Securing Cyber-Physical Social Interactions on Wrist-Worn Devices

YIRAN SHEN, School of Software & C-FAIR, Shandong University
BOWEN DU, Department of Computer Science, University of Warwick
WEITAO XU, Department of Computer Science, City University of Hong Kong
CHENGWEN LUO, College of Computer Science and Software Engineering, Shenzhen University
BO WEI, Department of Computer and Information Sciences, Northumbria University
LIZHEN CUI, School of Software & C-FAIR, Shandong University
HONGKAI WEN, Department of Computer Science, University of Warwick

Since ancient Greece, handshaking has been commonly practiced between two people as a friendly gesture to express trust and respect, or form a mutual agreement. In this article, we show that such *physical* contact can be used to bootstrap secure *cyber* contact between the smart devices worn by users. The key observation is that during handshaking, although belonged to two different users, the two hands involved in the shaking events are often rigidly connected, and therefore exhibit very similar motion patterns. We propose a novel key generation system, which harvests motion data during user handshaking from the wrist-worn smart devices such as smartwatches or fitness bands, and exploits the matching motion patterns to generate symmetric keys on both parties. The generated keys can be then used to establish a secure communication channel for exchanging data between devices. This provides a much more natural and user-friendly alternative for many applications, e.g., exchanging/sharing contact details, friending on social networks, or even making payments, since it doesn't involve extra bespoke hardware, nor require the users to perform pre-defined gestures. We implement the proposed key generation system on off-the-shelf smartwatches, and extensive evaluation shows that it can reliably generate 128-bit symmetric keys just after around 1s of handshaking (with success rate >99%), and is resilient to different types of attacks including impersonate mimicking attacks, impersonate passive attacks, or eavesdropping attacks. Specifically, for real-time impersonate mimicking attacks, in our experiments, the Equal Error Rate (EER) is only 1.6% on average. We also show that the proposed key generation system can be extremely lightweight and is able to run in-situ on the resource-constrained smartwatches without incurring excessive resource consumption.

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

This work is partially supported by National Natural Science Foundation of China under Grant 61702133, 91846205, the National Key R&D Program under Grant 2017YFB1400100 and China International Postdoctoral Exchange Program.
Authors' addresses: Y. Shen and L. Cui (corresponding author), School of Software and C-FAIR, Shandong University, 1500 Shunhua Road, Jinan, China, 250101; emails: yiran.shen36@gmail.com, clz@sdu.edu.cn; B. Du and H. Wen (corresponding author), Department of Computer Science, University of Warwick, 6 Lord Bhattacharyya Way, Conventry, UK, CV4 7EZ; emails: b.du@warwick.ac.uk, hongkai.wen@dcs.warwick.ac.uk; W. Xu, Department of Computer Science, City University of Hong Kong, 83 Tat Chee Ave, Kowloon Tong, Hong Kong; email: weitao.xu@cityu.edu.hk; C. Luo, College of Computer Science and Software Engineering, Shenzhen University, 3688 Nanhai Boulevard, Shenzhen, China, 518060; email: chengwen@szu.edu.cn; B. Wei, Department of Computer and Information Sciences, Northumbria University, Newcastle-upon-Tyne, UK, NE1 8ST; email: bo.wei@northumbria.ac.uk.

1550-4859/2020/04-ART19 $15.00
https://doi.org/10.1145/3378669

Additional Key Words and Phrases: Symmetric keys generation, device pairing, IMU sensors, principal component analysis

ACM Reference format:
Yiran Shen, Bowen Du, Weitao Xu, Chengwen Luo, Bo Wei, Lizhen Cui, and Hongkai Wen. 2020. Securing Cyber-Physical Social Interactions on Wrist-Worn Devices. *ACM Trans. Sen. Netw.* 16, 2, Article 19 (April 2020), 22 pages.
https://doi.org/10.1145/3378669

1 INTRODUCTION

Wrist-worn smart devices such as smartwatches and fitness bands are becoming ubiquitous: according to the latest global forecast [18], their market is set to triple its volume in the near future, reaching $32.9 billion by 2020. Instead of remaining as the companion devices of smartphones, now they are more independent, and capable of offering full-fledged functionalities. For instance, the recently announced Apple Watch Series 3 has the same level of connectivity including 4G LTE with smartphones, and can perform all kinds of tasks such as messaging, receiving/making calls, or streaming music without the presence of the paired phone [16]. Compared to smartphones, those wrist-worn wearables are often rigidly attached to the users' body and equipped with rich sensing modalities, which makes them the ideal pervasive platform to sense and interact with the internal and external user states. In the near future, they will become the key element in the cyber-physical ecosystem, enabling the next generation of applications in a broad spectrum of sectors such as intelligent mobility, smart spaces, and digital healthcare [13, 17, 23, 24, 32].

In this article, we leverage the unique advantages offered by wrist-worn smart devices, and explore the feasibility of using *handshakes*, a common form of physical contact between human beings, to enable secure data exchange between their devices (Figure 1). The intuition is that in many circumstances, we often shake hands and then exchange physical or digital tokens in order to connect with the others. For example, when meeting with someone for the first time, as a proper etiquette, we typically shake hands with each other, and then exchange business cards, save contact details on our phones, or even friend each other on social networks. Essentially, in those cases, handshakes can be viewed as the trigger for the subsequent data exchange activities, i.e., by shaking hands, both parties establish mutual consent to share information in the physical or cyber world. Therefore, we aim to integrate the process of exchanging private information with the actual handshakes so that when two users are shaking hands, their smartwatches or fitness bands can automatically communicate to each other and exchange data on-the-fly. We envision that in the future, users may have the option to configure their smartwatches or fitness bands to "socially discoverable" during handshakes, in the same way as the current wireless file sharing mechanisms such as the Apple AirDrop [29].

Although this vision is appealing, there are several challenges that need to be tackled. Firstly, the wireless communication channels established for data exchange have to be *secured*, since information such as contact details is sensitive, and wireless communications are often prone to impersonate attacks and eavesdropping attacks. Secondly, the data exchange process should require *minimal extra effort* from the users, i.e., they only need to shake hands normally without extra intervention such as pairing beforehand or entering the same PINs. Finally, the system should be efficient and lightweight enough to run *in-situ* on the resource-constrained smartwatches or fitness bands, but not to quickly drain the device battery.

Unfortunately, there is no existing solution that can address all three challenges at the same time. For instance, some existing products and patents [15, 22] can detect handshakes or high fives to exchange data such as social media information, but they either broadcast over open wireless channels, which can be easily intercepted [15], or rely on the cloud to verify keys [22], which

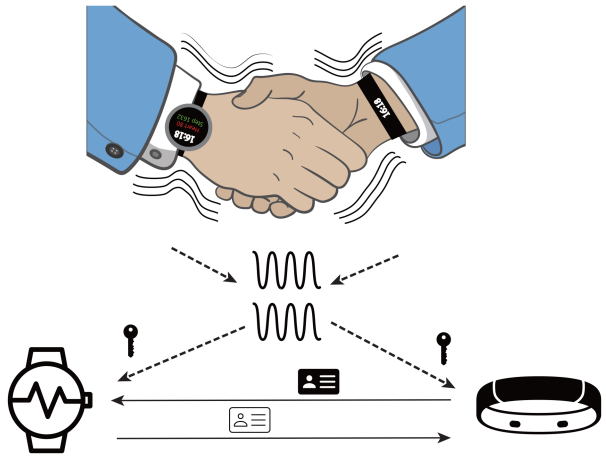


Fig. 1. The motion pattern induced by handshaking between two users can be captured by each of their wrist-worn smart wearables (e.g., a smartwatch and a fitness band as shown in this figure), and used to generate cryptographic keys for secure data exchange, e.g., sharing contact details or friending on social networks.

won't work without Internet connectivity and may incur undesirable delays. On the other hand, using key distribution protocols such as the Diffie-Hellman (D-H) key exchange [7] requires public key management infrastructure, which is computationally intensive and not feasible for real-time execution on wearables. The Near Field Communication (NFC) technology enables secure data communication between smart devices in the vicinity (normally within 20cm). However, for now, the security of NFC on smart wearables is largely guaranteed by the paired smartphones, such as passcodes or fingerprint authentication. As wearables can work independently of phone usage, the private data held by NFC can be vulnerable without the extra layer of protection.

To overcome the shortcomings of existing approaches, we propose the design and implementation of a novel key generation system that directly uses handshakes to encrypt data exchanged between wrist-worn smart devices. It exploits the fact that within an episode of handshaking, the two hands holding together should produce similar movements, which can be picked up by sensors (e.g., accelerometers) of smartwatches or fitness bands worn on the corresponding wrists of both users. With the captured motion signals, the proposed key generation system generates symmetric cryptographic keys on both sides, which are used to secure subsequent data exchange between the two devices. We proposed and published the original key generation system at the IEEE International Conference on Pervasive Computing and Communications (Percom) in 2018 [25]. However, in this journal version, we considered new practical threats when using our system, which includes passive impersonate attacks and mimicking eavesdropping attacks. Moreover, in the extended work, we provided significantly more comprehensive evaluations on the robustness of the system when the users are using different models of smart watches. We also evaluated the performance of the key generation system under the three types of attacks compared with the original publication. At last, we include new evaluation metrics, e.g., generated key entropy, to justify the quality of the keys.

Concretely, the technical contributions of this article are as follows:

—We propose a novel key generation system for secure data exchange between wrist-worn smart wearable devices via handshakes. To the best of our knowledge, this is the first work that explicitly uses physical contact (i.e., handshakes) to secure cyber contact (i.e., data exchange between smart devices) in a natural and user-friendly way.

—We implement the key generation system on off-the-shelf smartwatches, and propose a set of efficient algorithms that capture and process the motion signals of handshaking events, and generate symmetric keys in a distributed fashion to encrypt/decrypt data exchange. Our system is lightweight and can be always-on—it runs in real time on the resource-constrained smart wearables and incurs marginal overhead.
—We evaluate the key generation system extensively on datasets collected from real-world settings. The results show that it is able to generate keys of over 140 bits within 2s of handshaking, and the average key agreement rate between two legitimate devices are close to 100%. We also show that the proposed key generation system is resilient to runtime impersonate and eavesdropping attacks, where the Equal Error Rate (EER) for attackers mimicking the pattern of legitimate users is only 1.6%.

The rest of the article is organized as follows. Section 2 discusses the application scenarios of the proposed key generation system and describes the threat model. Section 3 presents the overview of the system, and Section 4 explains the details of the proposed approach that enables secure data exchange via handshakes. The proposed system is evaluated in Section 5, and related work is covered in Section 6. We conclude the article in Section 7 and discuss potential future directions.

2 APPLICATION SCENARIOS AND THREAT MODEL

2.1 Application Scenarios

Motivation: In this article, we aim to design a key generation algorithm for smart watches to enable secured Cyber-social interactions, such as friending on social networks (Facebook, Twitter, Instagram, etc.), bill splitting, or money transfer via digital wallets (e.g., Alipay). These interactions are now pervasively incorporated everyday into our social life. Considering the hardware specification and common use case of smart watches, the algorithm should be lightweight, secure, and require no further infrastructure. As far as we know, there are no existing solutions satisfying all the design factors. Specifically, the key distribution protocols such as D-H are computationally intensive; they also require public key management infrastructure to authenticate the identityor will be vulnerable to impersonate attacks [31]. NFC is a potential key exchange solution for smart watches and its short-range communication guarantees the security and authenticity. However, successful NFC communication requires at least one of the two smart watches working in active mode. According to our careful investigation on the specifications of smart watches available on the market, only very few models support NFC active mode in hardware level such as the Sony Smartwatch. But even if Sony Smartwatch is used, specific knowledge is required to enable NFC active mode as it is not supported by Android OS. Therefore, NFC is not a convenient choice at the current stage.

Our Solution: According to the above design factors, we propose a key generation system providing a natural and reliable way to secure data exchange between wrist-worn smart devices when shaking hands. It uses the motion signals induced by handshaking to simultaneously generating symmetric keys across different devices, and uses the keys to establish secure communication channels. In fact, the generated keys can also be used to encrypt wireless communications between other mobile devices of the users, such as phones, laptops, or even smart vehicles, as long as they remain paired or authenticated with the users' smartwatches or fitness bands. Therefore, the capability offered by the proposed key generation system can be orthogonal to devices types or communication modalities; this is a fundamental building block for many application scenarios. As per our evaluations in Section 5, our proposed key generation system is lightweight, secure, and requires no extra infrastructure. It only depends on acceleration data, which can be easily collected from most of the current commercialized smart watches. In practice, it can be

implemented as an OS level service, which empowers different apps to exchange information securely with other users' devices, such as friending on social networks (e.g., Facebook, Wechat) or money transfer via digital wallets (e.g., Alipay). In the following, we discuss two example applications of the proposed key generation system.

Friending upon Greeting: In many social events, connecting on social networks has become the convention after greeting with each other. With the proposed system, this process can be seamlessly done as two users shake hands, where they don't have to keep business cards or add each other manually on WeChat/WhatsApp. Concretely, during handshaking, a token containing the temporary authentication to connect the account details or friend request/confirmation will be encrypted at one of the two devices with the key generated by the proposed system, and transmitted through wireless channels. Then, the other device uses the symmetric key to decrypt the data to obtain the token to perform further actions such as updating the friending status accordingly. At last, temporary authentication contained in the token is deactivated after use to prevent offline attacks, such as through video recordings or exhaustive search.

Money Transfer via Digital Wallet: The proposed system can also be used to enable secure bill splitting or money transfer between users' smart watches. For instance, imagine Alice and Bob have lunch at a restaurant where splitting bills are not allowed. Alice would like to transfer money to Bob through digital wallet apps on their smart watches. By simply shaking hands, her digital wallet uses the generated key to secure a wireless channel and transmits a temporary token instantly. On Bob's side, his wallet app may use the symmetric key obtained from his smartwatch to decrypt and obtain the token. Then, his wallet app can use this token to request money from Alice's account. During this whole process, both of them do not need to even take their phones out.

2.2 Thread Model

The proposed system is designed to address *impersonation attacks* [10] and *eavesdropping attacks* [27], which are common security issues when transmitting data over a wireless medium. In this article, we assume the adversary has full knowledge of the system details and is able to sniff all the wireless traffic. However, we assume the legitimate devices have not been compromised, and it is impossible for the adversary to obtain the on-board motion data.

Impersonate Attacks: In such attacks, an adversary impersonates legitimacy in a communication protocol to extract private information exchanged by the legitimate devices. When he observes two legitimate users trying to build secured communication through handshakes, he follows their handshakes and uses the messages created by his own handshake movement to fool one of the two legitimate users to build "secured" communication channels with him. In this article, we consider two types of impersonate attacks—**mimicking attacks** are to mimic the very handshake movements between legitimate users wearing smartwatches or fitness bands, by himself or with another adversary. The recorded motion data is then used to generate the same cryptographic key and attack the encrypted message. While in **passive attacks**, the adversary is unable to observe the handshakes between legitimate users clearly so he shakes hands freely.

Eavesdropping Attacks: In eavesdropping attacks, the adversary only eavesdrops the messages exchanged between the two legitimate users without trying to impersonate into the data sharing protocol. We consider the adversary as an active copycat, who observes and eavesdrops the data exchange activities among nearby legitimate users. He mimics the handshakes of the legitimate users and tries to generate the same encryption key according to the reconciliation messages (see Section 4.3 for details) exchanged between the legitimate users. Once the adversary generates the same key as the legitimate users, he can easily extract the sensitive information from the encrypted messages.

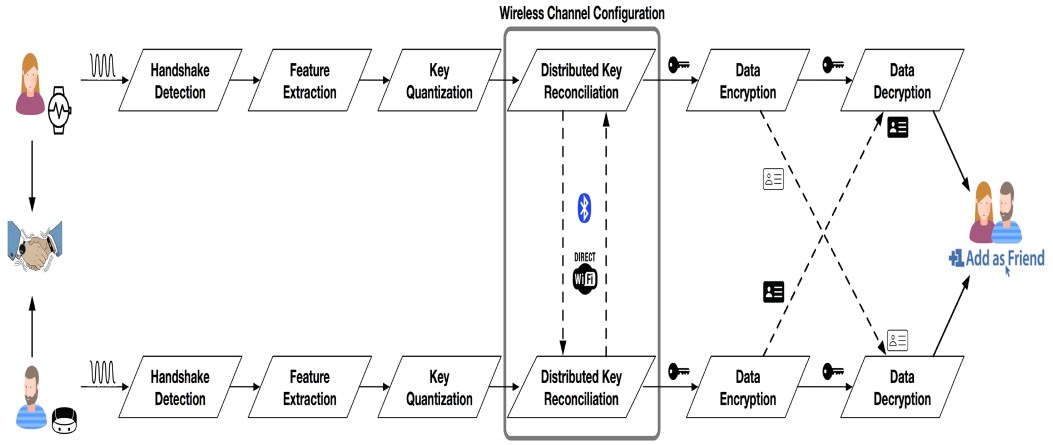


Fig. 2. The workflow of the proposed key generation system in the *friending upon greeting* scenario as discussed in Section 2.1.

It is also worth pointing out that, in some cases, the adversary may be able to record the handshakes using video cameras and infer the keys via sophisticated image processing and pattern recognition methods. The adversary may even try exhaustive search to decode the cached encrypted messages. However, the exchanged data is merely short-lived tokens, which would already expire before the adversary could decode the messages. Undertaking real-time video attacks is possible, but, in practice, the cost of launching such sophisticated attacks is much higher and the success rate is questionable.

3 SYSTEM OVERVIEW

In this section, we present an overview of the proposed key generation system. We consider the example of automatic friending on social networks as discussed in Section 2.1 and show how the proposed key generation works in practice. Figure 2 demonstrates the workflow of the system in this scenario.

Handshake Motion Detection: As two users start to shake hands, the key generation system captures the motion signals with the on-board Inertial Measurement Units (IMUs), which have been embedded in most of the current smart wearables. When significant motion is detected, a lightweight SVM-based classifier is applied on a short period of raw sensor data to determine if the handshake happens. If yes, a segment of acceleration data following the detected starting point is aligned and preprocessed, which is then used for the next step of key generation.

Key Generation and Reconciliation: With the preprocessed motion data, the key generation system firstly applies dimensionality reduction techniques to recover the dominant signal components. The extracted signals are then quantified and converted into bits by thresholding. Due to the noisy nature of motion signals, the system also incorporates a key reconciliation step with the other user's device, to discard the ambiguous bits and only keep the reliable ones to generate symmetric keys.

Wireless Channel Configuration: When a handshaking event is detected, the key generation system enables wireless radios on the smart wearables. As discussed in the previous section, the system is communication modality neutral—it relies on the underlying network service to discover nearby devices, and tries to establish a wireless communication channel with the correct peer (i.e., the one party involved in this handshaking event) using the generated keys.

Secure Data Exchange: Once an appropriate wireless communication channel is established, the proposed system uses the symmetric keys on both devices to secure data exchange between them. Concretely, we consider the standard messages encryption and decryption techniques, where the sender and receiver encrypts/decrypts the messages with their local keys, respectively. In this way, although the wireless medium can be sniffed, the adversary won't be able to decode the intercepted data without the correct keys.

Now, we are in a position to explain the details of the proposed secure data exchange approach.

4 SECURE DATA EXCHANGE VIA HANDSHAKES

In this section, we discuss the key components of the proposed key generation system, which enable secure data exchange via handshakes. We start with how to detect and capture motion data during handshaking in Section 4.1, and then describe our algorithms to generate and reconcile keys in Sections 4.2 and 4.3, respectively. Then, in Section 4.4, we show how to establish the correct wireless channel; finally, Section 4.5 discusses the process of data encryption and decryption in the proposed key generation system.

4.1 Handshake Motion Detection

In this article, we assume that both users involved in a handshaking event wear smart devices such as smartwatches or fitness bands, on the wrists of their dominant hands, i.e., the hands that are used during the handshakes. In this case, the motion of the handshakes can be recorded by the Inertial Measurement Unit (IMU) sensors, which are embedded in most of the current off-the-shelf smart wearables. Typically, they contain a three-axes accelerometer, and, optionally, gyroscope and magnetometer. In our system, we only use accelerometers since they are the most pervasive and very efficient, while gyroscope consumes significantly more energy. Our experiments show that, with sensor readings just from the accelerometers, the proposed system is able to generate robust enough cryptographic keys.

Figure 3 shows an example of the accelerometer sensor readings over three axes during a handshaking event (top three plots), where we can see that the handshake induces periodic patterns in the signal. On the other hand, the bottom plot of Figure 3 shows the squared magnitude of acceleration, which is computed by combining the squared data values of signals from all three axes. Clearly, the acceleration magnitude is very significant with respect to the background noise, and we see several peaks corresponding to the up and down movement during handshaking. Therefore, our system detects the first signal peak in a handshake event and uses it as the anchor point to align the sensor readings for the next step of key generation.

However, it is possible that other common motions may also cause significant peaks in the magnitude of the accelerations. Therefore, an activity recognition component is needed in our system to avoid false alarms. Activities or gesture recognition has been well-studied in the literature [33, 34, 34] and some of them become built-in functionalities of smartwatch OS, such as walking/step detection, raising hand detection, and so on. To recognize handshakes, we build a SVM-based classifier running in-situ on smart watches, which is designed and implemented in our previous work [12]. We considered four different types of activities that may happen when the people are using our system—walking, hand-raising, hand-waves, and handshaking. According to our evaluation on a collected dataset, the overall recognition accuracy is over 97% and the runtime of the classifier on smart watches is less than 9 *ms*.

4.2 Key Generation

Signal Feature Extraction: As shown in Figure 3, the raw acceleration signal has three axes, representing motion with respect to the body reference frame of the device. In practice, the devices

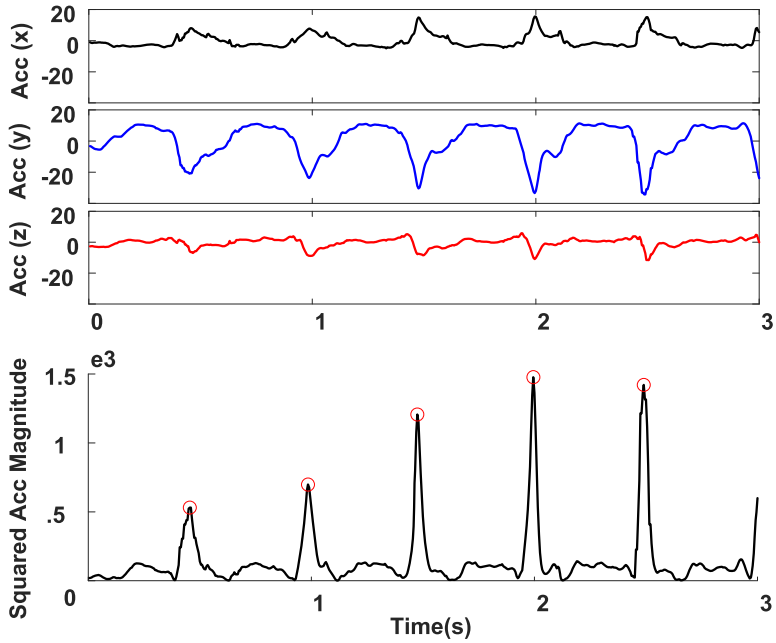


Fig. 3. The acceleration readings along three axis during 3s of a handshaking event (top three plots), and the squared acceleration magnitude (the bottom plot).

can be attached to the wrists of the users in arbitrary ways, to reliably generate symmetric keys across different devices, we need to transform the signal to reveal the dominant motion patters that are invariant to device attachment. On the other hand, although the acceleration magnitude is robust to different device posture, as shown later in Section 5.2 it contains much less information and thus impossible to generate high quality keys (i.e., in terms of bit rate). Therefore, in this article, we consider a Principal Component Analysis (PCA) based approach, which converts the three-axes acceleration into signals representing the dominant motion components.

Let us assume two users, *Alice* and *Bob*, have shaken hands with each other, and the motion data has been captured by both of their smart wearables. Suppose Alice obtains a data matrix $X \in \mathcal{R}^{M \times N}$ containing the accelerometer readings, where M is the number of axes and N is the number of data points from each axis. PCA finds a matrix $U \in \mathcal{R}^{m \times M}$ that projects the original data into a smaller subspace while retaining most of the information. Before computing the matrix U, the data matrix should be firstly centered, i.e., the mean value of each column in the data matrix X is subtracted:

$$X_C = X - \frac{1}{m} \sum_{i=1}^{M} X_i, \tag{1}$$

where X_i is the i_{th} row of X. After the data matrix is centered, the Eigenvalues and Eigenvectors matrices can be obtained by conducting EigenValue Decomposition (EVD) as,

$$\{U, E_{ordered}, V^T\} = EVD\left(X_C X_C^T\right) \tag{2}$$

$E_{ordered}$ contains the Eigenvalues on its diagonal, which are sorted in descending order according to their absolute values. U contains the corresponding Eigenvectors by columns. Then, we project

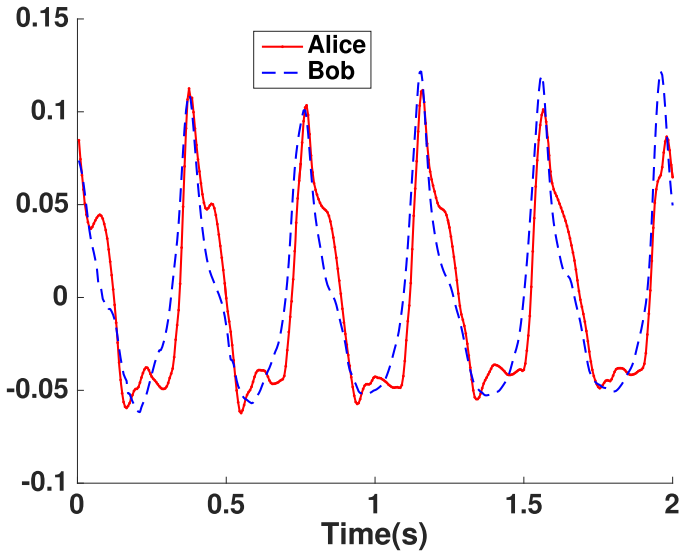


Fig. 4. The proposed PCA-based approach extracts dominant motion features from raw acceleration data, which are very consistent across the two users shaking hands with each other (Alice and Bob in this case).

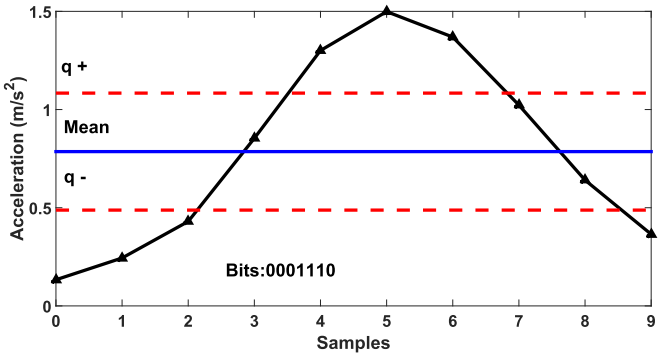


Fig. 5. Illustration of quantization process we use to generate bits stream; the figure was first presented in our previous work [31].

the original data matrix into the subspace defined by the U:

$$Y = U^T X, \tag{3}$$

where $Y \in \mathcal{R}^{m \times N}$. It is well known that the first principal component contains the largest part of information in the original signal. It corresponds to the largest Eigenvalue in $E_{ordered}$ and the first row of the matrix Y. Therefore, we use the first row of Y (denoted as Y^1) to represent the dominant motion signal features. Figure 4 shows an example of the computed features for both Alice and Bob. We see that although generated on different devices, they exhibit similar patterns and agree to each other very well. This confirms that it is clearly possible to use such features to generate symmetric keys across two different devices.

Signal Quantization: Now we need to convert the continuous motion signal features into discrete keys. Our system uses a similar signal quantization and bit extraction methods as in Refs [8] and [20], which quantify continuous signals into bit codes. As shown in Figure 5, the idea is to

firstly divide the time-varying continuous signals into small segments with the same length, e.g., 10 data points as in our experiments. Then, for each segment, we compute its upper (q_+) and lower (q_-) quantization thresholds as:

$$\begin{aligned} q_+ &= \mu + K\sigma \\ q_- &= \mu - K\sigma, \end{aligned} \tag{4}$$

where μ is the mean and σ is the standard deviation of the values of the data points within the segment. K is the quantization factor, which directly controls the values of the upper and lower thresholds. The thresholds, then, have significant impact on the performance of the key generation system via changing the bit generation rate and key agreement rate (we will discuss this with more details in Section 5.2).

In practice, those thresholds are used to determine the bit values at each position of the generated key. Specifically, a data point over δ_+ will be encoded as 1, while the one below δ_- will be 0. Any data points between the thresholds will be discarded to improve stability. After this quantization step, now Alice and Bob have their own keys locally at their devices. Without loss of generality, in the following, we denote the keys as $\mathcal{K}_{Alice}$ and $\mathcal{K}_{Bob}$.

4.3 Key Reconciliation

Theoretically, the generated keys can be used to encrypt and decrypt the messages if and only if they agree with each other, i.e., $\mathcal{K}_{Alice} = \mathcal{K}_{Bob}$. However, in practice, the keys generated from the above process may not be able to precisely agree with each other due to signal noise. In most cases, we would obtain two keys $\mathcal{K}_{Alice}$ and $\mathcal{K}_{Bob}$, where $\mathcal{K}_{Alice} \approx \mathcal{K}_{Bob}$. The disagreement often happens if the value of a data point is close to the thresholds, i.e., in the presence of small turbulence caused by motion noise, a bit at Alice's side might be discarded since the value is below the upper threshold δ_+, while remaining "1" at Bob's side.

To address this problem, we consider a key reconciliation approach [6, 30] to discard those ambiguous bits. The idea is to exchange the index of the valid bit positions to the other devices, and reach a mutual agreement on which bits should be used in the final keys. For instance, assume the key generated by Alice's devices is [1x0xx11x00], while for Bob it is [1x00x11xx0]; here, x means the position where no valid bit is present. Then, both Alice and Bob inform each other the positions of the valid bits, i.e., Alice sends $P_{Alice} = \{1, 3, 6, 7, 9, 10\}$, and Bob sends $P_{Bob} = \{1, 3, 4, 6, 7, 10\}$. Upon receiving the positions, they compare the received vector with the local one, and agree that only the bits that are valid according to both vectors should be used. In this example, the agreed positions should be $\{1, 3, 6, 7, 10\}$, so that the final symmetric keys are $\mathcal{K}_{Alice} = \mathcal{K}_{Bob} = [10110]$.

Note that during reconciliation, only the indices of the generated bits (either 1 or 0) are exchanged, but the true values of the keys are not disclosed to anyone (legal or illegal) overhearing the reconciliation. At this stage, it is possible that the two users haven't established a communication channel with each other. They just broadcast and receive the position vectors of valid bits to/from the nearby devices. For instance, there may be two pairs of users using the proposed key generation system in a close proximity, and, in the following, we explain how the system uses the received information to establish a secure wireless channel with the correct peer.

4.4 Wireless Channel Configuration

As discussed above, in the presence of multiple pairs of the system users, we need to correctly set up communication channels between the right peers, i.e., the two who are shaking each other's hands. One naive approach is to look at the received signal strengths (RSS), and choose the device with the strongest RSS to connect. However this is not robust enough in practice since RSS measurements are inherently noisy. The proposed key generation system considers a probe-based

approach. For instance, Alice might receive more than one position vectors P from different devices, including Bob's. Then it performs the above reconciliation step for each vector, and generates a list of candidate keys. Those candidate keys are then used to encrypt a pre-defined probe message whose content is known by everyone, and the encrypted messages (one for each candidate key) are replied back to the corresponding senders.

On the other hand, When Bob receives those probe messages, it tries to use the list of candidate keys to decrypt the messages accordingly. It should be that just one message can be successfully decrypted, which is the one sent by Alice. Now, Bob only needs to keep the candidate key from Alice, and reply an acknowledgment message to Alice, indicating that he is ready for further data exchange. In this way, the proposed key generation system establishes a wireless communication channel between Alice and Bob, who are shaking hands with each other, and identifies the correct keys for future data encryption and decryption.

Note that besides the two wearables that have generated the keys (i.e., the smartwatches or fitness bands worn by the users), such a communication channel could also be established between other devices (e.g., phones or laptops) belonging to the users, as long as they are paired or connected with those wearables in the same private (often ad hoc) network. This is straightforward since the wearables can simply share the generated keys with the other devices within the network, allowing those devices to directly perform the above probe process, and initialize data exchange with each other.

4.5 Secure Data Exchange

Given the symmetric keys and configured wireless communication channel, the proposed key generation systems on Alice and Bob apply the standard encryption and decryption methods using the keys generated distributedly to guarantee the confidentiality of the data exchange. The keys locally generated at both sides are very likely to be identical from the same handshaking motion so that the symmetric key encryption system can be achieved. In our implementation, we require the generated keys to contain at least 140 valid bits to ensure the level of security, and if unsuccessful, the system will ask the users to do another handshake. Then, we consider a fixed length of the first 128 bits to encrypt and decrypt messages at both sides.

Depending on different application scenarios, in some cases, such as automatic friending on social networks, the data exchanged is merely digital tokens, i.e., friend requests/confirmations. Therefore, after successful completion of the data exchange, the proposed system will also inform the social networks about this transaction via their Application Programming Interfaces (APIs), to update the friending status accordingly.

5 EVALUATION

5.1 Experiment Setup

5.1.1 System Implementation. We implement the key generation system on off-the-shelf smartwatches, which can run in real-time. In our experiments, we use Samsung Gear Live, which has a Quad-core 1.4GHz CPU, 512M RAM, and 300mAh battery, and run Android Wear OS. We set the accelerometer sampling rate to 200Hz, and uses the Bluetooth 4.0+ *InsecureRfcomm* to establish wireless communication channels.

5.1.2 Data Collection. We recruited 20 volunteers to participate in our experiments, containing 10 males and 10 females with ages ranging from 22 to 54. During each experiment session, we asked all participants to wear smartwatches on their right wrists, and randomly divided the 20 participants into five groups, each containing four people. Within each group, two of the participants were selected to be the legitimate users while the other two were the adversaries. We

used Samsung Gear Live for the above data collection. To show the robustness of our system on different devices, we also collected one dataset from the users wearing different devices. Ten subjects participated in collecting this dataset. During each data collection session, one subject was wearing a Sony Smartwatch while the other subject was wearing Samsung Gear Live.

Impersonate Mimicking Attack: When collecting the dataset associated with an impersonate attack, the legitimate users were asked to shake hands to exchange data, while at the same time the adversaries observed this and also shake hands with each other, trying to mimic the pattern of the handshake as much as they can. Then the adversary device will try to build a "secured" communication channel with one of the legitimate users through reconciliation.

Impersonate Passive Attack: The protocol is similar to an impersonate attack; the only difference is the adversaries will not mimic the handshakes.

Eavesdrop Attack: The data collection procedure is the same as the impersonate mimicking attack. But, the adversary device eavesdrops the reconciliation messages exchanged between the legitimate devices and determines the valid bits positions according to the eavesdropped messages. At the same time, the adversary will try to mimic the pattern of the legitimate users to generate the same encryption key to disclose the information from the following messages shared between the two legitimate devices.

We conduced over 1,000 sessions of experiments for each type of attack and have collected six datasets: three containing motion data from the legitimate users, while the other three consist of the data from the corresponding attacks. In the following, unless otherwise stated, we refer to those two datasets as the *legitimate dataset*, *impersonate mimicking dataset*, *impersonate passive dataset*, and *eavesdropping dataset*.

5.1.3 Evaluation Metrics. In this article, we evaluate the performance of the proposed key generation system with respect to the following metrics. *Generated Bit Rate* is the number of bits generated from the sensor readings per second. *Bit Agreement Rate* denotes the percentage of the matching bits of the two cryptographic keys generated by two devices during a handshaking event. *Generated Keys Entropy* is the entropy of the generated keys that measure the uncertainty; higher entropy means higher randomness of the key and better privacy, which has been used to estimate the quality of the keys generated by physical motions [30]. *Signals Coherence* is the empirical Cumulative Distribution Function (CDF) of the coherence of different motion signal features. *Key Success rate* is the percentage that the two keys generated via handshakes are identical. *False Acceptance Rate (FAR)* is a measure of the probability that a mimicking adversary generates an identical key to that of a legitimate device. *False Rejection Rate (FRR)* is defined as the ratio of the failed matching attempts via handshakes. It is obvious there is a tradeoff between FAR and FRR. *Equal Error Rate (EER)* measures the tradeoff between FAR and FRR, and it is the value of FAR or FRR when the two false rates are equal. We also evaluate the system overhead of the system by profiling the *Computation Time* and *Energy Consumption* of its key components on off-the-shelf smartwatches.

5.2 Experiment Results

5.2.1 Bit Rate vs. Signal Feature Extraction. The first experiment studies the impact of signal feature extraction approaches in terms of the generated bit rate. Figure 6 shows the mean and standard deviation of the generated bit rates of the proposed PCA-based approach with respect to the baseline method of using acceleration magnitude. First, we see that the proposed approach produces a significantly higher bit rate than using acceleration magnitude. For example, when $K = 0.7$, the average generated bit rate from our approach is about 120 bit/sec, which almost triples that of the acceleration magnitude based. In addition, we also see that as the quantization factor goes up, the generated bit rate drops for both approaches. This is expected since high quantization

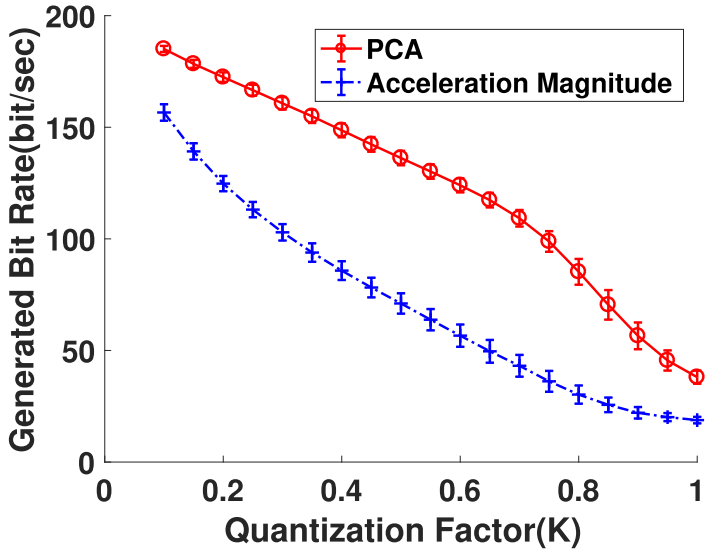
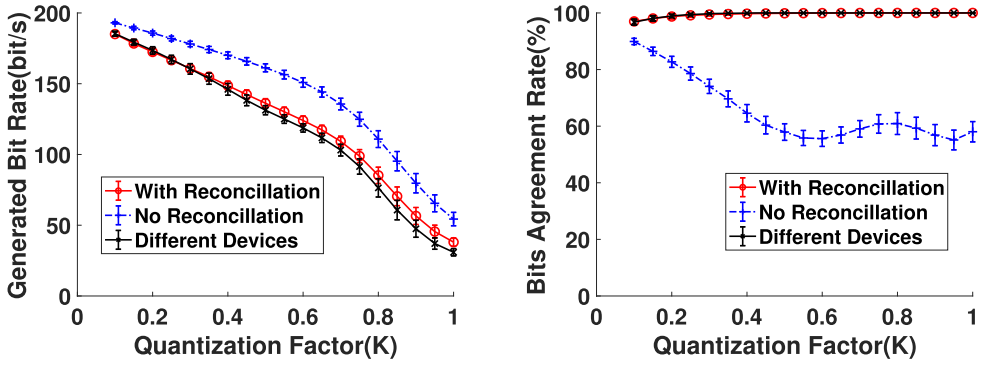


Fig. 6. The bit rate of keys under different quantization factor K, generated by the proposed PCA-based approach (the red line) and the baseline approach using acceleration magnitude (the blue line).



(a) Bit Rate (b) Bit Agreement Rate

Fig. 7. The impact of key reconciliation on (a) bit rate and (b) bit agreement rate of the generated keys.

factor means bigger thresholds, where the bits generation process is more prune to noise. However, the PCA-based approach degrades much more gracefully than the baseline, especially in the range of $[0.4, 0.8]$. This is because our approach can recover the most dominant motion patterns from the raw signal, and thus is inherently more robust to high quantization factors.

This experiment verifies the impact of the proposed key reconciliation method as discussed in Section 4.3 and robustness of the key generation algorithm when the two users using different devices. We consider both the generated bit rate and bit agreement rate as the metrics. As shown in Figure 7(a), we see that on one hand, reconciliation slightly reduces the generated bit rate, since it discards those ambiguous bits. However, this has a positive knock-on effect on the bit agreement rate. As shown in Figure 7(b), the bit agreement rate is significantly higher when key reconciliation is considered. In practice, for a reasonable K, we would prefer the high bit agreement rate introduced by reconciliation, since the generated keys are only valid if they are identical

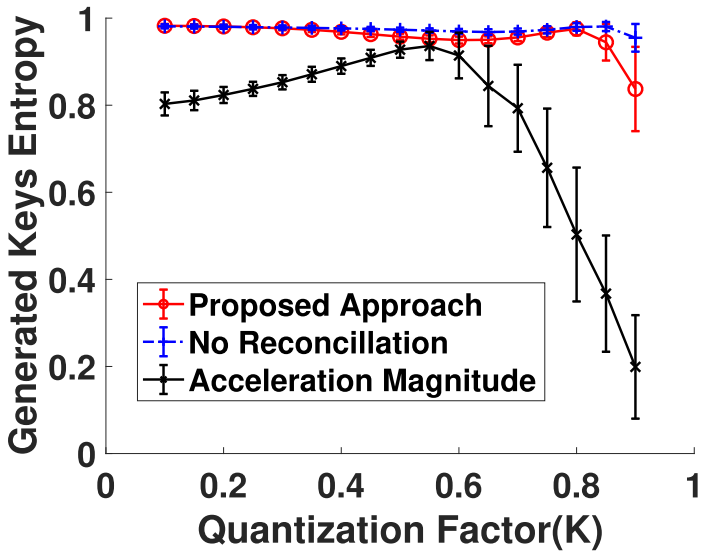


Fig. 8. Entropy of generated keys with different settings.

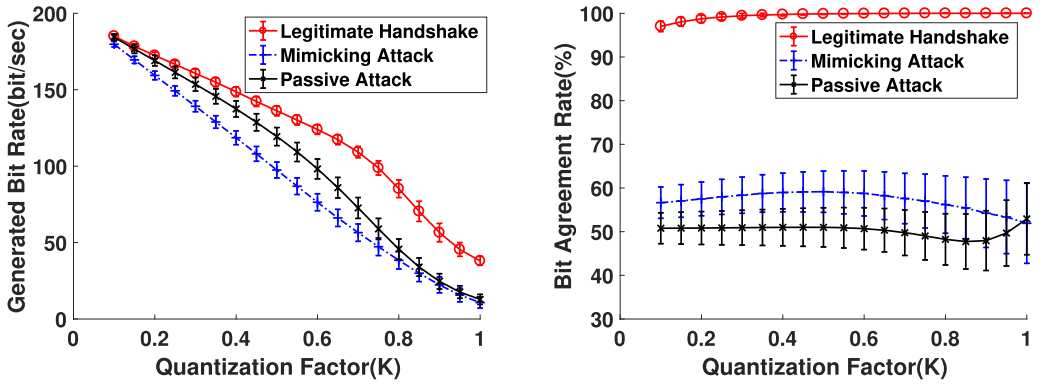
in every bit: if an agreement can't be reached, the users would have to shake hands again to regenerate the keys. In addition, the gap between with and without reconciliation in Figure 7(a) is only marginal especially when $K < 0.4$ and $K > 0.7$. At last, the results from "different devices" (two users are using different devices) are almost the same to the benchmark: generated bit rate only decreases a bit while the bits agreement rate achieves almost the same results. Therefore, we can claim that the users are not required to wear the same devices.

5.2.2 Entropy of Generated Keys. This experiment evaluates the quality of the generated keys in the sense of key randomness. Keys with higher randomness are more difficult to be hacked than those with lower randomness. To evaluate the randomness of the keys, we compute the entropy of the generated keys with different settings, i.e., the one utilizes all the proposed components in the system, the one applies all but reconciliation, and the one applies acceleration magnitude instead of PCA. The entropy is computed as the following equation:

$$S = -\sum_{i} p(\mathcal{K}_i) log_2 p(\mathcal{K}_i) \tag{5}$$

As shown in Figure 8, we can see that, compared with the approach without reconciliation, the proposed key generation system produces almost equivalent performance on entropy, but only slightly drops when $K > 0.9$. It is reasonable as reconciliation discards some bits which may reduce the randomness (or entropy). Meanwhile, we can observe that, compared with acceleration magnitude, PCA improves the entropy of the generated keys significantly. To prevent the system from using weak keys, we require the entropy of the generated key must be over a threshold (we use 0.9 in our system). The generated key will be checked if its entropy is over the threshold; otherwise, the key will be discarded.

5.2.3 Security Analysis on Impersonate Attacks. We first evaluate the performance of the system under impersonate attacks including (impersonate) *mimicking attacks* and *passive attacks*. We use both the legitimate and the two adversary datasets (impersonate mimicking dataset and



(a) Bit Rate (b) Bit Agreement Rate

Fig. 9. The impact of different quantization factor (K) on the system performance in terms of (a) bit rate and (b) bit agreement rate.

impersonate passive dataset) and vary the quantization factor K. The following three metrics, generated bit rate, bit agreement rate, and key success rate are considered in the experiments.

Generated Bits Rate vs. K: Figure 9(a) shows how the generated bit rate varies over K on all datasets. We see that, generally, a smaller K produces a higher bit rate, and handshakes made by legitimate users (i.e., from the legitimate dataset) achieve a higher bit rate than that of the adversaries. This is because legitimate handshakes tend to produce very similar motion signals across two devices; thus, only fewer bits in the generated keys will be thrown away during the key reconciliation step.

Bit Agreement Rate vs. K: On the other hand, for a bit agreement rate, as shown in Figure 9(b), legitimate handshakes have a slightly higher bit agreement rate as K goes up. However, for the handshakes produced by adversaries, the bit agreement rate first goes slightly up and then drops, and the variances becomes much bigger. Note that the average bit agreement rate from legitimate handshakes achieve almost 100% when K is larger than 0.7. More importantly, we see that the gap between legitimate and adversarial handshakes is very significant for all K. This implies that legitimate handshakes are very hard to be mimicked in real time, since it is very difficult for the adversaries to reliably generate high-quality keys. Besides, comparing different types of attacks the adversaries mimicking the handshakes tend to produce a slightly higher agreement rate than those conducting the attacks freely; however, the rate is still significantly lower than the legitimate handshakes.

Success Rate vs. K: Finally, we evaluate how a key success rate on a legitimate dataset, i.e., the generated two keys are identical, varies with respect to the quantization factor K. As shown in Figure 10, the key success rate increases as K, which reaches 100% when K is larger than 0.7. Therefore, in our case, a larger quantization factor K results in a higher bit agreement rate and key success rate, but has a negative impact on the bit rate. In addition, we observe that the bit rate and bit agreement rate of legitimate handshakes are consistently higher than that of the adversarial, which means we could reject the adversarial attempts with appropriate thresholds. Particularly, in the current key generation system, we use a bit rate threshold to decide if the generated key should be accepted as valid.

Resilience to Impersonate Attacks: In this experiment, we evaluate the performance of the proposed key generation system under mimicking and passive attacks. We assume that as two legitimate users are shaking hands, there are different types of adversaries who can sniff the wireless

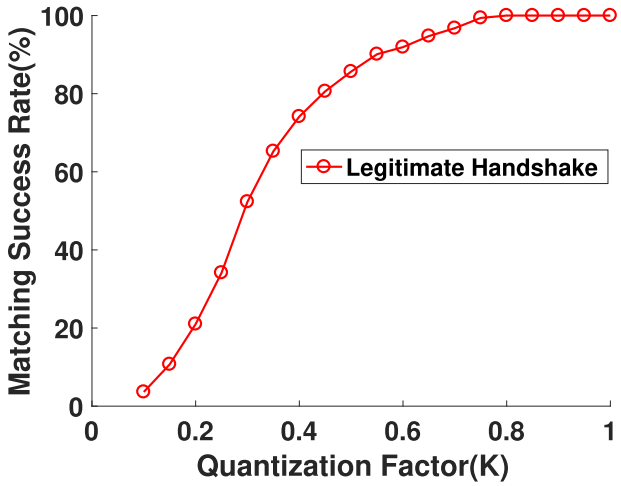


Fig. 10. Key matching success rate under various quantization factor value (K).

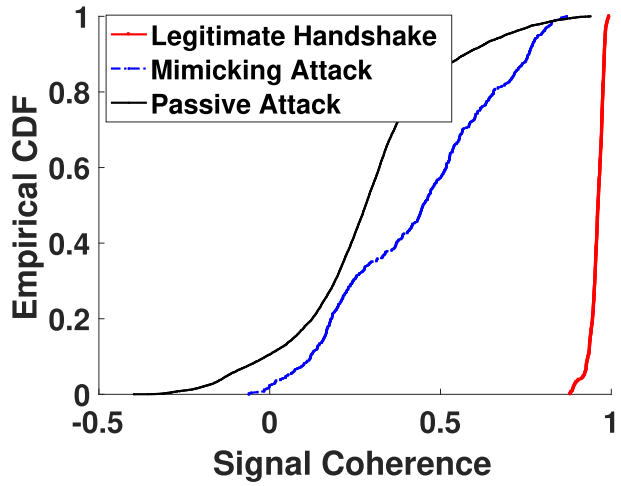
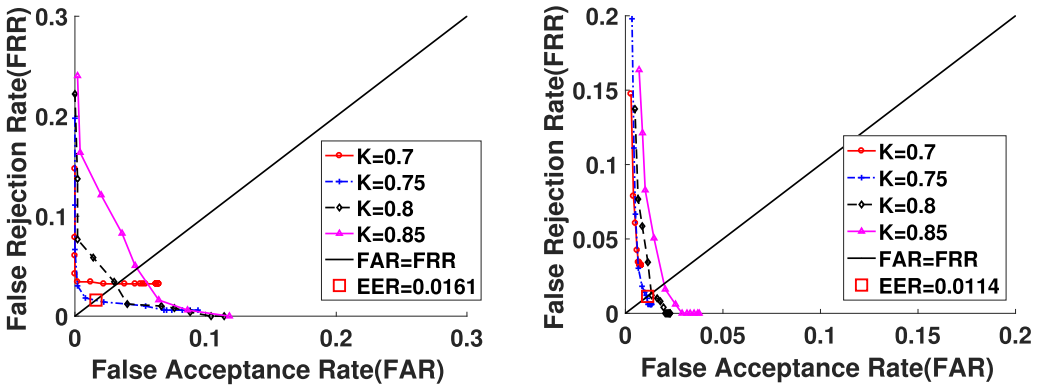


Fig. 11. The CDF of signal coherence between two signals generated by legitimate handshakes (red), by an adversary mimicking the legitimate handshakes (blue), and by an adversary shaking hands freely (black).

media, and mimic the handshaking patterns (mimicking adversary) or conduct their own handshakes freely (passive adversary) in real time, trying to produce the same cryptographic key. For the handshaking events, we first analyze the coherence between (a) signals produced by two legitimate devices; (b) signals produced by one legitimate device and the adversarial device that was imitating that handshake; and (c) signals produced by one legitimate device and the passive adversary's device. Figure 11 shows the empirical CDFs of signal coherence for the three cases. We see that the legitimate handshakes consistently produce many coherent signals: around 97% of the signals have coherence values over 0.9; while for mimicking adversary, around 97% is under 0.8; and for passive adversary, the results are even worse. This confirms that (a) during handshaking, the two legitimate devices tend to induce very similar motion signals; and (b) it is difficult to produce a similar key in real time without holding hands together, even if the adversary is able to mimic the patterns of the handshakes close by.



(a) Mimicking Attack (b) Passive Attack

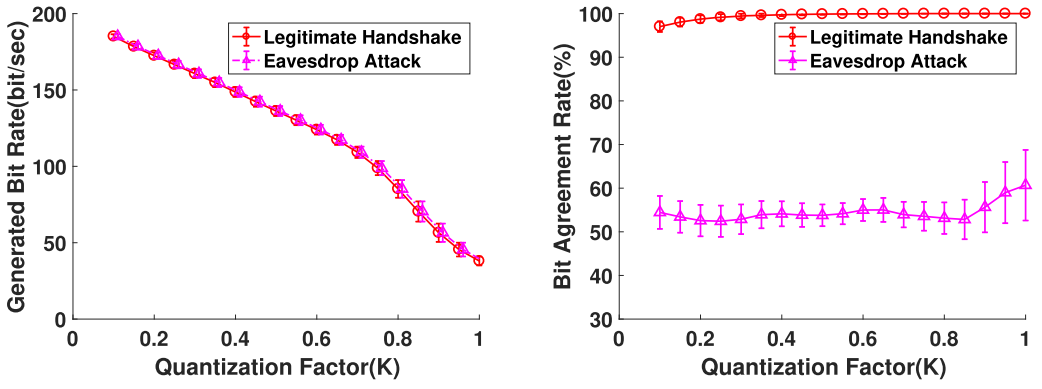
Fig. 12. FAR and FFR of the proposed system under different bit rate thresholds and quantization factor K under mimicking attacks and passive attacks. The best EER can be achieved in our experiments is 0.016 for mimicking attacks and 0.0114 for passive attacks (marked by the red rectangles).

Based on the above observations, now we evaluate the performance of the proposed system when facing mimicking attacks and passive attacks using metrics *FAR*, *FRR*, and *EER*. Generally, we would like to reduce both FAR and FRR, but in practice, they are often negatively correlated. As discussed in the previous experiments, we use the bit rate threshold to determine if a key should be accepted. Therefore, here, we vary the threshold (from 20 to 90) to see the correlation between FRR and FAR (see the lines in Figure 12(a) and (b) and Figure 12(a) and (c), where each dot is corresponding to a bit rate threshold value).

We also vary the quantization factor K, and plot the FAR vs. FRR curves for $K = \{0.7, 0.75, 0.8, 0.85\}$. As we can see, the proposed system works best with $K = 0.75$ under passive and mimicking attacks, where both FAR and FRR are lower (the curve is closer to the origin). We first investigate the results under mimicking attacks; in this case, the EER is 0.016 (see the rectangle marked in Figure 12(a)), which is the best balance between FAR and FRR. While for passive attacks, the EER is 0.0114 (see Figure 12(b)), which is slightly lower than that under mimicking attacks. As the system seems more vulnerable under mimicking attacks, we determine the parameters of our system according to the results of mimicking attacks. In practice, we typically require a lower FAR to make the system more conservative and robust against mimicking attacks, although this will increase the chances of falsely rejected legitimate handshakes. Concretely, in the current implementation we set the quantization factor K to 0.75, and the bit rate threshold to 70 bits per second. As shown in Figure 9(a), with these parameter settings, the system is able to generate a 128-bit key with only on average 1.3 seconds of handshaking, and we use device vibration patterns to inform the users if the data exchange is successfully completed or requires another go. It is worth noting that 1.3 seconds is longer than a normal handshake in social occasions; however, the users will benefit from the security add-on with little extra effort when they need to exchange information using their smart watches.

5.2.4 Security Analysis on Eavesdropping Attacks. We then evaluate the performance of the proposed system under eavesdropping attacks. We use the legitimate datasets and one adversary dataset (eavesdropping dataset) to evaluate its performance. We vary the quantization factor to compute the corresponding generated bit rate, bit agreement rate, and signal coherence.

Generated Bits Rate vs. K: We present how the generated bit rate of the legitimate handshakes and eavesdropping attacks varies over different K. It is apparent that the generated bit rate during



(a) Bit Rate (b) Bit Agreement Rate

Fig. 13. The (a) bit rate and (b) bit agreement rate generated by eavesdrop attacks benchmarked by legitimate handshakes.

eavesdropping attack is identical to the legitimate handshakes. This is because the adversary only overhears the communication between the legitimate devices but does not try to impersonate into the key generation protocol. The adversary determines the valid positions of the bits according to the reconciliation message exchanged between the legitimate devices; therefore, it produces the same bit rate to the legitimate handshakes.

Bit Agreement Rate vs. K: Though in eavesdropping attacks, the adversary produces high bit rate, it is impossible to decrypt the message without producing the identical key to the legitimate users. We compare the bit agreement rate produced by the legitimate handshakes and eavesdropping attacks. It is worth noting that, in the eavesdropping attack experiment, the adversary is able to mimic the legitimate handshakes close by as this produces higher agreement rate according to the evaluation in Section 5.2.3. The results of the bit agreement rate are presented in Figure 13(a). We can see that the average of the bit agreement rate produced by the adversaries fluctuates with the growth of K. The most important observation is that the gap between legitimate and adversarial handshakes is always significant through all K, and the agreement rate of the adversarial handshakes are unstable: it produces large variance. This indicates the eavesdropping attacks are nearly impossible to be successful.

Signal Coherence: At last, we compare the coherence between (a) signals produced by two legitimate devices and (b) signals produced by one legitimate device and the adversarial devices, which were mimicking the handshakes in eavesdropping attacks. Figure 14 demonstrates the empirical CDFs of eavesdropping attacks compared with the legitimate handshakes. We can easily confirm again that it is difficult to mimic the handshakes real time in eavesdropping attacks.

5.2.5 System Overhead. The final set of experiments evaluates the overhead of the proposed key generation system when running on off-the-shelf smart wearables. We profile the running time and energy consumption caused by key components of the system, including *Handshake Motion Capture*, *SVM-based Handshake Detection*, *Feature Extraction*, and *Key Generation*. The overhead of the remaining components such as data encryption and decryption can be application-specific, and is not considered in this article. In our implementation, we use the very lightweight Advanced Encryption Standard (AES) for data encryption/decryption, which only incurs negligible overhead according to our previous work [30].

Table 1 presents the detailed resource consumption of the three components obtained from Android APIs, averaging from 30 independent tests. We can see that the total running time is

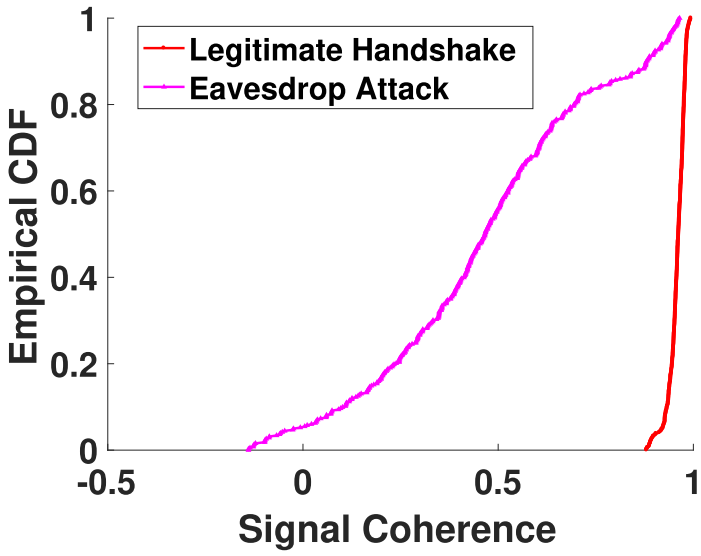


Fig. 14. The signals coherence under eavesdropping attacks.

Table 1. Overhead of the Proposed Key Generation System
Inters of Running Time (ms) and Energy Consumption (mJ)

Components	Time	Energy
Handshake Motion Capture	3.1 *ms*	2.04 *mJ*
SVM-based Handshake Detection	8.9 *ms*	5.86 *mJ*
Signal Feature Extraction	78.4 *ms*	60.35 *mJ*
Key Generation	89.6 *ms*	6.25 *mJ*
Total	180 *ms*	74.5 *mJ*

only 180 ms; thus, we could easily run the proposed system in real time. On the other hand, the average extra energy consumed by the key generation is 74.5 *mJ*. Since the system only works opportunistically when data exchange is required, its impact on the battery life should be limited. In fact, the battery capacity of Samsung Gear Live used in our experiments is 300 mAh (i.e., 4.32 *kJ*). If the targeted lifespan of the devices is one day (normally 12 hours of active usage), the proposed key generation system only accounts for 0.02% of the hourly budget (360 *J*), which is marginal.

6 RELATED WORK

Wrist-Worn Smart Wearables: The studies on smartwatches are booming in recent years and have produced many novel applications to improve the well-being of human's life. The sensor readings, e.g., accelerometer and gyroscope of the smartwatches, can easily track the motion of the user's arm and, therefore, are often used in activities recognition. For example, in Ref. [23], the authors proposed to use smartwatches to detect if the wearers took a foosball break at their workplace to maintain the work–break balance of the workers. Moreover, smartwatches were used to detect activities of the drivers in Ref. [13] to ensure the safety of the wearer during driving, and they could also be used to recognize the text inputs [2]. To improve the accuracy of activities recognition on smartwatches, the authors in Ref. [3] sought the power of deep learning techniques. Besides activities recognition, there is also some work concerning the security issue related to smartwatches.

For example, the sensors on mobile devices can be utilized as the sources of generating random numbers, which are key components for a number of different security-related applications [28]. In Ref. [33], the authors proposed Gait-watch, an authentication system exploiting the unique gait information of the smartwatch users to automatically detect the real user or malicious intruders.

Secure Wireless Communication between Smart Devices: A secure communication channel establishment between smart mobile devices requires multiple parties encryption key generation. Key generation methods are mainly based on the common information between the two parties sharing data. One of the mainstream studies focus on the properties of the wireless communication. For example, the Received Signal Strength Indicator (RSSI) of the wireless channel are frequently used to produce shared keys [9, 11, 19–21, 26] as the RSSI at both ends should be the same. However, the RSSI-based methods are only suitable for wireless devices exchanging packages frequently, but the application scenarios of the proposed system are for once-off data sharing. Another mainstream of key generation for mobile devices is utilizing the accelerometer readings. For example, in Refs [4], [5], and [14], the authors proposed to hold two mobile devices in one hand and shake them together; therefore, the accelerometer reading could be used to generate agreed keys. However, as our previous discussion in Section 1, these methods are not applicable for smartwatches belonging to different users. The previous work in Ref. [30] addressed the problem of automatic key generation for paring the on-body sensor networks by utilizing another natural pattern of humans, i.e., gaits. It focused on the mobile devices worn on the same human's body; therefore, it cannot be extended to this work. However, it is worth noting that we adopt the common quantization and key reconciliation mechanism proposed in previous work; however, we solve significantly different problems.

Data Exchange between Smart Wearables: Finally, there is some work exploring data sharing between smartwatches or waistbands. For example, the Nabu Smartband is able to share users social contacts via handshake; Apple Inc. drafted a patent [22] about exchanging information between devices in proximity when detecting a "greeting event" including handshakes. However, in both of these approaches, handshakes were just regarded as a hint to inform the devices there was a data sharing request; it did not consider using these handshakes to secure their data sharing process, which was the major contribution of this article. Besides, in a patent [1] submitted by Microsoft Inc., they proposed a new waist-worn hardware, which was able to transfer data via the human body; therefore, the secure data sharing could be achieved by physical contact like handshakes. However, it required extra hardware and could not be adopted by the majority of currently available smart wearables.

7 CONCLUSION

In this article, we propose the design and implementation of a novel key generation system that enables secure data exchange between smart devices via handshakes. The proposed system further blurs the boundaries between the physical and cyber worlds, as it uses the physical contact, i.e., handshakes between users, to bridge cyber contact, such as friending on social networks. Under the hood, our system uses the wrist-worn smart wearables, such as smartwatches or fitness bands, to capture the motion patterns induced by handshakes. Although belonging to different users, we show that the motion signals of the two hands shaking together are very similar. Based on this, we propose novel approaches to robustly generate and reconcile cryptographic keys on both sides, and use the pair of symmetric keys to establish a secure wireless communication channel in a distributed way. We evaluate the proposed key generation system extensively in real-world settings, and experimental results show that the proposed system (a) is able to reliably generate high-quality keys within less than two seconds of handshaking; (b) is very resilient to the mimicking attacks,

achieving only 1.6% Equal Error Rate; and (c) can run in real time on off-the-shelf smartwatches with very slim resource consumption.

The key quantization component in this article only considers binary quantization, which limits the key generation rate. Therefore, users need to pay more effort on handshaking compared with normal occasions. In the future, we will investigate a multi-level quantization strategy to significantly improve the key generation rate so that the key can be generated with significantly shorter handshakes.

REFERENCES

[1] Brian Amento, Kevin Ansia Li, Kermit Hal Purdy, and Larry Stead. 2014. Devices and methods for transferring data through a human body. US Patent 8,908,894.

[2] Luca Ardüser, Pascal Bissig, Philipp Brandes, and Roger Wattenhofer. 2016. Recognizing text using motion data from a smartwatch. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops'16)*. IEEE, 1–6.

[3] Sourav Bhattacharya and Nicholas D. Lane. 2016. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops'16)*. IEEE, 1–6.

[4] Daniel Bichler, Guido Stromberg, Mario Huemer, and Manuel Löw. 2007. Key generation based on acceleration data of shaking processes. *UbiComp 2007: Ubiquitous Computing* (2007), 304–317.

[5] Claude Castelluccia and Pars Mutaf. 2005. Shake them up!: A movement-based pairing protocol for CPU-constrained devices. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*. ACM, 51–64.

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

[7] Whitfield Diffie and Martin Hellman. 1976. New directions in cryptography. *IEEE Transactions on Information Theory* 22, 6 (1976), 644–654.

[8] Chitra Javali, Girish Revadigar, Ming Ding, and Sanjay Jha. 2015. Secret key generation by virtual link estimation. In *Proceedings of the 10th EAI International Conference on Body Area Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 301–307.

[9] Chitra Javali, Girish Revadigar, Lavy Libman, and Sanjay Jha. 2014. SeAK: Secure authentication and key generation protocol based on dual antennas for wireless body area networks. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer, 74–89.

[10] Wei-Chi Ku and Shen-Tien Chang. 2005. Impersonation attack on a dynamic ID-based remote user authentication scheme using smart cards. *IEICE Transactions on Communications* 88, 5 (2005), 2165–2167.

[11] Kai Li, Wei Ni, Yousef Emami, Yiran Shen, Ricardo Severino, David Pereira, and Eduardo Tovar. 2019. Design and implementation of secret key agreement for platoon-based vehicular cyber-physical systems. *ACM Transactions on Cyber-Physical Systems* 4, 2 (2019), 1–20.

[12] Chris Xiaoxuan Lu, Bowen Du, Hongkai Wen, Sen Wang, Andrew Markham, Ivan Martinovic, Yiran Shen, and Niki Trigoni. 2018. Snoopy: Sniffing your smartwatch passwords via deep sequence learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 152.

[13] Alex Mariakakis, Vijay Srinivasan, Kiran Rachuri, and Abhishek Mukherji. 2016. WatchUDrive: Differentiating drivers and passengers using smartwatches. In *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops'16)*. IEEE, 1–4.

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

[15] Razer Nabu. [n.d.]. Social Wearable Smartband with Display and Sensor. Retrieved from https://www2.razerzone.com/nabu.

[16] Apple Newsroom. 2017. Watch Series 3 brings built-in cellular, powerful new health and fitness enhancements. Retrieved from https://www.apple.com/au/newsroom/2017/09/apple-watch-series-3-features-built-in-cellular-and-more/.

[17] Swati Rallapalli, Aishwarya Ganesan, Krishna Chintalapudi, Venkat N. Padmanabhan, and Lili Qiu. 2014. Enabling physical analytics in retail stores using smart glasses. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. ACM, 115–126.

[18] Allied Market Research. 2016. Smartwatch Market is Expected to Reach 32.9 Billion, Globally, by 2020. Retrieved from https://goo.gl/DJjHYs.

[19] Girish Revadigar, Chitra Javali, Hassan Jameel Asghar, Kasper B. Rasmussen, and Sanjay Jha. 2015. Mobility independent secret key generation for wearable health-care devices. In *Proceedings of the 10th EAI International Conference on*

Body Area Networks. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 294–300.

[20] Girish Revadigar, Chitra Javali, Wen Hu, and Sanjay Jha. 2015. Dlink: Dual link based radio frequency fingerprinting for wearable devices. In *Proceedings of the IEEE 40th Conference on Local Computer Networks (LCN'15)*. IEEE, 329–337.

[21] Girish Revadigar, Chitra Javali, Weitao Xu, Wen Hu, and Sanjay Jha. 2016. Secure key generation and distribution protocol for wearable devices. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops'16)*. IEEE, 1–4.

[22] Brent W. Schorsch, David J. Shoemaker, Eugene Dvortsov, and Kamlesh Kudchadkar. 2013. Gesture-based information exchange between devices in proximity. US Patent App. 15/102,834.

[23] Sougata Sen, Kiran K. Rachuri, Abhishek Mukherji, and Archan Misra. 2016. Did you take a break today? Detecting playing foosball using your smartwatch. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops'16)*. IEEE, 1–6.

[24] Yiran Shen, Hongkai Wen, Chengwen Luo, Weitao Xu, Tao Zhang, Wen Hu, and Daniela Rus. 2018. GaitLock: Protect virtual and augmented reality headsets using gait. *IEEE Transactions on Dependable and Secure Computing* 16, 3 (2018), 484–497.

[25] Yiran Shen, Fengyuan Yang, Bowen Du, Weitao Xu, Chengwen Luo, and Hongkai Wen. 2018. Shake-n-shack: Enabling secure data exchange between smart wearables via handshakes. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*. IEEE.

[26] Lu Shi, Jiawei Yuan, Shucheng Yu, and Ming Li. 2013. ASK-BAN: Authenticated secret key extraction utilizing channel characteristics for body area networks. In *Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 155–166.

[27] Yi-Sheng Shiu, Shih Yu Chang, Hsiao-Chun Wu, Scott C.-H. Huang, and Hsiao-Hwa Chen. 2011. Physical layer security in wireless networks: A tutorial. *IEEE Wireless Communications* 18, 2 (2011).

[28] Kyle Wallace, Kevin Moran, Ed Novak, Gang Zhou, and Kun Sun. 2016. Toward sensor-based random number generation for mobile and IoT devices. *IEEE Internet of Things Journal* 3, 6 (2016), 1189–1201.

[29] Wikipedia. 2017. AirDrop—Wikipedia, The Free Encyclopedia. Retrieved September 20, 2017 from http://en.wikipedia.org/w/index.php?title=AirDrop&oldid=795508835.

[30] Weitao Xu, Chitra Javali, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. 2017. Gait-key: A gait-based shared secret key generation protocol for wearable devices. *ACM Transactions on Sensor Networks (TOSN)* 13, 1 (2017), 6.

[31] Weitao Xu, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. 2016. Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication. In *Proceedings of the 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 1–12.

[32] Weitao Xu, Yiran Shen, Neil Bergmann, and Wen Hu. 2017. Sensor-assisted multi-view face recognition system on smart glass. *IEEE Transactions on Mobile Computing* 17, 1 (2017), 197–210.

[33] Weitao Xu, Yiran Shen, Yongtuo Zhang, Neil Bergmann, and Wen Hu. 2017. Gait-watch: A context-aware authentication system for smart watch based on gait recognition. In *Proceedings of the 2nd International Conference on Internet-of-Things Design and Implementation*. ACM, 59–70.

[34] Hongyang Zhao, Shuangquan Wang, Gang Zhou, and Daqing Zhang. 2017. Gesture-enabled remote control for healthcare. In *Proceedings of the 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 392–401.

Received June 2019; revised January 2020; accepted January 2020