC(K_{Alice}, \delta_{Alice})$. In this case, the key generation process fails, and Bob will either notify Alice to restart the key generation process or consider Alice as an unauthorized device and deny all Alice's consequent requests, depending on application requirements.

The reconciliation process not only reduces the mismatch rate between Alice and Bob but also reveals partial information to an attacker, as δ_{Alice} is transmitted over a public channel and can be eavesdropped by an attacker. However, it can be theoretically proved that there are only $(n - k)$ bits of information leakage [Mathur et al. 2011]. Moreover, since the secret key is derived from a user's unique walking pattern, the attacker still cannot infer K_{Alice} by eavesdropping δ_{Alice}. To ensure that there is no partial information leakage, we can further apply the privacy amplification technique described in the following section.

5.3. Privacy Amplification

After reconciliation, Alice and Bob agree on a common secret key as $K'_{Alice} = K_{Alice}$. Simply concatenating the bits generated from each time window does not necessarily

produce a random secret key, as correlation between different steps may result in high correlation between key bits. Moreover, reconciliation leaks some information to an attacker. This issue can be addressed by privacy amplification techniques [Bennett et al. 1988]. In the system, we use a bitwise XOR function to combine keys generated from each direction and eliminate the correlation between them. Specifically, we interleave the bit streams from three directions in the time sequence and segment the concatenated keys into small windows with no overlap. Each window contains 30 bits, which is close to the bits generated in a gait cycle duration, as the evaluation results show in Section 6.5. Then we XOR two consecutive windows together to obtain the final key K''_{Alice}.

Another advantage of privacy amplification is that it diminishes the partial information revealed to Eve as discussed in Bennett et al. [1988]. In the reconciliation stage, Alice and Bob exchange messages over a public channel, and the publicly exchanged messages reveal a certain amount of information about the bit strings to Eve. To reduce the impact of the revealed information, the privacy amplification significantly improves the randomness of the keys generated, as the evaluation results show later in Section 6.5. Note that other privacy amplification methods, such as a universal hash [Bennett et al. 1988], can be employed to further enhance the randomness of the concatenated key. We refer the reader to Bennett et al. [1988] for more details.

After privacy amplification, the final key can be used by symmetric key algorithms such as AES to ensure secure communication between Alice and Bob. If the length of final key is greater than 128 bits, the first 128 bits are used.

5.4. CIA Properties of Gait-Key

As a security scheme, Gait-Key achieves CIA properties (confidentiality, integrity, and availability) by the following approaches:

—***Confidentiality*:** Data confidentiality is the key focus and is achieved through encryption after key generation.
—***Integrity*:** During key generation, integrity is achieved by the MAC; after key generation, with the key the data integrity can be easily achieved using any standard mechanisms, such as hashing and checksumming, which are beyond the scope of this article.
—***Availability (Anti-DoS attack)*:** During key generation, to prevent the adversary from modifying messages to fail the reconciliation between two legitimate devices, the MAC mechanism is used to ensure the integrity of the messages and to protect the availability of the key generation. After key generation, unauthorized communications can lead to denial-of-service (DoS) attacks, in which communications between legitimate devices are prevented and batteries are needlessly depleted [Rushanan et al. 2014]. To prevent such DoS attacks, Gait-Key only allows authorized communication through authentication achieved by the key generation techniques.

6. EVALUATION

6.1. Goals, Metrics, and Methodology

In this section, we evaluate the performance of the proposed key generation scheme. The goals of the evaluation are fourfold: (1) to determine the choice of the key parameters including the window size (W) and α in the quantization process, as well as the sampling frequency (F_s); (2) to evaluate the impact of different components in the workflow, including ICA, reconciliation, and privacy amplification; (3) to evaluate the impact of different body locations on bit agreement rate, including the head, chest, waist, and wrist; and (4) to evaluate the security of the scheme against various adversary attacks.

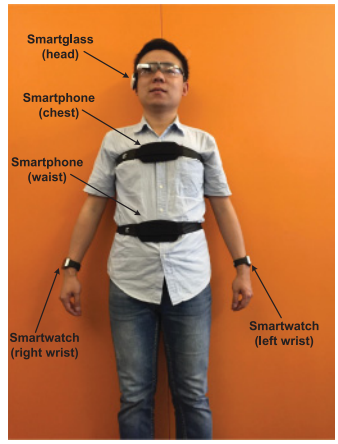


Fig. 11. Body locations for data collection.

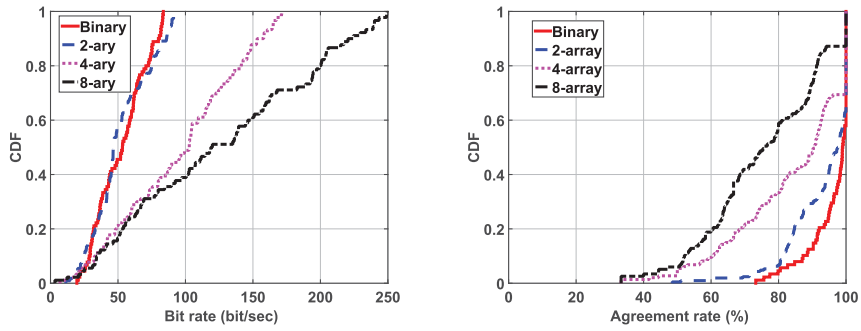
Data collection. The dataset used to evaluate the performance of the proposed system consists of 20 subjects (14 males and 6 females).[1] As shown in Figure 11, we collect acceleration data from the following body positions: head, chest, waist, and wrist. These positions represent the common locations of mobile devices and medical sensors (e.g., pacemaker). The sampling rate of all devices used in data collection is set to 100Hz.

During the data collection phase, the participants were asked to wear mobile devices as shown in Figure 11 and walk for about 5 minutes at their normal speed (0.7 to 1.1m/s). The data collection was performed both indoors and outdoors to capture different terrains in practical scenarios. Note that we do not consider data collection on different days or different walking speeds (slow, normal, and fast), as all devices worn by the subject are measuring the same gait signal simultaneously, which is different from the data collection requirements in the study of gait recognition. The detected peaks that indicate heel strikes are used to synchronize acceleration samples recorded on different devices and segment steps. For each device attached on one subject, we break the continuous acceleration values into segments according to heel-strike points; each segment contains 10 steps. The segments are used to generate keys and evaluate the following metrics.

Metrics. For a shared key generation protocol, we focus on the following three evaluation metrics:

—***Bit agreement rate*:** The bit agreement rate represents the percentage of bits matching in the secret keys generated by two parties. This metric evaluates the potential of Alice and Bob agreeing on the same key.
—***Bit rate*:** The bit rate denotes the average number of bits generated from the acceleration samples per unit time and is usually measured in bits per second (bits/sec). This metric evaluates how fast Alice and Bob can generate shared secret bits.
—***Entropy*:** Entropy is the measure of uncertainty or randomness associated with the generated bit strings. Entropy of a binary bit string varies in the range $[0, 1]$, and larger entropy indicates more randomness of the bit string.

[1]Ethical approval for carrying out this experiment has been granted by the corresponding organization (Approval Number HC15304).



(a) CDF of bit generation rate (b) CDF of bit agreement rate

Fig. 12. Binary quantization versus m-ary quantization.

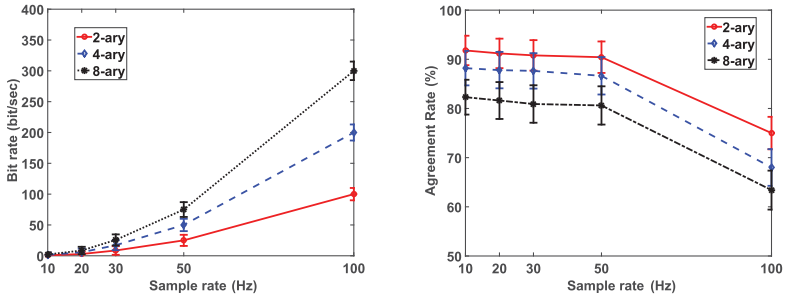
We examine the impact of parameters on the generated key by a systematic exhaustive search. We vary the respective parameters within a dedicated range (i.e., $W = 5, 10, \ldots, 50$, $\alpha = 0, 0.1, \ldots, 1$, and $F_s = 10, 20, 30, 50, 100$). The goal of the exhaustive search is to find the optimal combinations that achieve good performance in both bit agreement rate and bit rate. After choosing the best combination ($W = 50, \alpha = 0.9, F_s = 50$), we take turns to investigate the impact of each parameter on agreement rate and bit rate by fixing the other two parameters. Results are presented for the average values and 95% confidence levels of the performance metrics (bit agreement rate and bit rate).

6.2. Improvement of Multilevel Quantization over Binary Quantization

Since m-ary quantization can be used to generate keys with more bits, we compare its performance with the binary quantization method used in Walkie-Talkie [Xu et al. 2016a]. For evaluation purposes, we vary m from 2 to 8. Figure 12(a) plots the CDF of the bit generation rate under different methods. The "Binary" means the method used in Walkie-Talkie [Xu et al. 2016a], and the others indicate the method described in Section 5.1. Compared to the binary quantization method, the higher-level m-ary quantization can significantly increase the bit generation rate. Figure 12(b) is the CDF of the bit agreement rate between legitimate devices corresponding to the keys of Figure 12(a). Different from the bit generation rate, the bit agreement ratio decreases when higher quantization levels are used. This is because noise will produce more bit mismatches when the quantization level increases. The experimental results suggest that multilevel quantization can significantly increase the bit rate while decreasing the bit agreement rate. We also tried quantization levels larger than 8, which yield even a lower bit agreement ratio, so we limit our discussion to $m = 2$, 4, and 8 in this work.

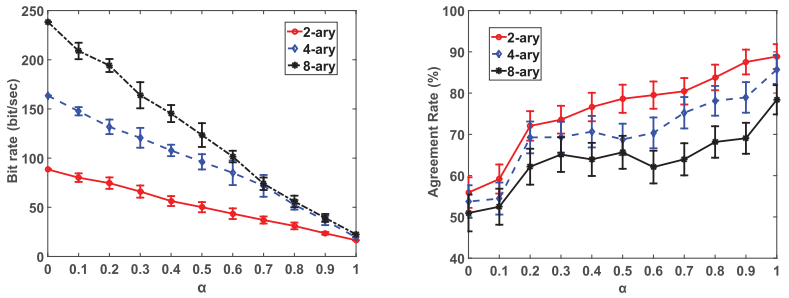
6.3. Parameter Selection

6.3.1. Impact of Sampling Rate. As mentioned previously, the initial sampling rate is 100Hz. We evaluate the impact of different sampling rates on the bit rate and bit agreement rate by downsampling F_s from 100Hz to 50Hz, 30Hz, 20Hz, and 10Hz, respectively. Figure 13(a) and (b) show the impact of F_s on the bit rate and bit agreement rate, respectively. We can see that the agreement rate between legitimate devices varies inversely with the sampling rate. The reason is that a higher sampling rate is able to record more acceleration values during the same period and thus improves the bit rate; however, it reduces bit agreement, as a higher sampling rate captures acceleration variation in more detail, leading to less similarity between legitimate devices.



(a) Impact of F_s on bit rate (b) Impact of F_s on agreement rate

Fig. 13. Impact of F_s.



(a) Impact of α on bit rate (b) Impact of α on agreement rate

Fig. 14. Impact of α.

Table II. Comparison of Different ECCs

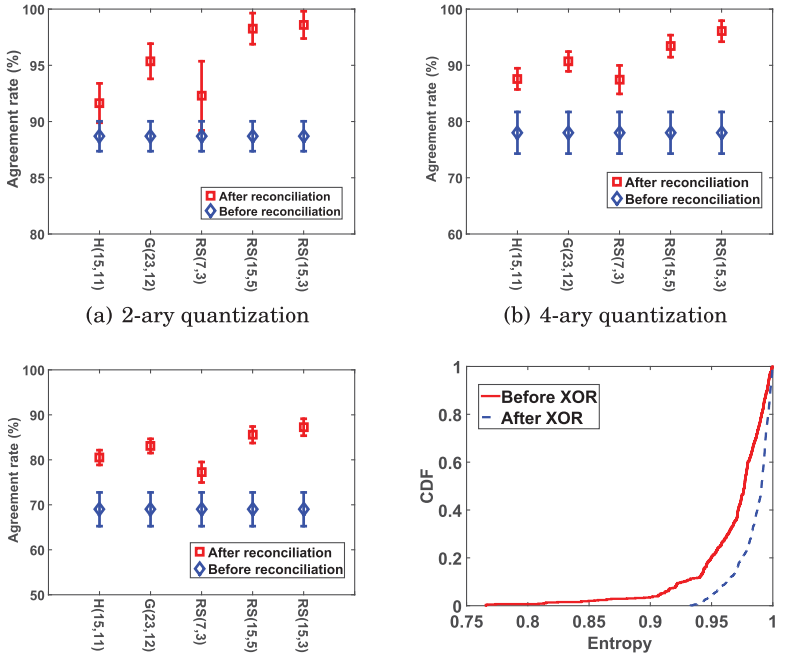
Code	n	k	r	Information Leakage
Hamming code	15	11	1	0.27
Golay code	23	12	3	0.48
RS(7,3)	7	3	2	0.57
RS(15,5)	15	5	5	0.67
RS(15,3)	15	3	6	0.8

6.3.2. Impact of α. We evaluate the impact of α to explore the trade-off between the agreement rate and bit rate. Figure 14(a) shows that the bit rate decreases as α increases. This is because the parameter α in Equation (7) decides the decision band to include or discard the acceleration measurements. A larger α means that more acceleration readings are discarded. This reduces the length of generated keys and decreases the bit rate. On the other hand, as shown in Figure 14(b), the bit agreement rate increases with increasing α because more mismatches in the decision band are excluded.

Apart from the sampling rate and α, we also investigated the impact of different window sizes when generating keys. We found that the moving window size W does not have much influence on the performance and that a moving window with a size of 50 is adequate for the proposed system.

6.4. Impact of Reconciliation

Reconciliation is used to correct errors between Alice's and Bob's keys. We examine the effectiveness of different ECCs under different quantization levels. The candidate ECC codes are the Hamming code, Golay Code, and Reed-Solomon (RS) code. Table II lists



(a) 2-ary quantization (b) 4-ary quantization

(c) 8-ary quantization (d) Impact of privacy amplification

Fig. 15. Evaluation results.

Table III. Comparison of Different Quantization Levels

	Bit Rate (bit/sec)	Time to Tenerate a 128-Bit Key	Probability of 100% Match
2-ary quantization	28	4.6s	98.3%
4-ary quantization	37	3.5s	92.4%
8-ary quantization	43	3s	72.1%

the parameters and properties of ECCs used in our evaluation (code word length n, code length k, error-correcting ability r). Figure 15(a) through (c) show the impact of ECCs on the agreement rate under different quantization levels, respectively. We can see a significant increase in the bit agreement rate after using the reconciliation technique. From the figures, we also find that an RS code with n =15, k = 3 achieves the highest bit agreement rate. One drawback of the reconciliation process is that it reveals some information to attackers, and this issue is solved by the privacy amplification process.

According to the preceding results, we choose RS(15,3) in our system and use it for the rest of evaluation. After determining the ECC, we examine the bit rate and match rate of different quantization levels. From Table III, we can see that a fast key generation rate is at the expense of the bit agreement rate. Overall, 2-ary quantization is a proper choice, and it can generate a common 128-bit key for two legitimate devices with 98.3% probability.

6.5. Improvement of Key Randomness with Privacy Amplification

We now examine how the XOR function in privacy amplification helps to enhance the randomness of the final key. Figure 15(d) shows the entropy of the final key after privacy amplification. From the results, we can see that the distribution of entropy is

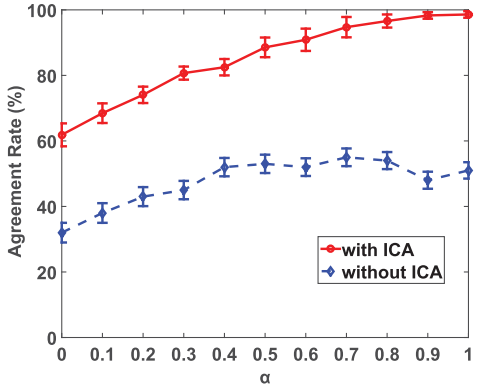


Fig. 16. Impact of ICA.

closer to 1 after the XOR operation. We also notice that the entropy of the final keys varies from 0.94 to 1, which in turn indicates that the proposed method can extract secret keys with good entropy. Note that a drawback of using the XOR function is that the bit rate is reduced by a factor of 2 (we XOR two consecutive windows together). According to the results in Table III, the bit rate of 2-ary quantization can still achieve 28 bit/sec after privacy amplification.

6.6. Improvement of the Bit Agreement Rate with ICA

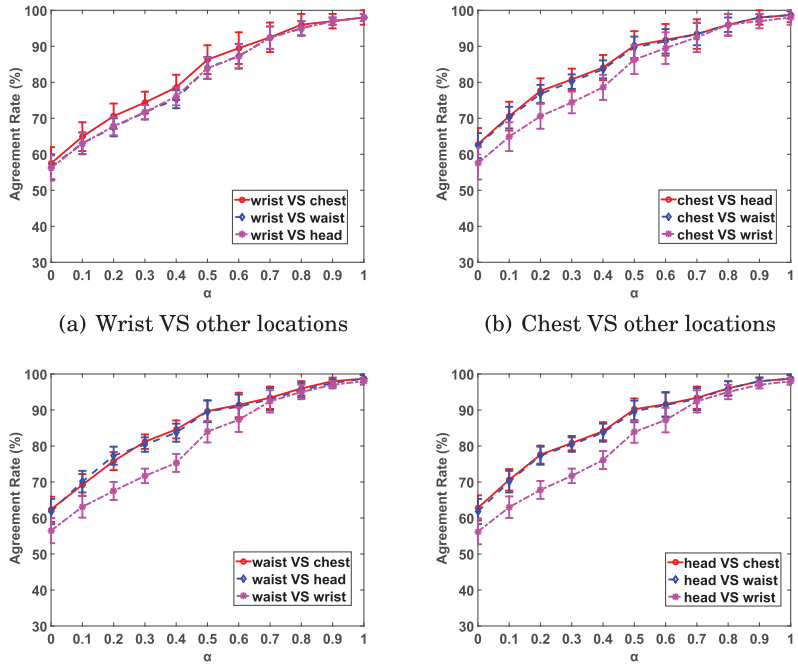
We examine whether the application of ICA can improve the agreement rate. As ICA is applied on acceleration signals recorded from the smartwatch only, we compute the bit agreement rate between keys generated from the smartwatch and devices placed at other locations by using raw acceleration values (without ICA) and extracted acceleration values (with ICA), respectively. From the results in Figure 16, we can see a significant improvement in the agreement rate after ICA. The maximum agreement rate of using raw acceleration values (without ICA) is near 50%, which is like a random guess between 0 and 1. The results suggest that applying ICA can extract walking signals from arm swing signals effectively and thus improve the agreement rate significantly.

6.7. Bit Agreement Rate of Devices on Different Body Parts

We evaluate how well the proposed method performs for each body part: wrist, chest, waist, and head. For each body part, we compare the keys generated from other locations with the keys generated from this location. For example, in terms of the wrist, we calculate the agreement rate by comparing the keys generated from the wrist with keys generated from other locations (e.g., waist, chest, and head), respectively. As shown in Figure 17, we notice that the pairs of waist-to-chest and chest-to-head achieve the best agreement rate. This result is intuitive, as sensors on the body trunk observe acceleration more similarly than sensors on the limbs.

6.8. Randomness of the Final Key

Guaranteeing that the generated keys are random is crucial because they are intended for use as a cryptographic key. To validate the randomness of the final key, we apply the NIST suite of statistical tests [Rukhin et al. 2001] to all the keys generated from our dataset. The NIST statistical test gives the p-values of different random test processes, and the p-values indicate the probability that the key sequence is generated by a random process. Conventionally, if p-value is less than 1%, the randomness hypothesis



(a) Wrist VS other locations

(b) Chest VS other locations

(c) Waist VS other locations

(d) Head VS other locations

Fig. 17. Bit agreement rate of different body parts.

Table IV. *p*-Values of NIST Statistical Testing

NIST Test	*p*-Value
Frequency	0.712248
FFT Test	0.557416
Longest Run	0.022491
Linear Complexity	0.380014
Block Frequency	0.978452
Cumulative Sums	0.986105
Approximate Entropy	0.996418
Non-Overlapping Template	0.332475

is rejected, which means that the key is not random. From Table IV, we can see that the *p*-values are all greater than 1% in the sense that the generated keys pass the random tests.

6.9. Security Analysis

We assume the presence of a passive adversary (eavesdropper) and an active attacker during an authentication session. The eavesdropper can listen to all communication between Alice and Bob and knows the bit generation algorithm. The active attacker has complete communication control (i.e., can jam, forge, and modify messages). Additionally, the adversary may mimic the walking style of the genuine user and start new protocol instances by injecting appropriate authentication request messages with multiple legitimate devices in parallel. We evaluate the robustness of the proposed system against the eavesdropper and active attacker by conducting the following two imposter attempt experiments:

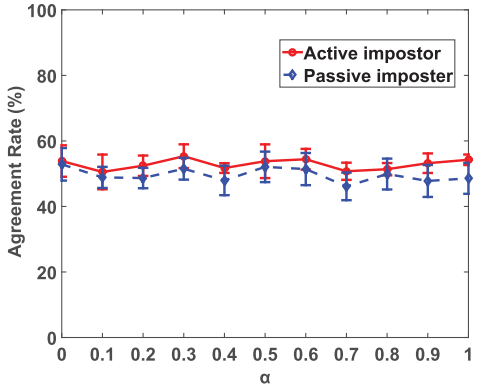


Fig. 18. Agreement rate of impostors.

Table V. Mutual Information Among Different Devices

	Alice Versus Bob	Alice Versus Active Attacker
Mutual Info. (bits)	1.42	0.21

—A passive impostor attempt is an attempt when an attacker tries to pair his device to a legitimate device by submitting his own walking signals.
—An active impostor attempt mimics the gait of the genuine user with the aim to pair with the devices of the genuine user.

The first experiment is conducted to evaluate the robustness to a passive imposter. For each location of one subject, we use the keys generated from the same location but from other subjects as passive imposter attempts. We then repeat this experiment by testing all locations of the 20 subjects in the dataset. To evaluate the robustness against the second imposter attack scenario, we group the 20 subjects into 10 pairs. Each subject was told to mimic his or her partner's walking style and try to imitate him or her. First, one participant of the pair acted as an attacker, the other one as a target, and then the roles were exchanged. The genders of the attacker and the target were the same. They observed the walking style of the target visually, which can be easily done in a real-life situation, as gait cannot be hidden. Every attacker made five active impostor attempts. Figure 18 plots the bit agreement rate of the passive imposter and active imposter, and we find that the agreement rate of an active attacker is slightly higher than that of a passive attacker, but there is no regular pattern for the agreement rate when α varies from 0 to 1. This phenomenon can be explained by two facts. For one, the unique walking pattern of the genuine user is difficult to mimic, and even an active attacker cannot produce similar walking patterns of the user. Therefore, an attacker cannot achieve a high agreement rate. Moreover, the RS code may introduce more mismatching bits if the number of mismatching bits exceeds the correcting ability due to its nonlinear nature.

To further quantify the amount of information that can be inferred from mimicking the gait, we calculate pairwise mutual information among different devices in Table V. We find that the legitimate devices on the same body can obtain 1.42 bits of information about the secret key. However, the active attacker can only get 0.21 bits of information. This result suggests that the legitimate devices obtain six times more information about each other than the attacker.

The individual nature of the walking gait provides our scheme security against passive eavesdroppers. Even if an active imposter can observe and try to mimic the

Table VI. System Overhead Measured on a Moto E2

	Computation Time (ms)	Energy Consumption (mJ)
ICA	105.7	71.2
Component Identification	2.6	1.5
Key Generation	310.5	125.6
AES Encryption	0.2	0.1
AES Decryption	0.2	0.1
Total	419.2	198.5

walking style of the target, the results in Figure 18 show that he or she still cannot obtain a common secret key. However, an active attacker can impersonate Alice or Bob in the reconciliation stage and insert false values. Gait-Key prevents such an attack by the MAC method described in Section 5.2. A further concern to all key agreement protocols is the man-in-the-middle (MITM) attack. An MITM attack against our scheme rarely occurs, as Alice and Bob exchange the offset (δ_{Alice} and δ_{Bob}) only instead of shared key during the reconciliation stage. Therefore, the shared key will not be compromised by MITM. Even if an active attacker who can obtain an approximately 50% agreement rate conducts a brute-force attack, he or she still cannot guess the same key, as the active attacker has no information about which bits are correct. Even a normal guesser can obtain a 50% agreement rate, as a cryptographic key contains 0 and 1 only. Therefore, he or she still needs 2^{128} attempts to guess the same 128-bit key, which is infeasible in real-world scenarios.

7. SYSTEM IMPLEMENTATION

To validate the feasibility of the proposed key generation approach on wearable devices, we implemented the whole system using an Android OS application.[2] The system is implemented in Java, and the implementation of FastICA is based on the Fastica Java library. The MAC algorithm described in Section 5.2 is implemented by keyed-hash message authentication code (HMAC-MD5). The sampling rate of the accelerometer is set as 50Hz, and BLE functionality is employed for wireless communication.

BLE is designed to provide significantly lower power consumption for devices with low power requirements. It introduces a new feature called *peripheral mode*, in which the data source can advertise and publish data without requiring to pair with the data requestor before hand. BLE peripheral mode is designed for devices with resource constraints and need to publish new data frequently. Therefore, we run the system in peripheral mode and advertise the data using broadcast packets. Bob organizes its data using the Generic Attribute Profile (GATT) and encrypts the data to publish by AES. All devices nearby, including adversaries, can receive the broadcast advertisements and read the publicly available data from Bob. However, only Alice on the same body can generate the same key for data decryption. In this way, the private data is protected from reading by unauthorized devices.

Table VI presents the system overhead (computation and energy consumption) of our system on a Moto E2 smartphone, which supports BLE peripheral mode. The computation time and energy consumption of each component are measured by averaging the results from running ICs separately and continuously for 5 minutes. Note that we do not consider the time for data collection (i.e., walking duration). The major components in Gait-Key—the source separation (including ICA and component identification) and key generation—take an average time of 108.3ms and 310.5ms, respectively. When the scheme is fully employed, the computation time and energy consumption are 419.2ms and 198.5mJ, respectively. The battery capacity of the Moto E2 smartphone

[2]A video demonstration of the system can be found at https://www.youtube.com/watch?v=YBFBJrNZy48.

is 2,390mAh (30.1kJ); therefore, the energy cost of Gait-Key amounts to 0.005‰ of the total energy supply. We assume the smartphone with a targeted lifespan of 1 day, which results in an energy budget of 1.25kJ per hour. To put this into perspective, with 5% of the budget per hour (62.5J), Gait-Key is capable of running approximately 317 times per hour (i.e., Gait-Key can continuously run every 12 seconds). These results demonstrate that the proposed key generation approach has a low system overhead and can run in real time on modern mobile devices.

8. RELATED WORK

In this section, we review the related work in the literature.

Applications of ICA. ICA has been successfully applied in numerous areas, such as biomedical signal processing [Srivastava et al. 2005] and speech separation [Schmidt and Olsson 2006]. De Moor et al. [2007] proposed using ICA to decompose maternal and fetal electrocardiograms recorded simultaneously from cutaneous electrodes placed on the mother's abdomen and chest. Other researchers have also applied ICA to remove artifacts from electroencephalogram signals [Srivastava et al. 2005; Delorme and Makeig 2004]. Other examples from the biomedical area are the studies by Calhoun et al. [2009] and McKeown and Sejnowski [1998], in which ICA was applied to functional magnetic resonance imaging data to separate different active components. In the speech separation area, ICA is used for extracting the interested speech signals from mixed signals [Schmidt and Olsson 2006; Liu et al. 2014]. The application of ICA on body sensor networks is an emerging field. Lo et al. [2006] applied ICA on body sensor signals to separate different sources of movement due to running and respiration. Atallah et al. [2009] used the ICA technique to detect walking gait impairment with an ear-worn sensor. In a work by Pendharkar et al. [2014], ICA was applied on accelerometer sensor attached on the heel to distinguish toe-walking gait from normal gait in idiopathic toe walker (ITW) children. In our study, we use ICA to separate accelerometer signals from different body movements such as arm swing and walk.

Key generation system for on-body devices. Many techniques exist that could be used to generate a shared secret key between two parties by exploiting the wireless channel information. Some of the examples are security mechanisms based on physical-layer characteristics. the received signal strength indicator (RSSI) has been proposed by researchers [Revadigar et al. 2015a, 2015b, 2016; Revadigar et al. 2015c; Javali et al. 2014; Shi et al. 2013]. However, these schemes are suitable for wearable devices that are frequently exchanging wireless packets. It is worth mentioning that Gait-Key utilizes several techniques used in physical-layer key extraction systems, such as the multilevel quantization method by Zeng et al. [2010] and the MAC method by Mathur et al. [2008]. In their work, they explore how to generate keys in wireless networks by using RSSI. The goal of our system is similar to their work in the sense that all of these systems aim to generate identical bit strings between two parties based on two correlated processes. However, we address a different problem in this work—how to generate keys for wearable devices by using gait. The potential of using acceleration to generate a shared key has not been well explored in the literature. The prior work that probably has the closest relation to ours is the study by Bichler et al. [2007], in which the researchers developed a method to generate a shared key based on acceleration data of shaking devices together.

Authentication system for on-body devices. Several previous works have used accelerometers to determine whether the devices are worn on the same body. Cornelius and Kotz [2012] proposed using coherence to analyze the similarity of acceleration signals from different devices and then decide whether two devices are carried on the same body. Compared to the research of Lester et al. [2004] and Cornelius and Kotz [2012], our work is significantly distinguished by exploiting gait information to

generate a secret key. The idea of shaking two devices together to pair them was first proposed by Holmquist et al. [2001]. Mayrhofer and Gellersen [2003] used the same technique but extended it to include secure authentication. Hinckley [2003] developed a similar method to pair devices that uses bumping rather than shaking together. These methods require the user to participate and shake/move the devices together, which is not suitable for many on-body devices, such as a pacemaker. Xu et al. [2017] proposed a gait-based authentication system by using kinetic energy harvesting. To automatically authenticate devices, other physical information can be used, such as accurate indoor locations [Luo et al. 2016a, 2016b]. The proposed scheme in this article can improve the user experience significantly, as walking is a normal activity, and two devices can be paired automatically when the user is walking.

Biometric-based authentication system. In identity management [Chen et al. 2011; Zheng et al. 2015], biometric recognition is the science of establishing the identity of a person using his or her anatomical and behavioral traits [Jain et al. 2008]. In this article, we have addressed a different problem (key generation) by using a biometric gait. Our work belongs to biometric cryptosystems (BCSs), which were developed for the purpose of either securing a cryptographic key using biometric features or directly generating a cryptographic key from biometric features. State-of-the-art BCSs proposed previously mostly utilize physiological modalities, such as the iris [Marino et al. 2012], face [Xu et al. 2016b], and fingerprint [Li et al. 2012]. Some studies have used behavioral biometrics such as signature [Maiorana 2010] and voice [Carrara and Adams 2010]. To the best of our knowledge, gait has not been well explored in BCS. In a similar work, Hoang and Choi [2014] used gait to encrypt a cryptographic key through a fuzzy commitment scheme [Juels and Wattenberg 1999]. In contrast, gait is explored to generate a cryptographic key directly in our work.

9. CONCLUSION

In this article, we propose and implement a key generation approach that exploits the acceleration signals produced by gait to establish a common cryptographic key between two legitimate devices. By exploiting BSS and incorporating a multilevel quantization mechanism, Gait-Key demonstrates superior effectiveness in performance. For example, when 2-ary quantization is employed, Gait-Key can generate a common 128-bit key for two legitimate devices in 4.6 seconds with 98.3% probability. Increasing quantization levels can improve the bit generation rate but will decrease the bit agreement rate. We also analyze the security against various attackers. The proposed method obtains a security advantage from the fact that different people have distinctive walking styles. Finally, we prototype the proposed scheme on the Moto E2 smartphone to demonstrate the feasibility on contemporary mobile devices.

ACKNOWLEDGMENT

We sincerely thank the anonymous reviewers for their insightful comments.

REFERENCES

Louis Atallah, Omer Aziz, Benny Lo, and Guang-Zhong Yang. 2009. Detecting walking gait impairment with an ear-worn sensor. In *Proceedings of the BSN Conference (BSN'09)*. IEEE, Los Alamitos, CA, 175–180.

Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. 1988. Privacy amplification by public discussion. *SIAM Journal on Computing* 17, 2, 210–229.

Daniel Bichler, Guido Stromberg, Mario Huemer, and Manuel Löw. 2007. *Key Generation Based on Acceleration Data of Shaking Processes*. Springer.

Vince D. Calhoun, Jingyu Liu, and Tülay Adali. 2009. A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data. *Neuroimage* 45, 1, S163–S172.

Brent Carrara and Carlisle Adams. 2010. You are the key: Generating cryptographic keys from voice biometrics. In *Proceedings of the PST Conference (PST'10)*. IEEE, Los Alamitos, CA, 213–222.

Jianyong Chen, Guihua Wu, and Zhen Ji. 2011. Secure interoperation of identity managements among different circles of trust. *Computer Standards and Interfaces* 33, 6, 533–540.

George C. Clark Jr. and J. Bibb Cain. 2013. *Error-Correction Coding for Digital Communications*. Springer Science & Business Media.

Cory T. Cornelius and David F. Kotz. 2012. Recognizing whether sensors are on the same body. *Pervasive and Mobile Computing* 8, 6, 822–836.

B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel. 1997. DAISY: A database for identification of systems. *Journal A* 38, 3, 4–5.

Arnaud Delorme and Scott Makeig. 2004. EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods* 134, 1, 9–21.

Whitfield Diffie and Martin E. Hellman. 1976. New directions in cryptography. *IEEE Transactions on Information Theory* 22, 6, 644–654.

Ken Hinckley. 2003. Synchronous gestures for multiple persons and computers. In *Proceedings of the the UIST Conference (UIST'03)*. ACM, New York, NY, 149–158.

Thang Hoang and Deokjai Choi. 2014. Secure and privacy enhanced gait authentication on smart phone. *Scientific World Journal* 2014, Article No. 438254.

Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W. Gellersen. 2001. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Proceedings of Ubicomp (Ubicomp'01)*. 116–122.

Aapo Hyvärinen. 1999. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks* 10, 3, 626–634.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. 2004. *Independent Component Analysis*. Vol. 46. John Wiley & Sons.

Anil K. Jain, Karthik Nandakumar, and Abhishek Nagar. 2008. Biometric template security. *EURASIP Journal on Advances in Signal Processing* 2008, 113.

Chitra Javali, Girish Revadigar, Lavy Libman, and Sanjay Jha. 2014. SeAK: Secure authentication and key generation protocol based on dual antennas for wireless body area networks. In *Proceedings of the RFID Workshop (RFIDsec'14)*.

Ari Juels and Martin Wattenberg. 1999. A fuzzy commitment scheme. In *Proceedings of the CCS Conference (CCS'99)*. ACM, New York, NY, 28–36.

Jonathan Lester, Blake Hannaford, and Gaetano Borriello. 2004. "Are you with me?"—using accelerometers to determine if two devices are carried by the same person. In *Pervasive Computing*. Lecture Notes in Computer Science, Vol. 3001. Springer, 33–50.

Peng Li, Xin Yang, Hua Qiao, Kai Cao, Eryun Liu, and Jie Tian. 2012. An effective biometric cryptosystem combining fingerprints with error correction codes. *Expert Systems with Applications* 39, 7, 6562–6574.

Yang Lin, Wang Wei, and Zhang Qian. 2017. Secret from muscle: Enabling secure pairing with electromyography. In *Proceedings of the Sensys Conference (Sensys'17)*. ACM, New York, NY.

Junliang Liu, Fengqin Yu, and Ying Chen. 2014. Speech separation based on improved fast ICA with kurtosis maximization of wavelet packet coefficients. In *New Perspectives in Information Systems and Technologies*. Vol. 1. Springer, 43–50.

Benny Lo, Fani Deligianni, and Guang-Zhong Yang. 2006. Source recovery for body sensor network. In *Proceedings of the BSN Conference (BSN'06)*. IEEE, Los Alamitos, CA, 1–4.

Chengwen Luo, Long Cheng, Mun Choon Chan, Yu Gu, Jianqiang Li, and Zhong Ming. 2016a. Pallas: Self-bootstrapping fine-grained passive indoor localization using WiFi monitors. *IEEE Transactions on Mobile Computing* PP, 99, 1–14.

Chengwen Luo, Hande Hong, Long Cheng, Mun Choon Chan, Jianqiang Li, and Zhong Ming. 2016b. Accuracy-aware wireless indoor localization: Feasibility and applications. *Journal of Network and Computer Applications* 62, 128–136.

Emanuele Maiorana. 2010. Biometric cryptosystem using function based on-line signature recognition. *Expert Systems with Applications* 37, 4, 3454–3461.

R. Alvarez Marino, F. Hernandez Alvarez, and L. Hernandez Encinas. 2012. A crypto-biometric scheme based on iris-templates with fuzzy extractors. *Information Sciences* 195, 91–102.

Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. 2011. Proximate: Proximity-based secure pairing using ambient wireless signals. In *Proceedings of the MobiSys Conference (MobiSys'11)*. ACM, New York, NY, 211–224.

Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. 2008. Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel. In *Proceedings of the MobiCom Conference (MobiCom'08)*. ACM, New York, NY, 128–139.

Rene Mayrhofer and Hans Gellersen. 2009. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing* 8, 6, 792–806.

Martin J. McKeown and Terrence J. Sejnowski. 1998. Independent component analysis of fMRI data: Examining the assumptions. *Human Brain Mapping* 6, 5–6, 368–372.

Peter Middleton, Peter Kjeldsen, and Jim Tully. 2013. Forecast: The Internet of Things, worldwide, 2013. Retrieved December 20, 2016, from https://www.gartner.com/doc/2625419/forecast-internet-things-world wide

Nesma Mohssen, Rana Momtaz, Heba Aly, and Moustafa Youssef. 2014. It's the human that matters: Accurate user orientation estimation for mobile computing applications. In *Proceedings of the MobiQuitous Conference (MobiQuitous'14)*. 70–79.

M. Pat Murray. 1967. Gait as a total pattern of movement: Including a bibliography on gait. *American Journal of Physical Medicine and Rehabilitation* 46, 1, 290–333.

Gita Pendharkar, Ganesh R. Naik, and Hung T. Nguyen. 2014. Using blind source separation on accelerometry data to analyze and distinguish the toe walking gait from normal gait in ITW children. *Biomedical Signal Processing and Control* 13, 41–49.

Girish Revadigar, Chitra Javali, Hassan Asghar, Kasper Rasmussen, and Sanjay Jha. 2015a. Mobility independent secret key generation for wearable health-care devices. In *Proceedings of the BodyNets Conference (BodyNets'15)*.

Girish Revadigar, Chitra Javali, Hassan Asghar, Kasper Rasmussen, and Sanjay Jha. 2015b. *Secret Key Generation for Body-Worn Devices by Inducing Artificial Randomness in the Channel*. nical Report UNSW-CSE-TR-201506. UNSW, Australia.

Girish Revadigar, Chitra Javali, Wen Hu, and Sanjay Jha. 2015c. DLINK: Dual link based radio frequency fingerprinting for wearable devices. In *Proceedings of the LCN Conference (LCN'15)*.

Girish Revadigar, Chitra Javali, Weitao Xu, Wen Hu, and Sanjay Jha. 2016. Secure key generation and distribution protocol for wearable devices. In *Proceedings of the PerCom Workshop (PerCom Workshops'16)*. IEEE, Los Alamitos, CA, 1–4.

Masoud Rostami, Ari Juels, and Farinaz Koushanfar. 2013. Heart-to-heart (H2H): Authentication for implanted medical devices. In *Proceedings of the CCS Conference (CCS'13)*. ACM, New York, NY, 1099–1112.

Nirupam Roy, He Wang, and Romit Roy Choudhury. 2014. I am a smartphone and I can tell my user's walking direction. In *Proceedings of the MobiSys Conference (MobiSys'14)*. ACM, New York, NY, 329–342.

Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. 2001. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Technical Report. DTIC Document.

Michael Rushanan, Aviel D. Rubin, Denis Foo Kune, and Colleen M. Swanson. 2014. SoK: Security and privacy in implantable medical devices and body area networks. In *Proceedings of the SP Symposium (SP'14)*. IEEE, Los Alamitos, CA, 524–539.

Mikkel N. Schmidt and Rasmus Kongsgaard Olsson. 2006. Single-channel speech separation using sparse non-negative matrix factorization. In *Proceedings of the INTERSPEECH Conference (INTERSPEECH'06)*.

Lu Shi, Jiawei Yuan, Shucheng Yu, and Ming Li. 2013. ASK-BAN: Authenticated secret key extraction utilizing channel characteristics for body area networks. In *Proceedings of the WiSec Conference (WiSec'13)*.

G. Srivastava, S. Crottaz-Herbette, K. M. Lau, G. H. Glover, and V. Menon. 2005. ICA-based procedures for removing ballistocardiogram artifacts from EEG data acquired in the MRI scanner. *Neuroimage* 24, 1, 50–60.

Weitao Xu, Guohao Lan, Qi Lin, Sara Khalifa, Neil Bergmann, Mahbub Hassan, and Hu Wen. 2017. KEH-Gait: Towards a mobile healthcare user authentication system by kinetic energy harvesting. In *Proceedings of the NDSS Conference (NDSS'17)*.

Weitao Xu, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. 2016a. Walkie-Talkie: Motion-assisted automatic key generation for secure on-body device communication. In *Proceedings of the IPSN Conference (IPSN'16)*. IEEE, Los Alamitos, CA, 1–12.

Weitao Xu, Yiran Shen, Neil Bergmann, and Wen Hu. 2016b. Sensor-assisted face recognition system on smart glass via multi-view sparse representation classification. In *Proceedings of the IPSN Conference (IPSN'16)*. IEEE, Los Alamitos, CA, 1–12.

Kai Zeng, Daniel Wu, An Chan, and Prasant Mohapatra. 2010. Exploiting multiple-antenna diversity for shared secret key generation in wireless networks. In *Proceedings of the IEEE INFOCOM Conference (INFOCOM'10)*. IEEE, Los Alamitos, CA, 1–9.

Hongying Zheng, Quan Yuan, and Jianyong Chen. 2015. A framework for protecting personal information and privacy. *Security and Communication Networks* 8, 16, 2867–2874.

Received July 2016; revised October 2016; accepted November 2016