LiReK: A Lightweight and Real-time Key Establishment Scheme for Wearable Embedded Devices by Gestures or Motions

Zitao Chen[a,e], Wei Ren[a,b,e], Yi Ren[c], Kim-Kwang Raymond Choo[d,a]

[a] *School of Computer Science, China University of Geoscience, Wuhan, P.R. China*
[b] *Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences (Wuhan), Wuhan, P.R. China*
[c] *School of Computer Science, University of East Anglia, Norwich, UK*
[d] *Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, USA*
[e] *Guizhou Provincial Key Laboratory of Public Big Data, GuiZhou University, Guizhou, P.R. China*

Abstract

With the recent trend in wearable technology adoption, the security of these wearable devices has been the subject of scrutiny. Traditional cryptographic schemes such as key establishment schemes are not practical for deployment on the (resource-constrained) wearable devices, due to the limitations in their computational capabilities (e.g. limited battery life). Thus, in this study, we propose a lightweight and real-time key establishment scheme for wearable devices by leveraging the integrated accelerometer. Specifically, we introduce a novel way for users to initialize a shared key using random shakes / movements on their wearable devices. Construction of the real-time key is based on the users' motion (e.g. walking), which does not require the data source for key construction in different devices worn by the same user to be matching. To address the known limitations on the regularity and predictability of gait, we propose a new quantization method to select data that involve noise and uncertain factors when generating secure random number. This enhances the security of the derived key. Our evaluations demonstrate that the matching rate of the shake-to-generate secret key is up to 91.00% and the corresponding generation rate is 2.027 bit/sec, and devices worn on human participant's chest, waist, wrist and carried in the participant's pocket can generate 4.405, 4.089, 6.089 and 3.204 bits random number per second for key generation, respectively.

Keywords: Lightweight; Key Management; Real-time; Body Sensor Networks; Embedded Devices

1. Introduction

Advances in both hardware (e.g. embedded wireless microelectronic components) and software have contributed to the popularity of wearable and embedded devices. These devices typically offer ubiquitous computing. For example, embedded sensors can be used to monitor the real-time physiological status of users and can be applied in a wide range of situations, such as in healthcare and allied health services (e.g. counting of steps, tracking of heart rate, and monitoring of glucose levels) [1, 2]. However, these devices are generally not designed with security in mind [3, 4, 5, 6]. The amount and nature of data and services these wearable devices can have access to (e.g. the user's private data), as well as the limitations of these devices (e.g. resource-constrained), require us to rethink how we design security solutions for wearable devices [7, 8, 9, 10].

Due to the limitations in the computational capability of the underlying hardware, a number of existing cryptographic solutions such as key establishment protocols may not be fit-for-purpose. For example, the Diffie–Hellman (DH) key exchange is usually employed in existing key management schemes. However, DH key exchange implementations require complex cryptography computations such as modular-exponentiation operation. Thus, the overheads may be beyond the existing capability of resource-constraint wearable devices.

It is important to be able to establish a secure session between two devices particularly those worn by the same user, as these wearable devices require frequent data exchange between devices (e.g. transmitting a user's health-related information such as glucose level and heartbeat counts between the smartwatch to a paired mobile device, so that the information can be sent to the hospital network). A successful compromise could have real-world implications. For example, if a malicious attacker successfully changes the glucose level or heartbeat counts of a particular user, this would result in delivery of the wrong medication or treatment plan and lead to fatality. Undeniably, security is an important factor to be considered in wearable devices, particularly those deployed in real-world applications. It is, therefore, unsurprising that designing lightweight and real-time cryptographic solutions such as key establishment protocols is of ongoing research interest [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21].

One challenge in designing lightweight cryptographic solutions such as key establishment protocols is providing an optimal security assurance without incurring excessive energy consumption. Therefore, we examine the potential of a lightweight key generation scheme designed for wearable devices, which provides real-time rekeying mechanism without high computational complexity. Periodic rekeying is introduced to increase the difficulties in successfully conducting a cryptanalytic attack (in comparison to using static key). In other words, a real-time key establishment scheme is more likely to be capable of withstanding attacks and providing long-term secure communication especially in Body Sensor Networks (BSNs) [22, 23].

The integrated accelerometer is commonly found on or can be easily integrated with most wearable devices. For example, a self-contained wearable cuff-less photoplethysmographic (PPG) based blood pressure monitor was developed in [24] and was integrated with two MEMS accelerometer to measure the hydrostatic pressure offset of the PPG sensor relative to the heart [25]. Popular consumer wearable devices (e.g. Apple watch, Garmin HRM-Tri heart rate monitor[1]) have integrated accelerometer. Thus, in this paper, we posit that the accelerometer embedded in wearable devices can be leveraged in the design of a lightweight and real-time key establishment scheme (hereafter referred to as LiReK). Specifically, the contributions of this paper are as follows:

1. We propose a novel method to generate a shared key on wearable devices by users who wish to establish a secure session by randomly shaking their devices in concert without the need to extract features from different dataset.
2. We design a lightweight bit-extraction algorithm based on the value-difference of neighboring samples. This allows us to correct the value deviation at the devices to extract shared stochastic data from the insecure sensing data.
3. We introduce a real-time key establishment scheme leveraging the original sensory data to generate secure random number for key construction without using any data transformation methods.

The organization of this paper is as follows. Related work is introduced in Section 2. The system model and adversary model are presented in Section 3. Section 4 describes our scheme, and its evaluation and analysis are presented in Section 5. We conclude the paper in Section 6.

2. Related Work

Motivated by the resource-constrained nature of the wearable devices, efforts have been devoted to designing lightweight and efficient key establishment schemes for such devices and BSNs in the literature. In [26], for example, the authors proposed a cloud-assisted key management scheme for BSNs, designed for both indoor and outdoor settings. The authors in [27] investigated the channel property by utilizing the signal strength fluctuation caused by incidental motion of the devices to achieve efficient key construction.

As integrated sensors in BSNs can be used to collect a user's physiological signals, such biometric measurements can be utilized for key establishment [23, 28, 29, 30]. In [23], for example, the authors proposed the Ordered-Physiological-Featured-based key agreement (OPFKA) scheme, which specifically employed the secret features in the physiological signal collected by the embedded sensors. Both studies in [28, 29] demonstrated how electrocardiogram (ECG) signals could be used as shared secret for key generation. Similarly, in this study, we leverage the accelerometer (also partly due to its broader adoption, as compared to other sensors). For example, Garmin HTM-Tri heart rate monitors are not integrated with the ECG sensor; thus, the schemes in [28, 29] would not be applicable for such devices.

One of the first studies to use one's gait characteristics via embedded sensors in on-body devices to establish cryptographic keys for these devices is that of Xu *et al.* [31]. To extract gait features from the sensing data in devices located in different body areas, the authors used the blind source separation techniques (e.g. Independent Component Analysis (ICA) and Fast Fourier Transform (FFT)) to process the sensing data. The matched data were then used to generate symmetric keys at both devices. Unlike the scheme of Xu *et al.* [31], we do not extract features from the sensory data to construct shared keys.

The use of gait characteristics for cryptographic key generation, such as the approach in [31], may not be suitable in a number of applications (e.g. key establishment). This is because gait features are generally associated with a user's habits (e.g. walking abnormalities may allow another person to infer the user's medical or health conditions, say gout). In addition, an attacker who knows the user reasonably well could have an advantage in "guessing" the right key (e.g. due to the degree of regularity, such as in the case of a person suffering from regular gout attacks). In the literature, researchers such as [32, 33, 34] have studied the viability of imitation attacks (e.g. the probability of gait features to be imitated for attacks). While the practicality of imitation attacks on real-world devices is yet to be determined, it is important to propose schemes resilient to imitation attacks. We remark that the scheme in [31] is not designed to mitigate imitation attacks (and we assume the possibility for a strong adversary to learn the gait features of any users in this paper). Also, the feature-extraction based methods in [31] may also impose constraints on the sensing data on different on-body devices. Once the users' motion present a certain extent of non-habitual (e.g. walking with sudden acceleration or deceleration), the data on different devices (e.g. the data from devices on the hand and on the waist, respectively) may fail to construct shared features for key generation (i.e. the study in [31] conducts experiments only on the scenarios of users' habitually walking).

[1] https://buy.garmin.com/en-GB/GB/p/136403

An inter-device authentication mechanism that requires users to shake their devices together was introduced by Rene *et al* [35], which inspires this work. Specifically, we explore the potential for generating symmetric key on users' devices from their arbitrary shaking. While Rene *et al* mainly focused on inter-device authentication, we seek to design a lightweight and real-time key establishment scheme in this paper. Specifically, their scheme can differentiate between devices shaken together by an individual and devices shaken separately for authentication. One of their proposed models, ShaVe, employs the DH key agreement and it is not introduced in our scheme. The another model, ShaCK, can be used to generate symmetric keys using feature-extraction based methods such as exponentially quantized FFT. Such an approach is similar to that of Xu *et al.* [31].

Random number generation (RNG) plays an important role in cryptosystem and it can be achieved by leveraging the integrated sensors on mobile devices. In [36], a sensor-based random number generator seeding scheme was proposed. To achieve secure seeding, the predictable (insecure) sensory data is removed and the data is then transformed into complex exponentials using a FFT. Finally, the seed for the random number generator is produced. The authors in [37] built a framework called SensoRNG to produce secure random number for mobile and IoT devices. Basically, the mixing algorithm is based on the analysis of data collected from the embedded sensors in 37 Android devices. The non-random bits are stripped before compressing the data into smaller data stream. The design of the algorithm is inspired by "S-box" used in asymmetric cryptography algorithms by further mixing the data, making the input data irreversible after seeding. Unlike the above mentioned schemes, the proposed scheme in this paper uses kinetic sensory data from the built-in accelerometer to provide secure random number for key establishment.

Mathur [38] proposed a key-extraction algorithm, level-crossing algorithm, for quantizing bit sequences from correlated stochastic data. Specifically, Mathur's algorithm computes two statistical thresholds. Samples above the upper threshold or below the lower threshold can be quantized into bits if they satisfy the defined requirement. Such an approach is considered in the designs of several other schemes such as [27, 31, 39]. In this paper, however, we propose a bit-extraction scheme based on the value difference of the neighboring samples rather than on the statistical analysis over the data as in [38].

3. Problem Formulation

In this section, we briefly describe how wearable devices can implement our proposed scheme before discussing the adversary model.

Let Alice and Bob denote two legitimate devices worn by the same user, and multi-factor noise denotes all uncertain factors contributing to the imprecision and randomness in the sensing data (e.g. outer noise, the intrinsic inaccuracy, and short-term deviation of the devices).

3.1. System Model

Generally, pre-shared based schemes relying on the pre-deployment of secret on devices can provide convenience to achieve end-to-end secure communication, which also eases the task of initializing a shared key before secure communication. However, in BSNs, using pre-shared keys may not be a feasible solution; thus, we should not assume that pre-shared keys are always available [40, 41]. When using pre-shared schemes, it is also inconvenient when adding a new device into BSNs that do not have a shared key with other devices (i.e. cannot achieve plug-and-play usage) [30].

While many wireless technology standards (e.g. Wi-Fi Protected Access (WPA)) use pre-shared key in their security protocols, such schemes may not be applicable for BSNs since there may not be a user-interface for users to input the pre-shared secret (e.g. the input of personal identification number (PIN) code in the 801.15.1 (Bluetooth) protocol).

Hence, it is reasonable that we do not assume the pre-shared key to be available. This is the approach we use in this paper.

Fig. 1 depicts a potential implementation model for wearable device users. Basically, prior to using the wearable devices, a user first shakes the devices together in an arbitrary pattern. The sensing data can then be utilized for initializing a shared secret key on the respectively shaken devices. With the availability of a shared key on the devices, the user can wear and begin using the devices, during which the wearable devices generate real-time keys by sensing the user's motion to secure their communications. We also remark that the data for real-time key construction in the different devices worn by the same user are not required to be matching (as shown in Fig. 1).

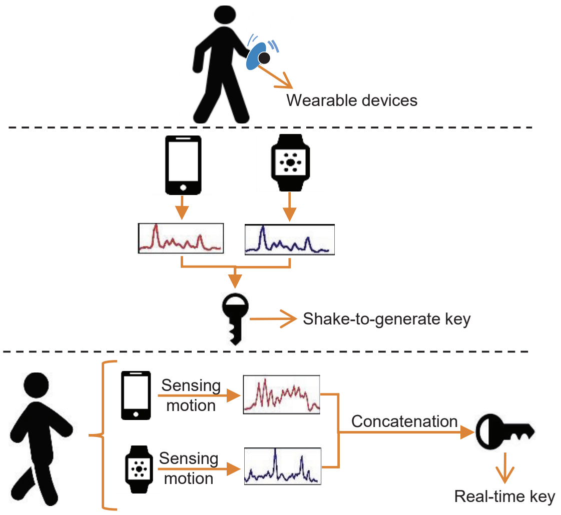


Figure 1: An implementation of the proposed scheme

3.2. Adversary Model

In this section, we identify four potential vulnerabilities that could be exploited by the adversary to compromise our scheme; namely: 1) imitating users' motion to counterfeit secret keys; 2) eavesdropping the communications between devices; 3) message modification and spoofing attacks; and 4) knowledge of the procedures and methods of the proposed scheme.

In our scheme, all of the secret keys are generated from users' motion or gestures. We assume that the users' motion to generate the secret keys are not secret (e.g. anyone in the vicinity of a user can observe the user's shaking of the devices). Thus, an adversary can observe users' gestures or motion and attempt to fabricate keys by imitation. While the adversary can imitate how users shake their devices, it is unrealistic for the adversary (or anyone) to reproduce the identical action (e.g. in terms of the strength and the exact movement such as hand swing and its angle), as unlike gait and other habitual motions, users' free-form shaking are arbitrary. However, we assume a strong adversary in the sense that the adversary may be able to successfully imitate users' gait features using other means (unlike the assumption in [31]).

During the reconciliation procedure when two shaken devices need to reconcile with each other the position of unqualified bits they have discarded, we assume their mutual communication is performed in an insecure channel. In other words, an adversary can eavesdrop on the message and the adversary might attempt to parse knowledge of the secret key from the captured message.

In addition to passive attacks, we assume that the adversary can conduct active attacks, such as modifying messages and injecting fabricated messages into the reconciliation message, to influence the outcome of the derived key. However, the factors not directly associated with the security during the generation of key (e.g. the adversary interrupts the communication between the legitimate devices and thus terminates the key establishment between devices, the strength of the encryption protocol, etc.) are outside the scope of this paper. In other words, the scheme in this paper is deemed secure, if and only, if the generated keys are secure. Apart from false injection, we assume that the adversary can perform man-in-the-middle (MITM) attacks, and impersonate a legitimate user.

As we will introduce later in this paper, our secure real-time key generation scheme is achieved using a combination of factors, each playing a different role in the implementation (e.g. the assignment of different thresholds to select or filter specific sensory data). And we assume all of these methods are not required to be kept secret, which means the adversary can gain full knowledge of the procedures and methods used in our scheme.

4. Proposed Scheme - LiReK

Our scheme is briefly depicted in Fig. 2 and the details of our scheme are elaborated in the following.

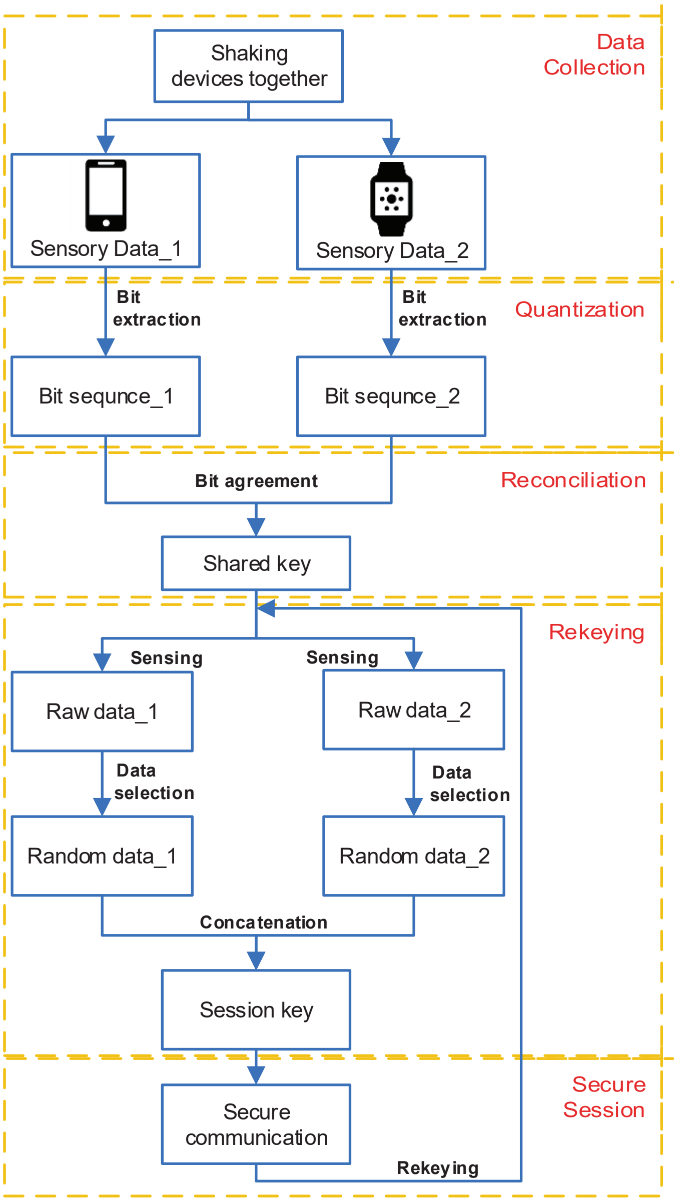


Figure 2: The proposed scheme – LiReK

4.1. Data Collection

Since all data are generated by the accelerometer in our scheme, we first look at how they obtain the acceleration data of devices. The underlying principle of the embedded accelerometer in mobile devices is described in Fig. 3. The acceleration in the motion direction is derived from the combination of measurement in 3 axes. Outcome of the measurement in the devices usually mix with a number of other factors, and the measurement in the X-axis can be expressed as [42]:

$$\tilde{a}_x = a_x + S_x a_x + B_f + n_x \tag{1}$$

where $\tilde{a}_x$ is the measurement in the X-axis provided by accelerometer, a_x are the applied acceleration acting along

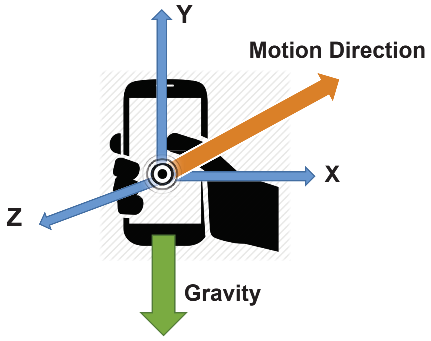


Figure 3: Underlying principle of accelerometer in mobile devices

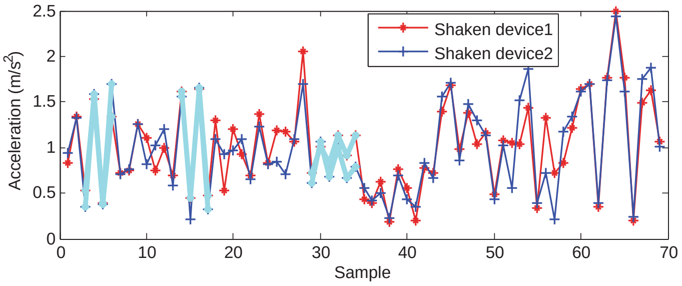


Figure 4: An example of original data recorded by the shaken devices. The denoted areas indicate the temporary regularity on user's shaking.

in X-axis, S_x is the scale factor error, B_f is the measurement zero-offset bias and n_x is the random noise. The measurement in Y-axis and Z-axis can be computed similarly. After obtaining the acceleration in 3-axis, the device can compute the acceleration in the motion direction by:

$$a_u = \sqrt{\tilde{a}_x^2 + \tilde{a}_y^2 + \tilde{a}_z^2}, \tag{2}$$

where a_u is the user driven acceleration, $\tilde{a}_x, \tilde{a}_y$ and $\tilde{a}_z$ are the measurements provided by the accelerometer in 3 axes respectively. We observe from Equation 1 that the acceleration measured by the sensors contains the original acceleration in its direction, as well as other factors which we denote as multi-factor noise. The latter affects the precision of the accelerometer's output. To correct this, a number of statistical and signal transformation methods (e.g. Median Absolute Deviation - MAD, and FFT) can be applied to extract the main features from the mixed data. For example, in [31] these approaches were utilized to extract features from data generated from human motion. However, such feature-selection based processes usually eliminate the influence of multi-factor noise during the selecting of features because the presence and influence of dominant features can outweigh those of multi-factor noise.

Thus, in this paper, we propose a new solution to generate a shared secret key without using such statistical and signal transformation methods to extract common features from the sensing data. A downside of not using the above mentioned statistical and signal transformation methods is the inability to acquire matching data in the devices that moved in different patterns (e.g. when a user is walking, data obtained from devices worn on the user's wrist and waist can present obvious distinction and both data will not be matching without feature selection). Compared with extracting features, a more straightforward approach to yield matching data in devices is to let the devices move in a similar pattern. Thus, we allow the users to be actively involved in the generation of the initialization key. To be specific, users first hold their devices together (on the same hand), and then shake them in an arbitrary pattern. The nearly same and random motion pattern can be recorded, which can yield similar data in the shaken devices for shared key establishment.

Since the wearable devices are generally portable (e.g. Apple watch, Garmin HRM-Tri heart rate monitor[2]), such operations will be trivial for the users to perform in almost any situation. An example of the original sensing data in different devices generated by a human's arbitrary shaking is shown in Fig. 4. Here, all original data discussed in our study are assumed to have been processed by the high-pass filter because the contribution of the force of gravity must be eliminated in order to measure the real acceleration of the devices [43]. And the sensing devices for all of the two-end data (Alice and Bob) are the same (ZTE U817 and Coolpad 5892). All the sampling data procedures given as examples in the following are the random motions for demonstration, unless otherwise specified.

Upon receiving the original data, a filter process is needed in order to increase the matching rate of derived bit sequence at both ends. We choose the simple Moving Average (MA) filter, which is a general filter method that can be used to smooth out short-term fluctuations between neighboring samples. It should not significantly affect the value-difference relation of neighboring samples, which is important during quantization. Therefore, we assign the filter parameter to be 2, which means each sample is only made an average smooth with its previous one. Fig. 5 depicts an implementation of MA filter over the sensing data (Fig. 4 represents the original data).

Another necessity to introduce the MA filter is that it enables our scheme to take into account the situation when a user's free-form shaking presents temporary regularity, which is possible in actual scenarios (i.e. it is unrealistic for users to always shake the devices in an arbitrary pattern and temporarily regular shaking is a reasonable assumption). However, this could be a vulnerability to be exploited by the adversary, who can more easily mimic the regular motion than other random motion. Such short-term periodic motion patterns can be presented as the samples denoted in bold lines in Fig. 4, where we can see the patterns of those samples showing distinct regularity.

[2] https://buy.garmin.com/en-GB/GB/p/136403

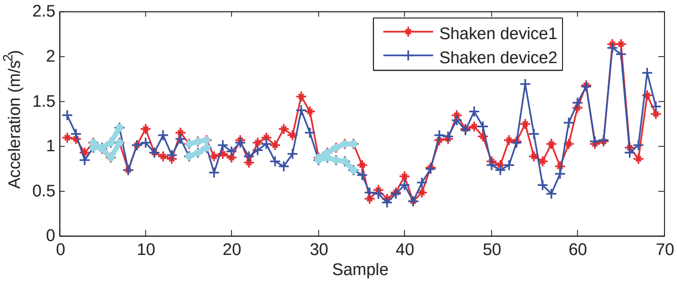


Figure 5: A comparative summary of sensing data at both ends after the implementation of Moving Average filter. The denoted areas are corresponding with that in the preceding figure.

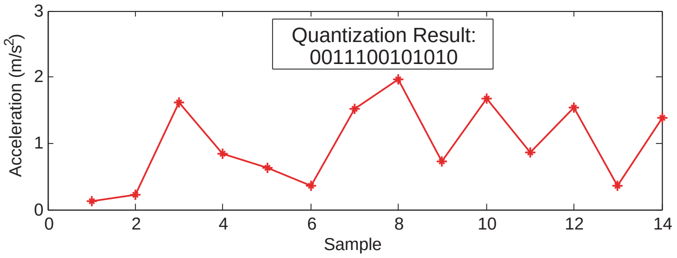


Figure 6: An implementation of the proposed value-difference based quantizer

The MA filter can be use to mitigate such potential risk, since the MA filter can smooth the value relation of the neighboring samples. Therefore, the value relation of the periodic samples becomes very similar and gentle after being processed (i.e. the value differences become trivial). This is due to the complementary nature of the regularity of the neighboring samples, which can be illustrated as the denoted samples in Fig. 4 and Fig. 5. As we will discuss in Section 4.3, these samples can be filtered during the reconciliation procedure; thus, enabling our scheme to generate a secure key even though the user has shaken the devices in an imitable pattern temporarily.

4.2. Quantization

We now describe our value-difference based quantization method, which is based on the comparison of values between a sample and its preceding sample, and we assume that both devices (Alice and Bob) have collected the same number of samples. The value of each bit extracted from the samples is determined by the value difference between each sample and its preceding sample. If the current sample is smaller than its previous sample, then the bit extracted from this sample is 1; otherwise, it is 0. Fig. 6 shows an implementation of our quantization method on the data collected from a random shaking (using Coolpad 5892 as the sensing device), and the description is provided in Algorithm 1.

Algorithm 1: Proposed value-difference based bit-extraction algorithm

```
Data: $cur\_sample, pre\_sample, new\_sample$
Result: Bit sequence
if $sensor\ updates\ new\_sample$ then
    $cur\_sample = new\_sample$;
    if $cur\_sample < pre\_sample$ then
        output 1;
    end
    else
        output 0;
    end
    $pre\_sample = cur\_sample$;
end
```

The proposed quantization allows us to more concretely determine the motion trend of the devices by investigating the value-difference relation of neighboring samples in the sensing data. For example, if the value differences of a series of successive samples with their previous ones are positive, then we can interpret that the device had been accelerating during that period. And the randomness of users' shaking devices can be perceived as the unpredictable acceleration or deceleration of the shaken devices. Thus, the extracted bit sequences are more related to the unpredictability of users' physical motion (i.e. the derived sequences are more unpredictable).

In general, deviation caused by multi-factor noise will not distinctly distort the value-difference relation of samples but only result in relative imprecision of the concrete value as the presence and influence of noise cannot outweigh those of the dominant motion features. Thus, it enables data at different ends to be extracted as the same bits, as long as they have positive or negative value differences with their previous samples (i.e. data in different ends are not required to be highly matching).

On the other hand, while the value deviation resulting from multi-factor noise generally causes indistinct influence, it also increases the difficulty to derive matching bit sequence at different ends. This is because some deviations are too trivial to be identified to have caused inconsistency in the bit sequence. For example, a trivial value difference of a sample with its previous one can be positive at one end and be negative at the other end, and thus the extracted bits will be 0 and 1, respectively. The users' shaking may also result in some small variation on the respective data, which can result in the above mentioned events (e.g. the variation on the shaken devices seen from Fig. 5). Therefore, we introduce a calibration method to filter these deviation, to be discussed next.

4.3. Reconciliation

A calibration filter process is introduced before the shaken devices can agree on a shared key. In Fig. 7, we depict a comparison of a bit sequence derived before and after the calibration filter. As observed from Fig. 7, the bit sequence extracted without calibration has some discrepancy in both devices, which are mainly due to the presence of multi-factor noise and they can draw stronger influence in

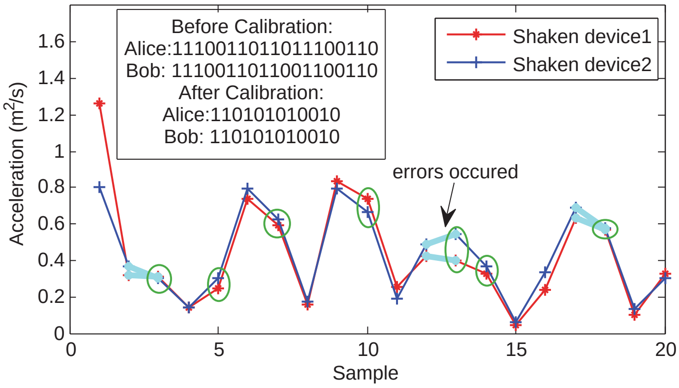


Figure 7: An implementation of calibration filter to filter mismatched bits in both ends. The bold lines denote the area(s) where errors are likely to occur. The samples denoted in green circle are discarded since their value differences with their previous ones are smaller than the defined deviation threshold (0.15).

those samples with small value difference with their previous ones. As denoted in bold lines of Fig. 7, samples that have indistinct value differences with their previous samples are a likely cause of mismatch in the sequence at both ends. Therefore, we define a deviation threshold to filter these mismatched bits.

A deviation threshold is defined, and if the value difference of a sample with its previous sample is smaller than the deviation threshold, then this bit would be discarded. It can be easily observed that the assignment of the threshold can influence both the bit matching rate and the length of derived bit sequence. A larger threshold can filter more mismatched bits. However, the matched bits will be discarded as well if their value differences with the previous samples are smaller than the defined threshold (as shown in Fig. 7); thus, decreasing the length of derived sequence and *vice versa* for a small threshold. Therefore, a moderate deviation threshold should be considered in order to balance the key matching rate with the key length. As the presence and influence of the deviation are unpredictable, we conducted experiments to study the distribution of the value difference of mismatched bits. The experiments were conducted by shaking in varying movements to simulate the real scenarios (e.g. the shaking contained relaxed movement, movements with temporary repeatedness, etc.). We collected the deviation value of 14100 mismatched bits and the range of the data was from $1.85{*}10^{-6}$ to $4.428378m/s^2$. The corresponding probability distribution is expressed as:

$$F(x) = \begin{cases} 0.7357 & deviation = 0.10 \\ 0.9130 & deviation = 0.15 \\ 0.9499 & deviation = 0.50 \\ 0.9913 & deviation = 1.50 \\ 1.0000 & deviation = 4.50, \end{cases} \tag{3}$$

where *deviation* is the defined deviation threshold, x are the mismatched bits whose value differences with their previous ones are smaller than *deviation*, and $F()$ is the function to compute the proportion of x in the mismatched bits.

Table 1: Loss rate of the matched bits to be discarded under different deviation threshold. The total amount of the matched bits is 9610 bits

Deviation	0.10	0.15	0.50	1.50	4.50
Loss Rate (%)	41.94	58.38	78.19	98.86	100.00

To evaluate the influence of different thresholds on the key length, we computed the loss rate, which is the percentage of the matched bits to be discarded. The results are shown in Table 1, where the data were ranged from $2 * 10^{-6}$ to $4.278323m/s^2$. As observed from Table 1, the threshold of 0.10 group can yield the longest sequence as its loss rate is the lowest among the groups. However, those mismatched samples with a value difference of smaller than 0.10 comprise only 73.57%, which can lead to a low key matching rate. Considering both the bit matching rate and the length of derived sequence, we chose 0.15 as the deviation threshold. Our evaluation given in Section 5.1 demonstrated that our scheme could maintain a stable and high matching rate by using the defined threshold of 0.15. A matched bit sequence after the implementation of the calibration filter can be seen in Fig. 7.

The situation when the motion pattern of the shaken devices presents short-term regularity (as we have mentioned in Section 4.1) can be addressed during this procedure. Because of the inter periodicity of the regular samples, the value differences of these data with their preceding ones become trivial after the implementation of the MA filter, as observed from Fig. 5. While the proposed calibration process can filter the samples with a value difference of smaller than the defined threshold, these insecure samples are discarded as well. Thus, we can ensure that the secret key derived from users' shaking are random (i.e. the derived key only adopts the "stochastic" data in the shaken devices while the "regular" data will be discarded).

After both devices have filtered their unqualified bits, the devices need to agree on the filtered bits since the position of these bits may vary from each other. We can simplify the procedure as follows: device 1 (Alice) locates and records the indexes of the filtered bits and these indexes will be sent as a filtering list to the other device (Bob), who operates in a similar fashion. After both Alice and Bob have sent and received a filtering list, they discarded all the unqualified bits and the remaining bits can form a shared key at both ends. However, as such procedure involves mutual communication, the following attacks should be considered. During reconciliation, an active attacker might attempt attacks such as message modification. To address such issues, we adopt the approach in [38], which proposed a scheme incorporating data-origin authentication and resilience to active attacks. Specifically, we modified the reconciliation procedure based on the scheme outlined in [38], and the extended mutual agreement is explained as follows:

1) After Alice has filtered the unqualified bits, she sends a list containing the indexes of filtered bits to Bob.

2) On receiving Alice's message, Bob is able to derive a key after filtering the bits specified in Alice's message. Then, Bob checks whether the length of derived key is sufficiently long. An agreed secure length threshold of secret key is predefined and publicly available (e.g. this value would be 128 if AES-128 is the encryption algorithm). Bob discards the received message and terminates the reconciliation if the derived key does not satisfy the length threshold (i.e. the derived key is insecure). Otherwise, Bob sends his filtering list to Alice along with a message authentication code (MAC), which envelopes Bob's filtering list and is encrypted using the derived key. MAC is employed here to authenticate the message sent from Bob and the keyed-hash message authentication code (HMAC) is preferred since it can verify both data integrity and data origin.

3) After receiving Bob's message, Alice can derive the key by discarding the unqualified bits included in Bob's message. Then, Alice uses the derived key to decrypt the received HMAC to verify the integrity of message and the identity of message sender. Thus, Alice is able to determine whether her derived key is a legitimate shared key.

4.4. Real-time Key Establishment

To achieve a real-time rekeying mechanism, there should be sufficient data (and sources) for session key generation. The quality of sequence comprising the cipher keys is a key factor for a secure cryptosystem as it determine the security of the keys. Ideally, random number generation is based on some physical processes with inherent randomness (e.g. white noise). In this section, we propose a method that utilizes the kinetic sensory data from sensing users' physical motion to provide secure random number for session key construction.

When a user is using the on-body devices, the user's motion can serve as a real-time data source for key construction in the devices. With the establishment of shake-to-generate key, we use the data generated from users' motion to construct real-time keys directly without additionally processing the sensing data. This avoids the overhead associated with data processing.

We first investigate the original data obtained from users' walking. Fig. 8 gives an example of the data generated from the sensing devices placed on four body areas during a human's regular walking (the pattern of the data may vary between different people, and the placement and types of the sensing devices are given in Section 5.1). Intuitively, there is no apparent regularity reported in these sensing data, mainly due to the fact that the original data were distorted by multi-factor noise; thus, concealing the underlying periodicity and regularity of gait.

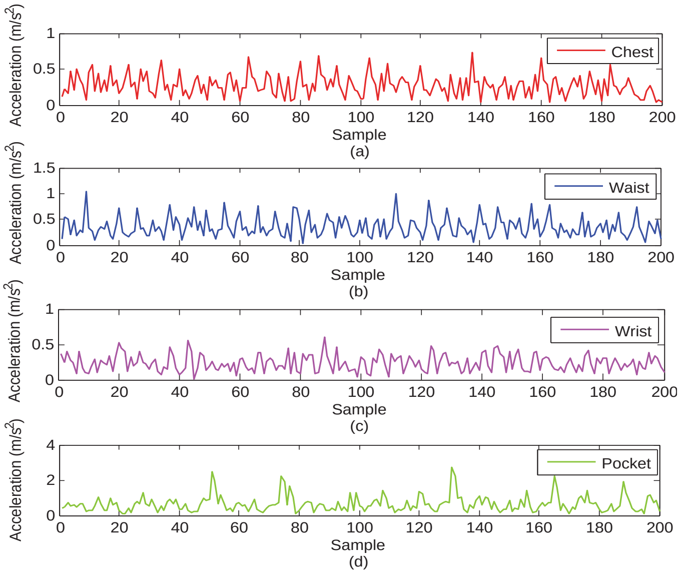


Figure 8: An example of real-time sensing data recorded by accelerometer in: (a) device worn on user's chest; (b) device worn on user's waist; (c) device worn on user's wrist; and (d) device carried in user's pocket during a user's regularly walking.

We used 6 of the NIST statistical tests (see Table 3 for the name of each test) to quantize the randomness of the bit sequence derived from these data [44]. We found that the data failed to pass the tests, which indicated the data could not be considered random (i.e. cannot be used for key generation). We attributed the failure to the influence of the features and regularity related to the gait. This could outweigh that of multi-factor noise, whose involvement in the derived key was indistinct by comparison. Thus, not all data derived from users' motion can be used for generating random number directly.

To address the limitations on regularity of the sensing data, we performed a specific data selection on the insecure data to select the stochastic and secure data for key establishment. As we have previously discussed, multi-factor noise generally yield stronger influence in the samples whose value differences with their preceding ones are trivial. The data that are subjected to the influence of multi-factor noise incorporate more randomness; thus, they are suitable to be used for random number generation. We defined a selection threshold to choose such data, whose value differences with their previous ones should be smaller than the defined threshold. We denoted the selected data as noise-dominance data where multi-factor noise could draw more influence. Thus, the selected data could not accurately reflect users' actual motion (i.e. these data remain more randomness). The remaining unselected data were not preferred since they mainly incorporated gait characteristics, which were inherently insecure.

Essentially, a smaller selection threshold could help to select data that have stronger entanglement with multi-factor noise. However, such threshold should not be too small, which might result in an increase in the randomness

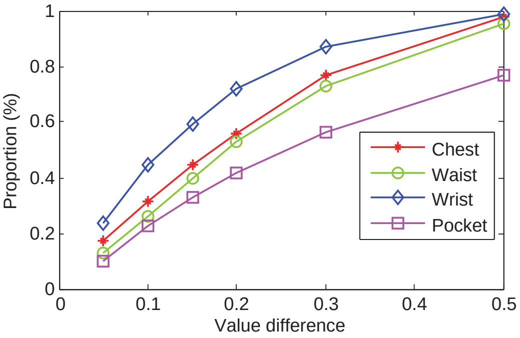


Figure 9: Distribution of value difference of the sensing data generated from devices located in different body areas

of the selected data while the amount of qualified data could diminish as well (i.e. only a small amount of data in the sensing data can be selected).

Based on the above, we performed the following experiments to investigate the distribution of the value difference between the sensing data to identify an optimal threshold. Fig. 9 presents the distribution of value difference of the data generated from human participant's regular walking, which was repeated 100 times and each experiment lasted 30s (the placement and types of the sensing devices are given in Section 5.1). The sensing devices were placed in four body areas, namely: chest, waist, wrist and pocket.

As mentioned before, the randomness of the derived sequence should be considered as well (e.g. a larger selection threshold may result in the selected data incorporating more data associated with gait features). Based on the results in Fig. 9 and our experimental results from 6 of the NIST randomness test suite over the data generated using different thresholds (from 0.05 to 0.3), we defined 0.15 as the selection threshold. A detailed evaluation on the data generating from using our defined threshold will be discussed in Section 5.2.

The above data selection is based on our proposed quantization method, and the procedures of real-time key establishment can be expressed as follows:

1) Both devices (Alice and Bob) acquire data from sensing the user's motions, where samples with a value difference of smaller than 0.15 with their previous one will be selected as noise-dominance data, which can be used for key construction. Algorithm 2 is a modified quantization method that can select and quantize the noise-dominance data.

2) The extracted bit sequence will be encrypted using the shared shake-to-generate key, before being sent to each other along with a message digest containing the hash value of the exchanged data for verifying the data integrity.

3) After validating the integrity of the received message, Alice and Bob can construct a new shared secret key to be used in the next session by concatenating their own sequence with the received authenticated sequence.

Algorithm 2: Modified value-difference based bit-extraction algorithm to identify and quantize noise-dominance data

```
Data: $cur\_sample, pre\_sample, new\_sample$
Result: Bit sequence extracted from
        noise-dominance data
if $sensor\ updates\ new\_sample$ then
    $cur\_sample = new\_sample$;
    if $pre\_sample - cur\_sample \leq$
      $0.15$ && $pre\_sample > cur\_sample$ then
        Output 1;
    end
    else if $cur\_sample - pre\_sample \leq$
      $0.15$ && $cur\_sample \geq pre\_sample$ then
        Output 0;
    end
    else
        No bit is extracted from $cur\_sample$;
    end
    $pre\_sample = cur\_sample$;
end
```

Under the assumption in this paper (i.e. the security of our scheme is equivalent to the strength of all the keys generated under the scheme, and other factors such as the reliability of the encryption protocols are not within consideration), once a secure shared session key has been successfully established, both devices can communicate with each other in an insecure channel using symmetric encryption algorithms such as AES-128. And the security during the secret message exchange in this procedure is determined by the encryption key (i.e. the shake-to-generate key), whose strength will be considered in Section 5.2.

Based on the above, our proposed key management scheme is achieved by a combination of procedures, where each procedure serves for different purposes (e.g. the reconciliation procedure to agree on a shake-to-generate key, the generation of noise-dominance data to provide the source of real-time key). Evaluation and analysis on the proposed scheme are to be discussed next.

5. Evaluation and Analysis

In this section, we first evaluate the utility of our scheme by investigating the matching rate of shake-to-generate key and, the generation rate of shake-to-generate key and random number from devices on different body areas. We then analyze the security and performance of the proposed scheme.

5.1. Utility Study

5.1.1. Design and Procedure

The sampling rate of the accelerometer determines the quality of the data stream (i.e. the performance of our scheme), and thus a moderate sampling rate should be considered.

In the procedure for constructing the shake-to-generate key, a higher sampling rate can result in a higher key generation rate as more data can be obtained by the accelerometer per second. However, an increase in the sampling rate may also result in a lower key matching rate because the sensors will be more sensitive to user motion. Thus, there may be more fluctuations in the sensing data and thus decreasing the consistency of the data to be used for shared key construction (i.e. the bit matching rate drops). If the key matching rate is too low, then users may have to repeatedly shake their devices to successfully establish a symmetric key; thus, affecting users' quality of experience.

Therefore, we need to strike a balance between data richness and useability. To choose a suitable sampling rate of the devices during the shake-to-generation operation, we evaluated the performance of the key generation rate and matching rate under three sampling rates, namely: 5Hz, 10Hz and 20Hz, in the experiments.

During real-time key construction, the reconciliation procedure is not required; hence, sampling rate in such scenario can be increased. We evaluated the sequence generated by the sensors running under 10Hz, 20Hz and 40Hz, respectively. Regardless of the generation rate, we first evaluated the security of data obtained from sensors working different sampling rates using 6 of the statistical tests as used previously. However, we found that the results from the groups of 20Hz and 40Hz were unsatisfactory (i.e. cannot pass the randomness test). This was because when the sampling rate was higher, the sensors could capture more detailed user motion information. For example, the periodic features of hand swing could be divided into successive samples captured by the sensors, which incorporated the characteristics of the motion into the data. And because of the high sampling rate, the value differences of neighboring samples were smaller comparing with using a lower sampling rate for sensing (e.g. the acceleration of a smartwatch has an increase of 1 m/s^2 in one second, the average value difference of samples generated from device working under 5Hz is 0.2 (1/5) and this value is 0.025 (1/40) when the sampling rate of device is 40Hz). It also echoed the findings from our experiments which showed that the generated binary sequence increases with a higher sampling rate. Thus, these samples suggested the regularity of users' motion were falsely chosen as noise-dominance data, and the randomness of the derived keys diminished. Therefore, we chose 10Hz as the sampling rate during the real-time key establishment.

Based on the above, the procedure for the utility study experiments were divided into two steps, as follows:

1. Human participant randomly shook devices together to initialize a symmetric key on the devices. The shaking movements were dynamic during which the shaking may be slow, intense, or even present a short-term repeatedness and the human participant could walk during shaking. Each group (5Hz, 10Hz and 20Hz) in the experiments had 50s, 25s and 10s of shaking for 100 times.
2. Human participant worn the sensing devices and walked for about 30s in a habitual manner for 100 times. Four typical body areas for placing wearable devices were studied; namely: chest, waist, wrist and pocket.
   The motion of habitual walking was specifically evaluated as it was the most common motion for general users and it is more likely to provide the weak data, in terms of the features and regularity of gait, compared with other types of motion (e.g. running).

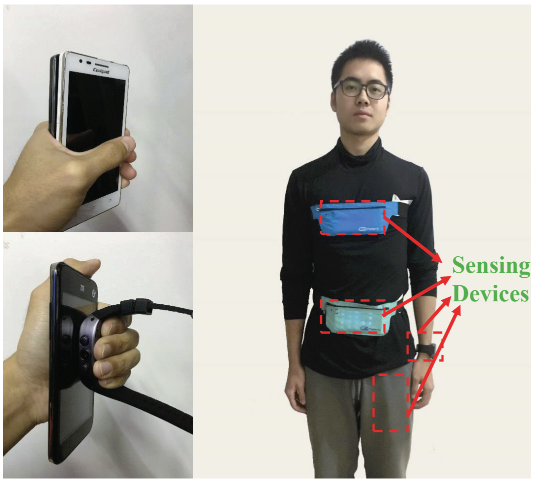


Figure 10: Demonstration of gestures to hold devices and placement of sensing devices in different body areas

Fig. 10 depicts how human participant held the devices before shaking them together, as well as the placement of the sensing devices located in different body parts during the experiments. We used a ZTE U817, two Coolpad 5892 smartphones and a Aomizi smartwatch as the sensing devices.

5.1.2. Findings

The bit matching rate of the key derived from random shaking and the corresponding generation rate computed in terms of our experimental results are presented in Table 2. The acceleration data during the experiments were recorded, where the acceleration ranged from $5.10{*}10^{-4}$ to $4.085041m/s^2$ (5Hz), $1.05 * 10^{-4}$ to $4.255811m/s^2$ (10Hz) and $8.04 * 10^{-5}$ to $3.584016m/s^2$ (20Hz), respectively.

We can see that while the sensors running under 10Hz and 20Hz could yield a longer bit sequence per second, the corresponding key matching rates are not satisfactory. Thus, the sampling rate of 5Hz is more appropriate due to the ability to achieve a higher bit agreement rate.

Fig. 11 presents the average generation rate of data from the on-body devices during the human participant's regular walking. The results varied between different body parts for their different motion patterns. The generation rates in the Chest group and Waist group were close, since their motion patterns were relatively similar. Devices worn

Table 2: Generation rates and matching rates of the shake-to-generate keys under different sampling rates

Sampling Rate	5Hz	10Hz	20Hz
Matching Rate (%)	91.00	79.00	61.00
Generation Rate (bit/sec)	2.027	5.374	10.42

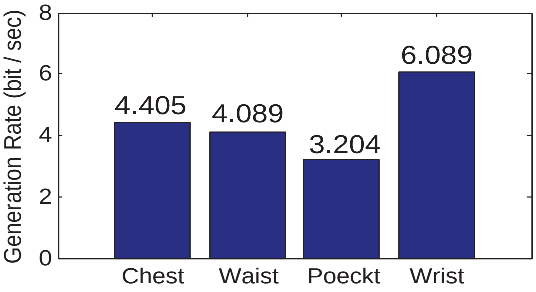


Figure 11: Generation rates of random binary sequence for real-time key construction from devices placed in different body areas

on the user's wrist could yield most noise-dominance data for key construction among the four groups. The findings in Fig. 11 also echoed those of Fig. 9 in Section 4.4, where we studied the distribution of value difference of sensing data to select a moderate selection threshold (e.g. 44.05% of the bits were selected as noise-dominance data in the Chest group as observed from Fig. 11, and this result was corresponding with the proportion of 44.81% in the Chest group in Fig. 9).

Apart from the above experiments, we also evaluated the performance of our scheme under other scenarios (e.g. taking up or down stairs, running, walking irregularly which included sudden acceleration or deceleration, etc.). The findings from these additional evaluations indicated that the on-body devices were able to generate the stochastic data for key construction. We also remark that the more non-habitual the users' motion are, the more secure the proposed scheme is as the real-time key is based on the physical motion whose unpredictability can contribute to the randomness of the derived key.

5.2. Security Analysis

We now demonstrate the security of our scheme in terms of the adversary model, and the security properties (i.e. imitation attacks, passive eavesdrop, active attacks and knowledge of all the procedures and methods).

Imitation Attacks. The shake-to-generate keys are assumed to be resilient to imitation attacks because the data derived in the shaken devices are related to a user's random shaking, and the user's active arbitrary motion are more capable of withstanding imitation attacks compared with walking or other habitual motion. The introduction of the MA filter can also mitigate potential vulnerability on users' temporary regular shaking. It, thus, increases the difficulty for imitation attacks. We conducted experiments to evaluate the possibility of counterfeiting a key by imitation. Videos were recorded during the random shaking. We then attempted to use the same motion pattern to derive the same sequence (key). However, all these attempts failed despite a few success in learning some temporarily regular motion patterns. These regular samples were discarded during the reconciliation procedure. In other words, the imitation attacks over the shake-to-generate key were not successful.

Our specific noise-dominance data selection can mitigate the potential vulnerability of the original data derived from users' gait. Even in the unlikely event that an adversary succeeds in reproducing the gait features of users' walking, the adversary is unable to compromise the key composed of noise-dominance data, since the compromised data related to the gait features have already been discarded and the compromised data would not be included into noise-dominance data. The generated noise-dominance data are user-specific. Specifically, though the data are noise-dominance, they are generated from specific user's motion and thus we can consider the users' physical motion as the seed to generate the random number. It, thus, can ensure the security of the session key, unless the adversary is able to reproduce the same noise-dominance data (i.e. the identical motion), which would be an unrealistic assumption.

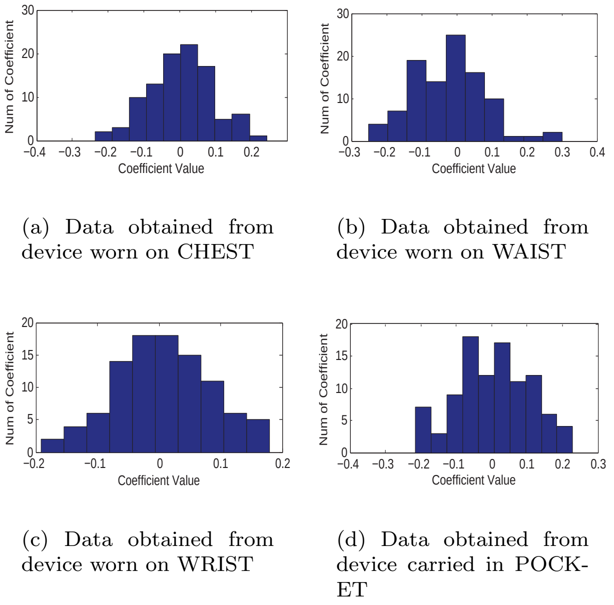
We also conducted experiments to evaluate the possibility on gait imitation attacks, in the sense whether it is possible for any parties (including the legitimate users themselves) to fabricate or "guess" a secret key by imitation. As we assume the possibility of a strong adversary successfully imitating the gait characteristics, we specifically let the human participant serve as both a legitimate user and an adversary trying to fabricate an identical key. This is because a user is likely to be a strong imitator of himself / herself. It can be realized that the requirement for the strength of the evaluated data was stricter compared with using a third party as the adversary.

The Phi coefficient [45] was employed to measure the association for the binary sequences extracted from the noise-dominance data and it can help us to have a better perception of the relationship between the successive bit sequences that came from the same users' walking. A strong association between the evaluated data could infer the possibility on the fabrication of secret key or leakage of knowledge of the key. The Phi coefficient, represented by the symbol ϕ, can be expressed as:

$$\phi^2 = \frac{\chi^2}{n} \tag{4}$$

where χ^2 denotes the Pearson's chi-squared statistic [46], and n is the total number of observations.

Fig. 12 presents the findings of the data from the experimental devices placed on the four body areas. Each group contains results of test over 100 data streams, whose length were 130, 120, 180 and 97 bits, respectively.



(a) Data obtained from device worn on CHEST

(b) Data obtained from device worn on WAIST

(c) Data obtained from device worn on WRIST

(d) Data obtained from device carried in POCKET

Figure 12: ϕ value of the Phi coefficient over the noise-dominance data generated from devices placed on different body areas

Generally, the interpretation of the Phi coefficient is similar to that of the Pearson correlation coefficient [45] and we adopt the rule of thumb for interpreting the size of correlation coefficient from [47]. If the absolute value of the coefficient value ϕ is smaller than 0.3, then the association between the variables would be interpreted as negligible. When $|\phi|$ ranges from 0.3 to 0.5, there is low positive (or negative) association between the data. The association between the bit sequences would be considered non-negligible if $|\phi|$ is larger than 0.5, which may indicate the weakness of the data for the high association.

It could be observed from Fig. 12 that most of the results were sampled from -0.2 to 0.2 and all of the them range between -0.3 and 0.3. Therefore, there are only negligible association between those sequences generated from the same user's regular walking. Based on the above results that even the sequences generated from the same user were negligibly associated, we can regard that any other parties who attempt to imitate the users' motion are likely to be frustrated.

In the very unlikely occasion when a user's gait was "identically" imitated (i.e. the session key was compromised), such security breach would not affect the next session as the compromised key would be revoked.

Passive Eavesdropping. When the reconciliation messages are available to the adversary, the adversary might attempt to parse knowledge of the derived key. The available information in the eavesdropped message is the position of filtered bits in both the communicating devices. However, these bits have been discarded and are not involved in the derived key. We also remind the reader that data in the reconciliation message originated from users' random shaking; thus, such data are unpredictable (due to the randomness of users' physically arbitrary motion). In other words, the adversary cannot gain any useful knowledge from the eavesdropped information.

Active Attacks. The adversary may attempt to perform active attacks by intercepting communications between legitimate devices. During the reconciliation procedure, the adversary may attempt to inject false message into the data exchanged between the two communicating devices. The qualified bits would be discarded if they were included in the adversary's injected message (thus resulting in a short-length key). If such data modification is moderate (i.e. not too much false message to be included in the exchange message), the length of the reconciled key should not be too short, and the key can still be adopted if and only, if the key is of sufficient length (i.e. over the threshold length). However, immoderate data manipulation by the adversary may result in a key with too short length, which is vulnerable to attacks (e.g. brutal attack) and thus the key in such case should be deemed useless. We use the check-on on the key length to address the above situation and thus only the key with sufficient length can pass the mutual agreement procedure.

The adversary may also perform MITM attacks, whose premise is that the adversary can successfully pretend as a legitimate user. Impersonating a legitimate user to communicate with any of the communicating devices requires the adversary to possess (i.e. fabricate) a list from which the targeted communicating device or its partner device can derive a secure key (i.e. a key with sufficient length) according to their own derived sequence. Our solutions are based on the solutions proposed in [38]: 1) check-on on key length and 2) HMAC verification. The first solution can prevent the adversary from "guessing" a list, from which the legitimate devices can generate a short-length key (short length is due to the fact that the list is formed by guessing). If a short-length key is passed, the adversary can "communicate" with the legitimate parties using the short-length key (i.e. the MITM attack succeeded). The second solution can verify both the data authentication and the identity of the message sender. Since the adversary does not know the sequence of the targeted device or its partner device, the above two measures ensure that the adversary is unable to fabricate such a list that can derive a shared long-length key, and the existence of a third party will be detected due to the employment of HMAC.

Knowledge of Methodology. While quantization and other procedures are crucial in constructing random and secure keys, the security of our scheme relies on the randomness in the data source for key construction, which come from users' physical motion. In other words, having knowledge of these procedures will not weaken the security of our scheme unless the adversary can produce the identical sensory data stored in the legitimate devices by imitation, which would be infeasible as discussed previously.

Randomness of Derived Key. The security of a cryptographic key can be measured by its randomness, which indicates the strength of the key statistically. We have discussed how our scheme can withstand the potential attacks and address the vulnerability of the insecure sensing data. To further evaluate the security of the keys generated under our scheme, we conducted statistical tests to validate the randomness of the key.

Specifically, we used the wide-adopted NIST statistical test suite for the validation of the random numbers (see [31, 36, 37, 38, 39] for example). There are 16 tests specified in this test suite and we ran 6 of them, namely: Frequency (Monobit) test, Frequency test within a block, Runs test, Serial test, Approximate entropy test and Cumulative sums test. These tests evaluate the randomness of the data from different aspects and we refer the interested reader to [44] for a detailed description of these tests. The test input followed the recommended input length specified in [44] and the remaining tests were not selected as the recommended length were too large (e.g. the Overlapping template matching test recommends a minimum of 10^6 bits input). We plan to obtain larger data pool in the future to complete the remaining tests.

The candidate sequence could be perceived as having passed the test if the p-value, output of the test, is greater than 0.01. Table 3 gives the percentage pass rate of the tested data, including the shake-to-generate keys, the selected sequence came from the devices carried in the pocket, worn on the wrist, waist and chest respectively.

Our experimental results in Table 3 and the above analysis could ensure the security of the key generated under our scheme, which validated the security of the proposed scheme to be used for key establishment for real-world application.

5.3. Performance Analysis

Compared with existing key generation schemes, two distinct features of our scheme are its *lightweight* design and *real-time* property.

Compared with the work in [31], the data process procedure in our scheme is much simpler while maintaining the strong security of the keys. We introduce a Moving Average filter once only (while [31] employed a combination of data transformation methods) to smooth the data derived from users' random shaking and the real-time keys are established directly from the original sensing data without the need for additional processing. Compared with Mathur's level-crossing algorithm in [38], our bit-extraction method is lightweight in terms of reducing storage requirement for the data (in Mathur's algorithm, all data should be stored on the devices for statistical analysis before quantization). The above design have contributed to the lightweight property of our scheme.

The real-time key construction is based on sensing the users' physical motion (e.g. walking). The proposed scheme is able to generate secure key continuously only if the accelerometer can sense the users' real-time motion. And thus the real-time key is available for the communication among the on-body devices.

6. Conclusion

Wearable device is a trend that is unlikely to fade anytime soon in our increasingly interconnected society, also evidenced by recent trends in smart cities, smart nations, etc. Existing wearable devices are generally resource-constraint and designing security solutions for these devices remains an ongoing challenge.

In this study, we proposed a lightweight and real-time key establishment scheme for wearable devices by leveraging the accelerometer embedded in these devices. We first introduced a novel shake-to-generation key establishment method. Unlike the usual approaches to extract features from the sensing data for key construction, our method can "create" stochastic features in the sensing data to establish secure keys by the users' randomly shaking their devices together. A lightweight value-difference based quantization algorithm was then presented, which allowed us to quantize the sensing data. The inter-relation of the data can be studied by investigating the value-difference relation of neighboring samples. Thus randomness of physical motion (e.g. users' random shaking) can be reflected in the derived bit sequence. The proposed data selection method can mitigate the potential for exploitation due to regularity and correlation in the sensing data.

Future work includes building a larger data pool. This would allow us to generate a more robust evaluation dataset (e.g. complete the remaining NIST tests). And the alignment of the devices to collect the same amount of data simultaneously will be further considered as well.

In the future, we also intend to extend the proposed lightweight and real-time key establishment scheme to other Internet of Things devices, such as those deployed in battlefields or military contexts (also known as Internet of Battlefield Things and Internet of Military Things). For example, can we leverage characteristics unique to the devices or their movements to generate secure session keys, which are also mission specific?

Acknowledgement

The research was financially supported by the Open Funding of Guizhou Provincial Key Laboratory of Public Big Data with No. 2017BDKFJJ006, and Open Funding of Hubei Provincial Key Laboratory of Intelligent Geo-Information Processing with No. KLIGIP2016A05.

References

[1] P. Castillejo, J.-F. Martinez, J. Rodriguez-Molina, A. Cuerva, Integration of wearable devices in a wireless sensor network for an e-health application, IEEE Wireless Communications 20 (4) (2013) 38–49.

Table 3: Percentage pass rate of the test over the shake-to-generate key, the noise-dominance sequence from devices on four body areas during human participant's regular walking. Each group represents 100 data streams whose length are 100, 130, 120, 97 and 180 bits, respectively

Test	Shake-to-generate key	Chest	Waist	Pocket	Wrist
Frequency (Monobit)	98.00%	97.00%	98.00%	98.00%	99.00%
Block Frequency	100.00%	98.00%	100.00%	100.00%	100.00%
Cumulative Sums	100.00%	96.00%	96.00%	98.00%	98.00%
Runs	98.00%	98.00%	100.00%	99.00%	96.00%
Approximate Entropy	99.00%	91.00%	95.00%	98.00%	97.00%
Serial	99.00%	95.00%	94.00%	98.00%	96.00%

[2] X.-F. Teng, Y.-T. Zhang, C. C. Poon, P. Bonato, Wearable medical systems for p-health, IEEE reviews in Biomedical engineering 1 (2008) 62–74.

[3] C. J. DOrazio, K. K. R. Choo, L. T. Yang, Data exfiltration from internet of things devices: ios devices as case studies, IEEE Internet of Things Journal PP (99) (2016) 1–1.

[4] C. J. D'Orazio, R. Lu, K.-K. R. Choo, A. V. Vasilakos, A markov adversary model to detect vulnerable ios devices and vulnerabilities in ios apps, Applied Mathematics and Computation 293 (2017) 523 – 544.

[5] Q. Do, B. Martini, K.-K. R. Choo, A data exfiltration and remote exploitation attack on consumer 3d printers, IEEE Transactions on Information Forensics and Security 11 (2016) 2174–2186.

[6] G. Sun, D. Liao, H. Li, H. Yu, V. Chang, L2p2: A location-label based approach for privacy preserving in lbs, Future Generation Computer Systems 74 (2017) 375–384.

[7] N. Sultan, Reflective thoughts on the potential and challenges of wearable technology for healthcare provision and medical education, International Journal of Information Management 35 (5) (2015) 521–526.

[8] M. Al Ameen, J. Liu, K. Kwak, Security and privacy issues in wireless sensor networks for healthcare applications, Journal of medical systems 36 (1) (2012) 93–101.

[9] P. Vijayakumar, S. M. Ganesh, L. J. Deborah, B. S. Rawal, A new smartsms protocol for secure sms communication in m-health environment, Computers & Electrical Engineering.

[10] Y. Yang, X. Zheng, X. Liu, S. Zhong, V. Chang, Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system, Future Generation Computer Systems.

[11] Y. Yang, H. Cai, Z. Wei, H. Lu, K.-K. R. Choo, Towards lightweight anonymous entity authentication for iot applications, in: Australasian Conference on Information Security and Privacy, Springer, 2016, pp. 265–280.

[12] Y. Yang, J. Lu, K.-K. R. Choo, J. K. Liu, On lightweight security enforcement in cyber-physical systems, in: International Workshop on Lightweight Cryptography for Security and Privacy, Springer, 2015, pp. 97–112.

[13] D. He, N. Kumar, H. Wang, L. Wang, K.-K. R. Choo, A. Vinel, A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network, IEEE Transactions on Dependable and Secure Computing.

[14] V. Chang, M. Ramachandran, Towards achieving data security with the cloud computing adoption framework, IEEE Transactions on Services Computing 9 (1) (2016) 138–151.

[15] V. Chang, Y.-H. Kuo, M. Ramachandran, Cloud computing adoption framework: A security framework for business clouds, Future Generation Computer Systems 57 (2016) 24–41.

[16] S. Namasudra, P. Roy, P. Vijayakumar, S. Audithan, B. Balusamy, Time efficient secure dna based access control model for cloud computing environment, Future Generation Computer Systems 73 (2017) 90–105.

[17] W. Ren, S. Huang, Y. Ren, K.-K. R. Choo, Lipisc: a lightweight and flexible method for privacy-aware intersection set computation, PloS one 11 (6) (2016) e0157752.

[18] W. Ren, R. Liu, M. Lei, K.-K. R. Choo, Segoac: A tree-based model for self-defined, proxy-enabled and group-oriented access control in mobile cloud computing, Computer Standards & Interfaces 54 (2017) 29–35.

[19] P. Vijayakumar, M. Azees, V. Chang, J. Deborah, B. Balusamy, Computationally efficient privacy preserving authentication and key distribution techniques for vehicular ad hoc networks, Cluster Computing (2017) 1–12.

[20] P. Vijayakumar, M. Azees, A. Kannan, L. J. Deborah, Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks, IEEE Transactions on Intelligent Transportation Systems 17 (4) (2016) 1015–1028.

[21] R. Amin, N. Kumar, G. Biswas, R. Iqbal, V. Chang, A light weight authentication protocol for iot-enabled devices in distributed cloud computing environment, Future Generation Computer Systems.

[22] C. C. Tan, H. Wang, S. Zhong, Q. Li, Ibe-lite: a lightweight identity-based cryptography for body sensor networks, IEEE Transactions on Information Technology in Biomedicine 13 (6) (2009) 926–932.

[23] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, 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.

[24] P. A. Shaltis, A. Reisner, H. H. Asada, Wearable, cuff-less ppg-based blood pressure monitor with novel height sensor, in: Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE, IEEE, 2006, pp. 908–911.

[25] S. Patel, H. Park, P. Bonato, L. Chan, M. Rodgers, A review of wearable sensors and systems with application in rehabilitation, Journal of neuroengineering and rehabilitation 9 (1) (2012) 21.

[26] J. Zhou, Z. Cao, X. Dong, N. Xiong, A. V. Vasilakos, 4s: A secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks, Information Sciences 314 (2015) 255–276.

[27] S. T. Ali, V. Sivaraman, D. Ostry, Zero reconciliation secret key generation for body-worn health monitoring devices, in: Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, ACM, 2012, pp. 39–50.

[28] M. Mana, M. Feham, B. A. Bensaber, Trust key management scheme for wireless body area networks., IJ Network Security 12 (2) (2011) 75–83.

[29] Z. Zhang, H. Wang, A. V. Vasilakos, H. Fang, Ecg-cryptography and authentication in body area networks, IEEE Transactions on Information Technology in Biomedicine 16 (6) (2012) 1070–1078.

[30] K. K. Venkatasubramanian, A. Banerjee, S. K. S. Gupta, Pska: Usable and secure key agreement scheme for body area networks, IEEE Transactions on Information Technology in Biomedicine

14 (1) (2010) 60–68.

[31] W. Xu, G. Revadigar, C. Luo, N. Bergmann, W. Hu, Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication, in: 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), IEEE, 2016, pp. 1–12.

[32] D. Gafurov, E. Snekkenes, P. Bours, Spoof attacks on gait authentication system, IEEE Transactions on Information Forensics and Security 2 (3) (2007) 491–502.

[33] A. Hadid, M. Ghahramani, V. Kellokumpu, M. Pietikäinen, J. Bustard, M. Nixon, Can gait biometrics be spoofed?, in: In Pattern Recognition (ICPR), 2012 21st International Conference on, 2012, pp. 3280–3283.

[34] R. Kumar, V. V. Phoha, A. Jain, Treadmill attack on gait-based authentication systems, in: Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on, IEEE, 2015, pp. 1–7.

[35] R. Mayrhofer, H. Gellersen, Shake well before use: Intuitive and secure pairing of mobile devices, IEEE Transactions on Mobile Computing 8 (6) (2009) 792–806.

[36] S. L. Hong, C. Liu, Sensor-based random number generator seeding., IEEE Access 3 (2015) 562–568.

[37] K. Wallace, K. Moran, E. Novak, G. Zhou, K. Sun, Toward sensor-based random number generation for mobile and iot devices, IEEE Internet of Things Journal.

[38] S. Mathur, W. Trappe, N. Mandayam, C. Ye, A. Reznik, Radio-telepathy: extracting a secret key from an unauthenticated wireless channel, in: Proceedings of the 14th ACM international conference on Mobile computing and networking, ACM, 2008, pp. 128–139.

[39] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, S. V. Krishnamurthy, On the effectiveness of secret key extraction from wireless signal strength in real environments, in: Proceedings of the 15th annual international conference on Mobile computing and networking, ACM, 2009, pp. 321–332.

[40] M. Li, S. Yu, W. Lou, K. Ren, Group device pairing based secure sensor association and key management for body area networks, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–9.

[41] J. Hyuk Park, S. Gritzalis, C.-H. Hsu, R. Roman, J. Lopez, Integrating wireless sensor networks and the internet: a security analysis, Internet Research 19 (2) (2009) 246–259.

[42] M. Park, Y. Gao, Error and performance analysis of mems-based inertial sensors with a low-cost gps receiver, Sensors 8 (4) (2008) 2240–2261.

[43] Sensorevent—android developers, `https://developer.android.com/guide/topics/sensors/sensors_motion.html`.

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

[45] H. Cramér, Mathematical Methods of Statistics (PMS-9), Vol. 9, Princeton university press, 2016.

[46] R. L. Plackett, Karl pearson and the chi-squared test, International Statistical Review/Revue Internationale de Statistique (1983) 59–72.

[47] D. E. Hinkle, W. Wiersma, S. G. Jurs, Applied statistics for the behavioral sciences.