Lin *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:13
https://doi.org/10.1186/s13638-019-1634-7

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Efficient physical layer key generation technique in wireless communications

Rushan Lin[1,2], Li Xu[1,2*], He Fang[3] and Chuan Huang[1,2]

Abstract

Wireless communications between two devices can be protected by secret keys. However, existing key generation schemes suffer from the high bit disagreement rate and low bit generation rate. In this paper, we propose an efficient physical layer key generation scheme by exploring the Received Signal Strength (RSS) of signals. In order to reduce the high mismatch rate of the measurements and to increase the key generation rate, a pair of transmitter and receiver separately apply adaptive quantization algorithm for quantifying the measurements. Then, we implement a randomness extractor to further increase key generation rate and ensure randomness of generated of keys. Several real-world experiments are implemented to verify the effectiveness of the proposed scheme. The results show that compared with the other related schemes, our scheme performs better in bit generation rate, bit disagreement rate, and randomness.

Keywords: Received signal strength, Key generation, Physical layer security

1 Introduction

In order to guarantee the confidentiality, integrity, and authenticity of communication, secret keys should be established to protect the wireless network. Currently, the public key cryptography has been well investigated in [1]. The crux of the implementation of public key cryptography is the key generation and exchange mechanism, which usually relies on the difficulty of certain mathematical problems, such as inverse operations, integer factorization, and discrete logarithm. However, conventional mechanisms may not be suitable for certain scenarios (e.g., sensor networks, cognitive radio networks, and vehicular networks) because low-energy devices cannot afford the high resource overheads. As an alternative method, the physical layer characteristics [2, 3]-based key generation has been designed. Different from traditional public key cryptography, the physical layer key generation schemes do not require expensive computation. It is possible to achieve information-theoretic security because the confidentiality of the generated key does not depend on the hardness of the computational problem but relies on the physical characteristics of the wireless fading channels [4]. More importantly, the physical layer key generation technique can dynamically establish a shared key between a pair of transceivers through exploring the channel reciprocity and variations. Depending on the channel reciprocity, transceivers can measure the statistical related channel state and extract the key. In contrast, time-varying reflects the changes in the channel state and affects the efficiency of key generation. The physical layer information measurement of the wireless channel can be collected using the channel state information (CSI) [5–9], received signal strength (RSS) [10–15], or phase [16–18]. Compared with the CSI and phase, the RSS-based key generation mechanism can be directly applied to the off-the-shelf wireless devices without any hardware modification. Therefore, in the physical layer key generation schemes, the RSS is the most widely used measurements to generate shared keys.

Although there have been many research works for key extraction based on RSS [10–15], unfortunately, existing methods may also suffer from low bit generation rate (BGR) and high bit disagreement rate (BDR). Note that the bit generation rate refers to the number of bits that can be generated each RSS sample. The bit disagreement rate is the proportion of the number of inconsistent bits

*Correspondence: xuli@fjnu.edu.cn
[1] College of Mathematics and Informatics, Fujian Normal University, Fuzhou, Fujian, China
[2] Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou, Fujian, China
Full list of author information is available at the end of the article

in the entire key length. This metric is the raw rate of quantization and encoding. Typically, it does not consider the leaked information during information reconciliation. All in all, the RSS-based key generation schemes usually fail to achieve high BGR with low BDR, which will cause excessive delay in key establishment.

To alleviate these problems, we propose a key generation scheme based on RSS, which can be utilized to efficiently generate shared keys between transceivers even if there are inconsistencies in their measurements. Specifically, our scheme composes of five components: channel probing, preprocessing, quantization & encoding, information reconciliation and randomness extractor and privacy amplification. In our scheme, the transceivers collect the RSS measurements during the channel probing and preprocess the measurements using the wavelet shrinkage based on rigrsure. A new quantization algorithm is developed to divide the quantization guard-band by different conditions for each interval, and it adjusts the correction of the measurements in the guard-band. Additionally, the information reconciliation is implemented in order to correct bit errors, and the randomness extractor is used to ensure the randomness of the generated key.

The contributions of this paper are summarized as follows:

- We propose a modified channel quantization with single guard-band (MCQSG) algorithm, which divides the quantization guard-band by different conditions for each interval. We choose to correct the values in the corresponding guard-band instead of directly discarding the measurements. In this way, we can resolve the contradiction between BGR and BDR on the quantization method. This allows our scheme to be used to generate shared keys between transceivers even in the complex environment where measurements are asymmetrical on the transmitter and receiver sides.
- A randomness extractor is proposed to ensure the randomness while improving bit generation rate. In detail, the randomness of keys is guaranteed by preventing the occurrence of multiple consecutive zeros or ones. Furthermore, the bit generation rate can be improved due to the increase in code length when running the randomness extractor. This scheme improves the bit generation rate under the premise of ensuring randomness, which greatly increases the speed of establishing keys between transceivers.
- We verify and evaluate the proposed scheme by collecting data from experiments and comparing our scheme with some existing schemes. The results of experiments demonstrate that the proposed schemes can be used to generate keys in the presence of too many differences measurements caused by complex, noisy environments. The results also show that the proposed scheme outperforms in terms of the bit generation rate and bit disagreement rate.

The rest of this paper is organized as follows. Related works are reviewed in section 2. Section 3 briefly introduces the key generation model. Section 4 presents the proposed scheme. Section 5 analyzes the performance of the proposed scheme in terms of following three metrics: bit generation rate, bit disagreement rate, and randomness. Finally, section 6 concludes the paper.

2 Related work

In recent years, many key extraction schemes have been proposed by exploiting different channel state [10–13, 19–24]. For instance, Mathur et al. [10] proposed a level-crossings key extraction algorithm that preserves only one bit from m consecutive 1s or 0s and discards other repeated $m - 1$ bits. However, it achieved low bit disagreement rate while sacrificing the bit generation rate. To solve this problem, Zhu et al. in [11] applied canonical correlation analysis to obtain the optimal weight coefficient of the sliding smoothing filter in order to improve the correlation of the measurements of transceivers and accelerate the bit generation rate. While Suman et al. proposed an adaptive secret bit generation (ASBG) scheme in [12], which was an improved version of the scheme [10]. Different from the Mathur quantizer, the measurements in [12] were divided into multiple blocks, and the quantizer extracted a bit from each measurement of each block. However, it depended on the further steps to eliminate the effects of correlated bits. Furthermore, this paper also introduced an adaptive secret multi-bit key generation scheme using the Gray code, which not only greatly improved the bit generation rate but also increased the bit disagreement rate. Li et al. [13] put forward the RSS-based high-bit rate, consistent, and random key extraction for VANETs scheme. Specifically, it used an inconsistency removal method to remove the inconsistent measurements between the two communication parties. This scheme was able to achieve 0-bit disagreement, but the bit generation rate becomes too low. They also proposed an n-dimension quantization method to reuse each RSS n times. As a result, it greatly compensated for the problem of low bit generation rate due to inconsistency removal.

In addition to the quantizer design, solving the boundary problem is also crux to achieve low bit disagreement rate in quantization-based scheme. In fact, channel measurements at transceivers usually lie close to the quantization boundaries, causing the two measurements to fall into the different quantization intervals. Hong et al. [19] first proposed utilizing sample and quantizer selection techniques to avoid this problem. And this problem can also be addressed by using channel quantization with guard-band (CQG) scheme [20] and guard band (GB) scheme [8], which set the quantized guard band at the quantization threshold to discard the measurements falling inside. They can effectively reduce the bit disagreement rate. The larger the guard-band setting, the lower the initial bit disagreement rate. However, the size of the guard-band seriously affected the bit generation rate. Because the CQG and GB reduced the initial bit disagreement rate by deleting measurements within the guard-band. Wallace and Sharma [21] and Patwari et al. [22] proposed channel quantization altering algorithm and multi-bit adaptive quantization algorithm, respectively. The core of the two algorithms was that the legal transmitter sent quantization error information on the common channel. Then, the receiver adaptively adjusted its quantization threshold according to the received information to reduce the BDR. In addition, a few research works have been proposed in the presence of channel estimation error and quantization error. For example, Zhang et al. [23, 24] evaluated the influence of channel estimation error caused by non-simultaneous measurement of the transceivers in the real environment.

The above mainly described the key establishment process between two nodes. In fact, these scenarios where a single user needs to establish keys with multiple users are quite common, e.g., roadside units (RSUs) with vehicles [25]. Alhasanat et al. [26] proposed a novel physical layer key distribution mechanism for IoT networks. Each node could decode the broadcast signal of the central signal through independent signal modulation type and order. This ingenious design allowed each node to simultaneously obtain a key of the desired length. However, when the network lacked a common channel, the central node could not directly broadcast signals to other nodes (e.g., cognitive radio networks involved channel application, allocation, and access). This is the limitation of the scheme. Our scheme is aimed at generating keys between transceivers, which is applicable to all networks environments.

3 Key generation model

This section briefly introduces the adversary model, the system model, the key generation process, and wavelet shrinkage based on rigrsure.

3.1 Adversary model

We consider a 3-node eavesdropping model. Alice and Bob are legitimate transceivers. Eve acts as a passive attacker who wants to eavesdrop confidential information between Alice and Bob. This model is shown in Fig. 1. Additionally, we assume that the key generation protocol is public and it can be accessed by the eavesdropper Eve. In order to implement key generation, the transmitter and receiver need to exchange information over the wireless channels. The purpose of the passive adversary Eve is to obtain the key generated between legitimate transceivers by making use of the eavesdropped information. We assume that the distances between the passive adversary and the transceivers are at least half the wavelength (i.e., $\lambda/2$). According to [10], when the eavesdropper is far away from the legal devices, the channel gain between the eavesdropper and the legitimate transceivers are independent. Therefore, the eavesdropper cannot obtain any useful channel gain between transceivers by making use of his own channel observations.

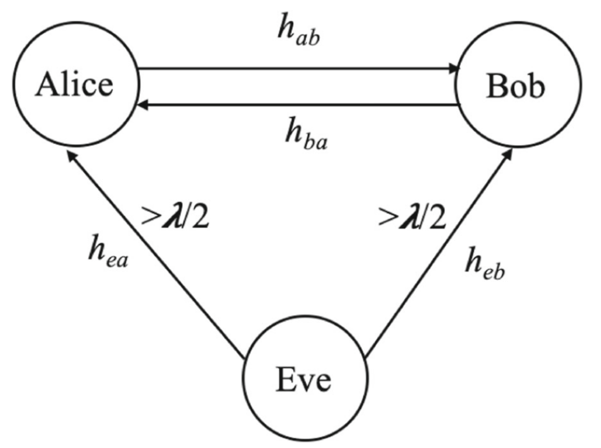


Fig. 1 Adversary model

3.2 System model

The wireless channel has the characteristics of reciprocity and time-varying, which ensures that the legitimate transceivers can exploit the wireless channel characteristics to extract the shared key. To facilitate understanding, we provide symbol definitions in Table 1. Due to hardware limitations, most of the existing communication systems are half-duplex, namely, transceivers cannot receive and transmit signals at the same time. We assume that a key needs to be established between two users, Alice and Bob. Therefore, the measurements between Alice and Bob can be represented as:

$$r_a(t_1) = s(t_1)h(t_1) + n_a(t_1) \tag{1}$$

Table 1 A summary of the symbol definition

Symbol	Meaning
$r(t)$	Received signal of $s(t)$
$s(t)$	Probing signal
$z(t)$	The effect of $n(t)$ on $h(t)$
τ	Coherence time
$n(t)$	Noise at time t

$$r_b(t_2) = s(t_2)h(t_2) + n_b(t_2) \tag{2}$$

where $r(t)$ signifies the signal received by the receiver; $s(t)$ denotes probing signal; $n_a\ (t_1\ ), n_b(t_2)$ represent noise components in Alice and Bob' s received signals, respectively; and t_1, t_2 are the time when Alice and Bob receive signals. Due to the influence of noise, $h(t)$ cannot be directly measured, and it can only be estimated as:

$$\widehat{h}_a(t_1) = h(t_1) + z_a(t_1) \tag{3}$$

$$\widehat{h}_b(t_2) = h(t_2) + z_b(t_2) \tag{4}$$

where $z_a(t), z_b(t)$ represent the effect of $n_a(t)$ and $n_b(t)$ on $h(t)$, respectively. According to (3) and (4), Alice and Bob can derive their channel measurements $\widehat{h}_a(t_1)$ and $\widehat{h}_b(t_2)$, respectively. Actually, half-duplex communication results in $t_1 \neq t_2$, while the noise around Alice and Bob is different, so $\widehat{h}_a(t_1)$ and $\widehat{h}_b(t_2)$ are in all likelihood unequal. But if the difference between t_1 and t_2 is less than the coherence time, the correlation between $\widehat{h}_a(t_1)$ and $\widehat{h}_b(t_2)$ is very high. That is $\widehat{h}_a(t_1) \approx \widehat{h}_b(t_2)$, if $|t_1 - t_2| < \tau$.

In order to ensure that the channel characteristics correlation measured by Alice and Bob is strong enough, it is needed to ensure that the time interval is small (smaller than the coherence time). Therefore, Bob should respond quickly after receiving quickly the probing signal. By repeatedly exchanging transmit probe signals over time-varying channel, Alice and Bob can derive a sequence of n channel estimates $\widehat{h}_a = (\widehat{h}_a[1], \widehat{h}_a[2], \cdots, \widehat{h}_a[n])$ and $\widehat{h}_b = (\widehat{h}_b[1], \widehat{h}_b[2], \cdots, \widehat{h}_b[n])$, respectively [10]. Theoretically, these sequences are highly correlated. During channel probing between Alice and Bob, the passive adversary Eve can overhear the probe signals. However, according to [10], if Eve is more than $\lambda/2$ away from Alice and Bob, she can have a low probability to derive correlated channel estimates between Alice and Bob. Therefore, Eve cannot exploit the eavesdropped signals to derive useful estimations shared between Alice and Bob, even if she knows the value of the probe signal $s(t)$.

3.3 Key generation process

Our key generation process is composed of five phases, including channel probing, pre-processing, quantization and encoding, information reconciliation, and randomness extractor and privacy amplification. Figure 2 overviews the five phases of our scheme. Specifically, channel probing is used to collect channel measurements between legitimate transceivers. The channel measurements can be received signal strength, channel state information, or phase, etc. In this paper, we design the key generation mechanism based on RSS. Information reconciliation [27] is used to correct the bits of inconsistent quantization results. Privacy amplification [14] is used to alleviate the leaked information or to strengthen the key.

Note that our work focuses on the three steps listed in the solid line boxes of Fig. 2.

- *Pre-processing.* In channel probing phase, Alice and Bob continuously collect RSS measurements by repeatedly sending probe signals. Then, we preprocess these measurements using wavelet shrinkage based on rigrsure in section 4.1.
- *Quantization and encoding.* We propose a modified channel quantization with single guard-band (MCQSG) algorithm. Then, the quantized measurements are encoded exploiting Gray code. We will explain the specific design in section 4.2.
- *Randomness extractor.* A randomness extractor is proposed, which cannot only guarantee randomness but also improve bit generation rate. We present the extractor in section 4.4.

3.4 Wavelet shrinkage based on rigrsure

Donoho and Johnstone [28] designed a function of unknown smoothness from noisy sampled data. So we can exploit the wavelet shrinkage to process the signal. In this process, the original signal is decomposed into various scales by wavelet decomposition. Then, wavelet coefficients belonging to noise in each scale are removed to retain and enhance the wavelet coefficients belonging to the signal. Finally, process wavelet coefficients are reconstructed to the denoised signal by wavelet inverse transformation. The basic flow is illustrated in Fig. 3. We use rigrsure to calculate the threshold. Rigrsure is an adaptive threshold selection using the principle of Stein's unbiased risk estimate.

In the above process, wavelet basis, decomposition scales and threshold selection rules are the key factors affecting the final denoising effect.

1. Wavelet selection. Different wavelet basis has their own characteristics in signal processing. None of them can achieve the optimal denoising effect for all kinds of signals. So we have to choose a wavelet basis that is most suitable for processing our signal.
2. Selection of decomposition scales. In wavelet decomposition, the choice of decomposition scale j is also a very important step. The larger the j is, the more distinct the different characteristics of noise and signal performance are, which is more conducive to separate them. On the other hand, the reconstructed signal distortion will be larger, which will affect the final denoising effect to a certain extent. Therefore, in the application, we must pay special attention to the contradiction between these two aspects and choose a suitable decomposition scale.
3. The threshold selection rule, which is based on the model $s(t) = f(t) + e$, where e is the white Gaussian

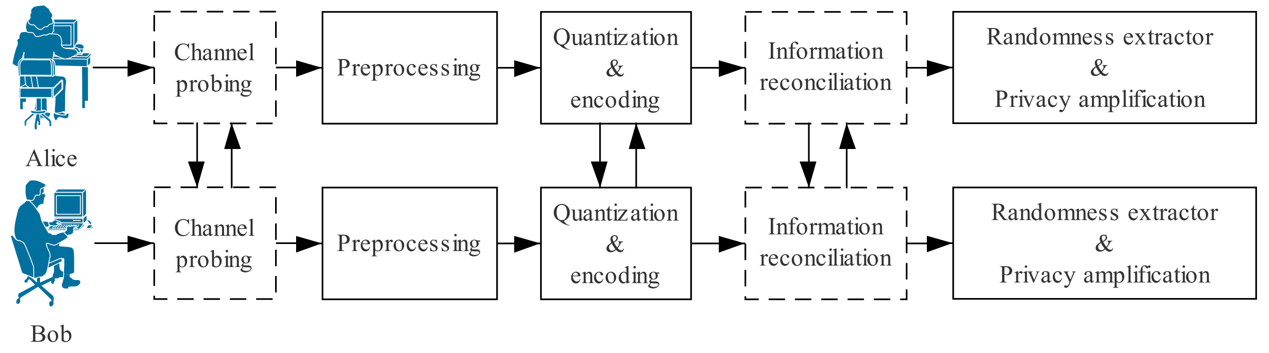

Alice
Bob
Channel probing
Preprocessing
Quantization & encoding
Information reconciliation
Randomness extractor & Privacy amplification
Channel probing
Preprocessing
Quantization & encoding
Information reconciliation
Randomness extractor & Privacy amplification

Fig. 2 Key generation process

noise and $f(t)$ is an instrumental signal. Therefore, the threshold value of noise in the wavelet domain can be eliminated by evaluating the wavelet coefficient or the original signal. The threshold selection rule we adopt is rigrusre. The specific steps of this method are as follows:

Step 1. Take the absolute value of each element in the original signal $s(t)$, sort it from small to large, and then square each element to get the new signal sequence.

$$\omega(k) = (\mathrm{sort}(|s|))^2, (k = 0, 1, \cdots, N_R - 1) \quad (5)$$

where N_R is the length of signal.

Step 2: If the threshold is the square root of the kth element of $\omega(k)$, i.e.,

$$\lambda_k = \sqrt{f(k)}, (k = 0, 1, \cdots, N_R - 1) \quad (6)$$

The risk generated by this threshold is

$$\mathrm{Risk}(k) = [N_R - 2k + \sum_{i=1}^{k} \omega(i) + (N_R - k)\omega(k)] / N_R \quad (7)$$

Step 3: According to the obtained risk curve *Risk*(k), the value corresponding to its minimum risk point is denoted as $k_{\min}$, and then the rigrsure threshold is defined as $\lambda_k = \sigma_n \sqrt{\omega(k_{\min})}$, where σ_n is the standard deviation of the noisy signal.

4 Efficient key generation process

In this section, we present an efficient key generation process. It is composed of five phases, including channel probing, pre-processing, quantization and encoding, information reconciliation and randomness extractor privacy amplification.

4.1 The process of pre-processing

In order to reduce the difference, Alice and Bob perform the same pre-processing on their measurements, respectively. In this work, we use the wavelet shrinkage based on rigrsure [28] to preprocess the sequences.

Following the steps in section 3.4, small-scale fluctuations are reduced based on wavelet shrinkage, which is mainly caused by the divergences between Alice and Bob's measurements. It can be seen from Fig. 4 that the Spearman correlation coefficient of original measurements is 0.698. The resulting bit disagreement rate might be very high if the original measurements are directly quantified. Therefore, we utilize different wavelets to process the measurements and observe the processing effect. For the results illustrated in Fig. 4, the Spearman correlation coefficient of the two groups of measurements processed by symlet wavelet (symN) is 0.862, which is the highest of all wavelets processing. This is due to the reason that symN is an approximate symmetric wavelet, which is an improvement of daubechies wavelet (dbN). Compared with dbN, the symN wavelet is consistent with dbN in

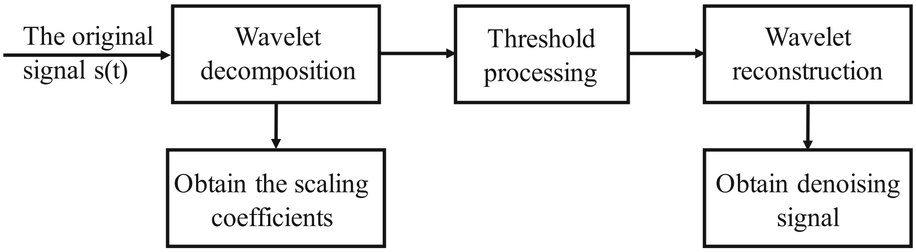


Fig. 3 Wavelet shrinkage flowchart

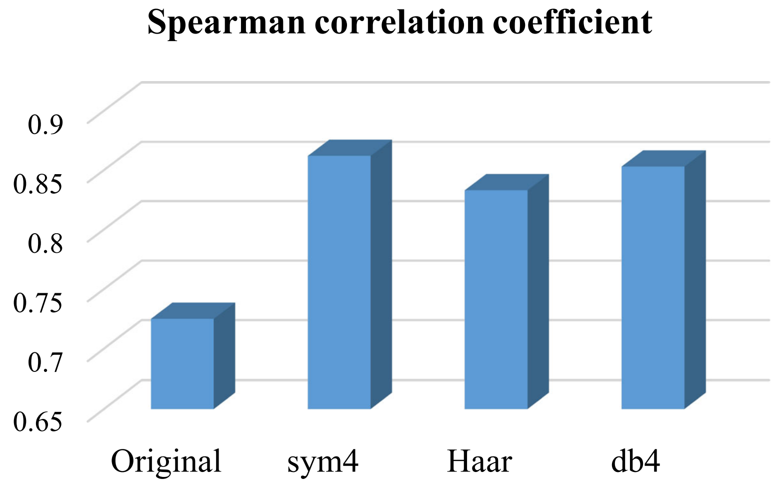


Fig. 4 Spearman correlation coefficient

terms of continuity, branch set length, and filter length. However, symN has better symmetry, which can reduce phase distortion during signal analysis and reconstruction to some extent. And Haar wavelet is the simplest and oldest orthonormal with compact support. In fact, Haar wavelet is the same as db1 wavelet. Note that N is the order of the vanishing of the wavelet function [29]. Therefore, we use symN wavelet to preprocess our original measurements. As depicted in Fig. 5, the curve of preprocessed measurements is smoother than the original measurements.

4.2 Quantization and encoding—an adaptively MCQSG algorithm

In this phase, to improve the bit generation rate and reduce the bit disagreement rate, we propose a modified channel quantization with single guard-band (MCQSG) algorithm. The MCQSG algorithm adaptively corrects the measurements in the guard-band by using the interactive correction information. In addition, our scheme is based on a single guard-band, that is, the guard-band of different intervals has different sizes, which do not cause repeated calculations at the boundary multiple times.

After preprocessing, Alice and Bob obtain their respective sequences. The resultant sequences are used to execute the following MCQSG algorithm.

Step 1: Divide the interval

(1) Intermediate interval. The input sequences of Alice and Bob are $\widehat{h}_a = (\widehat{h}_a[1], \widehat{h}_a[2], \cdots, \widehat{h}_a[n])$ and $\widehat{h}_b = (\widehat{h}_b[1], \widehat{h}_b[2], \cdots, \widehat{h}_b[n])$ respectively. Then, Alice and Bob perform the same operation. Let us take Alice as an example. Alice calculates the standard deviation std and mean value mean of the input sequences. Then Alice can get the upper and lower boundaries of the middle interval by computing:

$$\mathrm{th}_{N+1} = \mathrm{mean} - \alpha \times \mathrm{std} \tag{8}$$

$$th_{N+2} = \mathrm{mean} + \alpha \times \mathrm{std} \tag{9}$$

where N is the number of intervals on both sides; α symbolizes the quantization coefficient; th_{N+1} and th_{N+2} denote the lower and upper boundaries of the intermediate interval, respectively.

(2) Divide the intervals on both sides. Find the maximum value $\widehat{h}_a^{\max}$ and minimum value $\widehat{h}_a^{\min}$ of the input sequence $\widehat{h}_a$. According to formula (10), (11), and (12), the sequence is divided into $2N + 1$ intervals:

$$\mathrm{th}_{i+1} = \widehat{h}_a^{\min} + i \times \frac{(\mathrm{th}_{N+1} - \widehat{h}_a^{\min})}{N}, (1 \leqslant i \leqslant N-1) \tag{10}$$

$$\mathrm{th}_{N+2+i} = \mathrm{th}_{N+2} + i \times \frac{(\widehat{h}_a^{\max} - \mathrm{th}_{N+2})}{N}, (1 \leqslant i \leqslant N-1) \tag{11}$$

$$\mathrm{bin}_j = \begin{cases} [\widehat{h}_a^{\min}, \mathrm{th}_{j+1}], j = 1 \\ [\mathrm{th}_j, \mathrm{th}_{j+1}], j = 2, \cdots, 2N \\ [\mathrm{th}_{j+1}, \widehat{h}_a^{\max}], j = 2N \end{cases} \tag{12}$$

where th and bin denote the threshold and bin, respectively. When $N = 2$, the number of intervals is 5. Specifically, as shown in Fig. 6a, the bule line represents the boundary of the interval. and the red square represents the measured value. It shows that different measurements fall in different intervals. But many measurements fall near the boundaries of the intervals. The small shifts of RSS values around the blue line may cause Alice and Bob to divide the two corresponding RSS into different intervals. Therefore, we use *Step 2* to divide the guard-band for each interval.

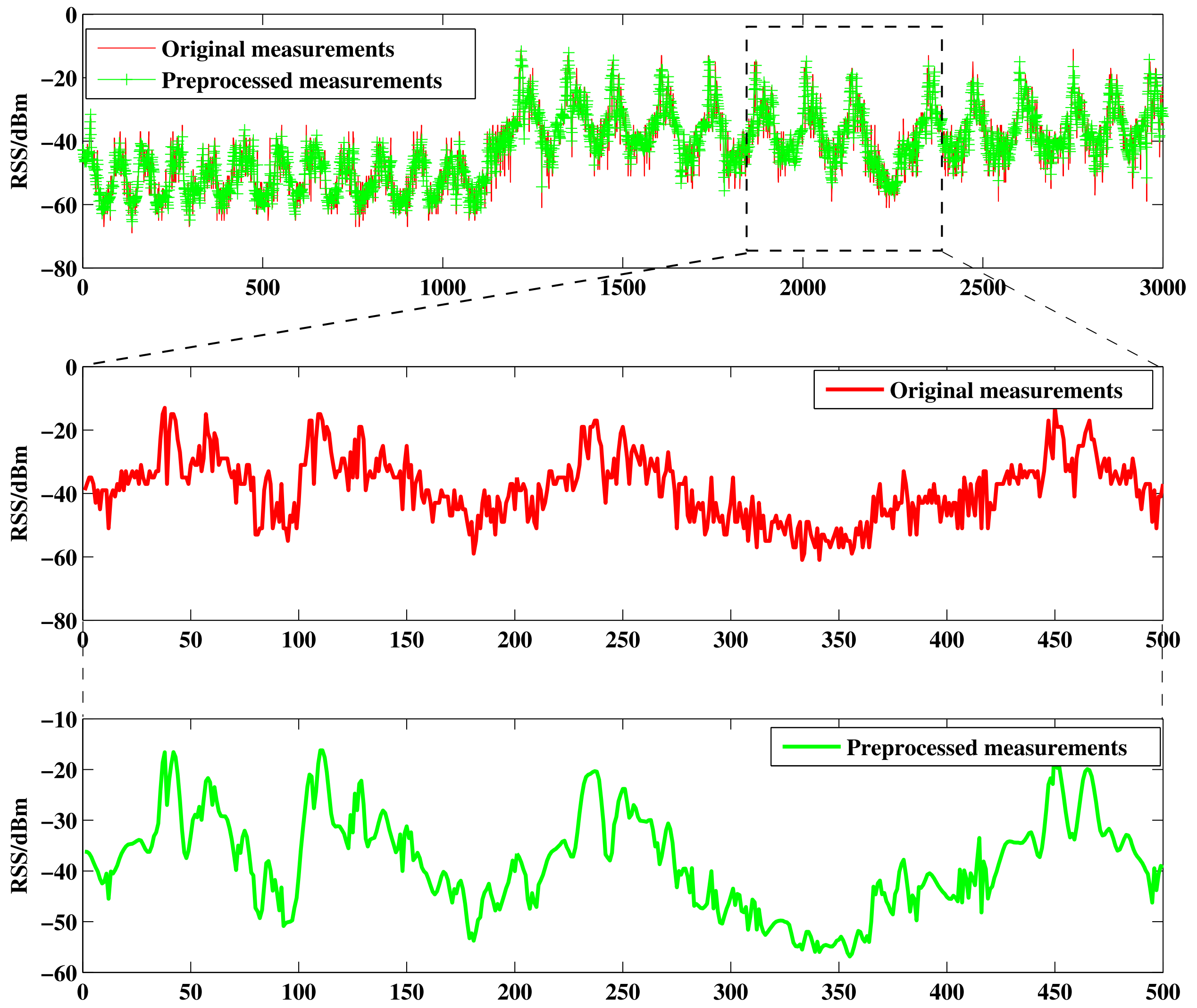
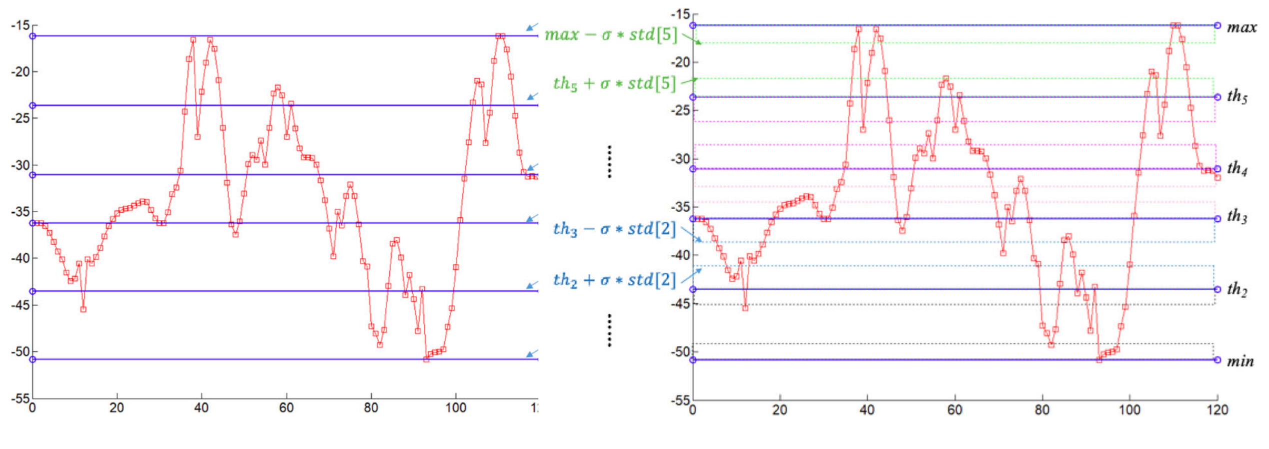


Fig. 5 The original and preprocessed measurements



(a) **Division of intervals** (b) **Division of guard-band**

Fig. 6 Basic principles of step 1 and 2

Step 2: Calculate the guard-band of each interval. Alice counts the distribution of the input sequences in each interval, and then calculates the mean mean $=$ $(\mathrm{mean}[1], \mathrm{mean}[2], \cdots, \mathrm{mean}[2N+1])$ and standard deviation std $=$ $(\mathrm{std}[1], \mathrm{std}[2], \cdots, \mathrm{std}[2N+1])$ of all the data in the corresponding interval, where $\mathrm{mean}[i]$ and $\mathrm{std}[i]$ denote the mean and standard deviation of ith interval, respectively. Similarly, we assume that $N = 2$, then the guard-band of each interval is shown in Fig. 6b. Specifically, the dashed box in the figure represents the guard-band. The measurements falling within the dashed box need to be processed according to *Step 3*. There are two guard-bands in each interval, $[\mathrm{th}_i, \mathrm{th}_i + \sigma \times \mathrm{std}[i]]$ and $[\mathrm{th}_{i+1} - \sigma \times \mathrm{std}[i], \mathrm{th}_{i+1}]$ respectively. Where $\mathrm{th}_i, \mathrm{th}_{i+1}$ denote the boundary of the ith interval, $\mathrm{std}[i]$ represents the standard deviation of the ith interval, and σ denotes the corrective coefficient.

Step 3: Correction information.

(1) Alice traverses all elements of its input sequences $\widehat{h}_a = (\widehat{h}_a[1], \widehat{h}_a[2], \cdots, \widehat{h}_a[n])$, and if $\widehat{h}_a[j] \in [th_i, th_i + \sigma \times std[i]]$ or $\widehat{h}_a[j] \in [th_{i+1} - \sigma \times std[i], th_{i+1}]$, $1 \leqslant j \leqslant n$, record the position j of $\widehat{h}_a[j]$ in $\widehat{h}_a$, and the difference $\Delta h = \widehat{h}_a[j] - \mathrm{mean}[i]$ between $\widehat{h}_a[j]$ and the mean value $\mathrm{mean}[i]$ of current interval, then add j to the set posA, and Δh to the set LA. Otherwise it will not be processed. As shown in Fig. 7. Finally, Alice sends LA and posA to Bob. The specific operation of step 3.1 is shown in the Algorithm 1.

(2) After receiving the correction information sent by Alice, Bob corrects the measurements of the position in its corresponding set LA, $\widehat{h}_b[\mathrm{pos}A(i)] = \widehat{h}_b[\mathrm{pos}A(i)] - \mathrm{LA}[i], i = 1, 2, \cdots, \mathrm{length(LA)}$. It is as shown in Algorithm 2. In this way, Bob can get the new sequence $\widehat{h}_b$. The new resultant sequence $\widehat{h}_b$ is used to perform the same operation as Alice. This is similar to Algorithm 1.

(3) After receiving the correction information sent by Bob, Alice performs the same correction operation as Bob in Algorithm 2.

Step 4: Encoding. After the correction information is quantized, Alice and Bob employ the same encoder to convert their original sequences into bit sequences. Gray code is any two adjacent codes with only a single binary digit difference. According to [30], if Gray code is used instead of other conventional binary codes to generate bit sequences, the bit disagreement rate can be reduced. Therefore, we apply Gray code for encoding in this paper. Finally, we can get the initial keys key_A and key_B.

4.3 Information reconciliation

Due to the characteristics of half-duplex communication mode adopted by wireless devices, Alice and Bob cannot measure the channel at the same time. In addition, noise factors lead to inconsistent quantization results after quantization and encoding. This requires the information reconciliation phase to correct the mismatched bits between the parties and generate the shared key. In this work, we employ Cascade protocol, a classical algorithm proposed in [27], which divides the bitstring into blocks of fixed length and checks the parities of each block pair. We continue to search until the error bits are found and corrected. According to the Cascade protocol, we can correct the error bits in the case of small amount of information leakage.

Algorithm 1: Find the measurements to be corrected

```
Input: $\widehat{h}_a$, Alice's standard deviation of each interval,
       mean of each interval, boundary threshold, the
       maximum value and the minimum value: $std$,
       $mean$, $th$, $\widehat{h}_a^{max}$, $\widehat{h}_a^{min}$
Output: $LA$, $posA$
$th[1] \leftarrow \widehat{h}_a^{min}$;
$th[2N+2] \leftarrow \widehat{h}_a^{max}$;
m=1;
for $i$=1 to length($\widehat{h}_a$ ) do
    for $j$=1 to 2N+1 do
        if $\widehat{h}_a[i] \in bin_j$ then
            if $(th[j] \leqslant \widehat{h}_a[i] \leqslant th[j] + \sigma \times std[j]) || ((th[j+1] - \sigma \times std[j]) \leqslant \widehat{h}_a[i] \leqslant th[j+1])$ then
                $LA[m] \leftarrow \widehat{h}_a[i] - mean[j]$;
                $posA[m] \leftarrow i$ ;//Record the index of the
                  data falling into the guard-band in the
                  $\widehat{h}_a$;
                $m \leftarrow m + 1$;
            end
        end
    end
end
Alice sends $LA$ and $posA$ to Bob on common channel.;
```

Algorithm 2: Correct the measurements

```
Input: $LA$, $posA$
Output: $\widehat{h}_b$
$m \leftarrow 1$;
for $i$=1 to length($\widehat{h}_b$ ) do
    if $posA[m] == i$ then
        $\widehat{h}_b[i] \leftarrow \widehat{h}_b[i] - LA[m]$;
        $m \leftarrow m + 1$;
    end
end
```

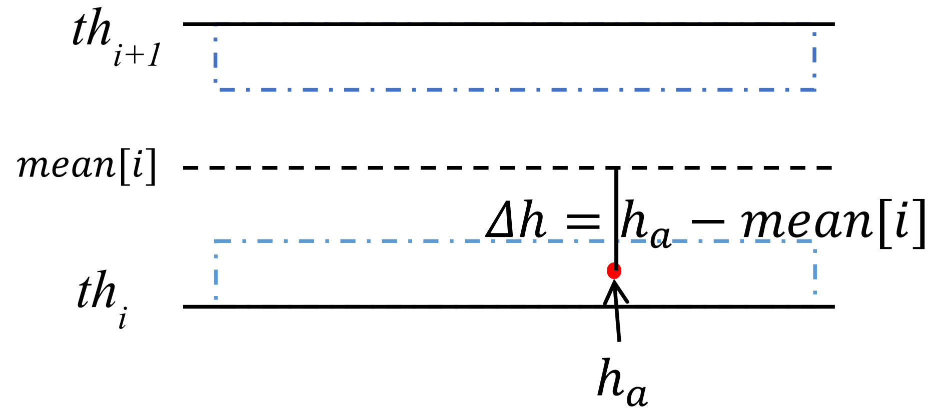


Fig. 7 Correct information

4.4 Randomness extractor and privacy amplification

After information reconciliation, Alice and Bob get the same initial key, but some of the bits might be exposed to the attacker. And when the channel changes slowly or even in a static environment, a large number of consecutive measurements will fall in the same interval. This situation will result in consecutive identical bit strings after quantization, which renders the initial key not perfect random, and the key cannot pass NIST test [31]. So it cannot be directly used as the shared key. In this work, we design a randomness extractor to ensure the randomness of the key and improve the bit generation rate. Our randomness extractor is designed based on a binary tree structure and generates the final key in the following three steps. The process is shown in Fig. 8.

Step 1: Record encoding type

The number of extracted bits ($G' = \lceil \log_2(2N+1) \rceil$) is determined according to the number of intervals. Then for the key, we record every G' bits. As shown in the Fig. 9, assuming $G' = 3$, every three bits are recorded once, and the number of each encoding type is counted. The last two bits are discarded. This is because they cannot form the required number of bits for the encoding type.

Step 2: Initial variable length coding

After counting the number of each encoding type, we sorted by the number from large to small. Note that if the number of an encoding types is 0, we should delete it. We use T' to indicate the number of encoding types. We assume that when $G' = 3$, the number of encoding types S_{000} is 0, and other encoding types are sorted by number as $S_{111} > S_{110} > S_{010} > S_{011} > S_{101} > S_{100} > S_{001}$. In this case $T' = 7$. Then, we will rank the first two encoding types as two leaf nodes, and then superimpose them from big to small according to one layer as the left node and the other layer as the right node. The code binary tree is exhibited in Fig. 10.

Step 3: From Fig. 10, we can get the initial variable length coding results as shown in Table 2. At this point, the situation is divided into two types.

Case 1 *When the value of T' is odd, we enter the sequences into the odd-encoding type extractor. That is, add a digit '0' to the highest bit of each initial variable length coding except for two leaf nodes. Then, we judge the two leaf nodes, and the number of E and D encoded in the sequence is denoted as* num_e *and* num_d *respectively. If* $\mathrm{num}_e - \mathrm{num}_d \geqslant \mathrm{num}_d$*, we will add a digit '0' to the highest bit of D-initial variable length code. Otherwise, the node of E remains unchanged. We assume* $\mathrm{num}_e - \mathrm{num}_d < \mathrm{num}_d$

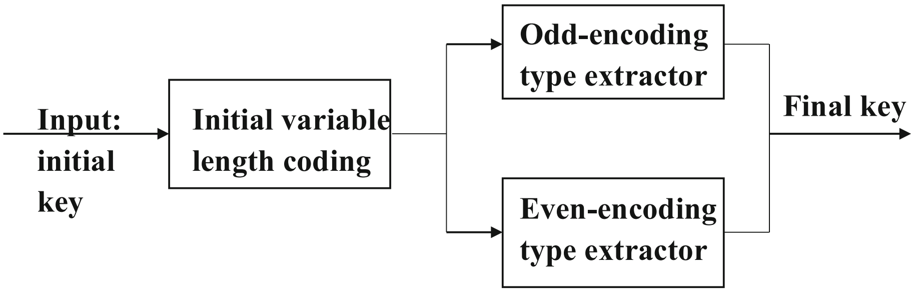

Input: initial key
Initial variable length coding
Odd-encoding type extractor
Even-encoding type extractor
Final key

Fig. 8 Randomness extractor

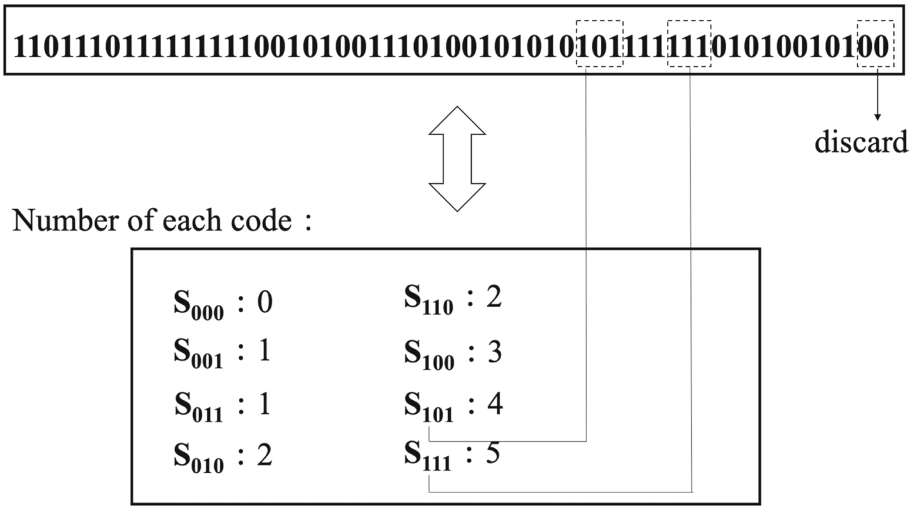


Fig. 9 Encoding type classification

. *As shown in Table 3, it is the final coding mode obtained by inputting the result of Table 2 to even-encoding type extractor. Finally, the initial key is converted to the final key.*

Case 2 *In contrast to Case 1, if the value of T' is even, we put it into the even-encoding type extractor, which adds a two-digit encoding '10' before the highest bit of any variable length code other than two leaf nodes. This can prevent the occurrence of consecutive multiple 0s or 1s. Finally, the initial key is converted to the final key.*

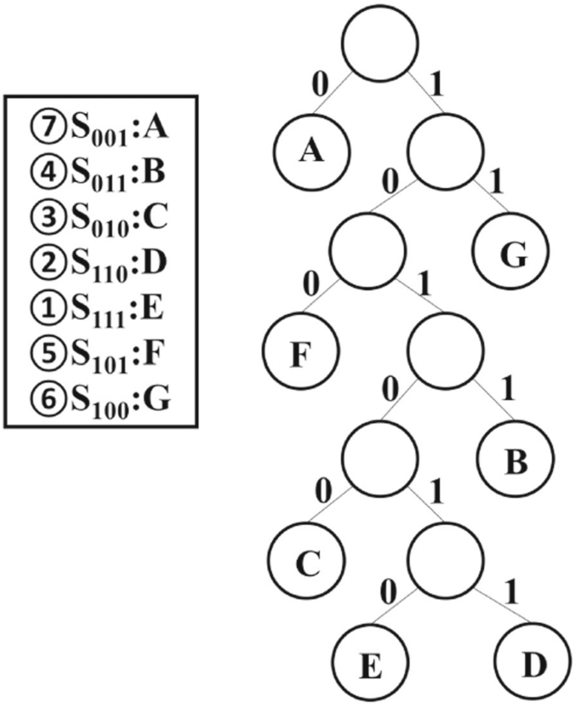


Fig. 10 Code binary tree

Now, the BGR has been improved due to the increase of encoding length and the transceivers have perfect random bits. However, during the exchange of information between Alice and Bob, the exchanged bits are exposed to the attacker. In this scheme, Alice and Bob use the same 2-universal hash functions to process the key, respectively. We exploit the method in [14]. Randomly select a hash function from 2-universal hash family containing all functions $h : \{1, \cdots, M\} -> \{0,1\}^m$.

$$h_{a,b}(x) = ((ax + b) \bmod p_M) \bmod m \tag{13}$$

where $a \in \left\{1, \cdots, p_{M-1}\right\}, b \in \left\{1, \cdots, p_{M-1}\right\}$; p_M represents a prime number larger than M. In addition, we divide the bit sequence into the blocks of size 256 bits, then $M = 2^{256}$. The value of m depends on the factors such as the entropy of the input bit sequence and the leaked information during information reconciliation. This is a complete description of the physical layer key generation scheme.

Table 2 Initial variable length coding

Encoding type	Result
S_{111}	101010
S_{110}	101011
S_{010}	10100
S_{011}	1011
S_{101}	100
S_{100}	11
S_{001}	0

Table 3 Final coding mode

Encoding type	Result
S_{111}	101010
S_{110}	101011
S_{010}	010100
S_{011}	01011
S_{101}	0100
S_{100}	011
S_{001}	00

5 Performance evaluation

5.1 Experimental

To verify the effectiveness of our scheme, we conduct experiments in an indoor environment. The devices are equipped with the rt2870 chip, the two communication parties Alice and Bob both use the Ralink driver and run on the Ubuntu system. All experiments use 802.11g, which operates at 2.4 GHz frequency. In this work, we conduct several experiments on mobile scenes in the laboratory. In each experiment, the experimenter carries a laptop with rt2870 chip either stationary or moving in a straight line. ICMP ping is used to collect packets, and the Radiotap header of each received packet can be used to extract the RSS. Bob measures and records the RSS measurements when he receives the packet from Alice. Then, he immediately replies Alice with a packet, who performs the same operation. As shown in Fig. 11, Alice is placed in the working position of the laboratory while Bob starts to move at a constant speed. Finally, as shown in the Fig. 12, we collected 3500 pieces of measurements to analyze the performance of our scheme. To evaluate our scheme, we use the following three metrics: (1) bit generation rate, (2) bit disagreement rate, and (3) randomness.

5.2 Bit disagreement rate

In our framework, we employ two methods to reduce the inconsistency of the keys. Firstly, we apply the wavelet shrinkage based on rigrsure to preprocess the measurements of both communication parties to improve the correlation between the two sequences in order to reduce the BDR. Secondly, the proposed MCQSG quantization algorithm is used to correct the measurement in the guard-band.

5.2.1 Influence of pre-processing on BDR

To analyze the performances of pre-processing, we compare the BDR of the original and preprocessed measurements during key generation, and the results are shown in Fig. 13. Note that n stands for the number of intervals divided and bit represents the number of bits quantized per RSS. As can be seen from Fig. 13, the BDR of the key generated by directly quantizing the original measurements is high.

5.2.2 Analysis of MCQSG

The MCQSG quantization algorithm we proposed can also reduce the BDR. Since Zhan and Yao [14] also used Gray code encoding, we compare the BDR with it in different cases. As shown in Fig. 14, the BDR of our scheme is far lower than that of theirs. Furthermore, the schemes we choose to compare with is adaptive secret bit generation (ASBG) scheme [12], ASBG_m [12], Abdelgader [15], and the scheme of Zhan and Yao [14]. The results in Fig. 15 show that our scheme achieves low BDR. Although ASBG has the lowest BDR, however, the BGR of ASBG is too low, and it makes the scheme impractical to implement. As a comparison, the BGR of our scheme is much higher than that of ASBG.

The probability of inconsistent bits between Alice and Bob is determined by two factors, namely the fluctuation

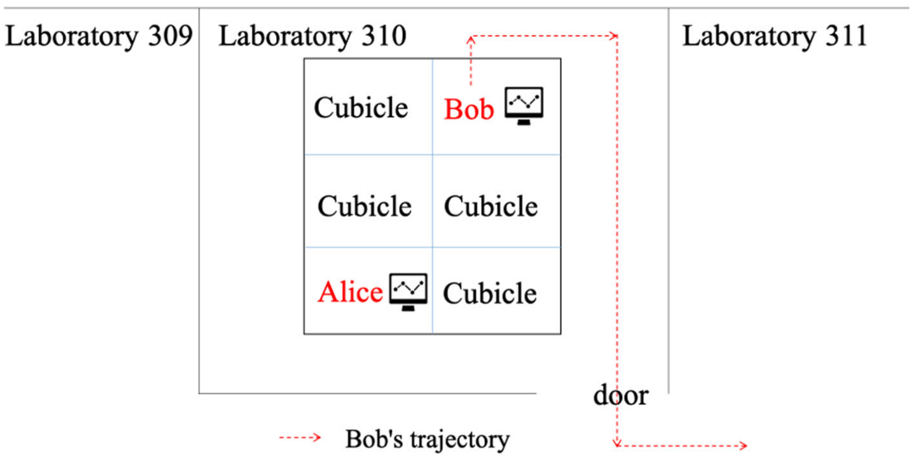


Fig. 11 Experimental environment

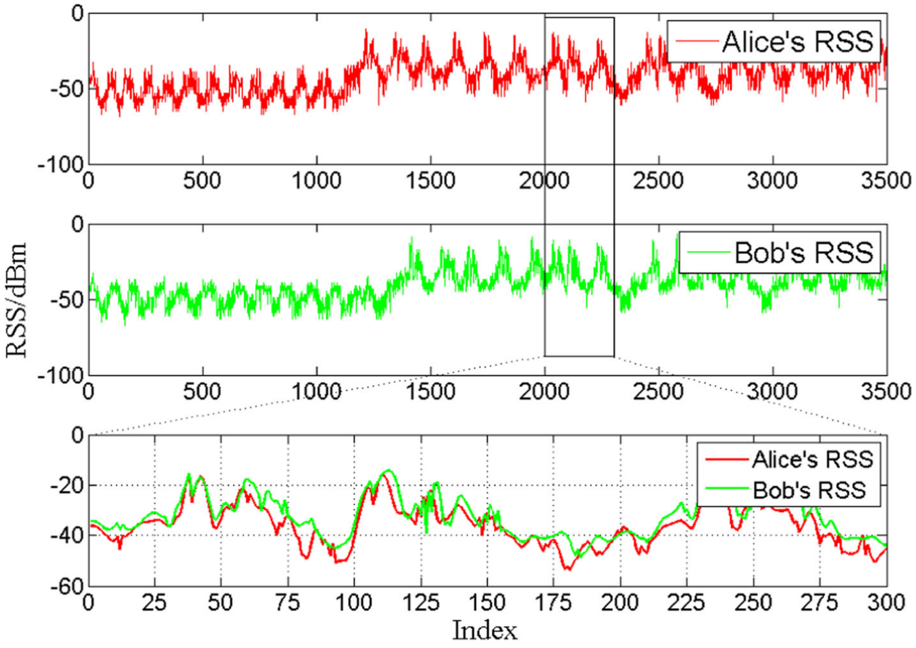


Fig. 12 RSS traces of Alice and Bob

factor α and corrective coefficient σ. In the following, we analyze their influence on BDR.

We assume that $n = 5$, bit = 3. First, the fluctuation factor α has a significant influence on the bit disagreement rate. The Fig. 16a shows that when $0 < \alpha < 0.5$, the bit disagreement rate is positively correlated with the value of α. However, the trend is slow. This indicates that the value of α has little effect on BDR in this case. When $\alpha > 0.5$, the BDR tends to decrease rapidly. The RSS measurements in the middle interval increases with the increase of α, which means that more RSS measurements are in the same interval, that is, the bit disagreement rate decreases. However, at the same time, the randomness of the key will be reduced because too many measurements fall into the same interval. This is determined that the value of α cannot be too large. We should find an optimal α to balance the randomness and BDR. The optimal value varies according to the number of partition intervals. Second, as the coefficient of modified RSS measurement, the corrective coefficient σ determines the number of

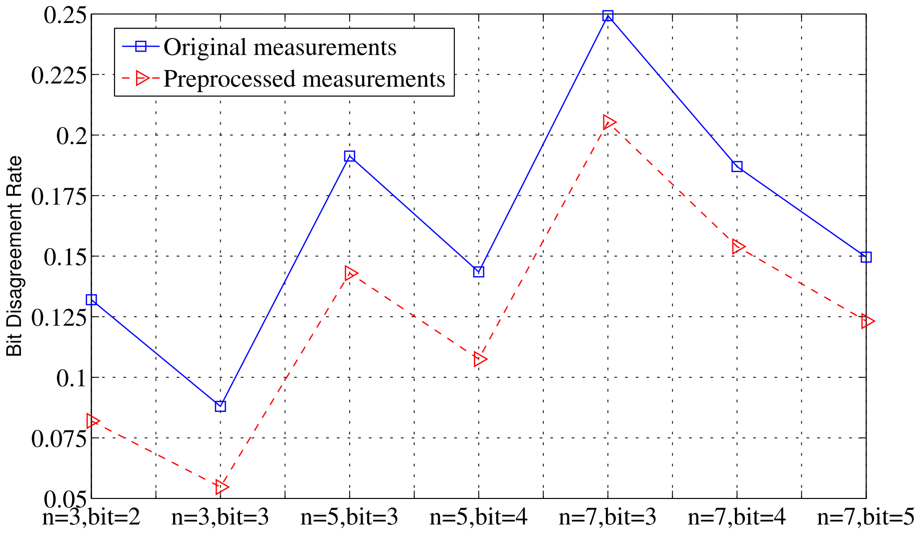


Fig. 13 BDR in different cases

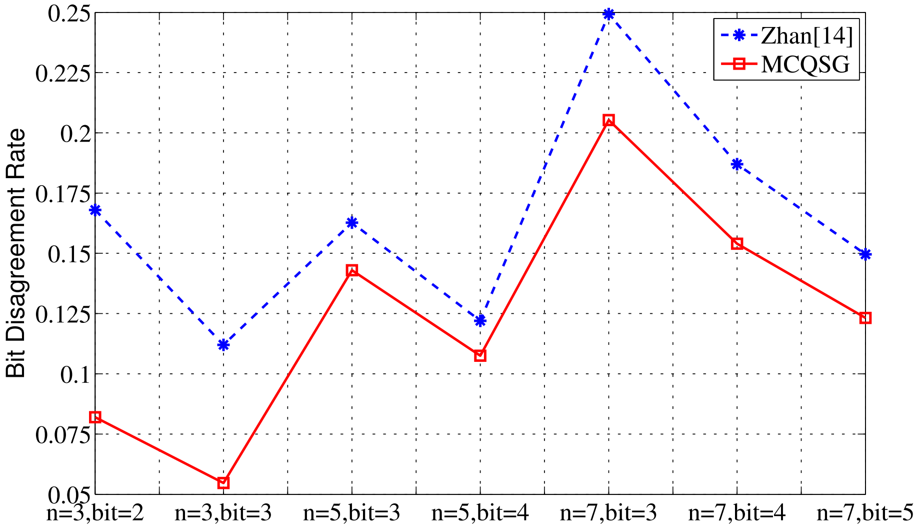


Fig. 14 Comparison of BDR of our and Zhan's scheme

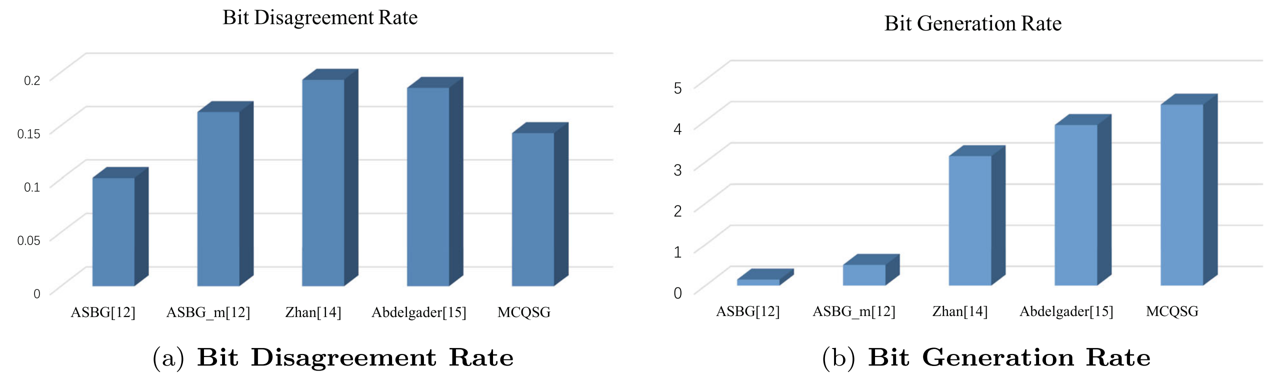
measurements that need to be modified. It is the main factor affecting the bit disagreement rate. The result of Fig. 16b shows that the bit disagreement rate decreases to a certain value with the increase of σ, and then it increases with the increase of σ. So we can find an optimal σ that minimizes the bit disagreement rate. We can figure out that the optimal case is $\sigma = 0.9$.

5.3 Bit generation rate

We define the metric that bit generation rate is the number of bit that each RSS can generate. We also compare our scheme with the schemes [12, 14, 15]. Figure 15b shows that the bit generation rate of our proposed scheme is much higher than other schemes. In our scheme, the number of quantized intervals is the main factor influences the bit generation rate. The more quantized intervals there are, the higher the bit generation rate will be. However, the probability of error also increases, which leads to an increase in the rate of bit inconsistency. Combining Fig. 15a and Fig. 15b, we can see that the scheme ASBG has a 13.09% bit disagreement rate, but its bit generation rate is only 0.146. Our scheme MCQSG's bit disagreement rate is 14.3%, while its bit generation rate reaches about 4.3967. In other words, ASBG's BDR is not much lower than ours, but the bit generation rate of our scheme is much higher. Thus, comparing with other schemes, in terms of BDR and BGR, our scheme performs better.

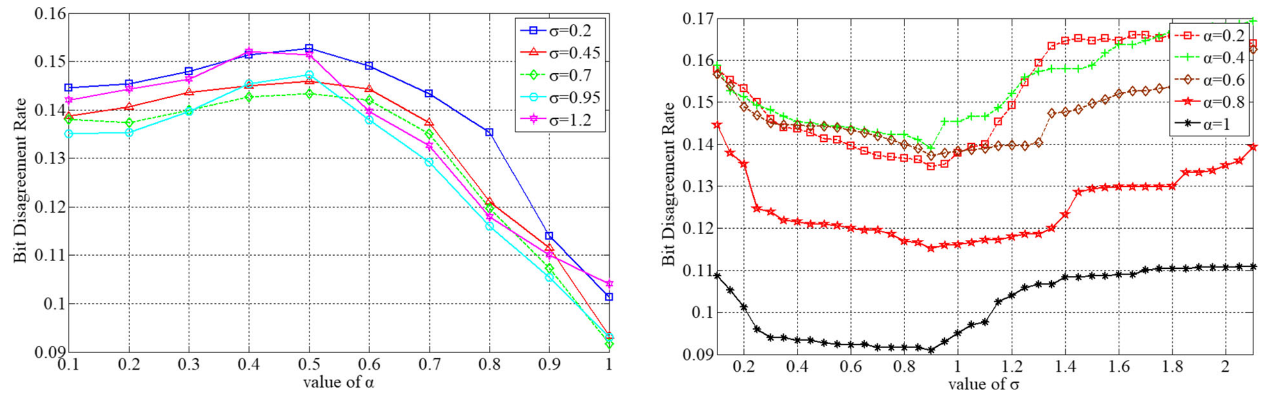
5.4 Randomness

Randomness is a significant metric to evaluate the generated key, because it reflects the difficulty of the attacker inferring the secret key. To ensure that the generated key is random, we use the standard randomness test suite from NIST [31] to evaluate the randomness of the key. NIST is a popular tool for verifying the randomness of the secret key. In this suite, there are 15 different statistical tests. In



(a) **Bit Disagreement Rate** (b) **Bit Generation Rate**

Fig. 15 BGR and BDR comparison of various schemes



(a) **Impact of α on the bit disagreement rate** (b) **Impact of σ on the bit disagreement rate**

Fig. 16 Impact of α and σ on the bit disagreement rate

this work, according to the bit size generated from our experiments, we exploit 5 NIST tests to evaluate the secret key. The test results for 4 different cases are showed in Table 4. All cases pass the test with p value greater than 0.01, which is used as the evaluation threshold. Therefore, these results prove that the keys generated in our scheme can be used to secure the communications.

6 Conclusion

In this paper, we proposed an efficient physical layer key generation scheme to generate shared keys between communication parties based on wireless channel reciprocity. In order to reduce error bits between measurement sequences of communication parties, we exploited the wavelet shrinkage based on rigrsure to pre-process measurements. Furthermore, a quantization scheme based on single guard-band was also implemented to reduce the bit disagreement rate. We also proposed a randomness extractor to further improve the bit generation rate and ensure the randomness of the key. To validate the proposed scheme, we implemented the wireless network card devices as the transmitter and receiver to conduct experiments. The results of experiments showed that our scheme could be used by the devices to generate keys. In addition, the comparisons demonstrated that the performances of our scheme were better than that of other related schemes; thus, our scheme could be wildly applied in wireless networks, such as cognitive radio networks.

Table 4 NIST statistical test suite results (p value >0.01)

TEST	$N = 3$	$N = 5$	$N = 7$	$N = 9$
Frequency	0.9261	0.8997	0.8446	0.4147
BlockFrequency	0.1867	0.7218	0.1619	0.3308
FFT	0.5164	0.3489	0.5421	0.4267
Runs	0.2367	0.3199	0.2202	0.3066
Approx. entropy	0.6794	0.4219	0.7169	0.3581

Abbreviations

ASBG: Adaptive secret bit generation scheme; BDR: Bit disagreement rate; BGR: Bit generation rate; CQG: Channel quantization with guard-band scheme; CSI: Channel state information; dbN: Daubechies wavelet; FIR: Finite impulse response; GB: Guard band scheme; MCQSG: Modified channel quantization with single guard-band algorithm; RSS: Received signal strength; symN: Symlet wavelet

Authors' contributions

All of the authors participated in the whole process of this research work and made considerable contributions, while with the following respective focus: (1) RL, LX, and HF designed and analyzed the efficient physical layer key generation technique in wireless communications. (2) CH participated in the discussion of protocol designed and modified the English expressions. All authors read and approved the final manuscript.

Funding

This work was supported in part by the National Natural Science Foundation of China (grant no. 61771140, U1905211), in part by Major program of Fuzhou Municipal Bureau of Science and Technology (grant no.(2017)325), and in part by Major science and technology project in Fujian province (grant no.2017H6005).

Competing interests

The authors declare that they have no competing interests.

Author details

[1]College of Mathematics and Informatics, Fujian Normal University, Fuzhou, Fujian, China. [2]Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou, Fujian, China. [3]The Department of Electrical and Computer Engineering, The University of Western Ontario, ON N6A 5B9, London, Canada.

Received: 1 July 2019 Accepted: 24 December 2019
Published online: 09 January 2020

References

1. K. Shim, A survey of public-key cryptographic primitives in wireless sensor networks. IEEE Commun. Surv. Tutorials. **18**(1), 577–601 (2016)
2. O. A. Topal, G. K. Kurt, B. Özbek, Key error rates in physical layer key generation: theoretical analysis and measurement-based verification. IEEE Wirel. Commun. Lett. **6**(6), 766–769 (2017)
3. H. Fang, X. Wang, L. Hanzo, Learning-aided physical layer authentication as an intelligent process. IEEE Trans. Commun. **67**(3), 2260–2273 (2019)
4. K. Zeng, Physical layer key generation in wireless networks: challenges and opportunities. IEEE Commun. Mag. **53**(6), 33–39 (2015)
5. J. Huang, T. Jiang, in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*. Dynamic secret key generation exploiting ultra-wideband wireless channel characteristics, (New Orleans, 2015), pp. 1701–1706. https://doi.org/10.1109/WCNC.2015.7127724

6. Y. Kong, B. Lyu, F. Chen, Z. Yang, The security network coding system with physical layer key generation in two-way relay networks. IEEE Access. **6**, 40673–40681 (2018)
7. J. Zhang, R. Woods, A. Marshall, T. Q. Duong, in *2015 IEEE Globecom Workshops (GC Wkshps)*. Verification of key generation from individual ofdm subcarrier's channel response, (San Diego, 2015), pp. 1–6. https://doi.org/10.1109/GLOCOMW.2015.7414111
8. Y. Peng, P. Wang, W. Xiang, Y. Li, Secret key generation based on estimated channel state information for tdd-ofdm systems over fading channels. IEEE Trans. Wirel. Commun. **16**(8), 5176–5186 (2017)
9. H. Fang, L. Xu, Y. Zou, X. Wang, K. R. Choo, Three-stage stackelberg game for defending against full-duplex active eavesdropping attacks in cooperative communication. IEEE Trans. Veh. Technol. **67**(11), 10788–10799 (2018)
10. S. Mathur, W. Trappe, N. B. Mandayam, C. Ye, A. Reznik, in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking - MobiCom '08*. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel, (San Francisco, 2008). https://doi.org/10.1145/1409944.1409960
11. X. Zhu, F. Xu, E. Novak, C. C. Tan, Q. Li, G. Chen, Using wireless link dynamics to extract a secret key in vehicular scenarios. IEEE Trans. Mob. Comput. **16**(7), 2065–2078 (2017)
12. S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking - MobiCom '09*. On the effectiveness of secret key extraction from wireless signal strength in real environments, (New York, 2009). https://doi.org/10.1145/1614320.1614356
13. X. Li, J. Liu, Q. Yao, J. Ma, Efficient and consistent key extraction based on received signal strength for vehicular ad hoc networks. IEEE Access. **5**, 5281–5291 (2017)
14. F. Zhan, N. Yao, On the using of discrete wavelet transform for physical layer key generation. Ad Hoc Netw. **64**, 22–31 (2017)
15. A. M. S. Abdelgader, L. Wu, in *2014 IEEE 17th International Conference on Computational Science and Engineering*. A secret key extraction technique applied in vehicular networks, (Chengdu, 2014). https://doi.org/10.1109/cse.2014.264
16. S. Althunibat, V. Sucasas, J. Rodriguez, A physical-layer security scheme by phase-based adaptive modulation. IEEE Trans. Veh. Technol. **66**(11), 9931–9942 (2017)
17. G. Li, A. Hu, C. Sun, J. Zhang, Constructing reciprocal channel coefficients for secret key generation in fdd systems. IEEE Commun. Lett. **22**(12), 2487–2490 (2018)
18. L. Cheng, L. Zhou, B.-C. Seet, W. Li, D. Ma, J. Wei, Efficient physical-layer secret key generation and authentication schemes based on wireless channel-phase. Mob. Inf. Syst. **2017**, 1–13 (2017). https://doi.org/10.1155/2017/7393526
19. Y. P. Hong, L. Huang, H. Li, Vector quantization and clustered key mapping for channel-based secret key generation. IEEE Trans. Inf. Forensic. Secur. **12**(5), 1170–1181 (2017)
20. A. Sayeed, A. Perrig, in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. Secure wireless communications: secret keys through multipath (IEEE, Las Vegas, 2008). https://doi.org/10.1109/icassp.2008.4518284
21. J. W. Wallace, R. K. Sharma, Automatic secret keys from reciprocal mimo wireless channels: measurement and analysis. IEEE Trans. Inf. Forensic. Secur. **5**(3), 381–392 (2010)
22. J. Croft, N. Patwari, S. K. Kasera, in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN '10*. Robust uncorrelated bit extraction methodologies for wireless sensors (ACM Press, Stockholm, 2010). https://doi.org/10.1145/1791212.1791222
23. J. Zhang, A. Marshall, R. Woods, Q. T. Duong, Efficient key generation by exploiting randomness from channel responses of individual ofdm subcarriers. IEEE Trans. Commun. **64**(6), 2578–2588 (2016)
24. J. Zhang, R. Woods, T. Q. Duong, A. Marshall, Y. Ding, in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. Experimental study on channel reciprocity in wireless key generation (IEEE, Edinburgh, 2016). https://doi.org/10.1109/spawc.2016.7536825
25. R. Jin, X. Du, K. Zeng, L. Huang, L. Xiao, J. Xu, Delay analysis of physical-layer key generation in dynamic roadside-to-vehicle networks. IEEE Trans. Veh. Technol. **66**(3), 2526–2535 (2017)
26. M. Alhasanat, S. Althunibat, K. A. Darabkh, A. Alhasanat, M. Alsafasfeh, A physical-layer key distribution mechanism for IoT networks. Mob. Netw. Appl. (2019). https://doi.org/10.1007/s11036-019-01219-5
27. G. Brassard, L. Salvail, in *Advances in Cryptology - EUROCRYPT '93*. Secret-key reconciliation by public discussion (Springer Berlin Heidelberg, Berlin, 1994), pp. 410–423. https://doi.org/10.1007/3-540-48285-7_35
28. D. L. Donoho, I. M. Johnstone, Adapting to unknown smoothness via wavelet shrinkage. J. Am. Stat. Assoc. **90**(432), 1200–1224 (1995)
29. R. S. Stanković, B. J. Falkowski, The haar wavelet transform: its status and achievements. Comput. Electr. Eng. **29**(1), 25–44 (2003)
30. S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, S. V. Krishnamurthy, Secret key extraction from wireless signal strength in real environments. IEEE Trans. Mob. Comput. **12**(5), 917–930 (2013)
31. A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report (2001). Booz-Allen and Hamilton

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.