Usable and Secure Pairing Based on Handshake for Wrist-Worn Smart Devices on Different Users

Xiaohan Huang[1], Guichuan Zhao[1], Qi Jiang[1,2(✉)], Xindi Ma[1], Youliang Tian[3], and Jianfeng Ma[1]

[1] School of Cyber Engineering, Xidian University, Xi'an 710071, China
609117168@qq.com, 1078161458@qq.com, {jiangqixdu, xdma}@xidian.edu.cn, jfma@mail.xidian.edu.cn
[2] Peng Cheng Laboratory, Network Communication Research Centre, Shenzhen 518055, China
[3] College of Computer Science and Technology, Guizhou University, Guiyang, China
youliangtian@163.com

Abstract. Wrist-worn smart devices are being used to share various sensitive personal information in various fields such as social, medical, sports, etc. Secure pairing establishing a trusted channel between the involved devices is a prerequisite to ensure data transmission security. Handshake has been proposed to realize secure pairing between devices worn by different users without pre-shared knowledge, the participation of third parties and complex user interactions. However, existing schemes cannot meet the practical requirement in terms of time delay and security. In this paper, we proposed a feasible handshake based secure pairing scheme, which utilizes the handshake acceleration data. Specifically, we quantify the features of acceleration data through random threshold values, to shorten the handshake time required for guaranteeing the length of the negotiated key. Besides, we propose an optimal feature selection algorithm that improves the success rate and security of the system. What's more, security analysis indicates that our solution can resist man-in-the-middle attacks. Experiments are performed on our scheme, which show that the proposed scheme is robust and secure. Users only need to take a few seconds to perform simple operations, and the devices can automatically pair securely.

Keywords: Handshake · Secure pairing · Smart wearable devices · Acceleration · Key negotiation

1 Introduction

Nowadays, wrist-worn smart devices are ubiquitous in our lives, the embedding of sensors (e.g., accelerometer, gyroscope and heartbeat detector) enable them widely used in health monitoring, activity recognition, and personal assistance. In some social occasions, an increasing number of people use their wrist-worn smart devices to sharing various personal information, such as business cards, music, and personal pictures, etc.

H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 349, pp. 493–510, 2021.
https://doi.org/10.1007/978-3-030-67537-0_30

A typical application scenario of employing wrist-worn devices for information sharing is shown in Fig. 1. Users wearing the devices are considered to be in secure domain, and they share data through wireless channels [1], such as Bluetooth and Wi-Fi.

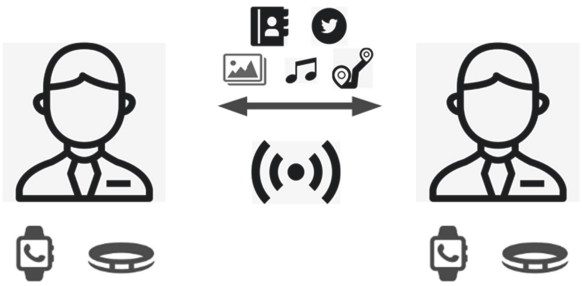

Fig. 1. The scenario of information sharing.

Unfortunately, the wireless channels for information sharing is inherently open due to its public nature, which could cause various types of attacks such as eavesdropping [2], tampering, and man-in-the-middle (MITM) attack [3–5]. Therefore, to share personal information securely, secure paring between two devices that have never known each other before is an urgent requirement, based on which a secure communication channel can be established to transmit sensitive information [6]. However, it faces the following challenges to fulfill secure pairing in this setting. First, two pairing devices on different users do not have any prior security context or a common point of trust, making the authentication between them difficult to achieve. Second, as for the users, almost zero effort should be required to complete secure pairing [1], since complicate user operations may cause users without security consciousness to skip the secure pairing procedure. Third, the energy consumption and time delay should be minimized to be fit the resource-constrained wrist-worn smart devices.

A number of secure pairing solutions have been proposed [7–12], while few of existing solutions caters to the scenario shown in Fig. 1, which is the concern of this paper. Although several works tried to address this issue [13–16], none of them satisfies both security and usability required by practical use, making them still have some distance from an ideal secure pairing scheme for wrist-worn devices on different users.

Handshake, a common form of physical contact between human beings, is one of the most promising solution to achieve secure pairing in this scenario. Handshake-based schemes usually takes handshake acceleration as the common input for secure pairing, however, existing methods cannot meet both security and usability requirements. For example, the scheme in [17] and [18] both have good usability, while their auxiliary data used for the key negotiating are not secure enough, which may reveal the secret key information. The scheme in [19] guarantees the security, while its key generation rate is low, requiring longer user handshake time, leading to poor usability.

Therefore, in this paper, we propose a novel handshake-based secure pairing scheme which achieves both the security and usability. Specifically, we obtain optimized features from the accelerometer data through feature extraction and feature selection, from which

we extract the similar witness and negotiate a symmetric key through fuzzy commitment, implementing secure pairing between the devices. In summary, the main contributions of this paper are outlined as follows.

- We propose a secure pairing scheme based on handshake acceleration, which enables secure pairing between wrist-worn smart wearable devices equipped by different individuals.
- We quantify the acceleration features through random threshold values, which increases the rate of witness generation thanks to making full use of the information contained in the acceleration data.
- We sort the acceleration features based on Euclidean distance and propose the optimal feature selection algorithm, which makes the witness generated by the two devices has more similarity. In addition, to ensure that the auxiliary data will not leak the acceleration information, we introduce the chaff features in the optimal feature selection algorithm.
- We evaluate the performance and security of our scheme by experiments and theoretical analysis. The results show that the proposed wrist-worn smart devices pairing scheme is robust and is able to resist both the passive attacks and active attacks. The key generation rate can reach 87bit/s, it only takes about three seconds to perform secure pairing.

The remainder of this paper is organized as follows: Sect. 2 reviews the related work. Section 3 introduces the required preliminary. Section 4 provides the details of our proposed scheme. In Sect. 5, the optimal feature selection algorithm is detailed. We evaluate the system by evaluating its performance in Sect. 6 and analyzing its security in Sect. 7, followed by the conclusion in Sect. 8.

2 Related Work

Most wearable devices today transmit data via short-range wireless communication technologies such as Bluetooth, Wi-Fi, etc., through which the data is easily leaked and tampered with [20]. Security is a matter of concern for information transmissions between them [21]. A simple method for pairing two devices is to have the users to enter the same password on both devices. However, it is shown that the passwords chosen by users are generally easy to guess. Another method is that, a random number is generated and displayed on the output interface of one device, then it is typed by the user on another device to be paired with [22]. This method is vulnerable to shoulder attacks and lacks user friendliness [23].

To avoid above problems, similar data matching based secure paring schemes have been explored, in which sensors embedded in devices are used to collect common context generated by users [24, 25]. Since shaking is a common behavior which can generate the same context between pairing devices, shaking based device pairing schemes [10, 11] have been proposed, in which the acceleration generated by shaking the devices simultaneously is employed to pair the devices. Specifically, the acceleration signals produced by the shaking patterns are pseudo-random, unique, and difficult to reproduce. Smart-Its

Friends [11] is the first shaking based scheme, which simply uses context matching to pair devices. However, it is vulnerable to MITM, as the context is transmitted in plaintext. ShaVe [10], which extends Smart-Its Friends, combines the session key generated by DH exchange protocol with the accelerometer data collected by shaking to form an acceleration time series, which are compared by both sides to fulfill secure pairing. However, the scheme entails much computational cost due to the encryption operations involved, and is still subject to MITM. Hence, secure pairing based on common context comparisons is not immune to attacks. To address this issue, shaking motion based secure pairing schemes [8, 10] have been proposed, which can not only accomplish device pairing, but also ensure secure transmission subsequently. ShaCK [10] and [8] extract features from the three-dimensional acceleration signal generated by the movement pattern of synchronous shaking devices, which are used to generate symmetric keys. However, they have security issues and are not suitable for the scenario shown in Fig. 1.

Recently, inspired by shaking based secure pairing, in [17–19], the device motion pattern generated by the handshake are used to establish a secure channel between devices, which is attractive for secure pairing between wrist-worn devices worn by different users. Regrettably, existing schemes still cannot satisfy both security and usability required by practical use. In [17], it is proposed to detect the handshake action in real time by extracting acceleration, gyroscope from multiple sensors, which is of high computation cost. Besides, check bits of the Hamming code and parity digit are used as auxiliary data to reconcile a session key, which will cause the leakage of the key information, reducing the security of the scheme. In [18], principal component analysis (PCA) is introduced to reduce the dimension of raw acceleration data. However, its key generation rate is low, since the processing of ambiguous bits in this procedure increases the pairing time. In [19], the perturbation vector based fuzzy cryptography (PVFC) is proposed to reduce the computing overhead. However, the method in [19] has low usability due to the long time delay incurred by the feature collection procedure.

3 Preliminaries

In this section, we introduce the important preliminaries of the proposed scheme.

3.1 Euclidean Distance and Hamming Distance

We first introduce two different types of distance used in this paper, Euclidean Distance and Hamming Distance.

Euclidean Distance refers to the straight-line distance between two points in Euclidean space. In this paper, we denote the Euclidean distance between two points x and y as $\Delta(x, y)$, which can be calculated by formula (1), where $|\cdot|$ represents the absolute value.

$$\Delta(x, y) = |x - y| \tag{1}$$

Hamming distance is a concept that represents the number of different bits corresponding to two bit strings of the same length. In this paper, we use $dis(str1, str2)$ to represent the Hamming distance between two bit strings $str1$, $str2$.

3.2 Fuzzy Commitment

Fuzzy commitment [26, 27], the combination of error correcting codes and cryptography, is a useful primitive for biometric authentication. In this paper, we utilize the Bose-Chaudhuri-Hocquenghem (BCH) code [28].

We denote the function of fuzzy commitment as $fc(\cdot)$, as shown in formula (2), where k is a randomly selected key, $h(\cdot)$ represents the anti-collision hash function.

$$fc(w, k) = (h(k), X) \tag{2}$$

X can be obtained by formula (3),

$$X = (c \oplus w) \tag{3}$$

where $\oplus$ represents XOR operation, and c is obtained by BCH encoding of k.

To decommit, $c^{'}$ is obtained from $w^{'}$ and X first (as shown in formula (4)), then $k^{'}$ can be recovered from $c^{'}$ by BCH decoding. Finally, the values of $h(k)$ and $h\left(k'\right)$ are compared to confirm whether k is successfully restored.

$$c' = (X \oplus w') \tag{4}$$

If $dis\left(w, w'\right)$ is lower than the error tolerance of the BCH code, then k can be recovered by w'.

4 Handshake Based Secure Pairing

In this section, we first give the overall system processes of the protocol, and then introduce the details of our handshake based secure pairing scheme. The devices on two sides are denoted as A and B.

The protocol includes four phases: data preprocessing, feature selection and reconciliation, witness generation and key binding, and key verification. In the data preprocessing phase, the acceleration data collected by the device is processed to reduce its dimension. In the feature selection and reconciliation phase, the optimal feature selection algorithm is employed to select the reliable features which are used to generate witness. In the stage of witness generation and key binding, the witness is generated by the extraction algorithm, then is bound with k to obtain auxiliary data. In the key verification phase, the device recovers the original key through the auxiliary data, then determines whether the pairing is successful by comparing the hash value of the key.

4.1 Data Preprocessing

In order to detect the start of the handshake accurately, users are required to press a button on the wrist-worn devices to indicate that it is about to start handshaking and devices pairing. Besides, 1–2 s of quiescence is needed before the handshake. It is worthy to note that these additional interaction can greatly simplify the detection of the start of the handshake and avoid other unnecessary monitoring process. Moreover, the required interactions are very simple and in line with the user's usage habits. After the data is

collected, dimensionality reduction and synchronization are performed to improve the probability of successful pairing.

In our scheme, the three-dimension accelerometer data generated by the handshake is used. Since the positions of the two devices worn on users' wrists vary from person to person, the collected three-dimension acceleration time series lack spatial alignment and cannot be compared directly. To this end, the root mean square of the X-axis, Y-axis, and Z-axis of accelerometer data (see Fig. 2(a)) is calculate to reduce the three dimension data to one dimension as shown in Fig. 2(b). We refer to the data obtained after processing as acceleration magnitude.

Furthermore, the accelerometer data, which are collected by devices independently, needs to be synchronized. As shown in Fig. 2(a), the handshake acceleration exhibits periodicity. Fig. 2(b) shows the acceleration magnitude, from which we can obviously find that the magnitude of acceleration fluctuates significantly. As it is verified that the time when the two sides generated data with almost zero acceleration magnitude is very close in [19], we take the intersection of the axis $x = 0$ and the acceleration magnitude curve of the first complete period (i.e., the acceleration magnitude in red arrow in the Fig. 2(b)) as the starting point. In this paper, the acceleration magnitude time series after synchronization is used as the feature sequence f.

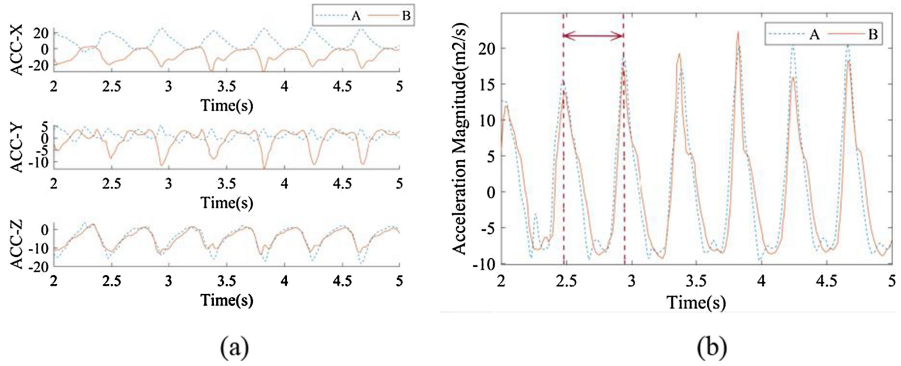

Fig. 2. Handshake acceleration.

4.2 Feature Selection and Reconciliation

The feature selection and reconciliation stage is shown in Fig. 3. After obtaining the feature f_A, Device A first selects a random threshold generation factor R_A and a random key k_A. Then, R_A and its identity ID_A are sent to device B. Then both sides start to agree on the optimal features, which is detailed in Sect. 5. Device A uses R_A and R_B respectively to calculate its own optimal feature indexes $I^A_{R_A}$ and $I^A_{R_B}$, and sends them to device B. After receiving the $I^B_{R_A}$, $I^B_{R_B}$ sent by device B, device A compares the preselected indexes I_R generated by both devices to obtain $I_R{}^*$, that is $I^*_{R_A} \leftarrow I^A_{R_A} \cap I^B_{R_A}$, $I^*_{R_B} \leftarrow I^A_{R_B} \cap I^B_{R_B}$. $x \cap y$ represents the intersection of x and y.

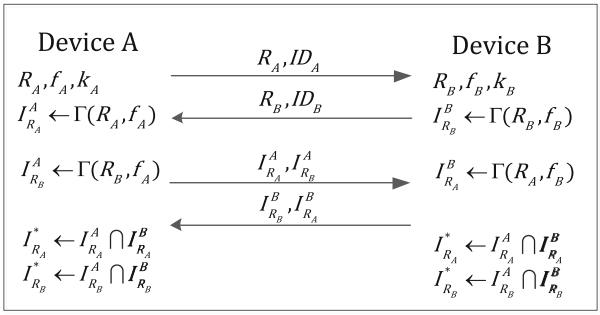

Fig. 3. Feature selection and reconciliation.

4.3 Witness Generation and Key Binding

The witness generation and key binding stage is given as follows, as shown in Fig. 4.

1. Device A calculates the witness w_{R_A}, w'_{R_B} respectively based on the preselected feature indexes $I^*_{R_A}$, $I^*_{R_B}$ obtained in the reconciliation phase, where $w_{R_A} \leftarrow EXT(I^*_{R_A}, R_A, f_A)$, $w'_{R_B} \leftarrow EXT(I^*_{R_B}, R_B, f_A)$. The witness generation algorithm EXT is shown in Fig. 5. At the beginning, w is set as an empty bit string $\varnothing$., and the fixed random number generation algorithm *rand* with the seed R is executed to generate random threshold sequence value $th = \{th_1, th_2, \ldots, th_n\}$. Next, the witness is gained by comparing the random threshold value with the feature sequence. If the feature value α_{index_i} is larger than the random threshold value th_{index_i}, a bit 1 is generated; otherwise, a bit 0 is generated, where $index_i$ is an element in $I^*_{R_A}$ or $I^*_{R_B}$ and the α_{index_i} is the value in f_A, | means bit concatenation. Finally, the witness w is compressed into a string of length *len*.

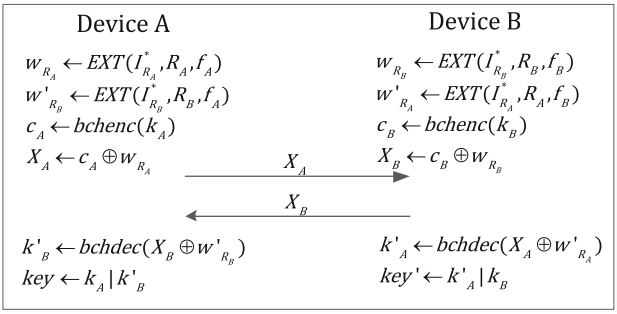

Fig. 4. Witness generation and key binding.

Algorithm EXT Witness Generation

Input:

$I^* = \{index_1, index_2, \ldots, index_m\}$

R

$f = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$

Output:

w

1: $w \leftarrow \emptyset$
2: $th \leftarrow rand(R)$
3: for $i = 0$ to m do
4: if $th_{index_i} - \alpha_{index_i} \geq 0$ then
5: $w \leftarrow w|1$
6: else
7: $w \leftarrow w|0$
8: end if
9: end for
10: $w \leftarrow w[1: len] \oplus w[len + 1: length(w)]1^{2*len - length(w)}]$

Fig. 5. Witness generation algorithm.

2. Device A calculates c_A by encoding the random key k_A with the BCH code, that is, $c_A \leftarrow bchenc(k_A)$, where $bchenc(\cdot)$ represents the BCH encoding function. The length of the information bits of the BCH code is equal to the length of k_A.
3. Device A generates auxiliary data X_A by binding the w_A with c_A, i.e., $X_A \leftarrow c_A \oplus w_A$, then sends the X_A to device B.
4. After receiving the X_B sent by device B, device A use w'_{R_B} generated in (1) to recover the random key k_B selected by the device B, that is $k'_B \leftarrow bchdec(X_B \oplus w'_{R_B})$, where $bchdec(\cdot)$ is the BCH decoding function.
5. The final key key is obtained by concatenating Device A's own random key k_A with the recovered key k'_B.

4.4 Key Verification

In the final phase, it is verified that the two legitimate devices have successfully extracted the same secret key key. The two devices first exchange the message authentication code (MAC), i.e., $M_A \leftarrow MAC(key|N_A|ID_A)$, $M_B \leftarrow MAC(key'|N_B|ID_B)$, where N is a random number to ensure real-time performance, and the ID indicates the identity of the device. After receiving M_B, N_B from the device B, device A uses its own key to calculate the $M'_B \leftarrow MAC(key|N_B|ID_B)$, then verifies whether $M'_B = M_B$. If the equation holds, the pairing is successful.

5 Optimal Feature Selection

After data processing (as described in Sect. 4.1), the noise in the acceleration data will affect the success rate of device pairing. Therefore, we propose the optimal feature selection algorithm, which can select reliable features from the feature sequence f for device pairing. In our algorithm, first the generated raw features are sorted, then the optimal features are selected from the sorted features, and finally the index I_R of the optimal features are sent as the auxiliary data to the pairing device.

5.1 Sort Based on Euclidean Distance

We find that the larger $\Delta(\alpha_i, th_i)$ is, the more robust the bit generated from the feature through experiments. Hence, sorting the Euclidean distance $\Delta(\alpha_i, th_i)$ of each feature in ascending order and then selecting those later features (features with larger Euclidean distance) is a feasible method to obtain reliable features. However, just sorting by $\Delta(\alpha_i, th_i)$ to select optimal features is not enough, since the sorting reliability can be still affected by the slight delay of one of the devices. In order to reduce the bias caused by delay, we propose an algorithm called minimum distance within the window. Specifically, we call $window_i = [\alpha_{i-\delta}, \ldots, \alpha_i, \ldots, \alpha_{i+\delta}]$ as a window of feature α_i with the size δ, and $min(\Delta(window_i, th_i))$ is the smallest α_i of $\Delta(\alpha_i, th_i)$ corresponding to $window_i$. We do not directly take $\Delta(\alpha_i, th_i)$ as the sorting criteria, but introduce the values δ before and after α_i to reduce the misleading caused by the slight lag, and use $min(\Delta(window_i, th_i))$ as the standard to sort.

5.2 Secure Feature Selection

To filter out the noisy features that are most likely to influence the success rate of device pairing, we propose the optimal feature selection algorithm Γ (see Fig. 6), where n is defined as the total number of original features, and q is defined as the number of optimal features selected by Γ, the corresponding optimal feature sequence is defined as Q.

We take the index i of each element in f and the corresponding $min(\Delta(window_i, th_i))$ as a key-value tuple $(i, min(\Delta(window_i, th_i)))$, and put them into a list, that is $add(list, (i, min(\Delta(window_i, th_i))))$, then the $list$ is sorted by $sort(\cdot)$ algorithm. As $min(\Delta(window_i, th_i))$ becomes smaller, the probability of the offset occurring between the α_i of devices will increase. We filter the feature sequence and select the last q key-value tuples in the sorted $list$ as the feature sequence Q for generating w. However, the direct exchange of the indexes i of the last q key-value tuples will expose the information of f, so we protect f by deleting some tuples in Q randomly, and we call these deleted tuples as chaff features. The algorithm $genchaff(\cdot)$ selects chaff features randomly from a sequence. In order to ensure that the minimum number of optimal features indexes selected by the devices on both sides is not less than len, that is, $Count(U_A \cap U_B) > len$, where $Count(U_A)$ and $Count(U_B)$ represent the number of features finally selected by users A and B, which is calculated by $q - cn$. Then $Count(U_A \cap U_B) > len$ can be simplified as:

$$Count(U_A)/Count(U_B) > \frac{n + len}{2} \tag{5}$$

and the value range of *cn* can be obtained:

$$cn < q - \frac{\mathrm{n} + \mathrm{len}}{2} \tag{6}$$

Finally, we arrange the key values of the tuples in Q in a sequential manner as the preselected feature indexes I_R.

Algorithm Γ Optimal feature selection

Input:

R

$f = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$

Output:

I_R

$th \leftarrow rand(R)$

$list \leftarrow \emptyset$

for $i = 0$ to n do

    $add(list, (i, min(\Delta(window_i, th_i))))$

end

$sort(list)$

$Q \leftarrow list(n - q + 1 : n)$

$chaff \leftarrow genchaff(Q)$

$Q \leftarrow Q \backslash chaff$

$I_R = getkeys(Q)$

Fig. 6. Optimal feature selection algorithm.

6 Performance Evaluation

6.1 Experiment Setup

In our experiments, the iPhones (iPhone6-CPU: A8 1.4 GHz, OS: iOS 11.0 and iPhone8-CPU: A11 2.74 GHz, OS: iOS 11.0) is tied to the volunteers' wrist to simulate the wrist-worn devices. We recruited 16 volunteers, containing 8 males and 8 females. Their ages are ranging from 22 to 25. In the following experiments, the handshake data is generated by volunteers shaking hands in pairs (one volunteer will shake with the remaining 15 volunteers), each handshake including 16 ups and downs and is repeated 5 times to ensure the accuracy of the data, the accelerometer sampling frequency is set to be 200 Hz. It is worth noting that the pairing of devices through a short-time handshake is user-friendliness. In our scheme, a handshake time of 3 to 4 s is enough to complete the pairing, which is convenient to user.

6.2 Witness Similarity

The success rate of secure pairing is directly related to the similarity of the witness w and w', which are generated from the optimal features. In this section, we first evaluate the sorting based on the minimum distance within the window, and then experimentally analyze the effect of window size δ and q on the similarity of the witness. In our scheme, we sort the features in ascending order based on $min(\Delta(window_i, th_i))$.

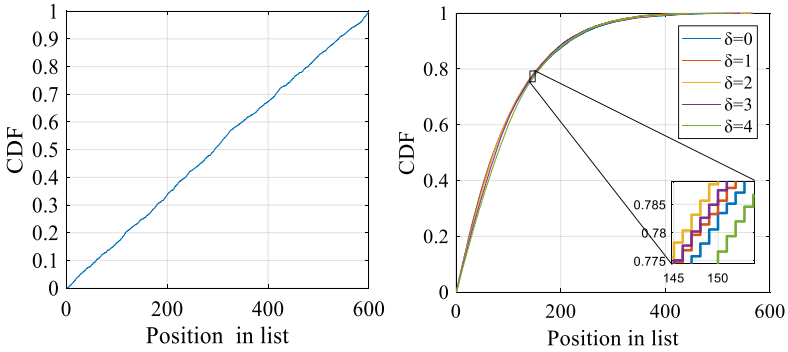

Fig. 7. CDF of indexes of the offset features.

Definition 1: If th_i and its corresponding feature values $\alpha_{A,i}$, $\alpha_{B,i}$ satisfy $(th_i - \alpha_{A,i}) * \left(th_i - \alpha_{B,i}\right) \leq 0$, we deem the index i of the th_i as the index of the offset feature.

The first 3 s of the synchronized acceleration magnitude is used as original feature data in our experiments, from which 600 acceleration magnitude features will be generated. The distribution of the indexes of the offset features is shown in Fig. 7. Fig. 7(a) shows the cumulative distribution function (CDF) of indexes of the offset features of the unsorted feature sequence, and Fig. 7(b) shows that of the sorted feature sequence. As can be seen, the indexes of the offset features are uniformly distributed before sorted, are mostly distributed at the front of the sequence after sorted. The smaller the index i is, the faster the CDF grows, and 80% of the indexes of the offset feature are ranging from 0 to 150. Besides, it can be seen from the partially enlarged diagram of Fig. 7(b) that in the range around 150, the CDF performance is better when $\delta = 2$.

In optimal feature selection algorithm, the window size δ and q are two important parameters which affect our feature selection directly, thus affecting the similarity of the witness w. We evaluate the similarity of the witness by changing the two important parameters, and the similarity of the two witness is measured by Hamming distance. Fig. 8 shows the variation of the average value of the $dis\left(w, w'\right)$ under different q and δ, from which we can see that as the number of selected feature q decreases, $dis(w, w')$ decreases exponentially. When q is less than 320, $dis(w, w')$ approaches 0, while the feature number q is selected to be less than 500, $dis(w, w')$ becomes smaller and the decreasing tendency slows down, which is less than 7. In addition, when q is taken around 500, the $dis(w, w')$ is the smallest when $\delta = 2$.

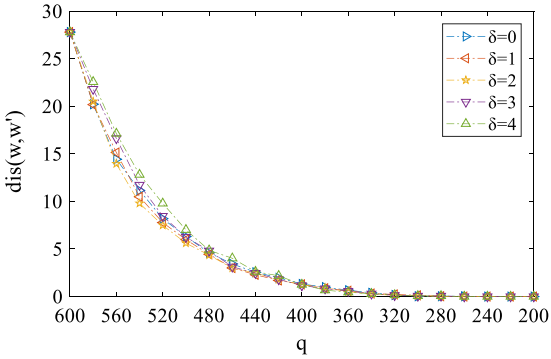

Fig. 8. Average $dis(w, w')$ with different q and δ.

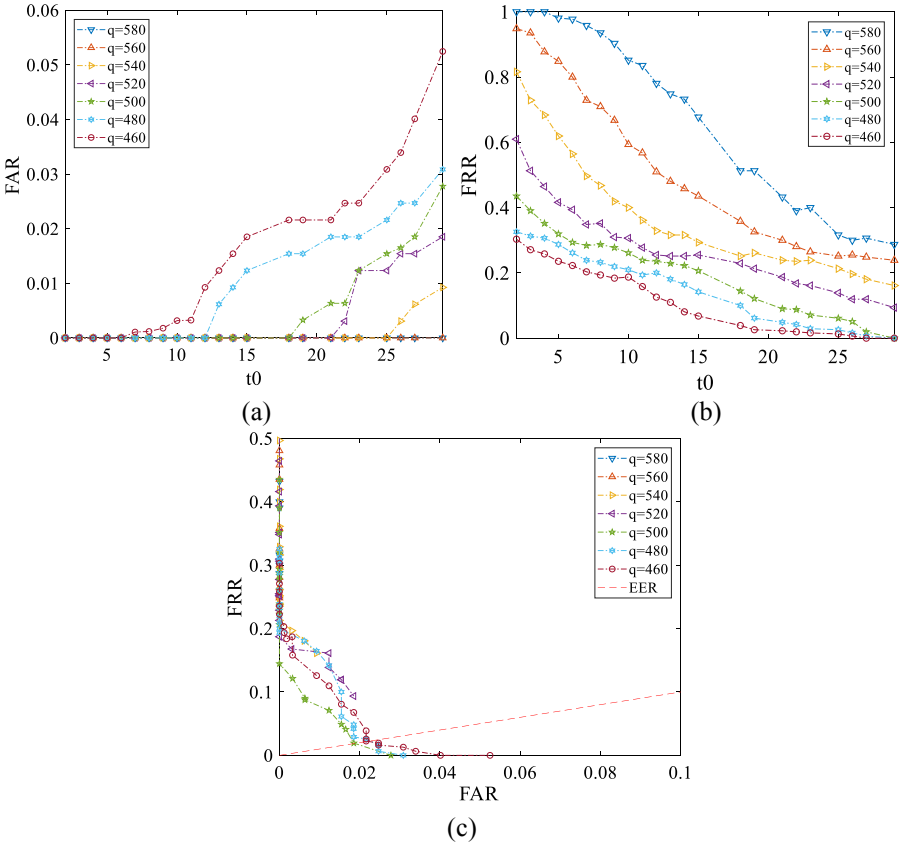

Fig. 9. FRR, FAR, EER of the scheme with different q and $t0$.

6.3 System Accuracy

System accuracy is the ability of the system to pair two devices correctly. It requires that only devices worn by users who shake hands with each other can be paired and generate the same symmetric key.

In our scheme, there are two main factor impacting system accuracy. One is the optimal feature selection algorithm, the second is the parameter of the BCH code. Since the feature selection algorithm directly affects the similarity of witness, and the similarity of witness is related to the accuracy of the BCH code, we evaluate the joint impact of q and different BCH codes of the system accuracy. We denote a BCH code as (ln, lk, $t0$) where ln represents the total length of the BCH code, lk represents the length of the information bits, and $t0$ represents the capability of noise tolerance.

The false rejection rate (FRR) and false acceptance rate (FAR) are used to evaluate the system accuracy. To ensure that the key can provide high security and filter out as much noise data as possible, we set the ln of a BCH code as 255.

In the experiment, the main concern is the variation of FRR and FAR and their correlation when different BCH codes are selected. We can see from Fig. 9(a) that the higher the value $t0$ is, the lower the FRR is. Conversely, from Fig. 9(b), we can see that the higher the value $t0$ is, the higher the FAR is. Noted that if the system accuracy is needed to be ensured, FAR and FRR must be reduced at the same time. As shown in Fig. 9(c), we can find an equilibrium point, i.e., equal error probability (EER) when $q = 500$.

6.4 Key Generation Rate

In this section, we evaluate the key generation rate and compare it with the state-of-the-art of acceleration-based pairing schemes, as shown in Table 1. In our scheme, the BCH code we choose is (255, 18, 131) and q is set as 100, as our scheme has better accuracy in this case. As can be seen in Table 1, our key generation rate is higher than other schemes, this shows that our solution quickly complete device pairing and session key negotiation while ensuring accuracy, which is more usable than other acceleration-based schemes.

Table 1. Key generation rate

Scheme	Generation rate/s
Walkie-talkie [12]	26/s
Inter-Pulse-Interval [9]	2 ~ 16/s
Bandana [7]	1 ~ 2/s
Shake to Communicate [19]	16 ~ 32/s
Shake-n-Shack [18]	25 ~ 48/s
SDP via Handshake [17]	80/s
Our proposed scheme	87/s

7 Security Analysis

In this paper, we consider two types of attackers, passive attackers and active attackers. Specifically, active attackers are divided into two classed: one is a MITM attacker who attempts to pair with a legitimate device by intercepting and tampering with the messages; another one is to obtain useful information by observing the handshake between two legitimate users. Passive attackers try to crack the pairing key by eavesdropping and monitoring the information transmitted in the open channel.

In order to evaluate the impact of the two attackers on the security of our system, we constructed experimental data sets for them respectively. The active attacker data set is generated by two illegal volunteers simulating the handshake of legitimate volunteers, and the passive attacker data set is the auxiliary data transmitted between legal devices during the pairing process.

7.1 Witness Security

To evaluate the security of w, we apply Shannon entropy to measure the randomness of w. Since w is a bit string of 0 and 1, the entropy of each bit is between 0 and 1. Fig. 10 shows the CDF of the average entropy of each bit in w. It can be seen that the average entropy of each bit is close to 1. For each set of experiments, the average Shannon entropy is greater than 0.92 and the largest one is up to 0.982. For the 255-bit witness, the entropy contained in w can be calculated by the following formula (7), which is 250.41. Therefore, we demonstrate that the generated w is secure.

$$\mathrm{E} = \sum_{i=0}^{m} e_i \tag{7}$$

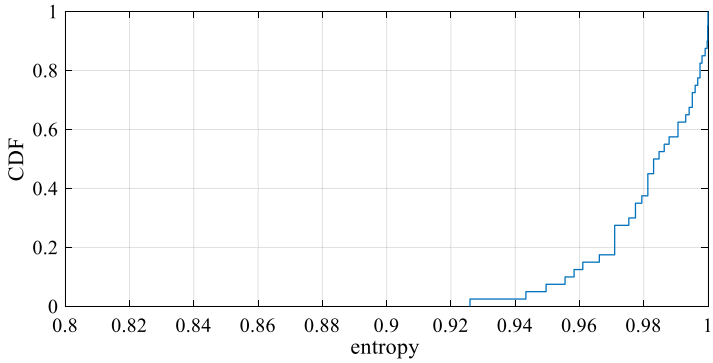

Fig. 10. CDF of the entropy of w.

7.2 Auxiliary Data Security

It is required that the transmitted auxiliary data I_R and X cannot reveal the information of the feature and the negotiated key, otherwise, it will be used by passive attackers.

In our scheme, we randomly delete a certain number of chaff features from the optimal q features, which makes it difficult for the attackers to distinguish the indexes of features that are close to the random threshold values. To guess the acceleration data from I_R, the attackers must identify the hidden chaff features from the missing indexes. The probability of finding all cn chaff features $P_{chafftuple}(cn)$ is obtained as follows:

$$P_{chafftuple}(cn) = 1/C_{n-q+cn}^{cn} \tag{8}$$

To find all the chaff features, the attacker needs at most $1/P_{chafftuple}(cn)$ attempts, so we define the security $S_{chaff}(cn)$ guaranteed by the chaff tuple as shown in formula (9).

$$S_{chaff}(cn) = -log_2 P_{chafftuple}(cn) \tag{9}$$

Take the BCH length len as 255 as an example. Fig. 11 shows the security that the chaff features can guarantee under different number of optimal features q. It can be seen that when $q = 520$, it reaches the peak. The security $S_{chaff}(cn)$ can reach up to 120 bits when q is in the range of 480 to 560. Therefore, our auxiliary data I_R is secure.

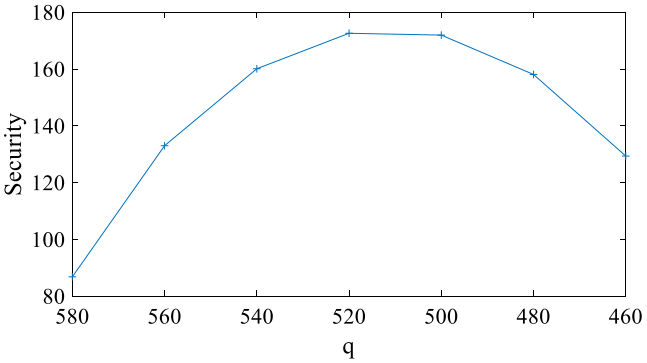

Fig. 11. Security guaranteed by chaff tuple.

The robustness of the auxiliary data $X = w \oplus bchenc(k)$ relies on the witness w and the random key k. The randomness of w is proved by experiments in Sect. 7.1, and the key k is randomly generated, so the $bchenc(k)$ obtained by BCH encoding is random. Therefore, the auxiliary data X is also secure.

7.3 Key Security

The session key key is concatenated by k_A,k_B, which are randomly selected by two devices, so its security depends on its length. As shown in Fig. 12, the length of the session key changes with the variation of noise tolerant ability $t0$ when the BCH code length len is 255. It can be seen that when the noise tolerance ability $t0$ is lower than 30, the length of the session key key is greater than 126. If the attacker wants to guess the negotiated key by brute force, it will take up to 2^{126} attempts.

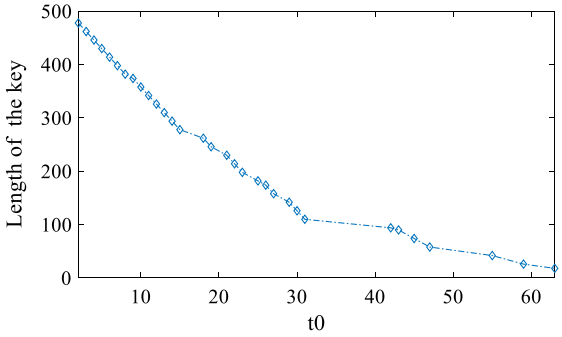

Fig. 12. Length of the session key *key* under different t0.

7.4 Mimicking Attack

The mimicking attacker observes the handshake between two legitimate users, and mimics it in real time in an attempt to obtain similar acceleration data collected by a legitimate device. We did imitation attack and obtained accelerometer value by imitating others handshake. As shown in Fig. 13, when mimics the handshake, there is a time lag from seeing the legitimate user's handshake to the react. Therefore, compared with a random handshake, it is more difficult for an attacker to generate a similar acceleration magnitude time series to pair with the legitimate device. Moreover, there is no success case that the mimicking attacker paired with a legitimated device in our experiments, which shows that our solution is suitable for practical applications.

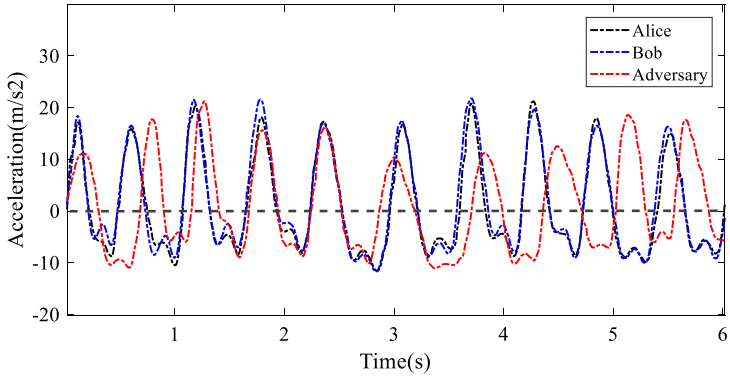

Fig.13. The acceleration magnitude gained by the mimicking attacker and the legitimate user.

8 Conclusion

In this paper, we proposed a robust and user-friendly secure pairing scheme for wrist-worn smart devices based on handshake acceleration, which is of high key generation

rate. In our scheme, we used the three-dimension inertial acceleration sensor on the off-the-shelf wrist-worn smart devices to record handshake patterns. We used the random threshold-based witness generation algorithm to improve the key agreement rate. To increase the success rate and ensure the security of the scheme, optimal feature selection algorithm, and BCH based fuzzy commitment are used. In the future work, we will improve our scheme by reducing system overhead and strengthening system security.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (61672413, U1708262, 61872449, 61772548, 61902290, 61772008), the project "The Verification Platform of Multi-tier Coverage Communication Network for oceans (LZC0020)", Scientific Research Program Funded by the Education Department of Shaanxi Province (Program No. 20JY016), the Fundamental Research Funds for the Central Universities, the Innovation Fund of Xidian University, Natural Science Foundation of Shaanxi Province (2019JM-109), Key Research and Development Program of Shaanxi (2019ZDLGY12–04, 2020ZDLGY09–06), China Postdoctoral Science Foundation (2018M640962).

References

1. Fomichev, M., Álvarez, F., Steinmetzer, D., Gardner-Stephen, P., Hollick, M.: Survey and systematization of secure device pairing. IEEE Commun. Surv. Tutorial **20**, 517–550 (2018)
2. Zhang, N., Wu, R., Yuan, S., Yuan, C., Chen, D.: RAV: relay aided vectorized secure transmission in physical layer security for internet of things under active attacks. IEEE Internet Things J. 8496–8506 (2019)
3. Mirzadeh, S., Cruickshank, H., Tafazolli, R.: Secure device pairing: A survey. IEEE Commun. Surv. Tutorials **16**, 17–40 (2014)
4. Zhang, N., Cheng, N., Lu, N., Zhang, X., Mark, J., Shen, X.: Partner selection and incentive mechanism for physical layer security. IEEE Trans. Wireless Commun. **8**, 4265–4276 (2015)
5. Chen, D., Zhang, N., Cheng, N.: Physical layer based message authentication with secure channel codes. IEEE Trans, Dependable Secure Comput (2018)
6. Jiang, Q., Zhang, N., Ni, J., Ma, J., Ma, X.: Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles. IEEE Trans. Veh, Technol (2020)
7. Schürmann, D., Brüsch, A., Sigg, S., Wolf, L.: BANDANA – body area network device-to-device authentication using natural gait (2017)
8. Groza, B., Mayrhofer, R.: SAPHE - simple accelerometer based wireless pairing with heuristic trees. In: Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia (2012)
9. Sun, Y., Wong, C., Yang, G.Z., Lo, B.: Secure key generation using gait features for body sensor networks (2017)
10. Mayrhofer, R., Gellersen, H.: Shake well before use: Intuitive and secure pairing of mobile devices. IEEE T. Mob. Comput. **8**, 792–806 (2009)
11. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H.-W.: Smart-its friends: a technique for users to easily establish connections between smart artefacts. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, pp. 116–122. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45427-6_10
12. Xu, W., Revadigar, G., Luo, C., Bergmann, N., Hu, W.: 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) 2016 15th

ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), p. 12 (2016)
13. Liu, H., Wang, Y., Yang, J., Chen, Y.: Fast and practical secret key extraction by exploiting channel response (2013)
14. Liu, H., Yang, J., Wang, Y., Chen, Y., Koksal, C.E.: Group secret key generation via received signal strength: Protocols, achievable rates, and implementation. IEEE T. Mob. Comput. **13**, 2820–2835 (2014)
15. Schürmann, D., Sigg, S.: Secure communication based on ambient audio. IEEE T. Mob. Comput. **12**, 358–370 (2013)
16. Yue, Q., Srinivasan, K., Arora, A.: Shape matters, not the size: a new approach to extract secrets from channel. In: ACM Workshop on Hot Topics in Wireless (2014)
17. Guo, Z., Gao, X., Ma, Q.: Secure device pairing via handshake detection. Tsinghua Science and Technology (2018)
18. Shen, Y., Yang, F., Du, B., Xu, W., Wen, H.: Shake-n-Shack: enabling secure data exchange between smart wearables via handshakes. In: IEEE International Conference on Pervasive Computing and Communications (2018)
19. Jiang, Q., Huang, X., Zhang, N., Zhang, K., Ma, X., Ma, J.: Shake to communicate: Secure handshake acceleration-based pairing mechanism for wrist worn devices. IEEE Internet Things **6**, 5618–5630 (2019)
20. Zhang, N., Lu, N., Cheng, N., Mark, J., Shen, X.: Cooperative spectrum access towards secure information transfer for CRNs. IEEE J. Sel. Areas Commun. **2013**, 2453–2464 (2013)
21. Zhang, N.,et al.: Physical layer authentication for internet of things via WFRFT-based Gaussian tag embedding. IEEE Internet Things J. (2020)
22. Christian, G., Kaisa, N.: Manual authentication for wireless devices. RSA Cryptobytes **7**(1), 29–37 (2004)
23. Chong, M.K., Mayrhofer, R., Gellersen, H.: A survey of user interaction for spontaneous device association. ACM Comput. Surv. **47**, 1–40 (2014)
24. Chong, M.K., Gellersen, H.: Classification of spontaneous device association from a usability perspective (2010)
25. Ming, K.C., Gellersen, H.: Usability classification for spontaneous device association. Pers. Ubiquitous Comput. **16**, 77–89 (2012)
26. Juels, A., Wattenberg, M.: A fuzzy commitment scheme (1999)
27. Jiang, Q., Chen, Z., Ma, J., Ma, X., Shen, J., Wu, D.: Optimized fuzzy commitment based key agreement protocol for wireless body area network. IEEE Trans. Emerg. Topics Comput. (2019)
28. Bose, R.C., Raychaudhuri, D.K.: On a class of error correcting binary group codes (1960)