Walkie-Talkie: Motion-Assisted Automatic Key Generation for Secure On-Body Device Communication

Weitao Xu[1,4], Girish Revadigar[2,4], Chengwen Luo[3,2], Neil Bergmann[1], Wen Hu[2]
[1]School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, Australia
Email: {w.xu3}@uq.edu.au {n.bergmann}@itee.uq.edu.au
[2]School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
Email: {girishr, wenh}@cse.unsw.edu.au
[3]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
Email: {csluocw}@gmail.com
[4]CSIRO Data61, Australia

***Abstract*—The ubiquity of wearable and implantable devices has sparked a new set of mobile computing applications that leverage the abundant information from sensors. For many of these applications, ensuring the security of communication between legitimate devices is a crucial problem. In this paper, we design *Walkie-Talkie*, 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, the accelerometer also captures motion signals produced by other body parts (e.g., swinging arms). It is shown that a Blind Source Separation (BSS) technique can extract the informative signal produced by the unique gait patterns. Our experimental results show that the keys generated by two independent devices on the same body are able to achieve up to a 100% bit agreement rate. 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.**

***Index Terms*—Secret key generation, Source separation, Wearable devices**

I. Introduction

With recent advances in wireless sensor networks and embedded computing technologies, on-body smart devices such as smartphones and smart watches have become increasingly popular and play significant roles in our daily lives. On the other hand, Implantable Medical Devices (IMDs) that can be used for continuous monitoring and treatment of chronic medical disorders are also becoming increasingly commonplace in the health-care sector to provide better health care services to patients. With the rise of these on-body IoT devices, secure data exchange among them becomes a significant problem. For example, smartphones need to frequently push notifications to devices such as smart watches, 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 paper, 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.

A. Motivation

This section discusses the benefits offered and applications enabled by the motion-assisted key generation technique proposed in this paper.

- **On-body Authentication.** By allowing secure communication establishment only between legitimate on-body devices using the unique body motion signals, Walkie-Talkie enables on-body device authentication without any intrusive manual assistance. Unlike state-of-the-art biometric authentication

978-1-5090-0802-5/16/$31.00 ©2016 IEEE

methods that use face and fingerprints, Walkie-Talkie reduces expensive computation as well as the manual user input required by conventional authentication approaches. This makes it a promising technique for light-weight *continuous authentication* for on-body IoT devices. This feature is desirable especially for wearable and implantable devices, which are usually *small*, *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*.
  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 [1], 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, e.g., user can pair two devices on their first use together and use the same key afterwards. 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. So it is crucial that the keys are generated on-the-fly only when they are authorized to communicate.

B. Challenges and Contributions

Gait refers to an individual's unique walking pattern [2]. The gait signal produced when a user is walking serves as a valuable signal for key-generation for on-body devices, since 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 (such as 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 is 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, Walkie-Talkie uses the Blind Source Separation technique described in Section IV to separate the signals produced from gait and arm swing, and use the common gait signal to generate key for secure communication for all on-body devices.

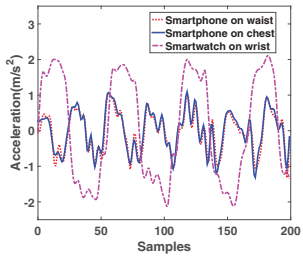


Fig. 1: Acceleration signal in the gravity direction captured by devices located at different body locations when a user is walking.

The second challenge is that the on-body devices are limited by their computational capacity and power supply. As described in [3], 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 paper are threefold:

- **Source separation for body motion signal**: By using Blind Source Separation to separate motion signals generated from different body movements, e.g., gait and arm swing motions, the proposed key generation approach achieves robust performance in generating keys for devices located at different body locations.
- **Shared key generation scheme**: We present a novel, light-weight key generation scheme for on-body IoT devices based on body motion signals. We experimentally demonstrate that the keys generated on two independent wearable devices on the same body can achieve up to 100% bit agreement rate, while the scheme also provides adequate security guarantees against impersonation attacks. By walking for 5s ($\approx$10 steps), the proposed key generation approach is able

to generate a 128-bit key with entropy varying from 0.93 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 paper is organized as follows. We introduce the user model and the adversary model in Section II. We specify the design overview in Section III, signal processing in Section IV, and key generation in Section V respectively. We then evaluate the performance of the proposed scheme and analyze security issues in Section VI, and present the system implementation in Section VII. Section VIII discusses the related work, and finally Section IX concludes the paper.

II. Model

Before discussing the framework of Walkie-Talkie, we first introduce the user model and the adversarial model.

A. User Model

We envision the use of Walkie-Talkie primarily for pairing wearable and implantable devices. Figure 2 illustrates a typical user model for on-body device communication in Walkie-Talkie. One morning, a user wants to pair his smart watch (Alice) with pacemaker (Bob) to read health information. The user launches Walkie-Talkie on the smart watch and walks several steps, and 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.

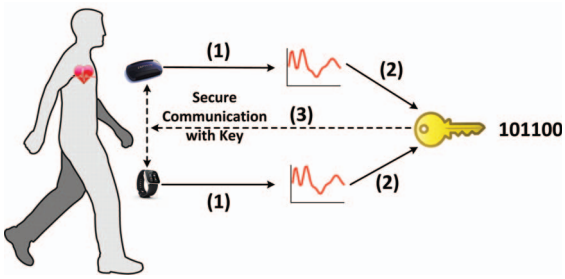


Fig. 2: Pacemaker and smart watch measure the gait signals simultaneously and use the gait signals to generate a shared secret key. The key is then used to ensure the security of communication between two parties.

B. 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 [4], 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, otherwise the protection of legitimate devices is impossible. We also assume that all the 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 this is a potential vulnerability of unknown severity and leave it as future work.

III. Design Overview

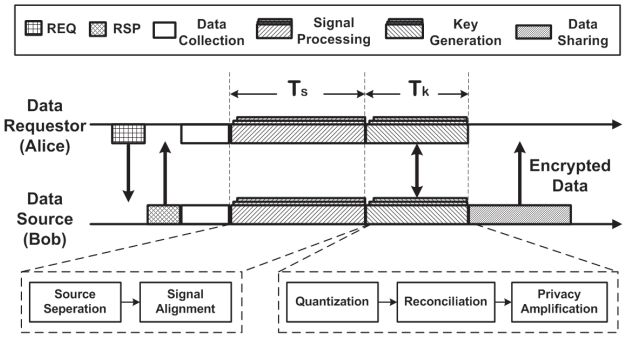


Fig. 3: Flowchart of the key generation scheme.

Figure 3 shows the work-flow of Walkie-Talkie. Suppose Alice (e.g., smart watch) wants to read data from Bob (e.g., pacemaker). Alice first broadcasts a *REQ* request to Bob. 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 Walkie-Talkie 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 will describe design details of each component. Table I summarizes the notation used in this paper.

TABLE I: A summary of the main symbol notations.

Symbol	Meaning
$Acc(t)$	raw linear acceleration data
A	mixing matrix
$S(t)$	independent components
W	unmixing matrix
$\tilde{S}(t)$	estimated independent components
$Acc^{'}(t)$	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

IV. Signal Processing

A. ICA-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 waist, chest, head and wrist. As described in Section I-B, 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. In order 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 paper, we apply independent component analysis (ICA) technique to separate signals from different body sources [5]. ICA is one of the most popular blind source separation (BSS) methods, which aims to separate the mixed signals into a set of independent sources (ICs) given very little information (or no prior information) about the source signals. Before applying ICA, we first justify that on-body accelerometer satisfies the conditions for ICA. 1) The acceleration from the different sources is mixed linearly at each sensor location, as we record the linear acceleration along 3 channels of the accelerometer sensor for each location. 2) The acceleration of arm swing is independent from that originating from heel strike. As stated in [2], the movement patterns of various parts of the body are independent, and gait is the total pattern of movement when they are integrated together. 3) Time delays in signal transmission through the body are negligible. 4) There are fewer sources than mixtures. For each location, we attach a 3-channel accelerometer sensor, thus we have an observation of 3 channels and the signals are mainly from two sources: arm swing and walking. 5) Statistical distributions of the acceleration values produced by body movement are not Gaussian [6].

Suppose a smart watch 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:

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

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 paper, we use FastICA (A fast fixed-point algorithm of independent component analysis) to solve the ICA model in Eq. 1, i.e., to estimate W. FastICA has been found to be 10-100 times faster than conventional gradient descent methods for ICA [6]. 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 Eq. 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 with the row representing the arm swing set to zero. Assume the second ICA component represents the signal from arm swing. $\bar{S}$ can then be written as:

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

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.

B. Identifying Motion Component

From the ICA model in Eq. 1, it can be seen that one cannot determine the order of the independent components, 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 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 which 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

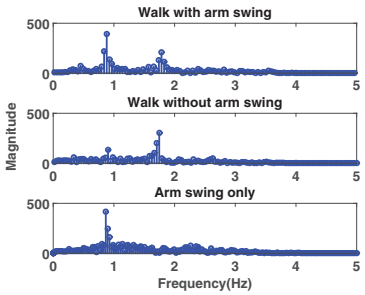


Fig. 4: Frequency of different activities.

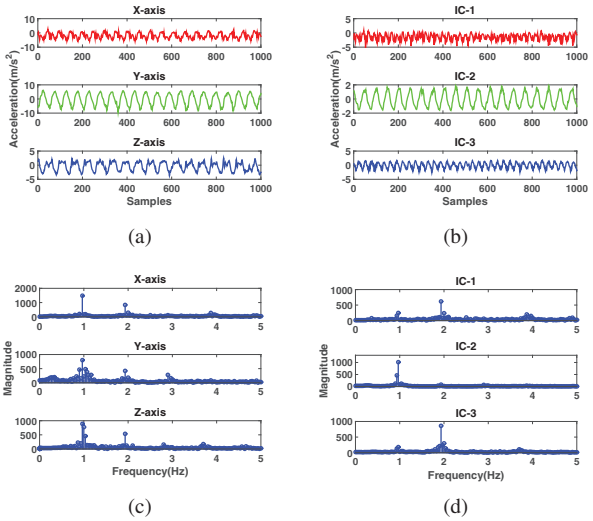


Fig. 5: ICA results: (a) Raw acceleration $Acc(t)$. (b) Estimated independent components $\tilde{S}(t)$. (c) Frequency of raw acceleration. (d) Frequency of estimated independent components.

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 of that of 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 left and right step can be deemed to be same. However, the arm swing signal only repeats every two steps as the smart watch is worn on one wrist of the user. We use this observation to identify the signal from arm swing and foot. Specifically, after obtaining $\tilde{S}(t)$ by Eq. 2, we perform a Fast Fourier Transform (FFT) on the three independent components (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 independent components (ICs) present different frequency distributions. The frequencies of IC-2 are concentrated on the fundamental frequencies. As discussed above, the reconstructed signal without arm swing motion can be obtained by setting the second row of the matrix $\bar{S}$ to zero (see Eq. 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 very similar to the readings on the chest, just 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 because the walking signal also contains a fundamental frequency component as shown in Figure 4.

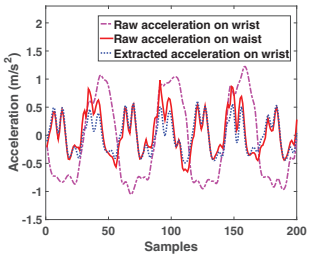


Fig. 6: Comparison of raw signal and extracted signal.

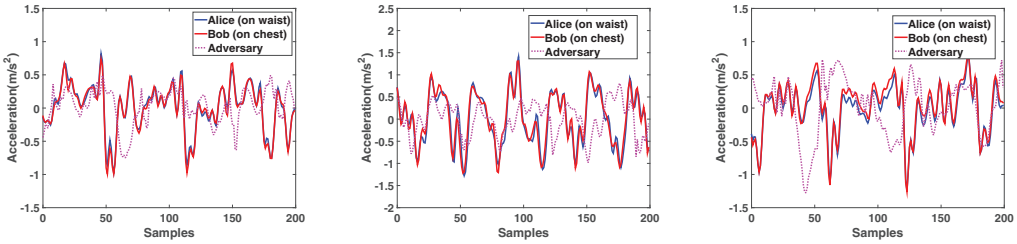
C. Signal Alignment

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

1) Temporal Alignment: As devices sample acceleration values independently, temporal synchronization is required for key generation. In this paper, we use an event-based approach in which devices detect the time point of a heel-strike event, and use 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 1.6-2.8 Hz [2]. 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 a *RSP* from Bob, both of them reach to 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.

2) Spatial Alignment: Walking is inherently a three-dimensional movement. 3D acceleration data independently



(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.

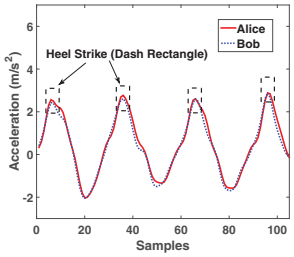


Fig. 7: The peak of acceleration along the gravity direction indicates a heel-strike on the ground.

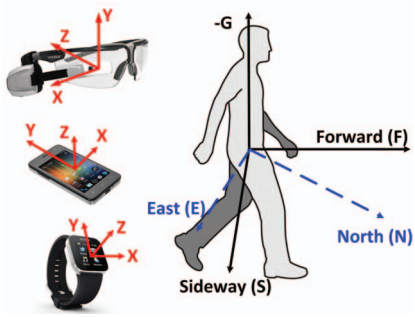


Fig. 8: The different coordinate systems

recorded at different locations lack spatial alignment and cannot be directly used to generate a shared secret key. We address this by transforming 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 the linear acceleration signals along three orthogonal directions of smartphone are Acc_x, Acc_y, and Acc_z respectively, the linear acceleration in the body reference system can be computed as:

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

where Acc_G, Acc_F, and Acc_S are linear accelerations along gravity 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 [7]. 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 [8]. In Walkie-Talkie we don't 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.

V. 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.

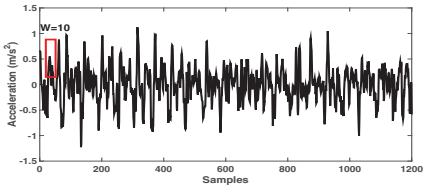
A. Quantization

We perform filtering, and 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 10 Hz [9]. Note that the cutoff frequency of this low-pass filter is different from that used for heel-strike mentioned in Section IV-C1. After filtering, the acceleration values are normalized to have zero-mean and unit length to alleviate the influence of different body locations. Then we employ the bit extraction mechanism described in [10], [11] to convert the acceleration values to bits. More specifically, we segment the

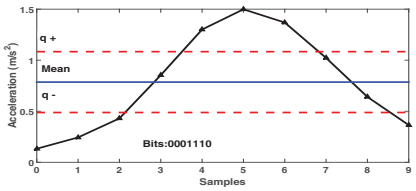
acceleration values with a moving window with no overlap (window size $W = 10$). Thereafter, for each window, we calculate two thresholds q_+ and q_- as follows:

$$q_+ = \mu + \alpha * \sigma, \quad q_- = \mu - \alpha * \sigma \tag{5}$$

where μ and σ are the mean and standard deviation of acceleration values in a particular window. Then the acceleration samples $> q_+$ are encoded as bit 1, and samples $< q_-$ are encoded as bit 0. The key is then extracted by combining the bits of each window together. The quantization process for each window is explained in Figure 10.



(a)



(b)

Fig. 10: Quantization process: (a) A sample of acceleration time series. (b) Secret bit extraction in a window of 10 samples.

At the end of this process, 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}.

B. Reconciliation

In practice, there may be some bit mismatches due to noise and we often get $K_{Alice} \approx K_{Bob}$. This is because the samples discarded during quantization at one party may be different from those at another. To increase the bit agreement rate, Alice and Bob need to exchange the sample indexes of generated bits. Only the bits corresponding to common sample indices are retained by both the parties to get the same key. For the sake of illustration, assume Alice and Bob each have 10 measurements. After quantization, Alice obtains "10110" and Bob yields "1001" (the length of K_{Alice} and K_{Bob} is not necessary to be same). Assume Alice generates bits at positions 1, 2, 3, 5, and 7, respectively. She sends $L_{Alice} = [1, 2, 3, 5, 7]$ to Bob. Bob observes these positions in his list and finds the bits are extracted at positions 1, 2, 4, and 5. He sends $L_{Bob} = [1, 2, 4, 5]$ back to Alice. Then they use the bits at positions 1,2 and 5 to generate the same key as $K^{'}_{Alice} = K^{'}_{Bob}$ ="101".

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 proposed in [12] to verify that the message has not been modified. Specifically, the MAC method contains the following three steps:

- To ensure the L_{Alice} is indeed sent from Alice, Bob computes the fraction of indexes in L_{Alice} that lies in L_{Bob}. If this fraction is less than $0.5 + \epsilon$ for some fixed ϵ ($\epsilon = 0.3$ in our system), Bob concludes 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 $K^{'}_{Bob}$ and replies to Alice with a message $\tilde{L}_{Bob} = \left\{\tilde{L}, MAC(K^{'}_{Bob}, \tilde{L})\right\}$, where $\tilde{L}$ contains those indexes lying in both L_{Alice} and L_{Bob}, and $MAC(\cdot)$ represents a message authentication code (MAC) algorithm [13].
- Upon receiving $\tilde{L}_{Bob}$, Alice computes $K^{'}_{Alice}$ and uses it for MAC verification. If Alice obtains $MAC(K^{'}_{Alice}, \tilde{L}) = MAC(K^{'}_{Bob}, \tilde{L})$, 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 $MAC(K^{'}_{Bob}, \tilde{L})$), modifying $\tilde{L}_{Bob}$ 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 $K^{'}_{Alice} \neq K^{'}_{Bob}$, Alice cannot obtain $MAC(K^{'}_{Alice}, \tilde{L}) = MAC(K^{'}_{Bob}, \tilde{L})$. In this case, the key generation process fails, and Alice will either notify Bob to restart the key generation process, or consider Bob as an unauthorized device and deny all Bob's consequent requests, depending on application requirements.

C. Privacy Amplification

After reconciliation, Alice and Bob agree on a common secret key as $K^{'}_{Alice} = K^{'}_{Bob}$. 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. This issue can be addressed by privacy amplification techniques [14]. In the system, we use a bit-wise 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 VI-D. Then we XOR two consecutive windows together to obtain the final key $K^{''}_{Alice}$ and $K^{''}_{Bob}$.

Another advantage of privacy amplification is that it diminishes the partial information revealed to Eve as discussed

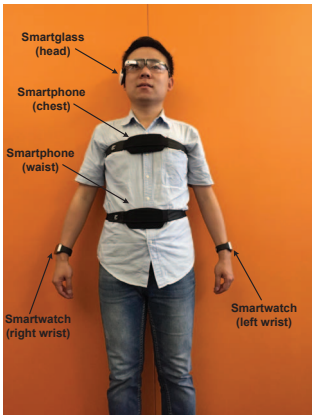


Fig. 11: Body locations for data collection.

in [14]. 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 in Section VI-D. Note that other privacy amplification methods such as a universal hash [14] can be employed to further enhance the randomness of the concatenated key. We refer the reader to [14] 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.

VI. Evaluation

A. 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 work-flow including ICA, reconciliation, and privacy amplification; 3) to evaluate the impact of different body locations on bit agreement rate including head, chest, waist, and wrist; 4) to evaluate the security of the scheme against various adversary attacks.

a) 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 100 Hz.

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 (0.7-1.1m/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 the 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 which 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.

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

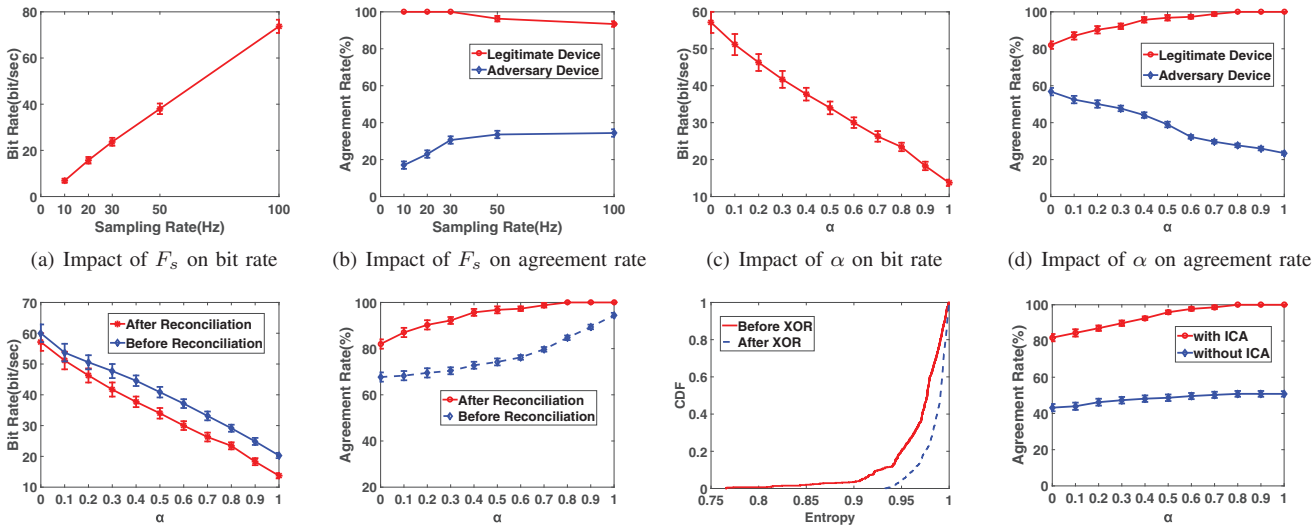
- **Bit agreement rate**: this represents the percentage of bits matching in the secret keys generated by two parties. This metric evaluates the potential of Alice and Bob agreeing on the same key.
- **Bit rate**: This denotes the average number of bits generated from the acceleration samples per unit time and is usually measured in bits per second (bps). This metric evaluates how fast Alice and Bob can generate shared secret bits.
- **Entropy**: This 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.

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 which concurrently maximize the agreement rate. After choosing the best combination ($W = 10, \alpha = 0.8, F_s = 30$), 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).

B. Parameter Selection

1) Impact of Sampling Rate: As mentioned above, the initial sampling rate is 100Hz. We evaluate the impact of different sampling rates on bit rate and bit agreement rate by downsampling F_s from 100Hz to 50Hz, 30Hz, 20hz and 10 Hz respectively. Figure 12(a) and Figure 12(b) show the impact of F_s on bit rate and bit agreement rate respectively. We can see that the agreement rate between legitimate devices varies inversely with sampling rate. The reason is that a higher sampling rate is able to record more acceleration values during the same period and thus improve 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.

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



(a) Impact of F_s on bit rate (b) Impact of F_s on agreement rate (c) Impact of α on bit rate (d) Impact of α on agreement rate

(e) Impact of reconciliation on bit rate (f) Impact of reconciliation on agreement rate (g) Impact of privacy amplification (h) Impact of ICA

Fig. 12: Evaluation results.

2) Impact of α: We evaluate the impact of α to explore the tradeoff between agreement rate and bit rate. Figure 12(c) shows that the bit rate decreases as α increases. This is because the parameter α in Eq. 5 decides the decision band to include or discard the acceleration measurements. A larger α means 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 12(d), the bit agreement rate increases with increasing α because more mismatches in the decision band are excluded. However, we notice that α has inverse impact on agreement rate for an adversary device. This is because more samples are discarded for quantization at all the devices when α increases. Therefore, the legitimate devices know the index numbers used and they exchange the index list during reconciliation, so the agreement rate increases. However, for the adversary, as the signal values itself are different from that of legitimate devices, the remaining extracted bits after dropping more samples in quantization will have lower matching rates.

Apart from 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 a moving window with size of 10 is adequate for the proposed system.

C. Impact of Reconciliation

Reconciliation is used to correct errors between Alice's and Bob's keys. Figure 12(e) and Figure 12(f) show the impact of reconciliation on the bit rate and agreement rate respectively. We can see a significant increase in the bit agreement rate after using reconciliation technique. One drawback of the reconciliation process is that it reduces bit rate as shown in Figure 12(e). Because the primary goal of a key generation protocol is to generate a shared common key, reconciliation is necessary for the proposed key generation approach.

D. Improvement of Key Randomness with Privacy Amplification

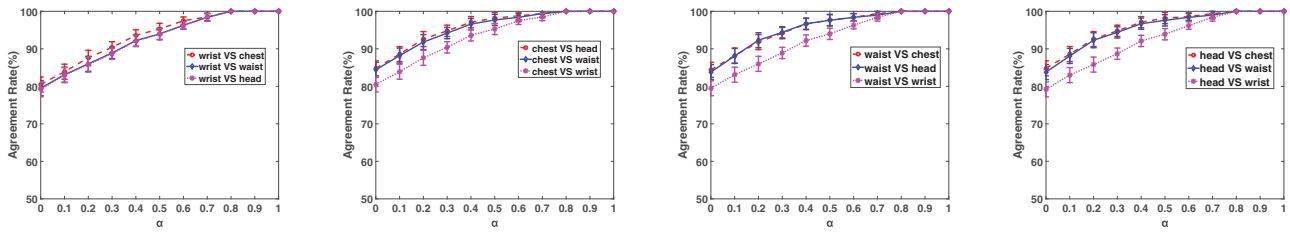
We now examine how the XOR function in privacy amplification helps to enhance the randomness of the final key. Figure 12(g) shows the entropy of the final key after privacy amplification. From the results, we can see that the distribution of entropy is closer to 1 after the XOR operation. We also notice that the entropy of the final keys varies from 0.93 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). As the results in Figure 12(e) show, the bit rate can still achieve 26 bit/sec after reconciliation and privacy amplification.

E. Improvement of 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 smart watch only, we compute the bit agreement rate between keys generated from smart watch 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 12(h), we can see a significant improvement in 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.

F. 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



(a) Wrist VS other locations (b) Chest VS other locations (c) Waist VS other locations (d) Head VS other locations

Fig. 13: Bit agreement rate of different body parts.

of wrist, we calculate the agreement rate by comparing the keys generated from wrist with keys generated from other locations (e.g., waist, chest, and head) respectively. As shown in Figure 13, 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. We also investigate the bit agreement rate of devices on different sides of the body. We compare the keys generated from one side of the body (left wrist) with keys generated from the other side of the body (right wrist). From the results in Figure 14, we find that the agreement rate of devices on different sides can still achieve 100% after performing source separation to extract the useful signals.

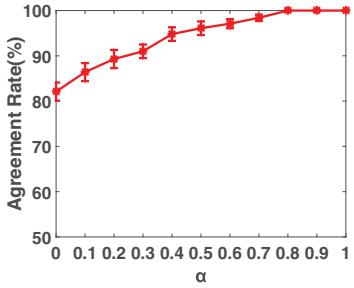


Fig. 14: Left wrist VS right wrist.

G. Randomness of the Final Key

Guaranteeing that the generated keys are random is crucial because they are intended for use as a cryptographic key. In order to validate the randomness of the final key, we apply the NIST suite of statistical tests [15] 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 is rejected which means the key is not random. From Table II, we can see that the p-values are all greater than 1% in the sense that the generated keys pass the random tests.

H. 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 the communications between Alice and Bob and knows the bit generation algorithm. The active attacker has complete communication control, i.e. can

TABLE II: P-values of NIST Statistical Test.

NIST Test	p-value
Frequency	0.606072
FFT Test	0.562699
Longest Run	0.027173
Linear Complexity	0.386887
Block Frequency	0.984496
Cumulative Sums	0.974180
Approximate Entropy	0.995898
Non Overlapping Template	0.302941

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.

- 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 the 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/her partner's walking style and try to imitate him or her. Firstly, 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 5 active impostor attempts. Figure 15 plots the bit agreement rate of passive imposter and active imposter, we can see that the agreement rate of passive impostor attempt is below 30% when $\alpha > 0.8$. Although an active imposter can improve the agreement rate significantly by mimicking the target's walking style, the agreement rate rises to only about 50% when $\alpha = 0.8$

(we set $\alpha = 0.8$ in the system). The results in VI-D show that Walkie-Talkie is able to generate 128 bits in about 5s. However, an active imposter can obtain 50% of the bits by imitating the user's walking pattern; therefore, the generated bits provide 64 bits of entropy, i.e., it takes about 10 seconds (about 20 steps) to generate a 128-bit secret key.

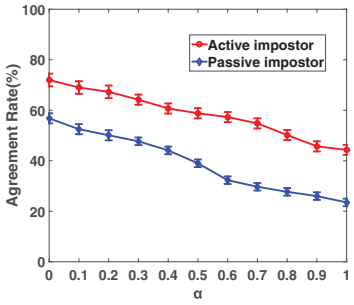


Fig. 15: Agreement rate of impostors.

The individual nature of walking gait provides our scheme security against passive eavesdroppers. Even if an active imposter can observe and try to mimic the walking style of the target, the results in Figure 15 show that he still cannot obtain a common secret key. However, an active attacker can impersonate Alice or Bob in the reconciliation stage and insert false values. Walkie-Talkie prevents such attack by the MAC method described in Section V-B. A further concern to all key-agreement protocols is the man-in-the-middle attack (MITM). A MITM attack against our scheme rarely occurs as Alice and Bob exchange the index of the generated key only instead of shared key during the reconciliation stage. Therefore, the shared key will not be compromised by MITM.

VII. 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. The system is implemented in Java and the implementation of FastICA is based on the Fastica Java library [16]. The MAC algorithm described in Section V-B is implemented by keyed-hash message authentication code (HMAC-MD5). The sampling rate of the accelerometer is set as 30Hz and Bluetooth Low Energy (BLE) functionality is employed for wireless communication.

Table III presents the system overhead (computation and energy consumption) of our system on a Moto E2 smartphone, which supports BLE peripheral mode. The major components in Walkie-Talkie: the source separation (including ICA and component identification) and key generation take an average time of 108.3ms and 208.1ms respectively. When the scheme is fully employed, the computation time and energy consumption are 316.8ms and 85.6mJ respectively. The battery capacity of the Moto E2 smartphone is 2390 mAh (30.1 kJ), therefore, the energy cost of Walkie-Talkie amounts to 0.002‰ of the total energy supply. We assume the smartphone with a targeted lifespan of one day which results in an energy budget of 1.25KJ per hour. To put this into perspective, with 5% of the budget per hour (62.5 J), Walkie-Talkie is capable of running approximately 730 times per hour, i.e., Walkie-Talkie can continuously run every 5 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.

TABLE III: System overhead measured on Moto E2.

	Computation time (ms)	Energy consumption (mJ)
ICA	105.7	71.2
Component Identification	2.6	1.5
Key Generation	208.1	12.7
AES Encryption	0.2	0.1
AES Decryption	0.2	0.1
Total	316.8	85.6

VIII. 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 [17] and speech separation [18]. The application of ICA on body sensor networks (BSN) is an emerging field. In [19], the authors applied ICA on body sensor signals to separate different sources of movement due to running and respiration. In [20], the researchers use the ICA technique to detect walking gait impairment with an ear-worn sensor. 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 physical layer characteristics based security mechanisms, e.g., Received Signal Strength Indicator (RSSI) have been proposed by researchers in [10], [21], [22]. However, these schemes are suitable for wearable devices which are frequently exchanging wireless packets. 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 [23], 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. There have been several previous works using accelerometers to determine whether the devices are worn on the same body. [24] proposed to use coherence to analyze the similarity of acceleration signals from different devices, and then decide whether two devices are carried on the same body.The idea of shaking two devices together to pair them was first proposed in [25]. [4] used the same technique but extended it to include secure authentication. [26] 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. The proposed scheme can improve 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. Biometric recognition is the science of establishing the identity of a person using his/her anatomical and behavioral traits [27]. In this paper, we have addressed a different problem (key generation) by using biometric gait. Our work belongs to biometric cryptosystems (BCS) 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 which were proposed previously mostly utilize physiological modalities such as iris [28], and fingerprint [29]. There are also some studies that use behavioral biometrics such as signature [30] and voice [31]. To the best of our knowledge, gait has not been well explored in BCS. In a similar work [32], the researchers used gait to encrypt a cryptographic key through a fuzzy commitment scheme [33]. In contrast, gait is explored to generate a cryptographic key directly in our work.

IX. Conclusion and Future Work

In this paper, 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. The proposed method obtains a security advantage from the fact that different people have distinctive walking styles. Evaluation results show that the keys generated by two independent devices on the same body are able to achieve up to a 100% bit agreement rate. We also analyze the security against various attackers. Finally, we prototype the proposed scheme on Motorola E2 smartphone to demonstrate the feasibility on contemporary mobile devices.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[2] M. P. Murray, "Gait as a total pattern of movement: Including a bibliography on gait." *American Journal of Physical Medicine & Rehabilitation*, vol. 46, no. 1, pp. 290–333, 1967.

[3] M. Rostami, A. Juels, and F. Koushanfar, "Heart-to-heart (h2h): authentication for implanted medical devices," in *CCS' 2013*. ACM, 2013, pp. 1099–1112.

[4] R. Mayrhofer and H. Gellersen, "Shake well before use: Intuitive and secure pairing of mobile devices," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 792–806, 2009.

[5] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley & Sons, 2004, vol. 46.

[6] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[7] N. Mohssen, R. Momtaz, H. Aly, and M. Youssef, "It's the human that matters: accurate user orientation estimation for mobile computing applications," in *MobiQuitous' 2014*, 2014, pp. 70–79.

[8] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Mobisys' 2014*. ACM, 2014, pp. 329–342.

[9] J. Lester, B. Hannaford, and G. Borriello, "Are you with me?-using accelerometers to determine if two devices are carried by the same person," in *Pervasive computing*. Springer, 2004, pp. 33–50.

[10] G. Revadigar, C. Javali, W. Hu, and S. Jha, "Dlink: Dual link based radio frequency fingerprinting for wearable devices," in *LCN' 2015*, 2015.

[11] C. Javali, G. Revadigar, M. Ding, and S. Jha, "Secret key generation by virtual link estimation," in *BodyNets' 2015*, 2015.

[12] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *Mobicom' 2008*. ACM, 2008, pp. 128–139.

[13] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Advances in Cryptology-CRYPTO*. Springer, 1996, pp. 1–15.

[14] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion," *SIAM journal on Computing*, vol. 17, no. 2, pp. 210–229, 1988.

[15] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," DTIC Document, Tech. Rep., 2001.

[16] http://sourceforge.net/projects/fastica/.

[17] G. Srivastava, S. Crottaz-Herbette, K. Lau, G. Glover, and V. Menon, "Ica-based procedures for removing ballistocardiogram artifacts from eeg data acquired in the mri scanner," *Neuroimage*, vol. 24, no. 1, pp. 50–60, 2005.

[18] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *INTERSPEECH' 2006*.

[19] B. Lo, F. Deligianni, and G.-Z. Yang, "Source recovery for body sensor network," in *BSN' 2006*. IEEE, 2006, pp. 4–pp.

[20] L. Atallah, O. Aziz, B. Lo, and G.-Z. Yang, "Detecting walking gait impairment with an ear-worn sensor," in *BSN' 2009*. IEEE, 2009, pp. 175–180.

[21] G. Revadigar, C. Javali, H. Asghar, K. Rasmussen, and S. Jha, "Secret key generation for body-worn devices by inducing artificial randomness in the channel," *Technical Report UNSW-CSE-TR-201506, UNSW Australia*, 2015.

[22] ——, "Mobility independent secret key generation for wearable helth-care devices," in *BodyNets' 2015*, 2015.

[23] D. Bichler, G. Stromberg, M. Huemer, and M. Löw, *Key generation based on acceleration data of shaking processes*. Springer, 2007.

[24] C. T. Cornelius and D. F. Kotz, "Recognizing whether sensors are on the same body," *Pervasive and Mobile Computing*, vol. 8, no. 6, pp. 822–836, 2012.

[25] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Ubicomp' 2001*. Springer, 2001, pp. 116–122.

[26] K. Hinckley, "Synchronous gestures for multiple persons and computers," in *Proceedings of the 16th annual ACM symposium on User interface software and technology*. ACM, 2003, pp. 149–158.

[27] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, p. 113, 2008.

[28] R. A. Marino, F. H. Alvarez, and L. H. Encinas, "A crypto-biometric scheme based on iris-templates with fuzzy extractors," *Information Sciences*, vol. 195, pp. 91–102, 2012.

[29] P. Li, X. Yang, H. Qiao, K. Cao, E. Liu, and J. Tian, "An effective biometric cryptosystem combining fingerprints with error correction codes," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6562–6574, 2012.

[30] E. Maiorana, "Biometric cryptosystem using function based on-line signature recognition," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3454–3461, 2010.

[31] B. Carrara and C. Adams, "You are the key: generating cryptographic keys from voice biometrics," in *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on*. IEEE, 2010, pp. 213–222.

[32] T. Hoang and D. Choi, "Secure and privacy enhanced gait authentication on smart phone," *The Scientific World Journal*, vol. 2014, 2014.

[33] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM conference on Computer and communications security*. ACM, 1999, pp. 28–36.