KEHKey: Kinetic Energy Harvester-based Authentication and Key Generation for Body Area Network

QI LIN, The University of New South Wales, Australia and CSIRO-Data61, Australia
WEITAO XU, City University of Hong Kong
GUOHAO LAN, Duke University, United States
YESHENG CUI, The University of New South Wales, Australia
HONG JIA, The University of New South Wales, Australia and CSIRO-Data61, Australia
WEN HU, The University of New South Wales, Australia and CSIRO-Data61, Australia
MAHBUB HASSAN, The University of New South Wales, Australia and CSIRO-Data61, Australia
ARUNA SENEVIRATNE, The University of New South Wales, Australia and CSIRO-Data61, Australia

For kinetic-powered body area networks, we explore the feasibility of converting energy harvesting patterns for device authentication and symmetric secret keys generation *continuously*. The intuition is that at any given time, multiple wearable devices harvest kinetic energy from the same user activity, such as walking, which allows them to independently observe a common secret energy harvesting pattern not accessible to outside devices. Such continuous KEH-based authentication and key generation is expected to be highly power efficient as it obviates the need to employ any extra sensors, such as accelerometer, to precisely track the walking patterns. Unfortunately, lack of precise activity tracking introduces bit mismatches between the independently generated keys, which makes KEH-based authentication and symmetric key generation a challenging problem. We propose KEHKey, a KEH-based authentication and key generation system that employs a compressive sensing-based information reconciliation protocol for wearable devices to effectively correct any mismatches in generated keys. We implement KEHKey using off-the-shelf piezoelectric energy harvesting products and evaluate its performance with data collected from 24 subjects wearing the devices on *different body locations including head, torso and hands*. Our results show that KEHKey is able to generate the same key for two KEH-embedded devices at a speed of 12.57 bps while reducing energy consumption by 59% compared to accelerometer-based methods, which makes it suitable for continuous operation. Finally, we demonstrate that KEHKey can successfully defend against typical adversarial attacks. In particular, KEHKey is found to be more resilient to video side channel attacks than its accelerometer-based counterparts.

CCS Concepts: • **Security and privacy**; • **Computer systems organization** → *Embedded and cyber-physical systems*;

Authors' addresses: Qi Lin, The University of New South Wales, School of Computer Science and Engineering, Building K17, Kensington Campus, Sydney, NSW, 2052, Australia, CSIRO-Data61, Australia, qi.lin@@unsw.edu.au; Weitao Xu, City University of Hong Kong, Department of Computer Science, Kowloon Tong, Hong Kong, weitaoxu@cityu.edu.hk; Guohao Lan, Duke University, Department of Electrical and Computer Engineering, 100 Science Dr, Durham, NC, 27708, United States, guohao.lan@duke.edu; Yesheng Cui, The University of New South Wales, School of Mechanical and Manufacturing Engineering, Building J17, Kensington Campus, Sydney, NSW, 2052, Australia, yesheng.cui@unsw.edu.au; Hong Jia, The University of New South Wales, School of Computer Science and Engineering, Building K17, Kensington Campus, Sydney, NSW, 2052, Australia, CSIRO-Data61, Australia, h.jia@unsw.edu.au; Wen Hu, The University of New South Wales, School of Computer Science and Engineering, Building K17, Kensington Campus, Sydney, NSW, 2052, Australia, CSIRO-Data61, Australia, wen.hu@unsw.edu.au; Mahbub Hassan, The University of New South Wales, School of Computer Science and Engineering, Building K17, Kensington Campus, Sydney, NSW, 2052, Australia, CSIRO-Data61, Australia, mahbub.hassan@unsw.edu.au; Aruna Seneviratne, The University of New South Wales, School of Computer Science and Engineering, Building K17, Kensington Campus, Sydney, NSW, 2052, Australia, CSIRO-Data61, Australia, a.seneviratne@unsw.edu.au.

2474-9567/2020/3-ART41 $15.00
https://doi.org/10.1145/3381754

Additional Key Words and Phrases: kinetic energy harvester, gait, key generation system, continuous authentication system, compressive sensing

ACM Reference Format:
Qi Lin, Weitao Xu, Guohao Lan, Yesheng Cui, Hong Jia, Wen Hu, Mahbub Hassan, and Aruna Seneviratne. 2020. KEHKey: Kinetic Energy Harvester-based Authentication and Key Generation for Body Area Network. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 1, Article 41 (March 2020), 26 pages. https://doi.org/10.1145/3381754

1 INTRODUCTION

The popularity of wearable devices is skyrocketing. While many of us are tracking our fitness and well-being using devices such as FitBit and Apple Watch, a recent survey [54] shows that the wearable market is beaming with hundreds of different types of products including smart glasses, smart jewellery, electronic garments, smart shoes, skin patches, and even implanted medical devices (IMDs). We are heading to a future where users are expected to have more than one wearable device continuously monitoring their bodies and providing advanced health and other services [24].

In such multi-wearable scenario, wearable devices may need to frequently exchange highly private sensory information for a range of tasks, such as data aggregation, sensing coordination and data relaying (from body to cloud). However, wireless communications among wearables are prone to various attacks, such as spoofing, man-in-the-middle (MITM) and eavesdropping attacks. Therefore, wearable devices need to establish continuous device authentication [17, 56, 63] for secure communications. Consider a scenario that a patient with chronic heart disease has a wearable/implanted heartbeat sensor. If the heartbeat rate drops below a certain value, this sensor will request all other on-body health sensors, such as blood pressure sensors, to capture patient's health data and transmit the data to a medical center via his/her smartphone. As a result, all wearable devices need to establish and maintain continuous device-to-device authentication for data synchronization. Moreover, continuous authentication also provides an additional line of defense by estimating whether their communication counterparts are legitimate at, ideally, every point in time [17]. Hence, we focus on continuous device authentication as the secure communication model in this paper.

Traditional approaches to achieving device-to-device authentication are based on a pre-deployed shared secret key or computation-intensive public key cryptography primitives, which do not scale with the increasing number of wearables and are not feasible for resource-limited wearable platforms. Therefore, wearable devices in body area network (BAN) need to agree on a symmetric key without a pre-shared secret or involving public key cryptography operations via an insecure channel to achieve device-to-device authentication. In the area of BAN security, exploiting biometrics as shared key is the one of the most popular approaches in establishing protected connections, as pairing devices needs no prior knowledge except the basic requirement that the devices can capture the same biometrics from the user body. For example, Electrocardiograms (ECGs) are a common choice for IMDs [52], where two IMDs can generate a common secret key from their independent observations of the same ECG. Similarly, recent works [53, 67] have demonstrated that two wearable devices can extract sufficient randomness and arrive at the same secret key by observing the gait of the user using acceleration sensors (also known as accelerometers). While the authors of [53, 67] demonstrated the feasibility of gait-based key generation, the power consumed by high-frequency accelerometer sampling is considered significant for small form factor wearables, making it unsuitable for continuous authentication systems. Although rechargeable wearable devices, such as smartphones, may not be affected significantly by power consumption, it is a critical issue for IMDs and smart wristbands, which are designed to operate for a long time without recharging [68]. Furthermore, a recent study revealed that accelerometer-based key pairing via gait observation is vulnerable to video side channel attacks [13].

Harvesting kinetic energy from human gait to power wearable sensors has become a hot research topic [31, 50, 65]. To reduce dependency on batteries, some commercial companies have already incorporated such energy harvesting hardware in their wearable products [1–5]. Motivated by this trend, we explore the feasibility of using kinetic energy harvesting (KEH) as a proxy to accelerometer for authentication and key generation. Different from accelerometer, KEH does not need power supply to operate. Consequently, a major advantage of KEH-based key generation is the potential for significant power saving arising from not sampling any accelerometer at all. Furthermore, as we will show in Section 6.5.1, it is more resilient to video side channel attacks than its accelerometer based counterparts. However, implementing a key generation system based on this novel sensor modality is non-trivial. This is mainly because KEHs are not designed as precise sensors in essence, which means that KEH-based authentication and key generation methods need to tackle more key mismatches due to the high level of noise inherent in KEH signal. To this end, we implement a state-of-the-art Compressive Sensing (CS)-based method [46] to correct key mismatches by introducing a sparse domain, which helps generate identical keys with high probability.

The contributions of this work can be summarized as follows:

- We conduct a world first study to experimentally analyze the potential of authentication and symmetric key generation from KEH patterns in wearable devices. Based on the outcome of this study, we propose KEHKey, a novel KEH-based authentication and key generation system for wearable devices. The proposed system can effectively generate symmetric keys from KEH-based gait signals collected from devices worn *on different body locations including head, torso and hands.*
- We provide a proof of concept implementation of KEHKey using off-the-shelf energy harvesting hardware. We experimentally show that KEHKey can successfully generate a shared 128-bit key by using a CS-based reconciliation method.
- Through a comprehensive attack analysis we show that KEHKey is highly robust, and is more resilient to video side channel attacks than accelerometer-based approaches.
- Finally, through a detailed power profiling study we show that KEHKey can reduce energy consumption by 59% compared to accelerometer-based approaches.

2 BACKGROUND

2.1 Kinetic Energy Harvesting

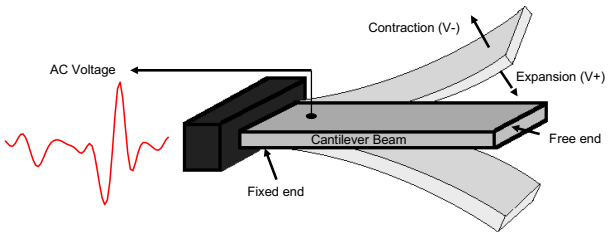


Fig. 1. Principle of converting kinetic energy into AC voltage using a piezoelectric KEH cantilevered beam. Generated voltage changes polarity as the beam alternately bends upwards and downwards.

KEH is a process of converting the environmental kinetic energy into electrical energy which can be further used to power electronics in many consumer devices. Our environment is full of kinetic energy sources such as natural seismic vibration, wind movement, sea waves, vehicular vibration, machinery vibration, infrastructure vibration, and human motion. Among these, human motion is the most relevant source for wearable devices because it means wearable devices can harvest energy directly from user activities.

The main component of a KEH system is a *transducer*, which converts motion into electrical voltage. Due to its simplicity and compatibility with micro electrical mechanical system (MEMS), *piezoelectric* is often used for KEH transduction [34]. Figure 1 shows a typical piezoelectric KEH transducer in the form of a cantilevered beam. One end of the beam is fixed to the device, while the other is set free to oscillate. When the piezoelectric material is subjected to a mechanical stress due to any motion or vibration, it expands on one side and contracts on the other. Positive charges accumulate on the expanded side and negative charges on the contracted side, generating an alternating current (AC) voltage as the beam oscillates around the neutral position. With the help of special circuits, known as *rectifier*, the AC voltage is converted into regulated direct current (DC) which is suitable to power various device components.

At any instant of time, the amount of AC voltage generated by KEH is proportional to the applied stress to the piezoelectric cantilever at that time, which means that a specific human activity, such as gait, would generate a specific AC voltage pattern over time. Thus, in theory, KEH AC voltage could act as a potential source for secret key generation as all wearable devices worn by the same user would use the same gait activity to generate their AC signal. However, to the best our knowledge, the true potential of KEH for generating secret keys has not been reported in the literature.

2.2 Compressive Sensing-based Reconciliation

KEHKey has a high bit mismatch rates when two legitimate parties generates their keys independently (see Section 3 later for more details). To address this challenge, we adopted the state-of-the-art CS-based reconciliation here to correct mismatches [46]. In this section, we describe this method in details. The symbol notations used is summarized in Table 1.

Table 1. Symbol notations

Symbol	Meaning
Φ	sensing matrix
N	length of secret key space
M	length of compressed key space
S	sparsity, number of non-zero elements
x_{Alice}, x_{Bob}	secret key after quantization
$\Delta x_{A,B}$	mismatches between Alice and Bob
$\Delta x_{A,E}$	mismatches between Alice and Eve
y_{Alice}, y_{Bob}	compressed key from secret key x
$\Delta y_{A,B}$	mismatches in compressed keys between Alice and Bob
$\Delta y_{A,E}$	mismatches in compressed key between Alice and Eve
S_{Alice}, S_{Bob}	sparsity of x_{Alice} and x_{Bob}
$S_{\Delta A,B}$	sparsity of mismatches between Alice and Bob
$S_{\Delta A,E}$	sparsity of mismatches between Alice and Eve
S'_{Alice}	sparsity of Alice in a sparsified transformed domain
Q, P	secure and effective bound for CS-based reconciliation method

2.2.1 Compressive Sensing Basis. Assume that we want to sample an unknown source or signal x with size of N with observation or sensing samplings y with size of M. Eq.(1) presents the linear relationship between x and y, where Φ ($\Phi \in \mathbb{R}^{M \times N}$) denotes the sensing matrix.

$$y = \Phi x. \tag{1}$$

Usually, we will need $M \geq N$ so that we can solve x from y; otherwise, Eq.(1) is under-determined can't be solved in general form. CS theory [8, 14, 42, 61] proposes a method to recover a higher dimension signal x from a

lower dimension observations y. When the unknown signal x is sparse (i.e., x contains zero value elements) and the sensing matrix Φ satisfied several conditions, x can be reconstructed from y by solving the ℓ_1 minimization problem.

Specifically, Φ is said to have restricted isometry property (RIP) when it satisfies the two conditions as follows. Firstly, x needs to be sampled incoherently. For example, Φ satisfies such condition when each element in Φ is ± 1 with equal probability, i.e., symmetric Bernoulli distribution. Secondly, the number of observations M must be larger than a certain threshold to capture all information in x. In CS, the sparsity S, which is the number of non-zero elements in a x ($x \in \mathbb{R}^N$), is used to indicate the amount of information in x. The quantitative threshold of M in Eq.(2) has been studied and proven in [14].

$$M > S * log(N/S). \tag{2}$$

When satisfying such condition, there is an overwhelming probability to reconstruct x from y. Eq. (2) is also known as the sufficient condition for CS reconstruction. Apart from the sufficient condition, the necessary condition was studied and proved in [61] as:

$$M > S. \tag{3}$$

It states that the number of observations M must be larger than the number of non-zero elements S, so that it is possible to reconstruct x. Otherwise, x has more information than the number of observations, it is impossible recover all information and reconstruct x.

2.2.2 CS-based Reconciliation. In symmetric key generation system, two legitimate parties (Alice and Bob) generate their keys independently. Usually, we will get two keys (x_{Alice} and x_{Bob}) that approximates to each other, but not exactly the same. As symmetric key generation requires bit-for-bit equal keys, an information reconciliation stage is usually applied to correct the mismatches. Applying the CS technique in information reconciliation was proposed in H2B [46]. The processing flow can be summarized as follows.

(1) Alice and Bob generate their keys x_{Alice} and x_{Bob}. They have no knowledge of their pairing partner's key but they only know that a legitimate partner will have a similar key as their own keys. They need to inform legitimate partner their keys via an insecure channel without leaking their keys.
(2) Alice and Bob compress their keys into y_{Alice} and y_{Bob} using Eq.(1). Their keys x_{Alice} and x_{Bob} generated from random source is *not sparse*. Therefore, it is difficult to reconstruct y_{Alice} from x_{Alice} directly. We will discuss the quantitative bound to ensure y_{Alice} cannot be recovered from x_{Alice} in Section 2.2.3.
(3) Alice sends her compressed key y_{Alice} to Bob. In CS-based information reconciliation, only one pairing party needs to solve the computational intensive ℓ_1 minimization problem. Assume that Bob is the resource-rich party. Bob will not transmit his key to Alice. Instead, he just needs to correct his key to match that of Alice.
(4) On receiving y_{Alice}, Bob calculates the mismatch vector between two compressed keys $\Delta y_{A,B} = y_{Alice} - y_{Bob}$. As Eq. (1) is a linear transform, we have $\Delta y_{A,B} = y_{Alice} - y_{Bob} = \Phi(x_{Alice} - x_{Bob}) = \Phi(\Delta x_{A,B})$, where $\Delta x_{A,B}$ is the mismatch vector between their original secret keys x_{Alice} and x_{Bob}.
(5) Bob reconstructs $\Delta x_{A,B}$ from $\Delta y_{A,B}$. Although x_{Alice} is not sparse and cannot be reconstructed from y_{Alice} directly, Bob can reconstruct $\Delta x_{A,B}$ from $\Delta y_{A,B}$ because $\Delta x_{A,B}$ is sparse; namely, $x_{Alice} \approx x_{Bob}$ and there are only a few non-zero elements in $\Delta x_{A,B}$.
(6) Finally, Bob corrects his key to match Alice's key with the mismatch vector $\Delta x_{A,B}$.

2.2.3 Security Analysis of CS-based Reconciliation. When transmitted via a public communication channel, y_{Alice} has potential to leak two kinds of information to a malicious eavesdropper Eve. The first is x_{Alice}, and the second is $\Delta x_{A,E}$, which is the mismatch vector between Eve's own key x_{Eve} and x_{alice}. Such two vulnerabilities can be addressed by selecting a proper M of Φ.

Given y and Φ are accessible to Eve via public communication channel and public knowledge respectively, she may derive the secret x by reconstructing x_{Alice} from either y directly or $\Delta x_{A,E}$ by following the the protocol in Section 2.2.2. As discussed in Section 2.2.1 earlier, the necessary condition to reconstruct x from y is shown in Eq.(3). We can then select the lower dimension M of sensing matrix Φ to be smaller than both the sparsity of x_{Alice} and the sparsity of $\Delta x_{A,E}$. Then, Eve cannot meet the necessary condition to reconstruct x_{Alice} or $\Delta x_{A,E}$. Note that a decrease in M also reduces the probabilities of successfully pairing between two legitimate parities, Alice and Bob.

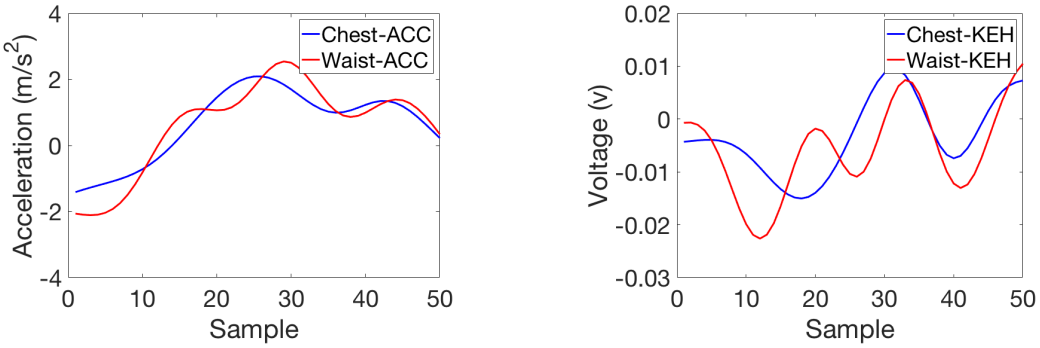
Therefore, CS-based reconciliation has two assumptions as follows. Firstly, Eve cannot derive a more similar key to Bob than Alice. Secondly, the mismatch vector is more sparse than the generated secret key itself.

To quantitatively selected a proper M, secure bound Q, effective bound P as well as a corollary $P < M < Q$ were defined in H2B [46], where $Q = min(S_{Alice}, S_{\Delta A,E})$ and $P = S_{\Delta A,B} * log(N/S_{\Delta A,B})$. Q is the minimal sparsity of x_{Alice} and $\Delta x_{A,E}$. $M < Q$ ensures that Eve cannot meet the necessary condition in Eq.(3). Recall that P is the sufficient condition of $\Delta x_{A,B}$ in Eq.(2). $M > P$ ensures that Bob meets the sufficient condition to reconstruct $\Delta x_{A,B}$. If we can find a M satisfying corollary $P < M < Q$, CS-based reconciliation is secure and effective.

When estimating the secure bound Q, three types of attack attempts, i.e., passive eavesdropping attacks, active mimicking attacks, and video side channel attacks, are taken into account to examine the minimum sparsity of $\Delta x_{A,E}$. They will be discussed in Section 4.2 and evaluated in Section 6.5.1 respectively.

Compared to original CS-based reconciliation in H2B, we also considered a third type of vulnerabilities. The original key x_{Alice} is not sparse in its original domain. However, after linear transform, it may be sparse in a sparsified domain and Eve can reconstruct x_{Alice} in the sparsified domain. In Section 6.5.1, we will examine a popular sparsified domain (i.e., Fourier Transform domain) and modify the secure bound Q to be $Q = min(S_{Alice}, S'_{Alice}, S_{\Delta A,E})$ where S'_{Alice} is the sparsity in the transformed domain.

3 PRELIMINARY ANALYSIS



(a) Accelerometer (Corr=0.94)

(b) KEH (Corr=0.63)

Fig. 2. Gait signal captured by KEH and accelerometer located on the chest and waist of the same user.

To empirically assess the key generation potential of KEH, we built prototype wearable devices by instrumenting SensorTags with piezoelectric cantilevers. The prototype also includes an accelerometer which is used to benchmark KEH performance against accelerometer signals. While details of the prototypes and data collection from volunteers are described later in Section 6.1.1, in Figure 2, we compare the gait signals recorded from KEH and accelerometer which are placed on chest and waist of the same user. Our first observation is that the pair of KEH signals are indeed correlated confirming that KEH output (e.g., voltage signal) can be potentially used as

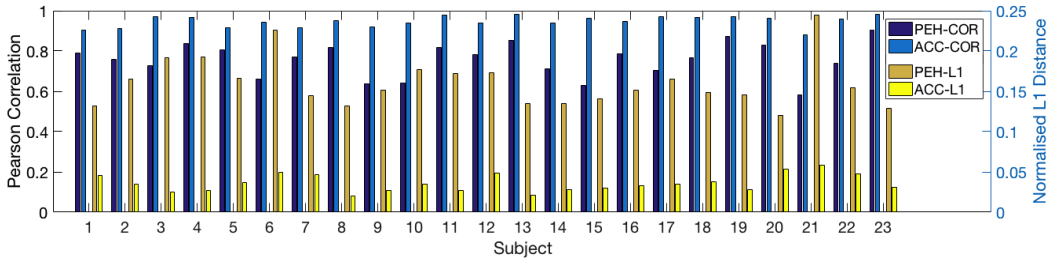


Fig. 3. Pearson correlation and ℓ_1 distance of gait signals observed at chest and waist by KEH and accelerometer.

a secret key source. We, however, also notice that compared to accelerometer signals, KEH AC voltage signal pairs are less close to each other, which means that generating identical keys at two different wearable devices would be more challenging for KEH. To gain a more quantitative insight, we compute Pearson correlation and the normalized ℓ_1 distance between two gait/acceleration signals from each of the 24 volunteers and plot them in Figure 3. Indeed, we find that KEH yields significantly lower correlation and higher ℓ_1 distance for every single participant. As a result, KEH faces significantly higher bit mismatch rates compared to accelerometer when the signals are quantized to bits.

There are two reasons for the high level noise in KEH AC voltage. Firstly, commercial accelerometers usually contain denoising component in their electrical circuits. By comparison, KEH devices are engineered for energy harvesting so they lack inherent denoising function. Secondly, KEH has an extremely narrow response bandwidth [33]. The harvested power peaks within this narrow bandwidth and falls rapidly outside this band. The resonant frequency of KEH module used in our prototype is 20Hz. Unfortunately, the frequency of human walking activity usually lies below 10Hz [43]. Even worse, the noise around 20Hz will be amplified. In comparison, accelerometers provide relatively flat response curves over frequency [23]. As a result, KEH voltage tends to be more noisy than accelerometer which becomes the key design challenge in KEHKey.

4 SYSTEM MODEL

Before discussing the framework of KEHKey, we first introduce the user model and the security model.

4.1 User and Trust Model

KEHKey exploits the heterogeneity of wearable devices architecture. Only one device out of two pairing parties, named Bob, needs to solve computational-intensive ℓ_1 minimization problem, the other device, named Alice, only needs to perform lightweight signal processing. Therefore, we can implement Bob-side KEHKey system on resource-intensive devices such as smartphones or smart watches, and Alice-side KEHKey system on resource-limit devices such as IMDs or wearable sensors.

Figure 4 provides an overview of KEHKey. A user has a pacemaker (Alice) which has a KEH installed, and a smartphone (Bob) which also has a KEH installed. Assume Alice would like to upload privacy-sensitive data to a data center. She has no knowledge of existence of Bob. She starts KEHKey process and broadcasts her request to all nearby devices, in order to find a potential pairing parties. Bob will respond to the request. On receiving response, Alice then starts pairing process with Bob. Two pairing parties wait for the user to walk several steps and generate keys independently. Only legitimate devices can generate the same symmetric key. If the user has another legitimate device that has Bob-side KEHKey system installed, Alice will start a parallel process pairing with both devices and terminate the connection if both pairings are successful.

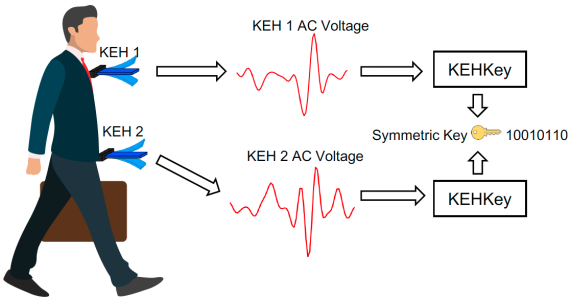


Fig. 4. An illustration of KEHKey. Two kinectic-powered wearable devices independently extract an identical 8-bit key from their respective energy harvesting patterns.

As discussed earlier, Alice and Bob may need to frequently exchange the data. KEHKey can provide continuous device-to-device authentication as long as the user is walking. When the user is stationary, KEHKey may either rely on the key generated in previous association (i.e., when she was walking), or give the user some hints (e.g. via the haptic feedback widely available on the smartphones and wearable devices) to walk[1]. Alternatively, users can use other KEH-based key generation methods such as [46] which utilizes heartbeat signals to generate keys.

We assume that all devices worn on the user are legitimate which means all the devices on user's body are trustworthy. The users are assumed to be capable of detecting such attacks if an attacker tries to modify user' on-body devices or place any adversary device on her so that attackers cannot modify or assess a legitimate devices before starting KEHKey.

4.2 Security Model

In the user scenario above, we identify the following desirable properties of secure communications.

- **Confidentiality**: Only Alice and Bob should be able to understand the contents of the transmitted message.
- **Integrity**: Alice and Bob want to ensure that the content of their communication not altered, either maliciously or by accident, in transit.
- **Authenticity** Both Alice and Bob should be able to confirm the identity of the other party involved in the communication.

Similar to [63], we *do not consider availability* in this work. This is because if an attacker (Eve) performs jamming attacks to block the communications, the protocol cannot agree on the same key. The prevention of jamming attacks is not the focus of this paper because many approaches have been proposed in the literature, and KEHKey can employ them to detect such attacks. Interested readers may refer to [60, 62] for more details.

We assume that Eve has the full knowledge of the key generation mechanism and has full control of the communication channel. Therefore, Eve may attempt to violate confidentiality via a MITM attack or eavesdropping, to compromise integrity by modifying messages, and to compromise authenticity via identity spoofing attacks.

Here, the confidentiality is achieved by encrypting the data using the keys generated by KEHKey with symmetric crytoghraphic primitives such as AES. The integrity is achieved by using message authentication code (MAC) which is sometimes also referred to message integrity code (MIC). The authenticity is based on the intuition that only legitimate devices on the same user's body can generate the same key. Therefore, as long as one party can generate the same key, its identity can be authenticated.

Based on the analysis above, the major concern of KEHKey is that whether Eve can derive the same secret key as Alice and Bob do by launching three types of attacks as following.

[1]Apple Watches remind users to be active for at last one minute once per hour via haptic feedback.

(1) **Passive eavesdropping attack.** The passive attacker knows the key generation mechanism and can monitor the communication between Alice and Bob. By eavesdropping the public components of key generation, Eve tries to generate a key based on her own gait information and use this key to pair with one or both of the legitimate devices.
(2) **Active mimicking attack.** The active attacker can observe and study the walking style of the user. She tries to impersonate the genuine user by mimicking the walking style of the genuine user.
(3) **Video side channel attack.** Researchers have shown that motion signal can be extracted from analyzing video [70]. Therefore, we assume the video attacker can take a video of the user and extract useful gait signal (i.e., acceleration signal) from the video clip. Then the attacker uses the extracted acceleration to pair legitimate devices [73].

5 DESIGN DETAILS

5.1 Overview

Figure 5 provides an overview of KEHKey. The protocol contains three main stages: sampling and smoothing, quantization, and information reconciliation. Suppose Alice and Bob are two KEH-embedded wearable devices worn on a user. At first, Alice and Bob record the user's gait signal simultaneously. Then, they quantize the samples into bit streams which contain '1' and '0' only. Finally, Alice and Bob apply a CS-based reconciliation approach to correct the mismatches between the initial bit streams, and generate the same key. The resourceful Bob-side device will handle the computational-intensive mismatches recovery. They will start the process from Stage sampling and smoothing again in the unlikely events of failing to generate the same key.

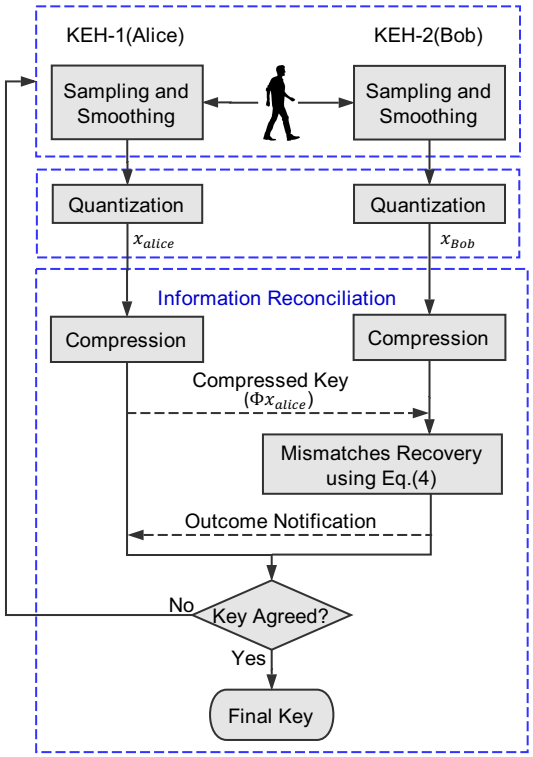


Fig. 5. KEHKey key generation protocol.

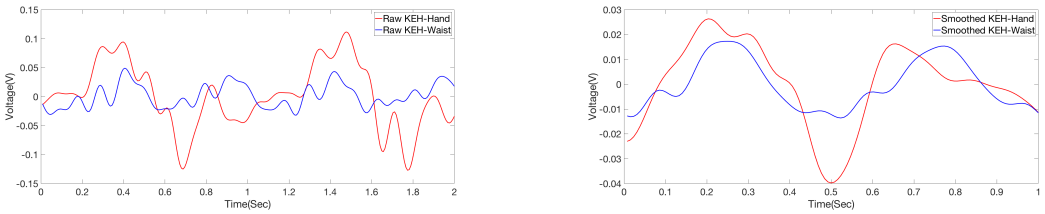
5.2 Sampling and Smoothing

One of the fundamental requirements of getting the required correlation to generate a key temporally, is accurate time synchronization of two devices as even a small offset can lead to a large decrease in correlation. On receiving the communication request/response, two wearable devices, Alice and Bob, collect gait signals independently and synchronize their clocks by observing the next *heel strike*, which can be detected accurately during each step [67]. Raw gait data collected are then normalized to have zero-mean and the same measurement units (e.g. International System of units).

One noise component in captured KEH signals is the limb swing. When KEHKey attempts to collect data from a device in user's hand or on user's arm (e.g. smartwatch) or leg, the energy generated from the limb swing will be added to the total KEH signal, making it difficult for the key pairing. The frequency of human walking steps usually lies between $1.6 - 2.8$Hz [43], and the useful human motion lies below 10Hz [35]. On the contrary, the frequency of the limb swing is usually half of the walking step frequency [30]. Figure 6(a) presents KEH signals captured in hand and on the torso of the same user with high frequency components filtered out. It is clear that the step frequency on torso is as twice as that on the arm. Therefore, we can filter out with a band pass filter if the user swings her arm or leg naturally. In KEHKey, we implement a 1 to 10Hz band pass filter to remove the limb swing component and the high frequency noise. As KEH signals are noisier than acceleration signals, we apply an exponential moving average (EMA) filter to further improve the smoothness of the KEH signals. Figure 6(b) shows the results of hand-to-torso pairing after we smooth the data using the bandpass filter and EMA. The correlation increases from 0.24 to 0.75, significantly improving the device pairing.

5.3 Quantization

Quantization basically refers to the process of converting the samples of gait signal into binary values, which ultimately constitute the bits of the symmetric key. This process is analogous to the widely used analog-to-digital (ADC) converter, except that not all samples are converted to bits and some of the samples are dropped. KEHKey uses the method in [40, 72] to quantize gait signals into bits series x_{Alice} and x_{Bob}.



(a) KEH signals with low pass filter (Corr=0.24)

(b) KEH signals with band pass filter (Corr=0.75)

Fig. 6. Hand-to-torso pairing of KEH signals captured on the same user.

The quantization boundary is the mean value in a sliding non-overlapping quantization window, so that each sample is quantized into one bit with equal probability, either 1 if the sample is larger than the boundary, or 0 if the sample is smaller than the boundary.

A guard band is inserted in the middle to avoid potential mismatches. KEHKey discards all bits that fall in guard bands. These bits are located near the quantization boundary, corresponding to potential mismatching bits. Guard bands are characterized by the guard band ratio α. The guard band ratio, which indicates the portion of the guard band, is a tunable parameter of system performance. As a result of a large guard band ratio, more bits are discarded as potential mismatches, and similarity of keys generated by two parties increases. A quantization example in Figure 7 illustrates this fact. In this quantization method, mismatching bits originate from two aspects.

First, some samples are discarded by one party while kept by the other party, such as sample 1 in Figure 7. Second, some samples are quantized into different bits by different parties, such as sample 7, 8 when $\alpha = 0.2$.

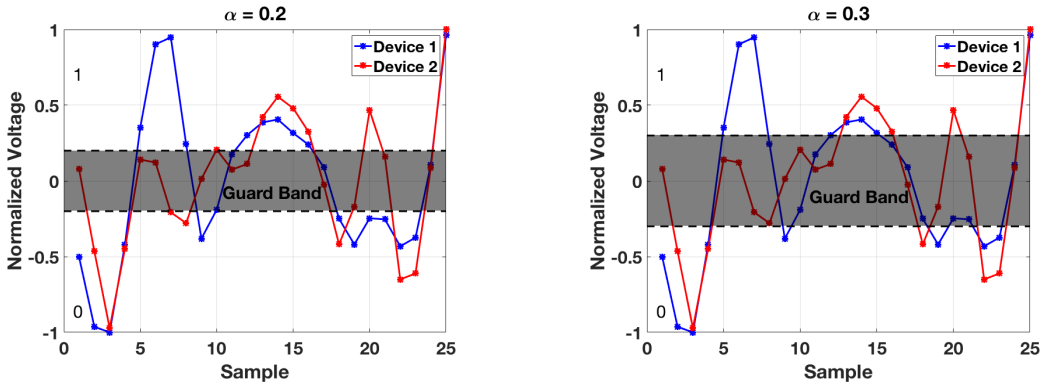


Fig. 7. Quantization example using guard bands. For $\alpha = 0.2$, there are 11/25 mismatches. For $\alpha = 0.3$, there are 8/25 mismatches.

5.4 Information Reconciliation with CS

As discussed in Section 2.2, we implement CS-based information reconciliation after quantization. We need to tune the system parameters N and M to satisfy the condition of CS-based reconciliation, i.e. $P > M > Q$, so that the system is secure and effective. Recommended by NIST [9], a symmetric key must has at least 128-bit length to prevent brute-force attacks so we select $N = 128$ and tune the sensing dimension M. The evaluation results of P and Q are shown in Section 6.2.3.

When y_{Alice} is sent via an insecure channel, Eve may modify y_{Alice} in reconciliation step to violate integrity. A MAC method is thereby implemented to verify the message [11]. Here, We code y_{Alice} to $L_{Alice} = \{y_{Alice}, MAC(k_{Alice}, y_{Alice})\}$, by treating k_{Alice} as the shared secret. On receiving L_{Alice}, Bob can obtain k_{Alice} as mentioned above and also protect the **integrity** and the **authenticity** of the key via the MAC method. If the message is modified, the derived x'_{Bob} will be different from k_{Alice} so that Bob cannot produce a correct L_{Alice} value. The message is thereby discarded.

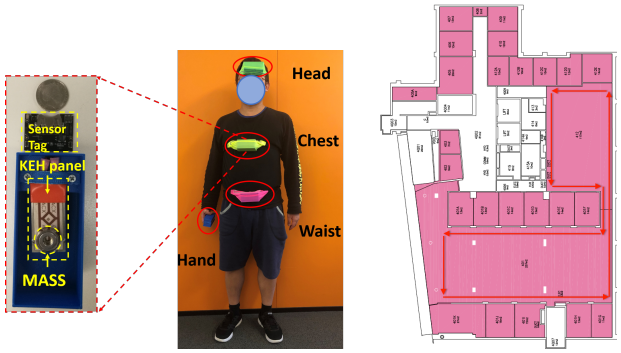
6 EVALUATION

6.1 Goals, Metrics and Methodology

The goals of evaluation are threefold: 1) to evaluate the performance of KEHKey; 2) to analyze the security of KEHKey against various attack attempts; 3) to evaluate power consumption saved by KEHKey compared to conventional accelerometer-based methods.

6.1.1 KEH Prototype. We built four prototype devices by instrumenting SensorTags[2] from Texas Instruments with PPA 1001 piezoelectric cantilevers produced by MIDE Technology [4]. We added weights to the piezoelectric cantilevers to reduce their resonance frequencies down to the minimum possible value of 20Hz, which is close to the vibration frequencies of human activities. The SensorTags feature a Cortex-M4 microcontroller and an MPU-9250 inertial measurement chip with a built-in 3-axis accelerometer[3]. Both Cortex-M4 and MPU-9250 have built-in

[2]SensorTag: http://www.ti.com
[3]MPU-9250: https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/



(a) KEH Prototype (b) Indoor Walking Path

Fig. 8. KEH Prototype and Data Collection.

analog-to-digital converters (ADCs) and the piezoelectric cantilever and the accelerometer were sampled by ADC at the highest possible rate of 128 Hz allowed by the Operating System (OS)[4] so we could later downsample the data as necessary[5]. KEH was sampled directly by the ADC inside Cortex-M4 so the microcontroller could access them directly. The accelerometer, on the other hand, was sampled by MPU-9250 and hence the microcontroller has to use the Inter-Integrated Circuit (I^2C) bus to obtain the accelerometer samples. All sampling data is saved on a on-board flash storage, which allows post processing of data in MATLAB. The size of each prototype device is 49 x 52 x 104 mm^3 as shown in Figure 8(a).

6.1.2 Data Collection. We collected KEH and accelerometer samples from 24 participants (20 males and 4 females, age: 18 - 44, height: 155 - 183cm, weight: 51 - 96kg)[6]. All participants wore four prototypes, on their chest, waist, head and hands as shown in Figure 8(c). The participants walked along an 88 meter indoor path shown in Figure 8(b) for two laps (a total of 176 meters) at their normal speed. Each device was started manually by pressing a button and terminated automatically after collecting 20, 000 samples from both KEH and accelerometer at 128 Hz. These samples were later processed to analyze the key generation performance of KEH.

6.1.3 Evaluation Metrics. Three metrics are selected to quantitatively evaluate the performance of KEHKey.

- **Bit agreement rate.** Bit agreement rate denotes the percentage of matching bits in keys generated by two parties. This metric evaluates the potential that two parties (either two legitimate devices or a legitimate and an adversary device) can generate the same key.
- **Bit rate.** Bit rate denotes the number of bits generated in unit time, measured in bits per second (bps). This metric evaluates how fast KEHKey can generate a secret key.
- **Entropy.** Entropy denotes randomness in generated keys. This metric indicates how difficult an attacker can guess the key.

6.2 Parameters Tuning

To tune KEHKey to the best performance, we examine impacts of deterministic factors in KEHKey protocol, including the sampling frequency in data collection step, the guard band ratio α in quantization step, and the sensing dimension M in reconciliation step. We select the data from 12 out of 24 participants randomly for

[4]Contiki OS: http://www.contiki-os.org
[5]Our evaluation showed that 20 Hz sampling rate provides good performance.
[6]Ethical approval for carrying out this experiment has been granted by the corresponding organization (Approval Number HC17008).

the purpose of parameter tuning. Evaluations of parameters are presented with the average values and 95% confidence intervals of the dataset.

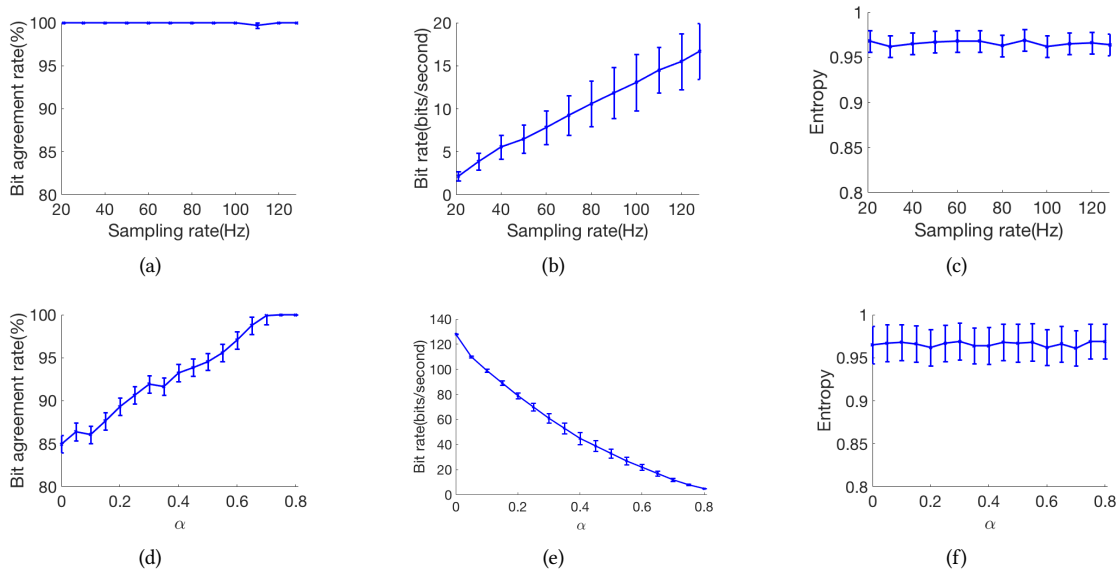


Fig. 9. Figure (a)-(c) show the impact of sampling rate on agreement rate, bit rate and entropy, respectively. Figure (d)-(f) show the impact of guard band ratio α on agreement rate, bit rate and entropy, respectively.

6.2.1 Sampling Rate. The sampling rate in the data sampler is 128Hz. For parameter tuning, we downsample the original signal from 128Hz to 20Hz.

Figure 9 shows the impact of sampling rate and the guard band ratio α on three evaluation metrics. In Figure 9(a) and 9(c), we can see that reducing sampling rate does not affect bit agreement rate and entropy significantly. In Figure 9(b), the sampling frequency has significant impact on the bit rate. The bit rate increases with the growth of sampling rate. This is intuitive because we can generate more bits with more samples. Based on the results in this experiment, we choose 128Hz sampling rate in KEHKey because it can achieve high bit rate while reaching high bit agreement rate and entropy.

6.2.2 Guard Band Ratio. As discussed in Section 5.3, the guard band is introduced to reduce mismatches because bits falling near the boundary of quantization levels are likely to be quantized differently (recall Figure 7). Meanwhile, it also determines how many bits are discarded. Therefore, there is a trade-off between bit agreement rate and bit rate. Figure 9(d) to 9(f) show the impact of α on bit agreement rate, bit rate and entropy, respectively. From Figure 9(d), we find that the average bit agreement rate increases from 85% to 100% when α increases from 0 to 0.8. In the meantime, from Figure 9(e) we can see that the bit rate decreases from 128bps to 5bps as the increase of α. Figure 9(f) shows that the entropy decreases slightly when α increases. Based on these results, we choose $\alpha = 0.7$ in KEHKey as the primary goal of a key agreement protocol is to generate the same key.

6.2.3 Dimensions of Sensing Matrix. KEHKey employs a CS-based method for reconciliation as discussed in Section 2.2. The deterministic parameter in CS-based method is the dimensions of sensing matrix, more precisely, the number of rows M.

Figure 10 shows the sparsity statistics of randomly selected participants in our dataset, where S_{Alice} denotes the sparsity of the original key and S'_{Alice} denotes the sparsity in Fourier Transformed domain of the key. As

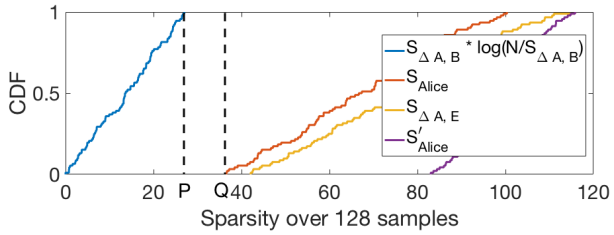


Fig. 10. Sparsity for CS-based reconciliation.

described in Section 5.4, M must satisfy $P < M < Q$ to be effective and secure (i.e., provide **confidentiality** to the key transmitted over public communication channels). We can see that the minimum possible value of $P = S_{\Delta A,B} * log(N/S_{\Delta A,B})$ is 28 and the maximum possible value of $Q = min(S_{Alice}, S'_{Alice}, S_{\Delta A,E})$ is 35. There exists a M that is larger than P and smaller than Q. As long as we can ensure the security of the key, M is desired to be as large as possible for better error correction capability. We select $M = 35$ so that CS-based reconciliation method can be effective and secure.

6.2.4 Overall Performance. After parameter tuning, we find the optimal combinations of parameters which are listed in Table 2. Under this parameter selection, we evaluate the performance of KEHKey based on the data from the remaining 12 participants.

The evaluation results show that KEHKey can successfully generate a pair of symmetric keys for legitimate devices at an average key generation rate of 12.57bps.

Table 2. Optimal parameters.

Sample rate(Hz)	Guard band α	M
128	0.7	35

6.3 Comparison of CS-based Reconciliation

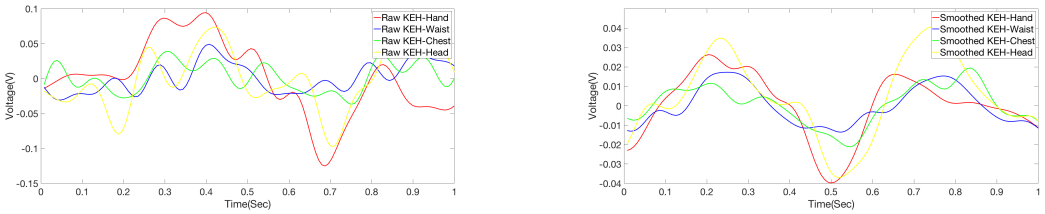
We compare the CS-based reconciliation with Error Correction Code (ECC)-based method which has been widely used in previous studies [15, 39, 67, 69]. The ECC is selected to be Reed-Solomon code with 15 bits codeword and 3 bits original message, i.e., RS(15, 3). The success rate of CS-based method is 100% while RS(15,3) can achieve 72.34% success rate only. Here, we use success rate as the evaluation metric, which means the probability of two parties deriving the same key.

6.4 Impact of Different Body Locations

Figure 11 shows the raw and smoothed gait signals on four different body locations. Four body locations have different different amplitude and waveforms with raw signals. After the frequency components smaller than 1Hz and larger than 10Hz are filtered out, the remaining components, which correspond to energy generated from gait cycle, have relatively high correlations. Their mutual correlations and average key agreement rate are shown in Table 3.

Table 3. Mutual correlation on four different body locations

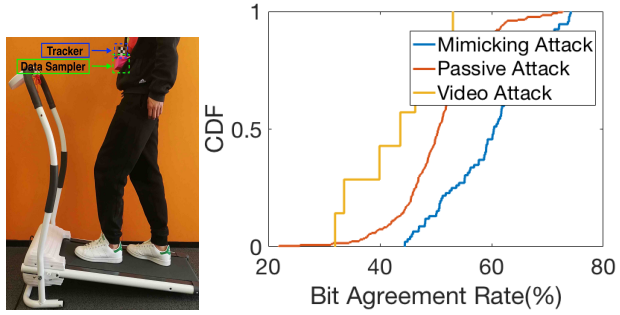
Location Pair	Hand-Waist	Hand-Chest	Hand-Head	Waist-Chest	Waist-Head	Chest-Head
Correlation	0.7556	0.6400	0.7208	0.7878	0.7117	0.6221



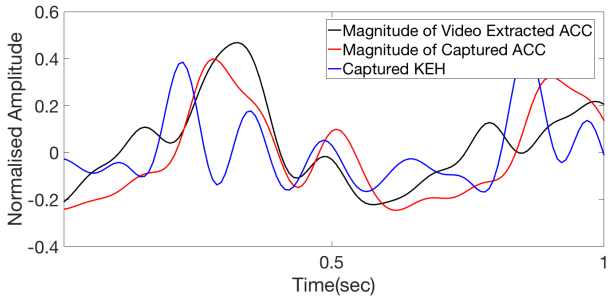
(a) Raw signals. (b) Smoothed signals.

Fig. 11. Gait signals captured on four different body locations.

6.5 Security Analysis



(a) Setup. (b) Agreement rate of attacks.



(c) Acceleration produced by the video.

Fig. 12. Attack attempts on KEHKey.

6.5.1 Resiliency against Attacks. As discussed in Section 4.2, Eve has full knowledge of KEHKey and can eavesdrop or modify the message transmitted between Alice and Bob. In this section, we examine the robustness of KEHKey against three attack attempts: passive eavesdropping attack, active mimicking attack, and video side channel attack.

The experiment for three types of attack attempts were undertaken as follows. For the passive attacks, we mutually paired the gait signal of different participants. For the active mimicking attacks, we grouped the 24 subjects into 12 pairs. Each subject was asked to mimic her/his partner's walking style and try to imitate her/him.

The genders of the subjects in a pair are the same and the attacker wore a data sampler at the same location as the genuine user.

A video side channel attacker firstly films genuine user's walking and tracks a legitimate device using a slow-motion camera, and then extract acceleration of the tracked devices from the video using computer vision-based tracking technique such as [73]. Finally, the attacker submits the extracted acceleration signals to pair the legitimate device. The dataset for video attacker was collected independently as shown in Figure 12(a). We recruited 7 participants (6 males and 1 females), and asked each participant to wear a data sampler on his/her waist and walk on a treadmill for 50 steps. A slow-motion camera at 60 frame per second were used for filming the whole walking procedure and tracking the data sampler worn on user's waist. *Many unrealistic advantages were deliberately given to the video attacker* to study the robustness of KEHKey against video attacks. For example, we ask genuine users to walk in a treadmill so that the attacker's camera had a very close observation on user's waist to generate an accurate acceleration signal of his/her gait; we asked each genuine user to wear an explicit tracker (see Figure 12(a)) so the feature-based computer vision gait tracking tasks are easier. Considering unrealistic advantages given to attackers, we anticipate that it will be significantly more difficult for the attacker to obtain similar results reported here.

Figure 12(b) shows results of different types of attackers' attempts, where KEHKey is tuned with $\alpha = 0.7$ and $M = 35$. We find that mimicking attacks can achieve the highest attacking possibility which is 74.17%. Passive attacks performed the second and the highest bit agreement rate is slightly lower than that of mimicking attacks. Video side channel attacks can at most achieve 53.06% agreement rate. Please note that the attackers are not aware which bits (i.e., the locations of the bits in the two keys) are matched, so the attackers will have 0 success rate.

A recent study [13] reported that a sophisticated adversary using video side channel attacks can exploit the vulnerabilities of four existing gait-based pairing systems using accelerometers and obtain the secret keys successfully. However, since KEH is a different sensing modality, video side channel attacks are not effective for KEHKey as shown in Figure 12(b). Figure 12(c) further compares the acceleration produced from video, the acceleration and the KEH readings captured using prototype devices simultaneously. All three signals were normalized and smoothed before converting into bits. We can see that the acceleration reproduced by video side channel attackers is close to the acceleration captured by accelerometers but very different from the KEH signal (*especially for the high frequency micro motions*). Further analysis using our dataset shows that the correlation between video-based acceleration and accelerometer-based acceleration is approximately 0.7 while the correlation between video-based acceleration and KEH signals is less than 0.3. It implies that *KEHKey is more resilient to the video side channel attacks than its accelerometer-based counterparts.*

6.5.2 Randomness of Secret Bits. To validate the randomness of the secret bit streams produced by KEHKey, the NIST suite of statistical tests (NIST-SP800-22) [10] are applied. P-values in Table 4 represent the probability that the bit streams are generated from a random process. If a p-value is less than a threshold (usually 1%), the randomness hypothesis is rejected. Table 4 shows that NIST tests of the bit streams of KEHKey have p-values larger than 1%. When there are multiple results such as the non-overlapping template tests, only the smallest P-value is listed.

6.6 Microbenchmarks

KEH is a *passive* device which does not require any power input. On the contrary, accelerometers are *active* devices, which must be supplied with power for it to take a measurement. As such, in theory, KEH should be more energy efficient for collecting the samples used for key generation, which we intend to study experimentally. On the other hand, the proposed compressive sensing involves complex computations, whose energy consumption must be analyzed to obtain a comprehensive picture of energy consumption for KEHKey.

Table 4. NIST Test Results

NIST Test	p-value
Approximate Entropy	0.024319
Block Frequency	0.010061
Cumulative Sums	0.271086
FFT Test	0.863372
Frequency	0.857972
Linear Complexity	0.051833
Longest Run	0.038054
Non overlapping Template	0.094303
Overlapping Template	0.017909
Rank	0.809631
Runs	0.112171
Serial	0.312711

Note that the CS computations in the proposed CS-based reconciliation needs to be executed only in *one of the two devices* trying to generate the symmetric key. As such, it makes sense to implement CS in more powerful wearable/personal devices, such as smartphone, smartwatch, or smartglasses when they try to pair with the more resource-constrained devices, such as smart necklaces, smart wristbands, or IMDs.

6.6.1 Energy Consumption in Resource-limited Devices. Considering the scenario that resource-limited devices perform data sampling, key compression, and data transmission tasks only and offload the computation-intensive key reconstruction task to their pairing parties. We implement all these tasks in our prototype device and measure its energy consumption. In our energy measurement setup (Figure 13), we connect the SensorTag to a Tektronix TBS-1052B oscilloscope through an external resistor. The SensorTag is controlled by the latest version of Contiki OS, which duty cycles the MCU to save power. For KEH sampling, MPU9250 and the I^2C bus are turned off. All unnecessary components of the SensorTag, including the SPI bus and other on-board sensors are powered-off. Although MPU9250 is a 9-axis sensor, combining a 3-axis gyroscope, 3-axis compass, and 3-axis accelerometer, we enabled only the 3-axis accelerometer and turned off the other two sensors. Since the instantaneous power consumption of the SensorTag fluctuates, we use the built-in function in the oscilloscope to measure the *average* power consumption.

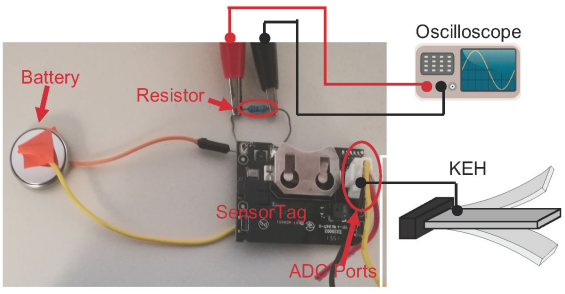


Fig. 13. SensorTag power consumption measurement setup with a resistance of 10Ω.

The measurement results of power consumption in sampling KEH and accelerometer with different sampling rates are given in Table 5. More specifically, as mentioned in Section 6.5, the KEH devices need to be sampled at 128Hz to achieve a successful paired key on 12.61 bit/second. This results in 33μW power consumption in sampling the KEH signal on a single device. For comparison, we consider the Gait-key case [67] in which two devices using the acceleration signal to generate keys. According to our experiments, only 20Hz sampling rate is needed to achieve a comparable performance to the KEH-based pairing. In particular, we found that accelerometer-based pairing is able to achieve 97.13% rate in generating successful paired key with 15.64 bit/second. Table 5, shows that 118μW of power consumption in sampling accelerometer acceleration signal at $20Hz$ sampling rate. The energy consumption for sampling are $33/12.57 = 2.62\mu$J/bit for KEH devices and $118/15.64 = 7.54\mu$J/bit for accelerometers respectively. The compressing matrix is a 35×128 Bernoulli matrix as mentioned in Section 6.5. The compressed key y_{Alice} has the size of 35 bytes and the MAC code has the size of 16 bytes. In our prototypes, the data are transmitted using a Bluetooth low energy (BLE) beacon. The energy consumption of compression and transmission are measured to be 72.08μJ and 28.6μJ respectively. Note that the energy consumption of compression and transmission is the same for KEH and accelerometer.

Considering a continuous authentication scenario, a device runs KEHKey and generates a 128-bit key for device authentication once per minute. The energy required for sampling, compression, and transmission once per minute is $7.54\times128+72.08+28.6 = 1065.8\mu$J for an accelerometer-based system and $2.62\times128+72.08+28.6 = 436.04\mu$J for KEHKey respectively. Therefore, KEHKey can reduce $(1065.8 - 436.04)/1065.08 = 59.06\%$ of energy consumption in a resource-limited device.

Table 5. Power consumption (μW) in data sampling with different sampling rates.

Sampling Rate	**KEH Signal**	**Accelerometer**
20Hz	10	**118**
100Hz	27	633
128Hz	**33**	858

6.6.2 Energy Consumption in Resource-rich Devices. Exploiting the heterogeneity of wearable devices architecture, we implement the proposed CS-based reconciliation method on three different powerful wearable devices, namely Vuzix Smartglass, Samsung Smartwatch, and Google Nexus 4 smartphone and measure energy consumption for each platform using a widely used open-source tool PowerTutor[7]. To get an insight to practical processing times of CS-based reconciliation, we also monitor the elapsed time from the console of the Eclipse development environment. Table 6 shows the average results of the time and energy consumption of one-time CS-based reconciliations from 30 repetitions (one-time reconciliation means one recovery process of the mismatch vector, which requires to solve a $\ell1$ minimization problem).

Table 6. Resource consumption of different devices.

Device	**Time(ms)**	**Energy(mJ)**
Vuzix Smartglass	102	33
Samsung Smartwatch	54	28.5
Google Nexus 4	62	48.7

[7]http://ziyang.eecs.umich.edu/projects/powertutor/

We can see that a single CS reconciliation can be executed in 100 ms or less and consumes only $48.7mJ$ at most, which means that even if a user runs KEHKey for 24 hours per day and generate keys once per minute, it will consume only 0.15% of the total battery energy per day in Google Nexus 4 (for a typical $3.8V$ and $2100mAh$ battery, which is equivalent to $2.9 * 10^7 mJ$).

6.7 Practical Key Generation Rate

In Section 6.2, the bit rate of KEHKey system is 12.57 bit/second. However, this is actually the key generation rate when user is walking. In practice, the number of walking steps varies for different users and during different time. A recent study [7] showed that factors including age, gender, body mass index, countries, and time of day can affect the number of walking steps within a unit time. Among all these factors, different time of day show the significant impact and a clear rules, so we want to model walking steps on this factor to represent the practical key generation rate in the real world.

6.7.1 Modeling Walking Steps in Different Time of Day. To evaluate the real key generation speed during different time in a day, we model the walking steps density in a day as a mixture Gaussian distribution of time. Assuming the average walking step of a user in a day is fixed, considering the time in a day T (from 0:00 to 24:00) as a random variable, a typical probability density function (PDF) of Gaussian distribution is depicted in Eq. (4).

$$f(T \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(T-\mu)^2}{2\sigma^2}} \tag{4}$$

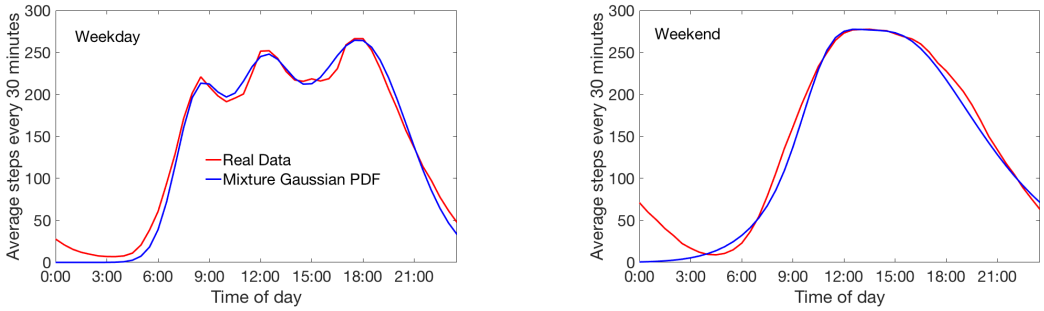
where μ, and σ indicate mean, standard deviation of time T respectively. Based on a public dataset [7], the walking step density, namely, average walking steps for 30 minutes, can be modeled as mixture Gaussian distribution comprised of three random Gaussian distributions. The model reflects a fact that walking activities of a normal user in a time usually centralize at three different time periods, namely morning, noon, and afternoon. Specifically, the density of walking steps for each time period roughly follows Gaussian distribution, with their corresponding the mean μ, and the standard deviation σ. Coupling three Gaussian distribution together with the weight w_i and the gain A, we can describe the walking step density in a day in Eq. (5). Normally, the weight w_i denotes the proportion of walking steps in three different time period and the gain A corresponds to the total walking steps in a day. Therefore, we can a set of ten parameters $\{A, w_i, \mu_i, \sigma_i, (i = 1, 2, 3)\}$ to describe the model of walking steps in different time in a day. With different age, gender, countries, etc., this set of parameters would change accordingly to show the accurate estimation of the real walking data.

$$D(T) = A \sum_{i=1}^{3} w_i f(T \mid \mu_i, \sigma_i^2) \tag{5}$$

Given the number of walking steps within a unit of time, we can further calculate the key generation rate in the corresponding time with Eq. (6), where ρ indicate the number of bits generated for each step. In KEHKey system, it is 6.98 bits/step.

$$BitRate(T) = \rho D(T) \tag{6}$$

6.7.2 Evaluation with Practical Dataset. To demonstrate the practicality of our model, we analyze the dataset provided with the article [7] of the average walking steps of people in U.S. during weekdays and weekends. This dataset is collected on smartphone from over 68 million days of activity by 717,527 individuals across 111 countries. We focused on walking data and extracted data from U.S. to plot the red curves in Figure 14(a) and Figure 14(b), which represent the average number of walk steps in different time of a day in weekdays and weekends respectively. We tuned the parameter set of mixture Gaussian distribution with the data and the results are listed in Table 7. For example, the gain A represents approximation of average steps in a day of this model, which is 3400 steps during weekdays, and 3500 steps during weekends. Three different weights w_i show



(a) (b)

Fig. 14. Modeling average walk steps on different time in a day with Mixture Gaussian distribution. Red line denotes real data, and blue line denotes estimation from mixture Gaussian distribution modelling.

Table 7. Tuned parameters for average walking steps of people in U.S.

Parameters	**Weekdays**	**Weekends**	**Weekdays High Walk-ability**	**Weekdays Low Walk-ability**
A (step)	3400	3500	3900	3200
w_1	0.175	0.0725	0.21	0.18
μ_1 (time)	8.3	11	8.4	8
σ_1 (hour)	1.3	1.5	1.3	1.5
w_2	0.28	0.6	0.19	0.31
μ_2	12	14	12	12
σ_2	1.8	4	1.4	2
w_2	0.545	0.3275	0.6	0.51
μ_2	17.8	18.8	17.5	17.5
σ_2	2.8	5	3	3

proportion of walking periods, which are {0.175, 0.28, 0.545} during weekdays, and {0.0725, 0.6, 0.3275} during weekends. These parameters reveal the fact that people tends to walk the most after works during weekdays and around noon during weekends. Three Gaussian distributions illustrate the walking steps of three different time periods. During weekdays, we can see walking steps in the morning period centralize at 8:18 am and have a 1.3 hours standard deviation. On the other hand, walking steps in the morning period during weekends centralize at 11 am and have 1.5 hours standard deviation. The model representation in blue curves in Figure 14(a) and Figure 14(b) show a good approximation to the real world data.

The current model is based on the global data of U.S., a more accurate model can be derived given sufficient data with narrower user range. For example, the paper [7] classify the people into high walk-ability and low walk-ability according to their city. We can use the high and low walk-ability data to derive new parameter settings respectively and approximate the real-world key generation rate more precisely. The tuned parameters of high and low walk-ability during weekdays are listed in Table 7, and the comparison between estimation and real-world data are illustrated in Figure 15.

On deriving the model representation of walking steps, we can further compute the practical bit rate using Eq. (6). For example, the bit rate for people in U.S. is 1.332 bit/second at 18:30 for people in a high walk-ability city during weekdays, and 0.875 bit/second at 18:30 for people in a low walk-ability city. We note that this is

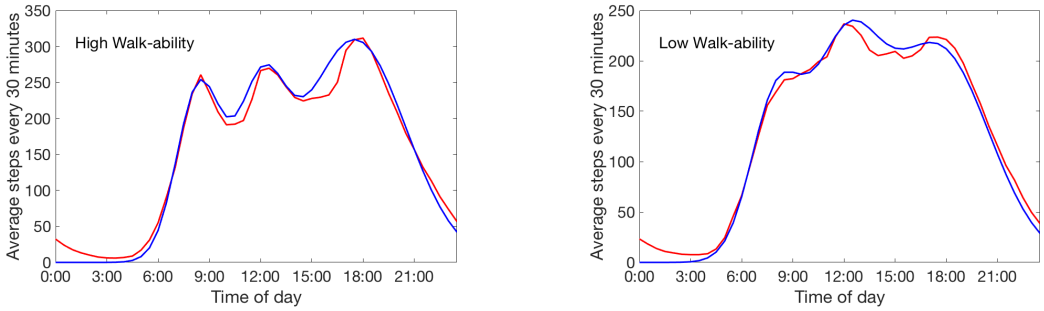


Fig. 15. High and low walk-ability models. Red line denotes real data, and blue line denotes estimation from mixture Gaussian distribution modelling.

the passive bit generation rate, and we can expect faster rates when a user walks consciously for faster key generation.

7 DISCUSSION

7.1 Comparison with State-of-the-art Key Generation System

Regardless of difference in working scenarios, underlying biometrics, as well as signal processing methods, symmetric key generation systems can be evaluated using bit rate, which indicates the key generation speed, and energy consumed to generate a 128-bit key. We implemented two state-of-the-art systems, H2B [46] and Gait-key [67] to evaluate the performance of KEHKey. When evaluating energy consumption, we implemented Gait-key and computational-intensive CS recovery in H2B or KEHKey on Google Nexus 4 as a prototype of resourceful devices. We used SensorTag as the prototype of the resource-constraint devices. The comparison results are revealed in Table 8. Gait-key (4-ary) based on in-built accelerometers can generate bits with fastest speed at 37 bit/s. H2B, limited to its biometrics, i.e. the IPI of heartbeat, can only generate key at 3 bit/s. One of the advantages of KEHKey is that it has the lowest energy consumption for a 128-bit key, consuming only 49 mJ on resourceful devices and 0.43 mJ on resource-constraint devices.

Table 8. Comparison of bit rate, energy consumption for state-of-the-art key generation system

System	bit rate (bit/s)	energy consumption (mJ)
Gait-key(2-ary)	28	198.5
Gait-key(4-ary)	37	198.5
H2B (resourceful device)	3	164
H2B (resource-constraint device)	3	21
KEHKey (resourceful device)	12.57	49
KEHKey (resource-constraint device)	12.57	0.43

7.2 Resiliency against Video Attack

In Section 6.5.1, we experimentally showed that KEHKey, as a different sensing modality to accelerometer, is highly resilient to video side channel attacks. However, future work on KEH operation mechanism is needed to analyze its fundamental limitations to such attacks.

7.3 Simultaneous Harvesting and Sensing

KEHKey targets wearable devices with KEH as their energy source. Although an KEH is used as a sensor not an energy harvester in the experiment, it is not clear whether the sensing function will be affected by energy harvesting function, if two functions are implemented together. An recent study [38] reveals that when KEHs are used for simultaneously sensing and energy harvesting, the amplitude of captured KEH signals may change while the curve pattern such as phase will remain the same. The quantization step (see Section 5.3) in KEHKey depends on curve pattern rather than absolute amplitude values and the KEH values are normalized inside a window (see Figure 7), so we believe that simultaneously energy harvesting will have little impact on KEHKey.

8 RELATED WORK

Key generation or channel authentication applications for BAN. Biometrics is the most popular trend in this direction. Among all biometrics, electrocardiograms (ECGs) are most common choice of key material especially for IMDs. This trend starts from a study showing that the time between interpulse interval (IPI) has a high level of randomness [45]. The following researches in this direction include H2B [46], Heart2Heart [51], OPFKA [25], IMDGuard [66], PSKA [57], etc. Besides ECG, other human motions, especially shaking two pairing devices together, are also suitable as shared information for key generation of on-body devices. Researchers extracted shared information from shaking activities from acceleration signals and paired two devices based on the shared information on either frequency domain [16, 41] or time-domain [12, 29]. Gait-based key system was recently developed [49, 53, 67]. Unlike IPI, stride length cannot be used as reliable randomness source, as this information can be easily leaked to a malicious attacker. The previous studies utilized the whole waveform of gait cycle as a weak random source for key generation. Other directions in this field include distance-bounding protocols, which relies on received signal strength (RSS) or other channel information [48] [55], out-of-band (OOB) authentication, which utilized auxiliary channels for authentication [21], etc.

Gait-based applications. Studies on gait have been well developed from the beginning of this century. In the early stage, gait recognition is mostly vision based. Captured in the form of video or images, both structural gait (or gait shape) and dynamic gait can be extracted for human recognition [22, 26, 32, 36]. Extended from these studies, dynamic gait features, which can be captured using accelerometer, were found to be sufficient to identify people [6]. This makes gait-based applications on wearable devices plausible as gait dynamics can be always captured by on-body accelerometers when needed. Ailisto et al. developed a user identification system, as the first work in this direction, and showed that time series of accelerometers signals can be directly used as features for user identification [6]. Gafurov et al. developed a gait-based user authentication system by attaching accelerometers in a fixed direction [18–20] . Vildjiounaite et al. developed an automatic user authentication system for wearable devices by recognizing the gait and the voice [58]. They used both time-series and frequency-domain template to match the gait signal. Recently, gait-based platform has changed from standalone accelerometer system to smartphones, which usually has accelerometers installed. With in-built accelerometers on smartphones, various user authentication systems have been developed by capturing gait dynamics [37, 44, 47].

KEH applications. The aim of introducing KEHs to wearable devices is to address the challenge of minimal maintenance and long operating life requirement in wearable devices. Therefore, the studies on this field focused on two directions: either optimizing energy harvesting of KEHs, or exploiting sensory features of KEHs to replace energy consuming wearable sensors. Researches on KEHs focusing on optimization of energy harvesting

include [59, 64, 71, 74]. Our work belongs to the second category, i.e. exploiting sensory features of KEHs. In this direction, Khalifa et al. [28] implemented human activity recognition (HAR) system, named HARKE, with KEHs to realize functionality of conventional accelerometers. Khalifa et al. [27] and Lan et al. [33] implemented KEHs to realize functionality of microphones as KEHs are able to sense incoming vibration so that the authors can detect hot words or maintain communication based on KEHs. Xu et al. [68] developed a gait recognition and wearer authentication with KEHs named KEH-gait. The gait signals extracted in their work are based on time-domain voltage readings of KEH. They segmented the signal into gait cycles and classified each gait cycle for wearer authentication.

New contributions of KEHKey. Among state-of-the-art works, KEH-gait [68], Gait-key [67], H2B [46], are closely related to KEHKey. KEH-gait presents the probability of using KEH to sample gait signals. Based on the feasibility, Xu et al. developed a user authentication system which exploits repeated patterns from gait dynamic as a fingerprint to identify users. KEHKey is a device-to-device authentication or symmetric key generation system aiming to pair two devices on the same body. Instead of exploiting repeated patterns as KEH-gait, KEHKey relies on temporal similarities among gait signals on different body locations, making it a different application of KEH. Gait-key is a gait-based key generation system relying on in-built accelerometers. By replacing accelerometers with KEHs, the energy consumption can be reduced in KEHKey as shown in Section 6.6. Lin et al. developed a heartbeat-based key generation system, named H2B, for body area network with piezoelectric sensor, where they introduced the CS-based reconciliation. As piezoelectric sensors are the main component of KEH used in KEHkey as discussed in Section 2, the scenario of H2B is similar to that of KEHKey in this context. However, instead of using IPI in heartbeat signals, we exploited gait as the key source and implemented different quantization methods for the key generation. In order to sample heartbeat vibrational signals, H2B has to use a small piezoelectric vibrational sensor to be attached to skin. In KEHKey, we don't have such limits. Instead, our system is based on an off-the-shelf KEH. By adding mass to it, the vibration on these sensors is significantly stronger than those used in H2B. Therefore, KEHKey can achieve higher probability of successful pairing and higher bit generation rate, as shown in Section 6.2. The disadvantage of KEHKey is that the users have to walk for key generation.

9 CONCLUSION

We have experimentally evaluated the authentication and key generation potential of piezoelectric KEH for BANs and found that KEH suffers from high key-disagreement rates. We have demonstrated that the key-disagreement problem of KEH can be effectively overcome with compressive-sensing-based information reconciliation, which can improve the agreement rate from 72.34% to 100% compared to conventional ECC-based reconciliation. Our evaluation results show that KEHKey is able to successfully generate the same key at the rate of 12.57 bits/second on different body locations including head, torso and hands. Our results have also confirmed that KEHKey is highly resilient against common attacks, including the video side channel attack, which is known to be effective for accelerometer-based key generation [13]. Finally, our power-profiling experiments have confirmed that KEHKey can potentially reduce energy consumption by 59% compared to the accelerometer-based approach, which makes it a suitable solution for continuous authentication and key generation in BAN.

REFERENCES

[1] Ampy. http://www.getampy.com/ampy-move.html//.
[2] BionicPower. https://www.bionic-power.com/. Accessed: 2017-10-21.
[3] InStep Nanopower. http://www.instepnanopower.com/Default.aspx. Accessed: 2017-10-23.
[4] Mide. https://www.mide.com/.
[5] ReVibe Energy. http://revibeenergy.com/. Accessed: 2017-10-21.
[6] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S.-M. Makela, "Identifying people from gait pattern with accelerometers," in *Defense and Security*. International Society for Optics and Photonics, 2005, pp. 7–14.

[7] T. Althoff, J. L. Hicks, A. C. King, S. L. Delp, J. Leskovec *et al.*, "Large-scale physical activity data reveal worldwide activity inequality," *Nature*, vol. 547, no. 7663, p. 336, 2017.

[8] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[9] E. Barker, "Nist special publication 800-57 part 1 revision 4, recommendation for key management part 1: General," *NIST*, 2016.

[10] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks *et al.*, "Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010.

[11] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Crypto*, vol. 96. Springer, 1996, pp. 1–15.

[12] D. Bichler, G. Stromberg, M. Huemer, and M. Löw, "Key generation based on acceleration data of shaking processes," *UbiComp 2007: Ubiquitous Computing*, pp. 304–317, 2007.

[13] A. Bruesch, L. Nguyen, D. Schürmann, S. Sigg, and L. C. Wolf, "Security properties of gait for mobile device pairing," *IEEE Transactions on Mobile Computing*, 2019.

[14] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[15] G. C. Clark Jr and J. B. Cain, *Error-correction coding for digital communications*. Springer Science & Business Media, 2013.

[16] C. Cornelius and D. Kotz, "Recognizing whether sensors are on the same body," in *International Conference on Pervasive Computing*. Springer, 2011, pp. 332–349.

[17] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE transactions on information forensics and security*, vol. 8, no. 1, pp. 136–148, 2013.

[18] D. Gafurov, E. Snekkenes, and P. Bours, "Gait authentication and identification using wearable accelerometer sensor," in *2007 IEEE Workshop on Automatic Identification Advanced Technologies*, June 2007, pp. 220–225.

[19] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor." *JCP*, vol. 1, no. 7, pp. 51–59, 2006.

[20] D. Gafurov, E. Snekkenes, and T. Buvarp, "Robustness of biometric gait authentication against impersonation attack," in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*. Springer, 2006, pp. 479–488.

[21] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 129–142.

[22] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[23] C. Harrison, D. Tan, and D. Morris, "Skinput: appropriating the body as an input surface," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 453–462.

[24] S. Hiremath, G. Yang, and K. Mankodiya, "Wearable internet of things: Concept, architectural components and promises for person-centered healthcare," in *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*. IEEE, 2014, pp. 304–307.

[25] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen, "Opfka: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2274–2282.

[26] A. Kale, A. Sundaresan, A. Rajagopalan, N. P. Cuntoor, A. K. Roy-Chowdhury, V. Kruger, and R. Chellappa, "Identification of humans using gait," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1163–1173, 2004.

[27] S. Khalifa, M. Hassan, and A. Seneviratne, "Feasibility and accuracy of hotword detection using vibration energy harvester," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A*. IEEE, 2016, pp. 1–9.

[28] S. Khalifa, G. Lan, M. Hassan, A. Seneviratne, and S. K. Das, "Harke: Human activity recognition from kinetic energy harvesting data in wearable devices," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1353–1368, 2018.

[29] D. Kirovski, M. Sinclair, and D. Wilson, "The martini synch," *Technical Report MSR-TR-2007-123, Microsoft Research*, 2007.

[30] M. Kubo, R. C. Wagenaar, E. Saltzman, and K. G. Holt, "Biomechanical mechanism for transitions in phase and frequency of arm and leg swing during walking," *Biological cybernetics*, vol. 91, no. 2, pp. 91–98, 2004.

[31] A. D. Kuo, "Harvesting energy by improving the economy of human walking," *Science*, vol. 309, no. 5741, pp. 1686–1687, 2005.

[32] T. Lam and R. Lee, "A new representation for human gait recognition: Motion silhouettes image (msi)," *Advances in Biometrics*, pp. 612–618, 2005.

[33] G. Lan, W. Xu, S. Khalifa, M. Hassan, and W. Hu, "Veh-com: Demodulating vibration energy harvesting for short range communication," in *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*. IEEE, 2017, pp. 170–179.

[34] E. Lefeuvre, A. Badel, C. Richard, L. Petit, and D. Guyomar, "A comparison between several vibration-powered piezoelectric generators for standalone systems," *Sensors and Actuators A: Physical*, vol. 126, no. 2, pp. 405–416, 2006.

[35] 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 *International Conference on Pervasive Computing*. Springer, 2004, pp. 33–50.

[36] Z. Liu and S. Sarkar, "Improved gait recognition by gait dynamics normalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 863–876, 2006.
[37] H. Lu, J. Huang, T. Saha, and L. Nachman, "Unobtrusive gait verification for mobile phones," in *Proceedings of the 2014 ACM international symposium on wearable computers*. ACM, 2014, pp. 91–98.
[38] D. Ma, G. Lan, W. Xu, M. Hassan, and W. Hu, "Sehs: Simultaneous energy harvesting and sensing using piezoelectric energy harvester," in *Internet-of-Things Design and Implementation (IoTDI), 2018 IEEE/ACM Third International Conference on*. IEEE, 2018, pp. 201–212.
[39] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, "Proximate: proximity-based secure pairing using ambient wireless signals," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 211–224.
[40] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *Mobicom*. ACM, 2008, pp. 128–139.
[41] R. Mayrhofer and H. Gellersen, "Shake well before use: Authentication based on accelerometer data," in *International Conference on Pervasive Computing*. Springer, 2007, pp. 144–161.
[42] P. Misra, W. Hu, M. Yang, and S. Jha, "Efficient cross-correlation via sparse representation in sensor networks," in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 2012, pp. 13–24.
[43] 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.
[44] C. Nickel, T. Wirtl, and C. Busch, "Authentication of smartphone users based on the way they walk using k-nn algorithm," in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*. IEEE, 2012, pp. 16–20.
[45] C. C. Poon, Y.-T. Zhang, and S.-D. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 73–81, 2006.
[46] L. Qi, W. Xu, J. Liu, A. Khamis, W. Hu, M. Hassan, and A. Seneviratne, "H2b: Heartbeat-based secret key generation using piezo vibration sensors," in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2019.
[47] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "Smartphone based user verification leveraging gait recognition for mobile healthcare systems," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*. IEEE, 2013, pp. 149–157.
[48] G. Revadigar, C. Javali, W. Hu, and S. Jha, "Dlink: Dual link based radio frequency fingerprinting for wearable devices," in *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*. IEEE, 2015, pp. 329–337.
[49] G. Revadigar, C. Javali, W. Xu, A. V. Vasilakos, W. Hu, and S. Jha, "Accelerometer and fuzzy vault based secure group key generation and sharing protocol for smart wearables," *IEEE Transactions on Information Forensics and Security*, 2017.
[50] L. C. Rome, L. Flynn, E. M. Goldman, and T. D. Yoo, "Generating electricity while walking with loads," *Science*, vol. 309, no. 5741, pp. 1725–1728, 2005.
[51] M. Rostami, A. Juels, and F. Koushanfar, "Heart-to-heart (h2h): authentication for implanted medical devices," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 1099–1112.
[52] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "Sok: Security and privacy in implantable medical devices and body area networks," in *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 2014, pp. 524–539.
[53] D. Schürmann, A. Brüsch, S. Sigg, and L. Wolf, "Bandana-body area network device-to-device authentication using natural gait," in *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*. IEEE, 2017, pp. 190–196.
[54] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A survey of wearable devices and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2573–2620, 2017.
[55] L. Shi, J. Yuan, S. Yu, and M. Li, "Ask-ban: authenticated secret key extraction utilizing channel characteristics for body area networks," in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*. ACM, 2013, pp. 155–166.
[56] F. Stajano, "The resurrecting duckling," in *International workshop on security protocols*. Springer, 1999, pp. 183–194.
[57] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "Pska: Usable and secure key agreement scheme for body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 1, pp. 60–68, 2010.
[58] E. Vildjiounaite, S.-M. Mäkelä, M. Lindholm, R. Riihimäki, V. Kyllönen, J. Mäntyjärvi, and H. Ailisto, "Unobtrusive multimodal biometrics for ensuring privacy and information security with personal devices," in *International Conference on Pervasive Computing*. Springer, 2006, pp. 187–201.
[59] T. Von Buren, P. D. Mitcheson, T. C. Green, E. M. Yeatman, A. S. Holmes, and G. Troster, "Optimization of inertial micropower generators for human walking motion," *IEEE Sensors journal*, vol. 6, no. 1, pp. 28–38, 2006.
[60] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey," *Security in distributed, grid, mobile, and pervasive computing*, vol. 1, p. 367, 2007.
[61] W. Wang, M. J. Wainwright, and K. Ramchandran, "Information-theoretic limits on sparse signal recovery: Dense versus sparse measurement matrices," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2967–2979, 2010.
[62] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," 2006.

[63] W. Xi, C. Qian, J. Han, K. Zhao, S. Zhong, X.-Y. Li, and J. Zhao, "Instant and robust authentication and key agreement among mobile devices," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 616–627.
[64] T. Xiang, Z. Chi, F. Li, J. Luo, L. Tang, L. Zhao, and Y. Yang, "Powering indoor sensing with airflows: a trinity of energy harvesting, synchronous duty-cycling, and sensing," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, p. 16.
[65] L. Xie and M. Cai, "Human motion: Sustainable power for wearable electronics," *IEEE Pervasive Computing*, vol. 13, no. 4, pp. 42–49, 2014.
[66] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li, "Imdguard: Securing implantable medical devices with the external wearable guardian," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1862–1870.
[67] W. Xu, C. Javali, G. Revadigar, C. Luo, N. Bergmann, and W. Hu, "Gait-key: A gait-based shared secret key generation protocol for wearable devices," *ACM Transactions on Sensor Networks (TOSN)*, vol. 13, no. 1, p. 6, 2017.
[68] W. Xu, G. Lan, Q. Lin, S. Khalifa, M. Hassan, N. Bergmann, and W. Hu, "Keh-gait: Using kinetic energy harvesting for gait-based user authentication systems," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 139–152, 2018.
[69] L. Yang, W. Wang, and Q. Zhang, "Secret from muscle: Enabling secure pairing with electromyography." in *SenSys*, 2016, pp. 28–41.
[70] G. Ye, Z. Tang, D. Fang, X. Chen, K. I. Kim, B. Taylor, and Z. Wang, "Cracking android pattern lock in five attempts," in *The Network and Distributed System Security Symposium*, 2017.
[71] J. Yun, S. N. Patel, M. S. Reynolds, and G. D. Abowd, "Design and performance of an optimal inertial power harvester for human-powered devices," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 669–683, 2011.
[72] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.
[73] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *European conference on computer vision*. Springer, 2012, pp. 864–877.
[74] J. Zhao and Z. You, "A shoe-embedded piezoelectric energy harvester for wearable sensors," *Sensors*, vol. 14, no. 7, pp. 12 497–12 510, 2014.