CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Conference Paper

i2Key: A Cross-sensor Symmetric Key Generation System using Inertial Measurements and Inaudible Sound

BO WEI
WEITAO XU
KAI LI*
CHENGWEN LUO
JIN ZHANG

*CISTER Research Centre

CISTER-TR-220203

2022/05/04

i2Key: A Cross-sensor Symmetric Key Generation System using Inertial Measurements and Inaudible Sound

BO WEI, WEITAO XU, KAI LI*, CHENGWEN LUO, JIN ZHANG

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: weitaoxu@cityu.edu.hk, kai@isep.ipp.pt, chengwen@szu.edu.cn, jin.zhang@szu.edu.cn

https://www.cister-labs.pt

Abstract

Networked devices, such as wearable devices, laptops, smart home appliances, etc., are ubiquitous nowadays. To secure communication among those devices, symmetric keys are widely used because of their feasibility in resource-constrained networked devices. The observations of sensors from independent devices have been adopted for symmetric key generation. The identical biometrics information or environment interference have been observed by sensors, and their corresponding patterns are used for key generation. Popular signals from networked devices are inertial measurements, sound, wireless signals, etc. The existing sensor-based key generation solutions use the same type of sensors for both devices. Different from the existing solutions, we are the first to propose a cross-sensor symmetric key generation system i2Key, where two devices collect inertial measurements from a motion sensor and inaudible sound from a microphone, respectively. A new coding framework is designed for general key generation. We also propose an efficient and accurate time synchronisation method for key generation. Additionally, a multi-tier key reconciliation method is suggested to improve key generation performance. By using the proposed architecture, the key generation rate is improved by up to approximately 40% compared with the situation without using it. We also perform security analysis and randomness analysis over the proposed method.

i^{2}Key: A Cross-sensor Symmetric Key Generation System using Inertial Measurements and Inaudible Sound

Bo Wei*
Lancaster University, UK
bo.wei@lancaster.ac.uk

Weitao Xu
City University of Hong Kong, China
weitaoxu@cityu.edu.hk

Kai Li
CISTER Research Centre, Portugal
kaili@isep.ipp.pt

Chengwen Luo
Shenzhen University, China
chengwen@szu.edu.cn

Jin Zhang
Shenzhen University, China
jin.zhang@szu.edu.cn

ABSTRACT

Networked devices, such as wearable devices, laptops, smart home appliances, etc., are ubiquitous nowadays. To secure communication among those devices, symmetric keys are widely used because of their feasibility in resource-constrained networked devices. The observations of sensors from independent devices have been adopted for symmetric key generation. The identical biometrics information or environment interference have been observed by sensors, and their corresponding patterns are used for key generation. Popular signals from networked devices are inertial measurements, sound, wireless signals, etc. The existing sensor-based key generation solutions use the same type of sensors for both devices. Different from the existing solutions, we are the first to propose a cross-sensor symmetric key generation system i^{2}Key, where two devices collect inertial measurements from a motion sensor and inaudible sound from a microphone, respectively. A new coding framework is designed for general key generation. We also propose an efficient and accurate time synchronisation method for key generation. Additionally, a multi-tier key reconciliation method is suggested to improve key generation performance. By using the proposed architecture, the key generation rate is improved by up to approximately 40% compared with the situation without using it. We also perform security analysis and randomness analysis over the proposed method.

KEYWORDS

Key generation, cross-sensor, inertial measurement units, sound, compressed sensing

1 INTRODUCTION

With the development of embedded systems and wireless communication techniques, the Internet of Things (IoT) and networked devices have been ubiquitous in many application scenarios, such as smart homes, factories, offices, farms, etc. With the ubiquitous deployment, wireless communication techniques, e.g. WiFi, Bluetooth, LoRa, etc., are commonly employed to share encrypted data. The cryptographic key agreement is, therefore, essentially required when communicating in the "open-air" environment. Even though the encryption method for communication is dominated by asymmetric key encryption (also known as public key encryption, such as RSA) in personal computers, mobile phones and servers, symmetric key cryptographic methods are still usually used by resource-constrained IoT devices due to their less computational complexity compared with public key cryptography. It has been empirically shown that the use of symmetric keys can significantly reduce energy consumption compared with that of asymmetric keys [50], which will be further confirmed in Section 4.9 Diffie-Hellman protocol is a common key establishment protocol over a public communication channel, but the use of Diffie-Hellman protocol usually requires additional certificate authority (CA) in practice. Otherwise, Diffie-Hellman protocol is vulnerable to man-in-the-middle (MITM) attack. Near field communication (NFC) is a popular authentication method in mobile devices. However, it has a very short communication range with no more than 5 cm. Another traditional yet common way for pairing devices is to let users manually choose nearby devices from the scanned list. However, the involvement of manual operation is not user friendly, and this is not feasible for screenless devices.

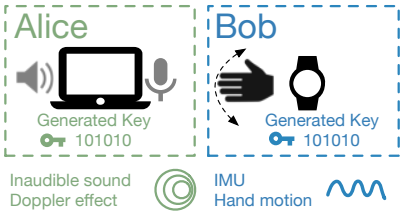


Figure 1: Proposed System: two devices use inertial measurements and inaudible sound, respectively, for symmetric key generation

Motivation: To explore the use of efficient and user friendly device paring methods, recent literature in the research community has focused on taking advantage of on-board sensors for symmetric key generation methods, such as wireless channel signals based [46, 49, 52], audio signal [19, 37] and inertial measurements based [18, 39, 51]. When using measurements from sensors, the symmetric key generation relies on similar observations from two devices. However, the key drawback of the existing systems is the requirement and assumption of the existence of the same sensors from independent devices. The hypothesis will bring barriers to the large scale of application and deployment of their key generation methods.

To generalise the key generation for the universal IoT devices, we propose to utilise two common signals, i.e. inaudible sound and

*Corresponding Author

inertial measurement units (IMUs), respectively, from two independent mobile devices, for a symmetric key generation method. Audio devices (including a microphone and a speaker) and IMUs are commonly equipped with mobile devices. Here are two motivation scenarios:

Motivation Scenario 1: one user wears one mobile or wearable device (e.g. a smartwatch) with an IMU and aims to pair with a laptop equipped with a speaker and a microphone by generating symmetric keys as shown in Figure 1.

Motivation Scenario 2: one user wears a smart band with an IMU but without a touch screen and intends to pair it with a smart TV with a microphone and a speaker to show step counts she has done today.

In these motivation scenarios, not both of devices have the same type of sensors (a microphone and speaker pair or an IMU). In Motivation Scenario 2, the wearable band does not include a touch screen, so the traditional password-based paring method cannot be used. Our proposed method aims to solve these issues in these common scenarios and make an contribution to a cross-sensor key generation method. Specifically, to enable the cross-sensor key generation, we obtain the movement of one wearable device using its integrated IMU. Simultaneously, its movement is also captured by inaudible sound from the other devices based on the Doppler effect. A novel cross-sensor symmetric key generation method is proposed to generate and correct symmetric keys.

To summarise, the main contributions of our papers are shown as follows.

- To the best of our knowledge, the proposed work is the *first* cross-sensor key generation solution, which enables key generation using two different types of sensors.
- A novel time synchronisation method along with a new coding scheme is suggested to encode two different types of measurements (i.e. IMU measurements and inaudible sound) and ensure the high-quality symmetric key generation.
- A two-tier key correction method is further proposed to remove and correct ambiguous bits in generated keys and thus improve the performance of key generation.
- Extensive experiments including evaluations by different users and in different hardware have been conducted to evaluate the effectiveness of the proposed method, which outperforms the existing methods. The randomness of generated keys has been verified using the NIST Statistical Test Suite. The system has also been developed in IoT devices to evaluate its energy efficiency.
- The security analysis has been conducted to show the proposed key generation mechanism is robust to common attacks, such as imitating attack and eavesdropping attack.

The rest of this paper is organised as follows. Section 2 shows the feasibility study and the challenges of the proposed method. In Section 3, we show our proposed cross-sensor key generation method in detail. Section 4 conducts extensive evaluations in real environments to demonstrate its efficacy and robustness. Security analysis is conducted in Section 5. Section 6 describes the related works. Finally, we concludes the paper in Section 7.

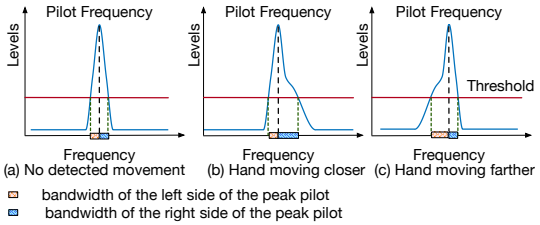


Figure 2: Doppler Effect: by using Doppler Effect, the inaudible sound can indicate the hand movements

2 FEASIBILITY STUDY AND CHALLENGES

In this section, we first study the feasibility of the use of inertial measurements and inaudible sound in independent devices for symmetric key generation and then analyse its challenges. IMUs in mobile devices are widely used to provide acceleration and angular acceleration measurements. Therefore, it is practicable to offer the movement of carrying hand based on proper processing. Wearing an IMU, this hand is supposed to move over the microphone of another device when its speaker continuously plays inaudible sound in one particular frequency (20 kHz in our implementation[1]). The microphone is also able to detect the movement of the vicinal hand based on the frequency shift according to the Doppler effect. Please note the hand tracking can also be done by playing sound in an audible frequency. The reason to choose an inaudible frequency is to avoid ambient background noise and its interference to the surrounding environment. Symmetric keys will be generated using measurements from inertial measurements and received inaudible sound frequency from two independent devices.

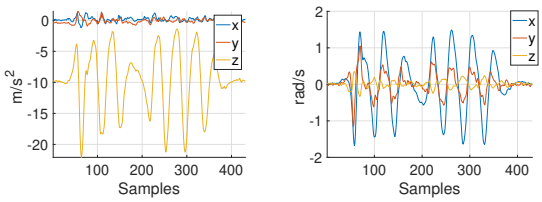
In the following part of this section, we conduct preliminary experiments and show the feasibility of the use of inaudible sound and inertial measurements to indicate hand movements. We first demonstrate the feasibility of the use of inaudible sound for the detection of hand movement. Figure 2(a) shows the frequency response without any interference from surrounding movements. i^{2}Key uses the Doppler effect theory for processing inaudible sound, where the moving object in the vicinity of the microphone and the speaker can interfere with the received frequency of sound. The measured frequency f_m by the microphone is calculated by Equation 1.

$$f_m = f_t \times \frac{(v_s + v_h)}{(v_s - v_h)} \tag{1}$$

where f_t is the actual frequency sent by the speaker, and v_s and v_h are the speed of sound in the air and nearby moving hand, respectively. Figures 2(b) and 2(c) show the frequency shift with the hand moving down (closer i.e. $v_h > 0$) and up (farther i.e. $v_h < 0$). The variance in the frequency response will be leveraged to indicate the movement of the hand.

Carrying one mobile phone with an IMU, a hand moves up and down over the microphone simultaneously. It is anticipated that its movement can be clearly indicated by IMU measurements. Figure 3 shows the acceleration and angular acceleration measurements from IMU when the hand moves up and down, which obviously indicates several peaks when the hand changes the movement direction. The gravity is reflected by the offset of acceleration measurements.

[1]Sound above 18 kHz is usually inaudible. 20 kHz is picked to further decrease the potential influence from other sources.



(a) Acceleration (b) Angular speed

Figure 3: IMU measurements when the carried hand moves up and down

With proper processing, an IMU is definitely able to indicate the hand movement.

i²Key uses the reciprocity between IMU measurements and inaudible sound signal. Both types of signals can indicate the motions of a tracked hand. However, there still remain challenges:

Challenge 1: The first challenge is the performance reduction caused by noise from low-cost IMU devices. Theoretically, the simple double integration of the acceleration and angular acceleration measurements can directly indicate the movement of the mobile phone. However, the measurements from low-cost IMUs integrated into mobile devices are not sufficiently accurate, which suffers from the non-avoidable noise from accelerometers and long-term shift from gyroscopes. Therefore, we will consider this in our general encoding framework, which will be introduced in Section 3.

Challenge 2: The time of the two devices is not strictly synchronised. Symmetric keys are generated by similar observations. Since the hand movement is usually very fast, time synchronisation between two sensors needs to be carefully considered. Without accurate time synchronisation, shifting measurements would create notable key mismatches when generating symmetric keys with measurements.

Challenge 3: The curve trend of measurements from two types of sensors cannot be exactly identical. It needs an innovative method to encode measurements from two independent devices to ensure the similarity of encoded measurements and the capability of generating symmetric keys in high performance.

3 METHOD

3.1 System Overview

Alice and Bob are the users of two legitimate devices which aim to create secure communication by using generated symmetric keys. We assume two devices in the vicinity are equipped with different sensors, i.e. a speaker and a microphone in Alice's device and an IMU in Bob's device. When Alice and Bob generate symmetric keys, Alice plays inaudible sound in 20 kHz and uses a speaker to capture the sound simultaneously. Wearing a device with an IMU, Bob waives the hand up and down at a random speed. Received inaudible sound and IMU measurements will be used to generate symmetric keys for secure communication in each independent device. Figure 4 shows the flowchart of the proposed system. During the key generation process, Bob first conducts a start gesture to acknowledge the system that the following measurements are used for generating symmetric keys. Additionally, the start gesture will be used for time synchronisation. The multi-bit quantisation

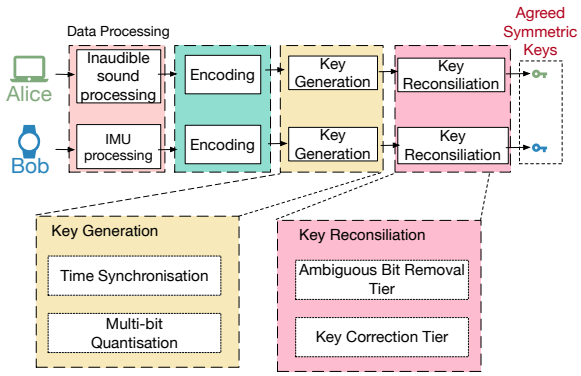


Figure 4: System overview: i²Key includes four steps to generate symmetric keys: data processing, encoding, key generation, and key reconciliation.

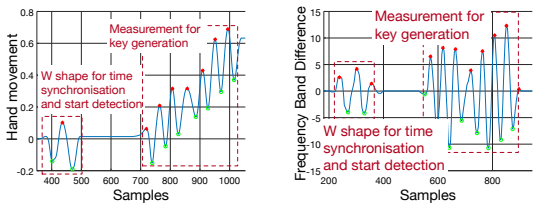
will be used to generate keys in both Alice and Bob independently. Finally, the proposed system will apply the two-tier key reconciliation method, including an ambiguous key removal tier and a key correction tier, to increase the key matching rate. Once secure communication is established, there is no need to conduct continuous sensing for this purpose anymore.

3.2 Inaudible Audio Processing

In this section, we will describe inaudible audio processing used by Alice for hand movement tracking and key generation. We adopt the signal processing method in [13] that uses the Doppler effect for the detection of the direction and speed of a moving hand. Section 2 has shown the feasibility of the use of inaudible sound for indicating the hand movement. In this section, we will reveal the sound processing method[2]. Alice continuously plays a pilot tone in an inaudible frequency band. We use 20 kHz in our implementation. Bob's moving hand in the vicinity of the speaker and microphone will cause the Doppler effect, and the induced shifted frequency will be reflected by the captured samples. Fast Fourier Transform (FFT) with n-point Hamming windows is applied to the captured samples by the microphone. Without any other interference, the number of bins neighbouring to the pilot tones (20 kHz) depends on the speed of moving hand according to the Doppler effect shown in Equation 1. In our implementation, we use the same setting in [13]. 1024-point Hamming windows are used with 66 neighbouring bins considered on the pilot frequency, assuming the speed of moving hand is no more than 6 m/s. These 66 neighbouring bins are scanned, and the bandwidth of the left side and the right side of the peak pilot will be calculated, respectively. The bandwidth difference $b_d = b_l - b_r$ will be employed for further processing, where b_l and b_r are the bandwidth of the left side and the right side of the peak pilot, respectively. Figure 5(b) shows b_d with respect to the time when the moving hand goes up and down. Local peaks (labelled by red and green spots) clearly shows the time points of the hand's changing directions.

Since inaudible sound is used to track hand movement, the ambient environment noise cannot affect captured inaudible sound. To limit the sensing range of the microphone and avoid the impact

[2]More details can be found in [13].



(a) IMU measurements (b) Sound measurements

Figure 5: IMU and sound measurements after data processing of ambient movement, the volume of the speaker needs to be tuned lower in a crowded environment.

3.3 IMU Processing

The same as the inaudible sound processing, the use of IMU aims to indicate hand movement. In this section, we will show the details of the used method of producing the hand trajectory using 6 degrees of freedom (DOF) IMU measurements. Without an elaborate stabilised platform, an IMU hand tracking in mobile devices is strap-down inertial navigation, i.e. the IMU is fastened directly to the user's hand, and three dimensions of acceleration measurements and three dimensions of gyroscope measurements are collected and used to indicate movement. Specifically, the following steps are used to obtain IMU positions. First, angular rates are integrated to obtain the attitude of IMU. Second, the attitude of IMU will be used to estimate the rotation of the IMU to transform the acceleration measurements (caused by both hand movement and gravity) from the body reference frame to the global reference frame. The gravity is then removed from the acceleration measurements in the global reference frame and the orientation of the device is derived. Finally, velocity and position are estimated using corrected acceleration in the global reference frame.

3.4 General Encoding Framework for Key Generation

Figure 5(a) and Figure 5(b) show movement estimates in z axis (perpendicular to the ground plane) from an IMU and inaudible sound, respectively, which clearly indicates that they have similar changing trend patterns. However, they cannot be exactly matched. The trend change of inaudible measurements is almost linear between the minimal peak and the maximum peak, while that of IMU signals is non-linear. This means the simple offset removal and normalisation cannot help these two types of measurements matched perfectly. To match these two types of measurements for symmetric key generation, we design a general encoding framework to make these two types of measurements matched and consistent.

Encoding method: Even IMU and inaudible measurements cannot matched perfectly, when looking at both types of measurements, intervals between local peaks are identical because they are simultaneously affected by hand movements. It can be anticipated that the local peaks, i.e. maximum and minimum values, should simultaneously happen. Therefore, we detect the local maximum and minimum values and record their event time. Then, encoding will be based on their event time.

Local peak detection: For both types of measurements, the local maximal and minimal values will be encoded as 1 and 0, respectively.

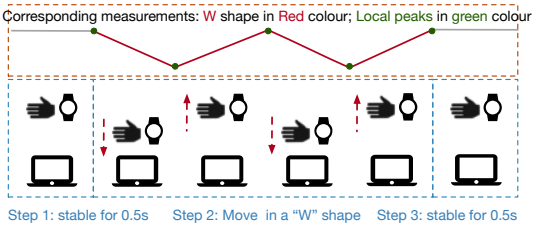
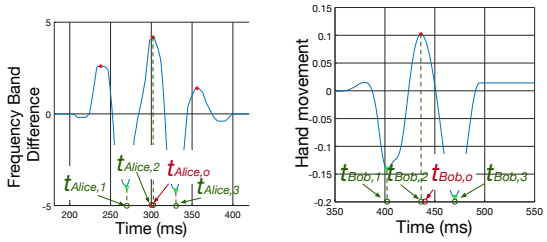


Figure 6: "W" shapes for the start gesture and time synchronisation:the hand, wearing one IMU, moves up and down, to make the measurements in temporal domain as W shape and, therefore, synchronise time between one pair of devices



(a) IMU measurements (b) Sound measurements

Figure 7: "W" shape for time synchronisation from IMU measurements and sound measurements

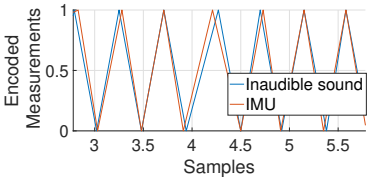


Figure 8: Synchronised Encoded Measurements after using "W" shape based time synchronisation method

Minimal peak prominence is used to prevent noise the small unexpected hand motions. Furthermore, we will use linear interpolation to encode the measurements of the immediate samples between local peaks, while original measurements will be abandoned. Figure 8 demonstrates the encoded measurements for symmetric key generation from Figure 5. Since the method is based on peaks of measurements, periodic measurements with peaks are needed for key generation.

3.5 Automatic and Accurate Time Synchronisation

Our proposed cross-sensor symmetric key generation system relies on the similarity of the encoded measurement at each time point. Therefore, accurate time synchronisation is of great importance to the cross-sensor key generation method. Network Time Protocol (NTP) [29] is a common time synchronisation protocol for networked devices. However, the empirical experiments show the accuracy of NTP can only achieve approximately 25ms by using an NTP server in a local area network [43]. This means that, with the

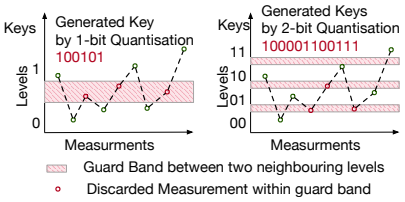


Figure 9: Multiple Bit Quantisation: measurements within guard band will be discarded.

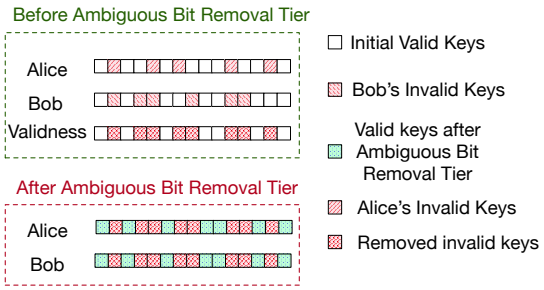


Figure 10: Example of ambiguous bit removal. Only valid bits in both Alice and Bob will be kept.

sampling rate of 100 Hz, there could be 2-3 measurement shift between a pair of devices using NTP, which will significantly decrease the key matching performance. Therefore, we propose a novel time synchronisation method based on measurement correlation. The proposed time synchronisation method is also responsible for start point detection based on a designed action without any package exchange between one pair of devices.

In our designed system, we use "W" shapes for the start gesture and the corresponding key points in "W" shapes for time synchronisation. Figure 6 shows the hand movements during conducting "W" shapes. Specifically, to detect the start point, Bob's hand that carries the IMU (1) keeps stable for more than half a second, (2) conducts a "W" shape in the air near the microphone (3) keeps stable again for more than half a second. The measurements from the IMU and sound are shown in Figure 5(a) and Figure 5(b), respectively. The processed measurements from both sensors can generate a "W" shape. For one "W" shape, there should be continuous 5 local peaks including 3 maximum local peaks and 2 minimum local peaks. Because the first and last local maximum peaks of the "W" shape are neighbouring to the steady measurements, they do not always appear. Therefore, we use three peaks in the middle to match two series of measurements.

Once "W" shapes are found in Alice and Bob, the time stamps of corresponding key points are used for time synchronisation. As shown in Figure 7, the time stamps of key points are represented as $t_{Alice,1}$, $t_{Alice,2}$, $t_{Alice,3}$, $t_{Bob,1}$, $t_{Bob,2}$ and $t_{Bob,3}$, respectively. The next step is to calculate origin time point for each device. $t_{Alice,2}$ and $t_{Bob,2}$ can be the origin time points. To further improve the time accuracy, we use the other two time stamps of key points in Alice and Bob, respectively. The origin time points for the devices are calculated in Equation 2.

$$t_{p,o} = \frac{t_{p,2} + (t_{p,1} + t_{p,3})/2}{2}, p \in \{Alice, Bob\} \quad (2)$$

$t_{p,o}$ is applied as the origin time point to calculate the time stamps of measurements from both devices. Figure 7 shows examples of origin points of Alice and Bob, and Figure 8 shows the synchronised encoded measurements after using "W" shape based time synchronisation method. The "W" shape based time synchronisation method usually takes no more than 2 seconds, so this would not cause inconvenience for users to conduct this extra action.

3.6 Key Generation

In this section, the proposed key generation method is introduced in detail. The multiple bit quantisation method [16] is used to generate symmetric keys to map the encoded measurements within the range between 0 and 1 to keys with binary bits. As illustrated in Figure 9, multiple quantised levels with the represent of binary bits are used to generate keys. The guard bands are employed between neighbouring levels to avoid key mismatch due to noise and improve the key generation rate. Specifically, we use $k-1$ guard bands for k-ary multiple bits quantisation. Each guard band is inserted between a pair of neighbouring bands. The keys are generated based on band B_i where measurements are located. $B_i = (b_u, b_{i-1} + s]$, where b_u is the boundary of the ith level. The length of the guard band s is calculated as $s = \frac{\alpha \times (b_u - b_{u-1})}{m}$, where α is the ratio of the guard band and m is the number of guard bands. The number of bits for each measurement is gauged as $n = log_2 m$. Figure 9 shows two examples when n is 1 and 2, respectively. Measurements falling into guard bands will be taken out without quantisation. A sequence of initial symmetric keys will be generated using this multiple bit quantisation. Now, we are ready to introduce the proposed key reconciliation method.

3.7 Two-Tier Key Reconciliation Method

The initial raw symmetric keys generated by Alice and Bob are usually not identical, so the reconciliation method is used to correct mismatched keys. In this paper, we propose a novel two-tier key reconciliation method, which includes the ambiguous bit removal tier and the key correction tier. Here we show the details of these two tiers.

3.7.1 Ambiguous Bit Removal Tier. The use of guard bands can help eliminate the effect of noise when generating keys, but the removal of measurements falling in guard bands in one device does not necessarily mean the corresponding measurements in the other device will not be included for coding either. In other words, the inconsistency will cause accumulating mismatches for the following generated keys. To solve this problem, we introduce an ambiguous bit removal method tier. The purpose of this tier is to find valid initial keys in both legitimate devices. Please note, this tier only cares about the locations of valid keys, but it does not correct keys. The next tier will be responsible for key correction.

Once legitimate devices exchange the location information of valid keys using "open-air" communication, both of them only select the shared locations of valid keys and discard invalid keys. One example of an ambiguous bit removal tier is shown in Figure 10. In this example, each measurement will generate 1 bit, 0 or 1.

There are 8 measurements. Alice and Bob generate keys at locations of {1,3,4,5,6,7} and {2,3,4,5,7,8}, respectively. The shared locations are {3,4,5,7}. The keys generated at the shared locations are picked for further processing in the next tier. Suppose Alice and Bob's initial generated keys are "0<u>10</u>01<u>0</u>" and Bob's initial key is "0<u>110</u>01", respectively. The underlined bits are at common valid locations. Afterwards, Alice and Bob will have the key "1000" and "1100", respectively. One mismatched bit still exists after the first tier, which needs further processing in the next key correction tier.

3.7.2 Key Correction Tier. Noise from both devices unavoidably causes key mismatching even after the ambiguous bit removal tier. The following key correction tier is used to correct the mismatched keys for legitimate devices. The compressed sensing based key correction method [21] is employed in this paper in order to (1) increase privacy for the "open-air" wireless communication and (2) reduce the size of the exchanged messages. The exchange of the keys relies on wireless communication. The use of compressed sensing for key correction will encrypt the exchanged initial keys with a random matrix for privacy protection and transmitted message size reduction. The sent encrypted keys can only be solvable by the use of ℓ_1 minimisation when the mismatched keys are sparse. More details regarding compressed sensing can be found in [9]

Here, we introduce the details of the compressed sensing based key correction method. The feasibility of the use of compressed sensing theory for key correction is based on the fact of the sparsity of mismatched keys and the use of ℓ_1 minimisation. Suppose the initial valid keys after the ambiguous bit removal tier from Alice and Bob are k_{Alice} and k_{Bob}. After the use of multiple bit quantisation and ambiguous bit removal, the difference between k_{Alice} and k_{Bob} is small, i.e. $\Delta k = k_{Alice} - k_{Bob}$ is sparse with only few non-zeros elements to be corrected, which is the condition in the compressed sensing theory that ℓ_1 minimisation can recover the difference Δk. To correct mismatched keys, one random project matrix R with elements in symmetric Bernoulli distribution is kept and used in both legitimate devices, Alice and Bob. Alice applies the project matrix R on k_{Alice} and obtains an encrypted message $y_{Alice} = Rk_{Alice}$, and y_{Alice} is then sent to the other legitimate device Bob using the wireless communication. Once Bob receives the message y_{Alice} from Alice, he uses the same method to obtain y_{Bob}, i.e. $y_{Bob} = Rk_{Bob}$ and calculates $\Delta y = y_{Alice} - y_{Bob}$, so $\Delta y = Rk_{Alice} - Rk_{Bob} = R(k_{Alice} - k_{Bob}) = R\Delta k$. Δk is sparse due to the low percentage of the mismatched keys. Therefore, Δk can be recovered by ℓ_1 minimisation based on compressed sensing theory according to Equation 3.

$$\arg\min_{\Delta k} \|\Delta k\|_1 \quad \text{subject to } \|\Delta y - R\Delta k\|_2 < \epsilon. \tag{3}$$

, where ϵ is the noise. Δk is then used by Bob to achieve the agreed key as Alice, i.e. $k'_{Bob} = k_{Bob} \oplus \Delta k$.

The MAC is used on the final generated keys by Alice and Bob in our implementation, to avoid Eve spoofing one legitimate device and modifying exchanged messages. This also helps evaluate the effectiveness of generated keys. Similar implementations have also been used in the existing systems [50, 51, 53]. To be specific, during the final key correction step, Alice sends her compressed message key $y_{Alice} = Rk_{Alice}$ along with MAC message $MAC(k_{Alice}, y_{Alice})$ to Bob. Once Bob finishes the key correction using the compressed sensing theory, he uses his generated key k'_{Bob} to verify if both (1) $MAC(k_{Alice}, y_{Alice})$ and $MAC(k'_{Bob}, y_{Alice})$ and (2) y_{Alice} and Rk'_{Bob} are the same. If they are not both the same, it means (1) Alice's message is modified by an attacker; or (2) Bob is failure to generate an identical key. In either situation, Alice and Bob will need to first generate a new symmetric key. If the final generated keys are not agreed again, users will be informed to cease the key generation due to the existence of an attacker[3]. Furthermore, the universal hash function is applied to the agreed keys for privacy enhancement. Please note that the use of MAC and universal hash function does not necessarily make the proposed method immune to attacks. In Section 5, security analysis will be detailed to show the effectiveness of the proposed method against common attacks.

4 PERFORMANCE EVALUATION

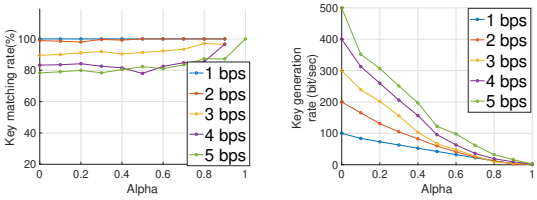
In this section, we will evaluate the proposed key generation method. In the following part, we will introduce experiment setup, performance metrics and analyse the performance of our proposed method. We further evaluate the effectiveness of the proposed method by using multiple users and hardware from other vendors. Moreover, we also show the randomness of the generated keys.

4.1 Experiment Setup

A speaker and a microphone from a Macbook Pro (Alice) are used for sending and capturing inaudible sound, and IMU from an iPhone (Bob) is used to provide motions. Please note that any computing devices with speakers and microphones can act as Alice and any mobile devices with IMUs, such as smart watches or wearable bands, can act as Bob. In Section 4.7, we will further use a smart TV and a wearable band to test the proposed method and show its effectiveness and generality in various hardware. In our implementation, the sample rate of the microphone is set as 44 kHz as the standard setting. The inaudible sound is played at 20 kHz. The bandwidth difference is gauged in 100 Hz. The sampling rate of IMU is set as 100 Hz as well. The volume of the MacBook Pro is set as the medium level. The hand with the iPhone conducts the gestures near the right-hand side of the keyboard near the microphone with a height of approximately 10-20 cm when waving the hand. We decrease the volume of the speaker from the highest level to limit the sensing range and thus the possibility of being influenced. We collect measurements by following steps in Figure 4 and keys are generated using the proposed method.

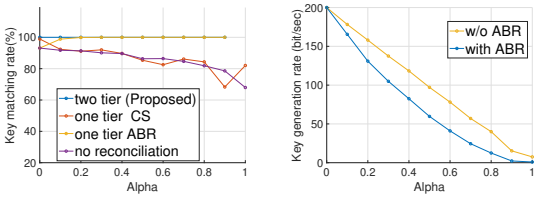
4.2 Goals, Metrics and Methodology

The goal of our evaluation is to show the performance and robustness of our proposed methods. In this section, we will use the following metrics to evaluate our proposed method and show the key generation performance: (1) **Key agreement rate**: the percentage of matching keys generated by Alice and Bob. (2) **Key generation rate**: the key generation speed that is measured by the number of generated bits per second (bit/sec).



(a) Key Matching Rate (b) Key Generation Rate

Figure 11: The effect of bit per sample (bps): (a) key matching rates with respect to the percentage of guard band. (b) key generation rates with respect to the percentage of guard band



(a) Key Matching Rate (b) Key Generation Rate

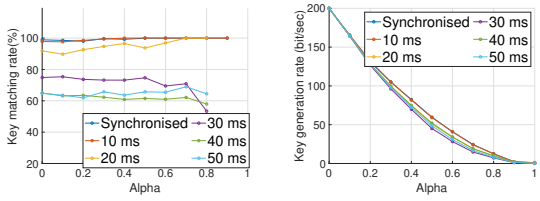
Figure 12: Comparison with the state-of-the-arts [38] (labelled as one tier ABR), [21] (labelled as one tier CS) and no reconciliation : (a) key matching rates with respect to the percentage of guard band. (b) key generation rates with respect to the percentage of guard band (with or without ambiguous bit removal (ABR))

4.3 Impact of Bit per Sample

We demonstrate the effect of bit per sample in this section. We use a window size of 100. The key matching rate and key generation rate using varying bits per sample (from 1 to 5) are shown in Figure 11. When using 1 or 2 bits per sample for key generation, the key matching rates can archive nearly 100% for different α values. The use of more bits per sample will increase the key generation rates. Figure 11 shows the key generation rate decreases with the increase of α because more samples are within guard bands and discarded. When using 2 bits per sample, the key matching rates are very close to that of the use of 1 bit per sample and its key generation rate is higher than that of 1 bit per sample. Therefore, we use 2 bits per sample as the default parameter for the multi-bit quantisation method. [21, 38, 44, 51] that realised IMU based or heartbeat-based key generation use 1 bits per sample with 50-200 Hz sampling rate. Therefore, the key generation rate (considering bit/sample $\times$ sampling rate) in the proposed method (2 bit/sample with 100 Hz sampling rate) is comparable with the state-of-the-arts.

4.4 Comparison with the State-of-the-arts: the Effect of Reconciliation Method

In this section, we will compare our proposed two-tier reconciliation method with the state-of-the-arts [21, 38][4]. [38] uses ambiguous bit removal (shown as "one-tier reconciliation ABR" in Figure 12) and [21] uses compressed sensing reconciliation (shown as "one-tier reconciliation CS" in Figure 12), respectively. We also show the performance without any reconciliation. In this experiment, we use the default parameters, i.e. the window size 100 and 2 bits per sample, for multi-bit quantisation. The performances when using different reconciliation methods are demonstrated in Figure 12. It can be seen from Figure 12(a) that the key matching rates are nearly 100% with the use of the two-tier reconciliation. The performance with only ambiguous bit removal is close to the use of the two-tier reconciliation with α more than 0.2. However, without any guard band ($\alpha = 0$), the key matching rate is approximately 10% less than that using the two-tier reconciliation method. When looking at the performance of no reconciliation and one tier with only compressed sensing reconciliation, their key matching rates drop significantly. Without any reconciliation, the key matching rate is approximately 90% when α is 0 and drops to approximately 65% when α is 1. It cannot improve the performance by only using compressed sensing reconciliation. This evidences that the use of the ambiguous bit removal tier can significantly improve the key matching rate. With the combination of the additional compressed sensing reconciliation tier, the key matching rates will be further increased. The advantage of our proposed method can be further seem in Figures 14 and 16[5], where the key matching rates are significantly increased up to approximately 20% and 40% compared with only using ambiguous bit removal only and without conciliation. Figure 12(b) shows the key generation rates with and without ambiguous bit removal. It is expected that the use of ambiguous bit removal will decrease the key generation rate. As shown in Figure 12(b), when α is 0.5, the key generation rate drops from 100 bit/sec to approximately 60 bit/sec.



(a) Key Matching Rate (b) Key Generation Rate

Figure 13: The effect of time synchronisation: (a) key matching rates with respect to the percentage of guard band. (b) key generation rates with respect to the percentage of guard band

4.5 Effect of Time Synchronisation

Figure 13 shows the key matching rates and key generation rates for synchronised measurements using our proposed method. We also manually shift these two signals to make them unsynchronised

[3]Due to the good performance shown in Section 4, it is hardly possible that generated keys are not agreed in two continuous instances without the interference of an attacker. The number of trial could be set higher for more failure tolerance of key generation.

[4][21, 38] do not use exact same sensors as the proposed system. We use their key generation methods with our collected measurements for fair comparison.

[5]Details are shown in Section 4.6 and Section 4.7

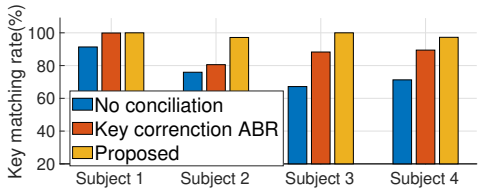


Figure 14: Key matching rate by different users

with a shift from 10 ms to 50 ms. Figure 13(a) shows key matching rates with the use of the proposed time synchronisation method and the manual shifts. It can be shown that, with our proposed time synchronisation method, the key matching rates are nearly 100%. With 10 ms shift, the key matching rates are not affected significantly. However, the key matching rate drops by approximately 10% when the shift 20 ms. With the time shift is 30 ms and above, the key matching rates are below 80%. As discussed in Section 3, the empirical experiments show the accuracy of NTP can achieve 25 ms with an NTP server in the local area network. This means our proposed method can improve the key matching rates by 10%-20% compared with the use of the common NTP time synchronisation method for symmetric key generation.

4.6 Effect of User Variance

In this section, four subjects (including the one who did the above experiments) test our proposed systems to show usability and feasibility. Different subjects[6] have different gesture waving amplitude, speed, heights and distances between the mobile phone and the laptop.

Figure 14 shows the key matching rates when four subjects use our proposed method i^{2}Key for key generation. All the users perform the "W" shape start detection successfully after guidance. In previous experiments, the key matching rate becomes 100% when α increases to 0.2 while the use of 0.2 as α can also maintain a good key generation rate. α is set as 0.2 in this experiment. The results by Subject 1 is from the above experiment. When no key conciliation is used, the key matching rate is 91.36%. The use of the ambiguous bit removal increases the key matching rate to 99.89%, and the further use of compressed sensing makes the key matching rate 100%. When Subject 2, Subject 3 and Subject 4 use the system without any key conciliation, they achieve lower key matching rates, i.e. 67.18%, 75.93% and 71.28%, respectively. When applying the ambiguous bit removal method, they can achieve 88.28%, 80.56% and 89.45%, respectively, and they are still not as good as that by Subject 1 with the first ambiguous bit removal tier. However, when looking at the performance of the proposed i^{2}Key, they all achieve competitively nearly 100% key matching rates. The key generation rates for these four subjects are 130.89 bit/sec, 130.66 bit/sec, 123.11 bit/sec, and 130 bit/sec, respectively. This experiment confirms that (1) the use of our proposed i^{2}Key can increase the key matching rate significantly; (2) the proposed i^{2}Key can be used by different users, which verifies its usability and practicality.

4.7 Effect of Device Variance

[6]The profiles (Gender, Age) of users: User 1, Male, Age 35; User 2, Female, Age 33; User 3, Female, Age 58; User 4, Male, Age 59.

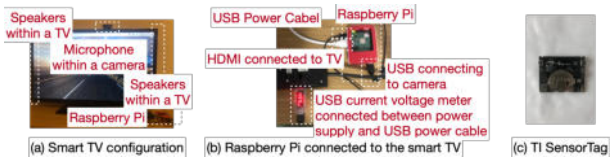


Figure 15: (a) The configuration of a smart TV (a normal TV connected to a Raspberry Pi). A microphone and a speaker are equipped. (b) the configuration of a Raspberry Pi connected to the TV. (c) A TI Sensortag equipped with an IMU.

To further validate the feasibility and robustness of the proposed method, we use two additional devices, i.e. Raspberry Pi [30] with a microphone and speaker and a TI CC2650 SensorTag [3] with an IMU, which are from different hardware vendors (rather than Macbook Pro and iPhone from Apple Inc.) to collect inaudible sound and IMU measurements, respectively. We use Raspberry Pi to set up a smart TV and use our proposed method to generate symmetric keys between the smart TV and a wearable band made from SensorTag with an IMU. Figure 15 shows the setup of the smart TV, and Raspberry Pi is powered by a 5V USB power source. One USB voltage and current meter is used between the Raspberry Pi and the power source to measure the energy consumption. More details about the energy consumption will be shown in Section 4.9. An HDMI cable is used to connect a TV to Raspberry Pi, so that the speaker in the TV can be used in our system. A Logitech camera [1] is connected to Raspberry Pi through a USB cable with its integrated microphone. The CC2650 SensorTag is a low power IoT device with an IMU. We use Contiki OS [10] to program a SensorTag to let it have 128 Hz sensing frequency, and then downsample the measurements from the SensorTag to 100 Hz[7].

Because we have two devices for sensing inaudible sound (i.e. a Macbook Pro and a smart TV) and two devices for collecting IMU measurements (i.e. an iPhone and a Sensortag), we show the performance in four combinations as shown in Table 1. In this experiment, we also use α 0.2 as the previous experiments. The results of Experiment 1 are again from the previous experiment with the Macbook Pro and iPhone, which achieves 91.36%, 99.89% and 100% key matching rates without any reconciliation, with the first tier and with two tiers in our proposed i^{2}Key. When using other combinations in Experiments 2, 3, and 4 without key reconciliation, the key matching rates are much lower, i.e. 64.84%, 65.62%, and 78.90%, respectively. This is because (1) the IMU in Sensortag lacks calibration compared with iPhone with a required calibration phase in many applications; (2) the distance between the microphone and speaker of the smart TV is farther than those in Macbook Pro. These two facts bring more noise and interference to the system, which results in the performance decrease without any key reconciliation. The use of the first ambiguous bit removal tier increases their key matching rates to 84.37%, 82.02%, and 85.93%, respectively. The performances are enhanced significantly by using our proposed system when the key matching rates can all achieve 100%, and their corresponding key generation rates for these four experiments are all more than 130.89 bit/sec. This experiment again confirms that

[7]The tick interrupt is used in the implementation using Contiki OS, so the sensing frequency cannot be set to 100 Hz directly

Table 1: Experiment Setup Using Different Devices

	Inaudible Sensing	IMU
Exp 1	Macbook Pro	iPhone
Exp 2	Macbook Pro	SensorTag
Exp 3	Smart TV	iPhone
Exp 4	Smart TV	SensorTag

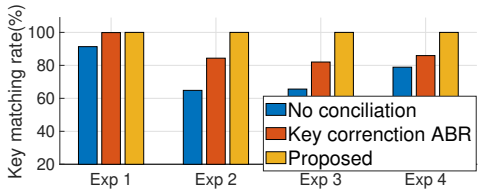


Figure 16: Key matching rate in different experiment setups

(1) the significant performance enhancement thanks to the use of our proposed i²Key; (2) its feasibility to use that to any hardware from other vendors; and (3) its capacity for large scale deployments.

4.8 Randomness of Keys

In this section, we test the randomness of the generated 256-bit keys. NIST Statistical Test Suite [34] is used to test the randomness. NIST returns p-values as results for randomness indications, where p-values above 0.01 demonstrate sufficient randomness of tested keys. p-values and entropy are shown in Table 2, which shows keys generated from our proposed methods are highly random.

Table 2: Results of NIST test and entropy.

NIST TEST	p-value
Monobit test	0.237
Frequency within block test	0.384
Runs test	0.119
Longest run ones in a block test	0.213
DFT test	0.916
Non overlapping template matching test	0.999
Serial test	0.181
Approximate entropy test	0.182
Cumulative sums test	0.143
Random excursion test	0.514
Random excursion variant test	0.308
Entropy	0.790

4.9 System Implementation

To show the feasibility of i²Key in IoT devices and compare its energy consumption with existing solutions, we implement and test one prototype of i²Key on one Raspberry Pi 3B+ used in Section 4.7 as a smart TV box (shown in Figure 16).

Raspberry Pi 3B+ is equipped with a 1.4GHz CPU, and we use Debian Linux based Raspberry Pi OS. The prototype was implemented in Python. ℓ_1 Homotopy [6] is employed as its efficiency in embedded systems [43]. The generated key length is 256 bits.

Table 3: Processing Time and Energy Consumption

	i²Key	RSA
Processing Time (ms)	811	92,161
Energy Consumption (mJ) Key Generation	202	23,040
Energy Consumption (mJ) with Sensor	4568	-

In this section, we use the popular public key cryptography method RSA as the benchmark to compare with our proposed method. We use RSA to generate one 256 bit key, the same length as the proposed method. The key generation of RSA is also implemented in Python using RSA library[8] for fair comparison. We calculate the processing time and energy consumption for both i²Key and RSA in Python. Figure 16 shows the setup of the implemented system. We execute both methods 100 times and calculate the average processing time. The current is measured by a USB voltage and current power meter connected between the power source and Raspberry Pi (as shown in Figure 16(b)). Ohm's law is then used to calculate the energy consumption. Table 3 shows the results of processing time and energy consumption of the proposed i²Key and RSA. i²Key needs 811 ms to generate a 256 bit symmetric key, while RSA needs 92,161 ms to generate the same length of the key. It can be seen that the proposed key generation method i²Key in the key generation stage costs only no more than 1/100 processing time and energy compared with the popular RSA. According to the measured processing time for each step, in our method, most of time is consumed by the necessary encoding method for cross-sensor measurements (i.e. peak detection and interpolation), while the use of two tiers can cost no more than 1/50 processing time. In other words, we use approximate 50 ms out of 811 ms for two tier reconciliation compared with the state-of-the-art with only ambiguous bit removal or only compressed sensing based method, but enhance the key matching rates significantly.

Sensors also cost energy. The power consumption of an IMU chip MPU9250 (used in Sensortag) is 0.013 W [2], and the power consumption of a microphone is no more than 0.05 W [43]. The power consumption of a speaker varies, and the miniature speaker can cost as low as 2 W [4]. High specification speakers cost more energy but they are usually powered by a separate independent power source (such as a power cable). The key generation rate is no less than 120 bit/sec using 2 bit per sample and α 0.2 in our experiments, which takes no more than 2.13 seconds (256 bit / (120 bit/sec)) to collect measurements to generate 256 bits. Therefore, When considering the power of used sensors, the energy cost from IMU is 10.65 mJ and that from a pair of a microphone and a speaker is 4366 mJ. Therefore, the total energy consumption is no more than 4568 mJ (4366 mJ + 202 mJ). It means that the proposed method still costs no more than 1/3 of energy compared with RSA with the consideration of the sensor energy cost. Please note in many situations used sensors is on even when the proposed method is not used, so there is no additional energy cost in that situation. This also confirms that symmetric keys are more suitable for IoT devices due to the limited energy and processing power in IoT devices.

[8] https://pypi.org/project/rsa/

5 SECURITY ANALYSIS

Eavesdropping and imitating are common attack models, which are illustrated in Figure 17. Eve has an adversary IoT device (either equipped with a speaker and microphone pair or an IMU) that has a good knowledge of the technical and implementation details of the used key generation mechanism used by legal devices Alice and Bob. In this paper, we consider four common attacks.

Imitating attack (Attack 1 and Attack 2): When Bob waves his hand to interfere with inaudible sound played by Alice and generate symmetric keys, Eve observes Bob's hand movements and its relative location to Alice's speaker and microphone. After Bob finishes his movements, Eva wears an IMU on one wrist and tries to imitate Bob's movement over her own device with a speaker (Attack 1) and microphone (Attack 2). Keys will be generated from Eve's IMU measurements and/or inaudible sound. When conducting Imitating Attack 1 and Imitating Attack 2, Eve can receive exchanged messages between Alice or Bob for time synchronisation and reconciliation as we assume that the attacker Eve has the full knowledge of the communication and reconciliation mechanism of the legal devices.

Eavesdropping attack (Attack 3): Intending to have the same symmetric key as Alice and Bob, Eve eavesdrops on messages transmitted in wireless communication between Alice and Bob and tries to use the known key generation mechanism to recover their generated keys. When conducting this attack, Eve knows the random metric used by Alice and Bob for compressed sensing. When conducting Attack 3, the initial keys from the attacker are all the same (all 0's or all 1's). Conducting Attack 1, Attack 2 and Attack 3, the attacker could be in the vicinity to observe.

Eavesdropping attack on measurements (Attack 4): Eve uses a camera to record Alice's movements and analyse the recorded video to obtain accurate measurements. To avoid her exposure, Eve stays in a distant spot beyond Alice and Bob's communication range when recording Alice's movement. Because sensor measurements are not sent in wireless communication, one way for eavesdropping attack to sensor measurements is to use one camera and sophisticated computer vision method to recover the movements. Because the recovery of the movements using a computer vision method is out of the scope of this paper, we assume the attacker can obtain the perfect IMU measurements as IMUs from Bob[9] and would use that to generate keys. The use of a camera by an attacker nearby is noticeable and impractical. Like when using the password, the legal user can always cover conducted gestures when paring devices in a crowded area. Therefore, we assume the attacker has to use the camera in a distant spot, which makes it beyond the wireless communication range of legal devices[10]. In other words, when conducting Attack 4, the attacker, Eve, cannot conduct any reconciliation with Alice. Once Eve finishes generating keys using Attack 4, she can bring one device with her to try to decrypt messages in the vicinity of Alice and Bob. Please note that we do not consider the situation that both measurements are eavesdropped and Eva is within the communication range of legal devices where Eve can receive exchanged messages for reconciliation as a legal device and discover the key. One user in a crowded environment is suggested to cover their hands' movements using the other hand when initiating a new connection with the other devices to avoid attackers obtaining keys, like a similar and common precautionary method of inputting passwords.

The eavesdropping attacks can be seen as passive Man-In-the-Middle attack. To avoid the awareness of Alice and Bob, Eve does not interrupt the key generation process or intend to conduct Denial-of-Service (DoS) attack. This assumption has also been utilised in recent literature [20, 44, 46, 51].

Figure 18 shows key matching rates and key generation rates under these attacks. Compared with nearly 100% key matching rates with legal devices, key matching rates of Eve's can only achieve approximately 65% when conducting Attack 1, Attack 2 and Attack 3. When conducting Attack 1 and Attack 2, this is not easy to strictly mimick the other users' gestures, which generates non-identical measurements. When conducting Attack 3, too many bits (approximately 50%) are different from that generated the legal devices. Since the bit difference between the attackers and the legal devices are not sparse, the system cannot recover legal key correctly using compressed sensing theory. When Eve conducts Attack 4, once Eve eavesdrops on the sensor measurements using a camera along with a sophisticated method, Eve tries to generate keys using the measurements but without any reconciliation since she is beyond the wireless communicating range. with α is 0, the no ambiguous bit will be removed from Alice, so the similarity of measurements from Alice and Eve will make the key matching rate approximately 85%. However, with the increase of α , the attacker cannot generate matched keys as she does not know the valid bits from inaudible sound due to the lack of reconciliation. When looking at the key generation rates in Figure 18(b), the key generation rate of Attack 4 is higher than others. However, it still cannot achieve the competitive key generation rate.

Figure 19 shows the false reject rate of the proposed method and the false accept rates from all kinds of attacks. Using our proposed system, the false accept rates from all kinds of attacks are all 0. This means that our proposed method can effectively reject the common attacks. Different from [38] with the use of a threshold 70% of match successful rate to decide the key effectiveness, when using our proposed, only two keys are totally matched, Alice and Bob can use that key. Even with this more strict condition, the false reject rate is usually less than 5%. Although the attacker can achieve approximately 65%~80% key matching rate, the probability of generating a same 256-bit key is as low as 0.65^{256}~0.8^{256} ($1.28e^{-48}$~$1.55e^{-25}$). The 0 false accept rates in Figure 19 confirm the fact. We can realise this thanks to the high key matching rate compared with the state-of-the-arts, which makes our system more resilience to adversaries. This further confirms that the proposed i^{2}Key is robust to the common imitating and eavesdropping attacks.

6 RELATED WORKS

By taking advantage of the ambient or active interference from the surrounding environment, symmetric key generation methods have been used for IoT Device-to-Device communication. In this section, we show the details of these methods.

[9]It is more complicated to generate the inaudible measurements because it needs to consider the relevant distance among the hand, the microphone, and the speaker.
[10]The legal device can limit the wireless communication range by limiting the transmission power.

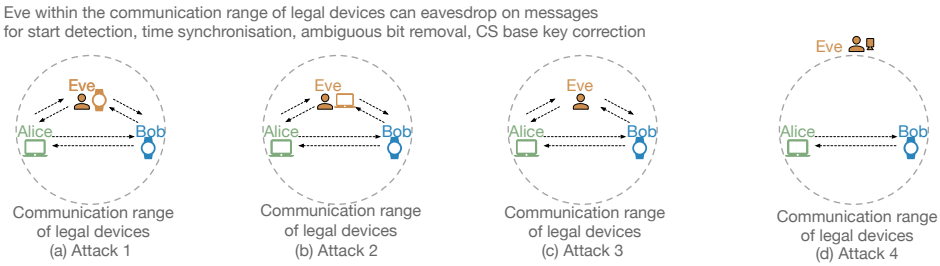
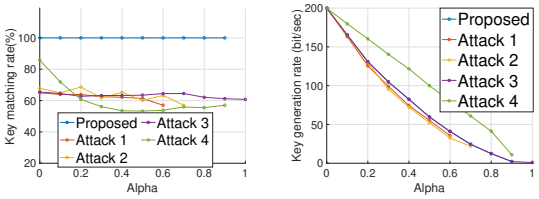


Figure 17: Attack models



(a) Key Matching Rate (b) Key Generation Rate

Figure 18: Attacks Analysis: (a) key matching rates with respect to the percentage of guard band. (b) key generation rates with respect to the percentage of guard band

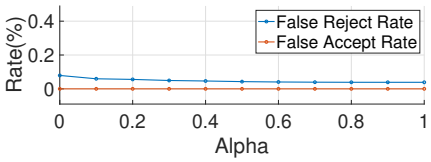


Figure 19: The false reject rates of the proposed method and the false accept rate from all kinds of attacks (they are all 0) using different α

Similar observations can be collected by devices in the vicinity, which is employed to produce symmetric keys. In the literature, there are several types of used measurements for symmetric key generation, such as radio signal [25, 42, 46, 55], audio [36] and surrounding context [28]. Specifically, [42] uses a public cryptographic key exchanging method, named Diffie-Hellman protocol, on surrounding radio observations to verify the physical proximity and pair devices for communications. The use of the Diffie-Hellman protocol is removed in [25, 55], which suffered from reduced key generation rate, though. [28] considered surrounding context for key generation, but it needed a relatively long duration for the collection of the required information.

The acoustic signal has already been utilised for device authentication and key generation [7, 14, 19, 24, 36, 37, 47]. In [24], signal propagation of inaudible sound has been analysed and audio channel taps were used to generate keys. The reciprocity from sound pressure level is used for the key generation as well in [7]. Han et al. [14] used heterogeneous IoT devices, including microphones, in a house based on different sensory data for authentication. Recently, acoustic signal is also employed to attack IoT devices [8, 33, 41, 54]. Additionally, acoustic signals have been used for key generation for mobile devices [7, 14, 19, 24, 36, 37, 47, 50]. Different from their solutions, additional to pairing different devices, we directly use cross-sensors for symmetric key generation.

An IMU, as a common sensor in mobile and wearable devices, is also a popular choice to generate keys. The simultaneous motions detected by different wearable IMUs are studied for this aim. [15, 27] secured the IMU measurements for authentication. IMUs were also used for continuous key generation for body wearable devices and device paring [35, 38, 40, 51].

Due to the ubiquity of the wireless radio equipped in the IoT devices, the wireless radio information has been taken advantage of for symmetric key generation. The popular solutions include ZigBee based [16, 23], WiFi based [22, 26, 45, 46], and 5G based [17]. Received Signal Strength Indicator (RSSI) is popular characteristics for the key generation with a fallback of low key generation. Channel State Information (CSI) from Physical layer based method [46] increased the key generation rate due to its higher resolution compared with RSSI. Other new sensors are also used for the same purpose, such as electromyogram sensors [52] and electrode sensors [31], and heartbeat sensors [21, 32, 48]. FastZIP [12] has designed a fast device paring mechanism, while our proposed method focuses on the cross-sensor key generation.

[5] designed a motion based time synchronisation method between a camera and an IMU using a particle filter. [11] mimicked GPS clocks for the time synchronisation between Lidar and an IMU. Different from their method, our proposed system proposes a simple yet effective "W" based time synchronisation method.

7 CONCLUSION

We are the first to perform a study on the feasibility of the use of cross-sensors for key generation mechanism i²Key. Signal processing methods have been applied on both IMU measurements and inaudible sound measurements to enable this cross-sensor key generation. A novel time synchronisation method and compressed sensing based key reconciliation method have been investigated in the paper. Extensive evaluations have been conducted to show the efficacy of the proposed method i²Key with high key matching rates. Furthermore, we verify the randomness of generated keys from the proposed method and perform security analysis to show the robustness of the proposed method i²Key.

ACKNOWLEDGMENTS

We thank the anonymous shepherd and reviewers for their helpful and constructive feedback on this paper.

REFERENCES

[1] [n.d.]. Logitech C270 HD Webcam, 720p Video with Noise Reducing Mic. https://www.logitech.com/en-gb/products/webcams/c270-hd-webcam.960-001063.html?crid=34

[2] [n.d.]. MPU-9250 Product Specification. https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf

[3] [n.d.]. SensorTag. https://www.ti.com/tool/CC2650STK

[4] [n.d.]. Uxcell a15112300ux1550 2W 40mm Diameter 8 Ohm Internal Mini Magnet Speaker Loudspeaker. https://www.amazon.co.uk/a15112300ux1550-Diameter-Internal-Speaker-Loudspeaker

[5] Peter Aerts and Eric Demeester. 2019. Time Synchronisation of Low-cost Camera Images with IMU Data based on Similar Motion.. In *ICINCO (2)*. 292–299.

[6] M Salman Asif and Justin Romberg. 2014. Sparse Recovery of Streaming Signals Using ℓ_1-Homotopy. *IEEE Transactions on Signal Processing* 62, 16 (2014), 4209–4223.

[7] Dania Qara Bala and Bhaskaran Raman. 2020. PHY-Based Key Agreement Scheme using Audio Networking. In *COMSNETS*. IEEE, 129–136.

[8] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden voice commands. In *USENIX Security Symposium*. 513–530.

[9] David L Donoho. 2006. Compressed sensing. *IEEE Transactions on information theory* 52, 4 (2006), 1289–1306.

[10] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. 2004. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *IEEE LCN*. IEEE, 455–462.

[11] Marsel Faizullin, Anastasiia Kornilova, and Gonzalo Ferrer. 2021. Open-Source LiDAR Time Synchronization System by Mimicking GPS-clock. *arXiv preprint arXiv:2107.02625* (2021).

[12] Mikhail Fomichev, Julia Hesse, Lars Almon, Timm Lippert, Jun Han, and Matthias Hollick. 2021. FastZIP: faster and more secure zero-interaction pairing. In *MobiSys*. 440–452.

[13] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Soundwave: using the doppler effect to sense gestures. In *CHI*. 1911–1914.

[14] Jun Han, Albert Jin Chung, Manal Kumar Sinha, Madhumitha Harishankar, Shijia Pan, Hae Young Noh, Pei Zhang, and Patrick Tague. 2018. Do you feel what I hear? Enabling autonomous IoT device pairing using different sensor types. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 836–852.

[15] Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W Gellersen. 2001. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Ubicomp*. Springer, 116–122.

[16] Suman Jana, Sriram Nandha Premnath, Mike Clark, Sneha K Kasera, Neal Patwari, and Srikanth V Krishnamurthy. 2009. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Mobicom*. ACM, 321–332.

[17] Long Jiao, Ning Wang, and Kai Zeng. 2018. Secret Beam: Robust Secret Key Agreement for mmWave Massive MIMO 5G Communication. In *IEEE GLOBECOM*. IEEE, 1–6.

[18] Rong Jin, Liu Shi, Kai Zeng, Amit Pande, and Prasant Mohapatra. 2015. Magpairing: Pairing smartphones in close proximity using magnetometers. *IEEE Transactions on Information Forensics and Security* 11, 6 (2015), 1306–1320.

[19] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. Sound-proof: usable two-factor authentication based on ambient sound. In *24th USENIX Security Symposium*. 483–498.

[20] Qi Lin, Weitao Xu, Guohao Lan, Yesheng Cui, Hong Jia, Wen Hu, Mahbub Hassan, and Aruna Seneviratne. 2020. KEHKey: Kinetic Energy Harvester-based Authentication and Key Generation for Body Area Network. *ACM IMWUT* 4, 1 (2020), 1–26.

[21] Qi Lin, Weitao Xu, Jun Liu, Abdelwahed Khamis, Wen Hu, Mahbub Hassan, and Aruna Seneviratne. 2019. H2B: Heartbeat-based secret key generation using piezo vibration sensors. In *IPSN*. 265–276.

[22] Hongbo Liu, Yang Wang, Jie Yang, and Yingying Chen. 2013. Fast and practical secret key extraction by exploiting channel response. In *INFOCOM*. IEEE, 3048–3056.

[23] Hongbo Liu, Jie Yang, Yan Wang, and Yingying Chen. 2012. Collaborative secret key extraction leveraging received signal strength in mobile wireless networks. In *INFOCOM*. IEEE, 927–935.

[24] Youjing Lu, Fan Wu, Shaojie Tang, Linghe Kong, and Guihai Chen. 2019. FREE: A Fast and Robust Key Extraction Mechanism via Inaudible Acoustic Signal. In *Mobihoc*. ACM, 311–320.

[25] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. 2011. Proximate: proximity-based secure pairing using ambient wireless signals. In *Mobisys*. ACM, 211–224.

[26] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. 2008. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *Mobicom*. ACM, 128–139.

[27] 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.

[28] Markus Miettinen, N Asokan, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and Majid Sobhani. 2014. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *CCS*. ACM, 880–891.

[29] David Mills. 1992. *RFC1305: Network Time Protocol (Version 3) Specification, Implementation*. RFC Editor.

[30] Raspberry Pi. [n.d.]. Raspberry Pi 3 Model B. https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

[31] Marc Roeschlin, Ivan Martinovic, and Kasper Bonne Rasmussen. 2018. Device Pairing at the Touch of an Electrode.. In *NDSS*, Vol. 18. 18–21.

[32] Masoud Rostami, Ari Juels, and Farinaz Koushanfar. 2013. Heart-to-heart (H2H): authentication for implanted medical devices. In *CCS*. ACM, 1099–1112.

[33] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. 2017. Backdoor: Making microphones hear inaudible sounds. In *Mobisys*. ACM, 2–14.

[34] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. 2001. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Technical Report. Booz-Allen and Hamilton Inc Mclean Va.

[35] Dominik Schürmann, Arne Brüsch, Stephan Sigg, and Lars Wolf. 2017. BANDANA—Body area network device-to-device authentication using natural gAit. In *PerCom*. IEEE, 190–196.

[36] Dominik Schürmann and Stephan Sigg. 2011. Secure communication based on ambient audio. *IEEE Transactions on Mobile Computing* 12, 2 (2011), 358–370.

[37] Jiacheng Shang and Jie Wu. [n.d.]. AudioKey: a usable device pairing system using audio signals on smartwatches. ([n. d.]).

[38] 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 Transactions on Sensor Networks (TOSN)* 16, 2 (2020), 1–22.

[39] 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 *PerCom*. IEEE, 1–10.

[40] Yingnan Sun, Charence Wong, Guang-Zhong Yang, and Benny Lo. 2017. Secure key generation using gait features for body sensor networks. In *BSN*. IEEE, 206–210.

[41] Chen Tao, Longfei Shangguan, Li Zhenjiang, and Jamieson Kyle. 2020. Metamorph: Injecting Inaudible Commands into Over-the-air Voice Controlled Systems. In *NDSS*.

[42] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal De Lara. 2007. Amigo: Proximity-based authentication of mobile devices. In *Ubicomp*. Springer, 253–270.

[43] Bo Wei, Mingrui Yang, Yiran Shen, Rajib Rana, Chun Tung Chou, and Wen Hu. 2013. Real-time classification via sparse representation in acoustic sensor networks. In *SenSys*. 1–14.

[44] Yuezhong Wu, Qi Lin, Hong Jia, Mahbub Hassan, and Wen Hu. 2020. AutoKey: Using Autoencoder to Speed Up Gait-based Key Generation in Body Area Networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–23.

[45] Wei Xi, Xiang-Yang Li, Chen Qian, Jinsong Han, Shaojie Tang, Jizhong Zhao, and Kun Zhao. 2014. KEEP: Fast secret key extraction protocol for D2D communication. In *IWQoS*. IEEE, 350–359.

[46] Wei Xi, Chen Qian, Jinsong Han, Kun Zhao, Sheng Zhong, Xiang-Yang Li, and Jizhong Zhao. 2016. Instant and robust authentication and key agreement among mobile devices. In *CCS*. ACM, 616–627.

[47] Pengjin Xie, Jingchao Feng, Zhichao Cao, and Jiliang Wang. 2018. GeneWave: Fast authentication and key agreement on commodity mobile devices. *IEEE/ACM Transactions on Networking (TON)* 26, 4 (2018), 1688–1700.

[48] Fengyuan Xu, Zhengrui Qin, Chiu C Tan, Baosheng Wang, and Qun Li. 2011. IMDGuard: Securing implantable medical devices with the external wearable guardian. In *INFOCOM*. IEEE, 1862–1870.

[49] Weitao Xu, Sanjay Jha, and Wen Hu. 2018. Lora-key: Secure key generation system for lora-based network. *IEEE Internet of Things Journal* (2018).

[50] Weitao Xu, Zhenjiang Li, Wanli Xue, Xiaotong Yu, Bo Wei, Jia Wang, Chengwen Luo, Wei Li, and Albert Y Zomaya. 2021. InaudibleKey: Generic Inaudible Acoustic Signal based Key Agreement Protocol for Mobile Devices. In *IPSN 2021*. 106–118.

[51] 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 *IPSN*. IEEE, 1–12.

[52] Lin Yang, Wei Wang, and Qian Zhang. 2016. Secret from muscle: Enabling secure pairing with electromyography. In *Sensys*. ACM, 28–41.

[53] Kai Zeng, Daniel Wu, An Chan, and Prasant Mohapatra. 2010. Exploiting multiple-antenna diversity for shared secret key generation in wireless networks. In *INFOCOM*. IEEE, 1–9.

[54] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *CCS*. ACM, 103–117.

[55] Jiansong Zhang, Zeyu Wang, Zhice Yang, and Qian Zhang. 2017. Proximity based IoT device authentication. In *INFOCOM*. IEEE, 1–9.