Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 6980619, 13 pages
https://doi.org/10.1155/2023/6980619

*Research Article*

# Wireless Key Generation Scheme Based on Random Permutation and Perturbation in Quasistatic Environments

**Liquan Chen**,[1,2] **Yi Lu**,[1] **Tianyu Lu**,[1] **Zhaofa Chen**,[1] **and Aiqun Hu**[1,2]

[1]*School of Cyber Science and Engineering, Southeast University, Nanjing 211102, China*
[2]*Purple Mountain Laboratories, Nanjing 211111, China*

Correspondence should be addressed to Liquan Chen; lqchen@seu.edu.cn

Received 13 August 2022; Revised 19 January 2023; Accepted 8 February 2023; Published 27 April 2023

Academic Editor: Zhao Li



The wireless key generation using wireless channel reciprocity has attracted considerable attention in the past two decades. However, there are many challenges for the key generation in quasistatic wireless environments. The key generation rate (KGR) in a quasistatic environment is low, and the randomness of the key is insufficient, which is difficult to meet the secure communication requirements. To tackle these issues, a random permutation and perturbation-based wireless key generation (RPP-WKG) scheme is proposed to improve the KGR and randomness in quasistatic environments. Unlike existing key generation schemes, the RPP-WKG scheme allows the two legitimate users to generate the same secret key based on their random permuted channel measurements. Besides, the perturbed key sequence will be obtained by combining the initial key generated after quantization and the permutation order sequence through the XOR operation. Simulation results show that the proposed RPP-WKG scheme can generate secret keys with a high generation rate, sufficient randomness, a low mismatch rate, and a low correlation coefficient in quasistatic environments.

## 1. Introduction

With the rapid development of wireless communication techniques and the wide use of the Internet of Things (IoT) devices, establishing an encrypted and secure communication link between two IoT devices has become an urgent need [1–5]. The traditional encryption methods widely used at present are symmetric or asymmetric cryptographic algorithms. Symmetric cryptographic algorithms usually rely on preshared secret keys, which are not suitable for distributed IoT devices [6]. Asymmetric cryptography requires complex mathematical algorithms. However, due to the limited computing power of IoT devices and the difficulty in establishing a public key infrastructure between devices, these asymmetric cryptographic algorithms are not suitable for secure communication between lightweight IoT devices. In recent years, wireless key generation schemes based on physical layer channel characteristics have received extensive attention due to their low computational complexity and high security. Traditional cryptographic mechanisms can be supplemented and enhanced by taking advantage of the inherent physical properties of wireless channels [7]. Wireless key generation based on physical layer channel reciprocity is a promising solution for secure communication between IoT devices [8–10].

Generally, a wireless key generation scheme contains four steps: channel sampling, quantization, information reconciliation, and privacy amplification [11]. Among the four steps, quantization converts channel measurements into binary bit sequences, which is the core function of the wireless key generation scheme. Various channel characteristics can be used for quantization, such as received signal strength (RSS), channel state information (CSI), time delay amplitude, phase, and angle-of-arrive (AoA) [12–15].

However, unsynchronized channel sampling in time-division duplex systems and environmental noise will impair channel reciprocity in the real system. The nonreciprocity in the channel measurements can be further amplified by ambient noise, adversely causing the inconsistent quantization result between two users. To mitigate the nonreciprocity of a wireless channel, many researchers have proposed solutions. For example, Li et al. [16] designed a mean-value

quantization scheme for RSS to improve the key generation rate (KGR). Zhao et al. [17] proposed performing group quantization and adaptive quantization on the collected RSS measurements. Margelis et al. [18] used discrete cosine transform (DCT) on channel observations to reduce the mismatches caused by quantization. Liu et al. [19] designed a bipartite graph matching-based wireless key generation method to avoid quantization.

In some IoT application scenarios, such as environmental monitoring and smart home, the IoT devices are fixed and the surrounding wireless environment changes very slowly [20, 21]. In these scenarios, wireless channels between the communication users are quasistatic. The KGR based on the characteristics of this quasistatic wireless channel is very low, which is difficult to meet the secure communication requirements. The reason for low KGR is due to the long channel coherence time in the quasistatic channel, and the secret keys are generated within the coherence time, so the similarity of the secret keys is high. At the same time, ambient noise will also cause key inconsistency. Therefore, an efficient and robust solution is required to achieve a low key mismatch rate (KMR) in a quasistatic environment. Various schemes have been proposed to overcome the challenges of wireless key generation in quasistatic environments. [22, 23] proposed key generation protocols with the aid of a reconfigurable intelligent surface (RIS) to boost KGR in quasistatic environments. [24] used singular value decomposition techniques to reconstitute the wireless channels to improve the randomness of the wireless channels. In [25], the two legitimate users independently generated local randomness to be used together with the uniqueness of the wireless channel coefficients in order to enable high-rate secret key generation.

To mitigate the effect of channel nonreciprocity, we use principal component analysis- (PCA-) based processing on the sampled channel measurements. Li et al. [26] proposed two realization algorithms of PCA for preprocessing: PCA algorithm with interaction and PCA algorithm without interaction. The corresponding eigenvalues and eigenvectors of the two legitimate users, Alice and Bob, are different due to the deviation. Alice can send her eigenvectors to Bob via a public channel and both of them use it for signal reconstruction, which is named as the PCA algorithm with interaction. Alice and Bob can also calculate their own eigenvectors and eigenvalues and use their eigenvectors for signal reconstruction without any interaction, which is called the PCA algorithm without interaction. Although the PCA algorithm with interaction can obtain a relatively higher key agreement than the PCA algorithm without interaction, information leakage will be caused by the transmission on an insecure public channel. When the eavesdropper, Eve, obtains enough information such as eigenvalue and eigenvector, he/she can find the secret key by a brute-force search. Li et al. [26] assume Eve can only obtain eigenvectors instead of the covariance matrix, resulting in a low information leakage ratio. In this paper, since broadcasting eigenvectors on a public channel still has security risks, we recommend the two legitimate users perform a processing algorithm based on PCA without interaction on their original channel measurements after channel sampling.

In a quasistatic channel, the secret keys extracted from channel measurements not only have a relatively low KGR but also have poor randomness. The use of PCA processing on the CSI matrices of legitimate users can only obtain good feature amplification and deredundancy effects, but the KGR cannot be improved by PCA processing. In order to solve the problems of the low KGR and the poor randomness of the secret keys, we focus on the preprocessing algorithm of channel measurements and propose a random permutation and perturbation-based wireless key generation (RPP-WKG) scheme, which provides high randomness and low correlation for secret keys. Based on the RPP-WKG scheme, we develop a secret key generation method that is aimed at extracting secret keys from channel measurements at a low KMR and high speed. CSI is chosen as the channel measurement in this paper because the existing work has shown that CSI could provide more channel characteristics than RSS does. The main contributions of this paper are summarized as follows:

(1) A new and practical RPP-WKG scheme is proposed. Based on the scheme, we can mitigate the impact of the quasistatic channel and generate secret keys with high randomness and low correlation

(2) We propose an efficient and secure permutation method, which can help legitimate users perform the same random permutation on their respective CSI to acquire new random sources with high randomness and great fluctuations. In addition, the length of the permutation order can be adjusted by the number of CSI segments. The random sources can be used as the new channel measurements to generate secret keys

(3) A minimum weight-based matching method is proposed to reduce KMR in the RPP-WKG scheme. Legitimate users can obtain an agreement on the permutation order of CSI without revealing it. The permutation order will be obtained by finding the correspondence between the users' CSI, and it can be used as a source of the secret keys

(4) We propose a random perturbation generation method based on the permutation order agreed by the two legitimate users. The correlation between the secret keys is reduced by performing an XOR operation on the random perturbation sequence and the initial key, and the randomness and KGR are further improved

*1.1. Notation and Outline.* Unless otherwise specified, we use the following notations throughout the manuscript: Upper bold-face letters denote matrices and lower bold-face letters denote vectors. Light-face letters denote scalars. Numeral subscripts of matrices and vectors, if needed, represent their sizes. $\mathbf{I}$ denotes the identity matrix. Matrix superscript $(\cdot)^H$ denotes conjugate-transpose. The $E\{\cdot\}$ denotes ensemble expectation. The $\mathrm{vec}\{\bullet\}$ is the straightening operation by row.

The remainder of this paper is organized as follows: In Section 2, the system model and the related formulations are presented. The basic key generation steps are also introduced in this section. In Section 3, we describe the proposed RPP-WKG scheme in detail. The performance results are evaluated extensively in Section 4. In Section 5, we summarize the paper.

## 2. System Model

*2.1. Channel Estimation.* Figure 1 illustrates the system model of a wireless key generation system in the smart home: In an orthogonal frequency division multiplexing (OFDM) communication system, Alice and Bob establish secret keys in the time division duplex (TDD). They take advantage of the reciprocity and time variability of wireless channels to generate consistent security keys at both ends and update them continuously. Eve has a potential security threat to the communication between Alice and Bob.

During the channel sampling process, Alice and Bob alternately transmit pilots to each other. Alice sends a channel probing signal at time slot 1, and Bob receives the signal and stores it locally. Bob sends a channel probing signal at time slot 2, and Alice receives the signal and stores it. Meanwhile, Eve eavesdrops on the signals from Alice and Bob in two-time slots and tries to decrypt the message.

In this paper, we use the CSI as the channel measurements. We assume that the difference in measured values caused by delay and hardware fingerprints has been removed by methods such as interpolation transformation and hardware calibration. The matrices $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ of size $N \times K$ are defined as the channel measurement matrices of Alice and Bob after channel sampling, where $N$ is the number of subcarriers and $K$ is the number of samples. The relationship between $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ can be expressed as $\boldsymbol{H}^B = \boldsymbol{H}^A + \boldsymbol{W}$, where $\boldsymbol{W}$ represents the observation deviation caused by the measurement noise and the noise remaining in the calibration process. $\boldsymbol{W}$ is independent of $\boldsymbol{H}^A$ and considered to follow a complex Gaussian distribution.

*2.2. Problem Formulation.* According to the principle of channel reciprocity, the channel response of Alice and Bob should be highly correlated in practice. Since the ambient noises are usually considered to follow complex Gaussian distribution, the received channel measurements $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ should also be highly correlated. Based on the above theories, traditional wireless key generation methods allow Alice and Bob to extract the same secret keys by quantizing each channel measurement in $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$, respectively. However, $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ could be easily affected by random ambient noise and nonsimultaneous channel probing, resulting in inconsistent quantization results and mismatched secret keys between two users.

Besides, the wireless environments between two legitimate devices change slowly in the smart home application scenario, which will result in the two adjacent channel samples in a coherence time being very similar. Figure 2 shows the CSI sampled under the quasistatic environments, which

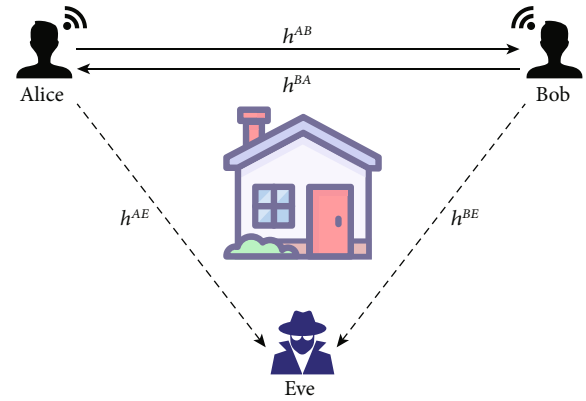



Figure 1: Channel estimation in a smart home.

is in an OFDM model with 56 subcarriers. The SNR in the scenario is 40 dB, and the sampling interval is 0.5 ms. The $x$-axis and $y$-axis of Figure 2 represent the real and imaginary parts of the CSI parameter. It can be seen that the CSI measured from two adjacent samples are very similar. This will result in the two generated keys being very similar or even identical. Overall, the above challenges demonstrate the need for a new key generation scheme to achieve efficient key generation in a quasistatic environment.

*2.3. Basic Steps of Wireless Key Generation.* Generally, the generation of secret keys based on channel measurements between two legitimate users includes four steps.

*2.3.1. Channel Sampling.* To initiate the key generation, Alice and Bob sample the channel through multiple rounds of probe packet exchanges [27, 28], each controlled within a coherence time to ensure channel reciprocity. After each user receives the probe packets, the channel measurements are extracted from the probe packets to construct reciprocal channel matrices $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ for Alice and Bob, respectively. The channel sampling process is completed after a sufficient number of probe packets are collected.

*2.3.2. Quantization.* After channel sampling, Alice and Bob need to adopt the same quantization scheme on channel measurements to obtain the initial keys. The quantization process is an analog-to-digital conversion process, which converts the CSI estimated by the legitimate communication parties into a sequence of key bits [29].

*2.3.3. Information Reconciliation.* Due to the interference, estimation error, and other facts, the initial keys quantized by Alice and Bob may have inconsistent bits. The main purpose of information reconciliation is to correct the inconsistent bits in the secret keys of the two legitimate users without divulging the key information as much as possible [30, 31]. After information reconciliation, inconsistent bits are eliminated and both Alice and Bob will obtain the consistent initial keys.

*2.3.4. Privacy Amplification.* Eve can eavesdrop on the information about the secret keys during the communication between Alice and Bob. Privacy amplification needs to be

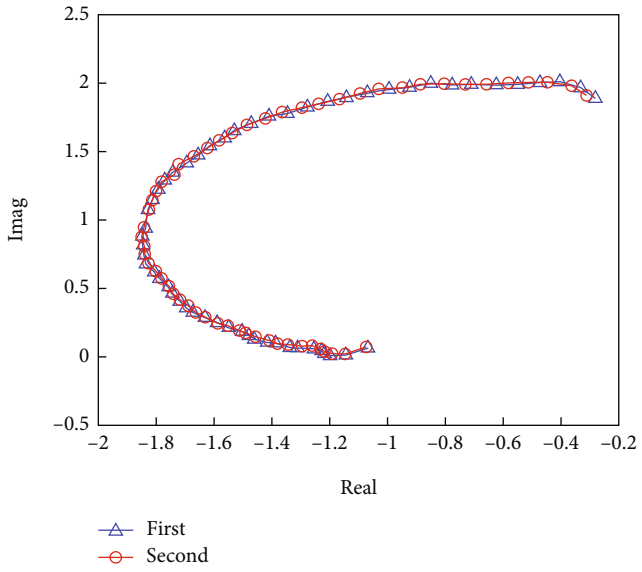



Figure 2: The results of two adjacent channel sampling.

performed to eliminate the information eavesdropped on by Eve [32–34]. After privacy amplification, Alice and Bob will obtain the final secret keys to encrypt their messages.

## 3. The Proposed RPP-WKG Scheme

*3.1. RPP-Based Key Generation.* The basic idea of the RPP-based wireless key generation algorithm is to permute channel measurements randomly and match the sorted channel measurement values between pairs of reciprocal users. Then the two reciprocal users perform the wireless key generation scheme according to the channel measurements after permutation and the agreed permutation order. As shown in Figure 3, Alice and Bob collect their respective channel measurement matrices $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ of size $N \times K$ in the channel sampling stage. Alice and Bob then perform PCA processing without interaction on their respective channel measurement matrices; the channel measurement matrices after PCA processing are $\boldsymbol{Y}^A$ and $\boldsymbol{Y}^B$ of size $P \times K$ and consist of $P$ groups of samples. For the accuracy of statistical information, the number of sample groups and dimensions should satisfy $K \geq P$.

$$\begin{aligned} \boldsymbol{Y}^A &= \left[\boldsymbol{y}_1^A, \boldsymbol{y}_2^A, \cdots, \boldsymbol{y}_P^A\right]^H, \\ \boldsymbol{Y}^B &= \left[\boldsymbol{y}_1^B, \boldsymbol{y}_2^B, \cdots, \boldsymbol{y}_P^B\right]^H. \end{aligned} \tag{1}$$

After PCA processing, Alice applies random permutation to her channel measurement matrix $\boldsymbol{Y}^A$. The channel measurement matrix after permutation is $\widehat{\boldsymbol{Y}}^A$. The permutation order **PO** is determined by Alice according to the size of the matrix $\boldsymbol{Y}^A$. After the straightening transformation of $\widehat{\boldsymbol{Y}}^A$, Alice then sends the permutated channel measurements to Bob via a public channel without revealing the permutation order. Once receiving the permutated channel measurements, Bob can infer the permutation order by finding the correspondence between $\widehat{\boldsymbol{Y}}^A$ and his own channel measurement matrix $\boldsymbol{Y}^B$ through channel reciprocity. Bob then performs the same permutation on $\boldsymbol{Y}^B$ and gets the new channel measurement matrix $\widehat{\boldsymbol{Y}}^B$.

Meanwhile, Alice and Bob use their respective reconstructed signal matrices after random permutation $\widehat{\boldsymbol{Y}}^A$ and $\widehat{\boldsymbol{Y}}^B$ to perform the quantization operation. The permutation order **PO** participates in key generation as a source of randomness perturbation. Last, Alice and Bob perform the information reconciliation and privacy amplification on the origin secret keys to further eliminate occasional errors and generate secret keys with high randomness. The details of these components are elaborated as follows.

Some notions and their descriptions used in the following sections are listed in Table 1.

*3.2. Sampling and Preprocessing Model.* In the channel sampling phase, Alice and Bob each send pilots to each other and estimate CSI. A vector of length $N$ for the $k$-th channel estimate can be written as

$$\boldsymbol{h}_k^u = \boldsymbol{h}_k + \boldsymbol{n}_k^u, \tag{2}$$

where $u = \{a, b\}$, $a$ and $b$ denote Alice and Bob, respectively, $\boldsymbol{h}^k$ follows complex Gaussian distribution, and $\boldsymbol{n}^u$ is independent and identically distributed zero-mean complex Gaussian noise with variance $E\{\boldsymbol{n}_k^u(n_k^u)^H\} = \sigma_n^2 \mathbf{I}_N$. After $K$ channel samplings, Alice and Bob can construct the channel measurement matrix $\boldsymbol{H}^u$ as

$$\boldsymbol{H}^u = [\boldsymbol{h}_1^u, \boldsymbol{h}_2^u, \cdots, \boldsymbol{h}_K^u], \tag{3}$$

where $\boldsymbol{h}_k^u$ and $\boldsymbol{h}_l^u$ are assumed to be independent and identically distributed, $k, l \in [1, 2, \cdots, K]$. The subscript is omitted for simplicity, and we define the channel signal-to-noise ratio (SNR) as

$$\mathrm{SNR} = \frac{E\{\boldsymbol{h}^H \boldsymbol{h}\}}{N\sigma_n^2}. \tag{4}$$

After channel sampling, Alice and Bob will get their respective channel measurement matrices $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ for further preprocessing. The signal preprocessing process is divided into two steps: PCA processing without interaction, random segmentation, and permutation.

*3.2.1. PCA Processing without Interaction.* Figure 4 shows the process of PCA. In PCA processing without interaction, Alice and Bob calculate the transformation matrices according to the following steps:

(1) Alice and Bob perform eigenvalue decomposition of their covariance matrices $\boldsymbol{R}^A$ and $\boldsymbol{R}^B$, respectively, where $\boldsymbol{\Lambda}^A$, $\boldsymbol{\Lambda}^B$ are eigenvalue matrices and $\boldsymbol{U}^A$, $\boldsymbol{U}^B$

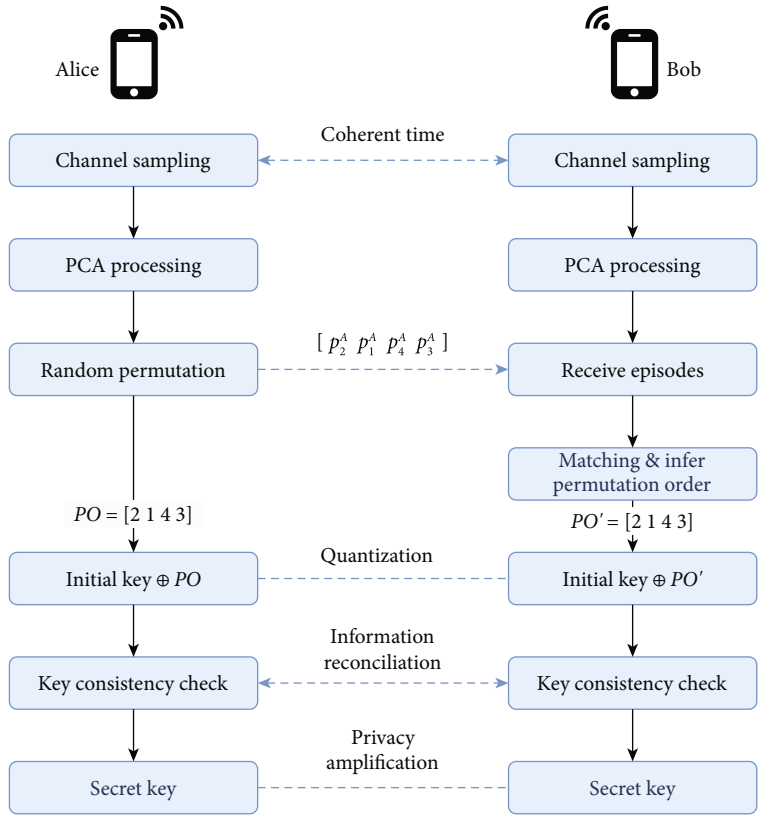



Figure 3: System flow chart.

Table 1: Notion list.

| Notation | Descriptions |
|---|---|
| $H$ | Channel measurement matrix |
| $W$ | Observation deviation |
| $Y$ | Channel measurement matrix after PCA processing |
| $R$ | Covariance matrix |
| $U$ | Eigenvector matrix |
| $\Lambda$ | Eigenvalue matrix |
| $T$ | Transformation matrix |
| $P$ | Segmented channel measurement sequence |
| PO | Permutation order |
| RS | Random perturbation sequence |
| IK | Initial key |
| PK | Perturbed key |

are eigenvector matrices. $\boldsymbol{R}^A$ and $\boldsymbol{R}^B$ are given by

$$\begin{aligned}\boldsymbol{R}^A &= \boldsymbol{U}^A \boldsymbol{\Lambda}^A \left(\boldsymbol{U}^A\right)^H,\\ \boldsymbol{R}^B &= \boldsymbol{U}^B \boldsymbol{\Lambda}^B \left(\boldsymbol{U}^B\right)^H.\end{aligned} \tag{5}$$

(2) Alice and Bob sort their eigenvalue matrices and eigenvector matrices in descending order of eigenvalues, respectively. The eigenvalue matrices after sorting are $\tilde{\boldsymbol{\Lambda}}^A, \tilde{\boldsymbol{\Lambda}}^B$, and the eigenvector matrices after sorting are $\tilde{\boldsymbol{U}}^A, \tilde{\boldsymbol{U}}^B$.

(3) Alice and Bob select the first $P$ eigenvectors of their eigenvector matrices to construct the transformation matrices $\boldsymbol{T}^A$ and $\boldsymbol{T}^B$, where $P$ is the number of eigenvectors agreed upon by Alice and Bob in advance

Alice and Bob transform their channel measurement matrices $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ by using the transformation matrices; the matrices after signal reconstruction are $\boldsymbol{Y}^A$, $\boldsymbol{Y}^B$, which are given by

$$\begin{aligned}\boldsymbol{Y}^A &= \left(\boldsymbol{T}^A\right)^H \boldsymbol{H}^A,\\ \boldsymbol{Y}^B &= \left(\boldsymbol{T}^B\right)^H \boldsymbol{H}^B,\end{aligned} \tag{6}$$

where $\boldsymbol{Y}^A = [\boldsymbol{y}_1^A, \boldsymbol{y}_2^A, \cdots, \boldsymbol{y}_K^A]$ and $\boldsymbol{Y}^B = [\boldsymbol{y}_1^B, \boldsymbol{y}_2^B, \cdots, \boldsymbol{y}_K^B]$ are the reconstructed signal matrices.

*3.2.2. Random Segmentation and Permutation.* To further increase the complexity and randomness of the collected channel measurements, we perform a random permutation on the channel measurement matrices $\boldsymbol{Y}^A$ and $\boldsymbol{Y}^B$. Figure 5 shows the effect of permutation on the CSI measurements. For ease of calculation and matching, Alice and Bob straighten $\boldsymbol{Y}^A$ and $\boldsymbol{Y}^B$ by row to make them two $1 \times S$, ($S$

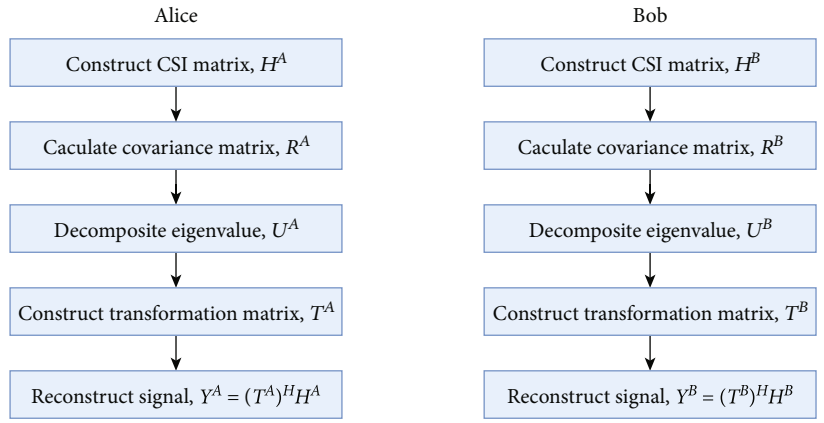



Figure 4: PCA processing steps of CSI.

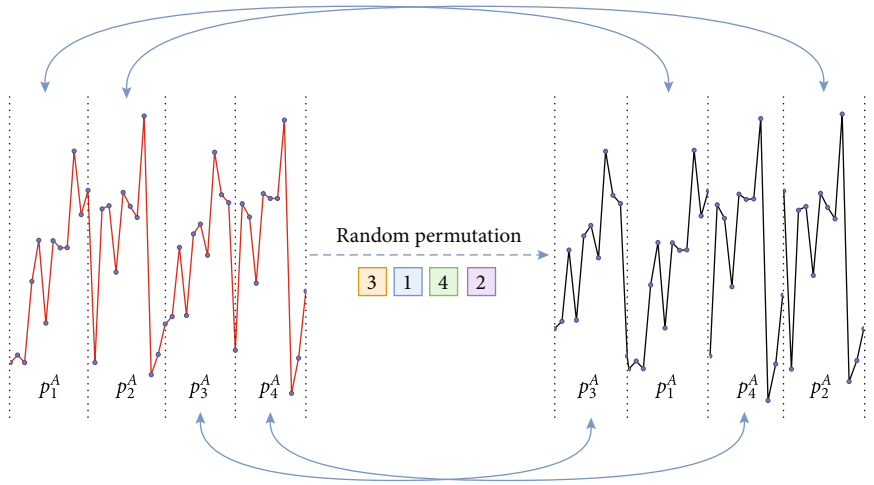



Figure 5: Random permutation on CSI.

$= P \times K$) dimensional vectors $\boldsymbol{v}^A$ and $\boldsymbol{v}^B$

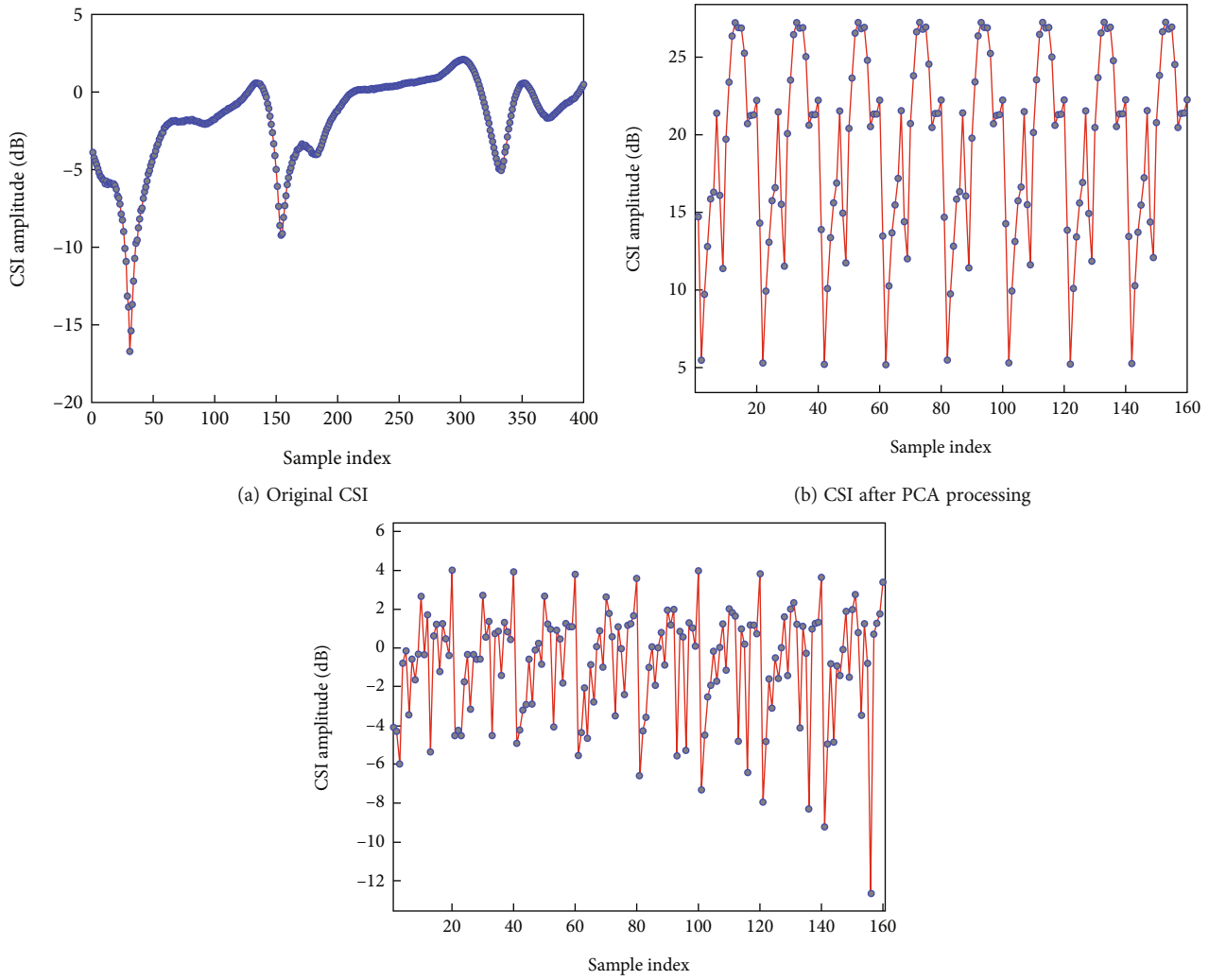
$$\begin{aligned} \boldsymbol{v}^A &= \mathrm{vec}\{\boldsymbol{Y}^A\}, \\ \boldsymbol{v}^B &= \mathrm{vec}\{\boldsymbol{Y}^B\}, \end{aligned} \tag{7}$$

where vec$\{\bullet\}$ is the straightening operation by row.

Alice and Bob segment their vectors $\boldsymbol{v}^A$ and $\boldsymbol{v}^B$ into $M$ episodes of the same length, $\boldsymbol{P}^A = [\boldsymbol{p}_1^A, \boldsymbol{p}_2^A, \cdots, \boldsymbol{p}_M^A]$ and $\boldsymbol{P}^B = [\boldsymbol{p}_1^B, \boldsymbol{p}_2^B, \cdots, \boldsymbol{p}_M^B]$, where $\boldsymbol{p}_m^A$ and $\boldsymbol{p}_m^B$ are the $m^{\text{th}}$ episode with length $L = S/M$. Given the segmented channel measurement sequence $\boldsymbol{P}^A$, Alice comes up with a permutation order $\mathbf{PO} = [k_1, k_2, \bullet\bullet\bullet, k_M]$ and applies permutation to $\boldsymbol{P}^A$ to create a new channel measurement sequence $\widehat{\boldsymbol{P}}^A = [\boldsymbol{p}_{k_1}^A, \boldsymbol{p}_{k_2}^A, \cdots, \boldsymbol{p}_{k_M}^A]$, where $k_m \in [1, M]$ is the original index of the episode $\boldsymbol{p}_{k_m}^A$ in $\boldsymbol{P}^A$. Alice then sends $\widehat{\boldsymbol{P}}^A$ to Bob without revealing the permutation order via the public channel, which potential attackers listen to. Each $\boldsymbol{p}_k^B$ in $\boldsymbol{P}^B$ can always find the reciprocal $\boldsymbol{p}_{k_m}^A$ in $\widehat{\boldsymbol{P}}^A$ even permutated due to channel reciprocity. Bob can infer the permutation order $\mathbf{PO} = [k_1, k_2, \bullet\bullet\bullet, k_M]$ of $\widehat{\boldsymbol{P}}^A$ by finding the perfect match between the episodes in $\widehat{\boldsymbol{P}}^A$ and $\boldsymbol{P}^B$ with the minimum discrepancy and use **PO** as part of the secret key. Bob performs the same permutation on $\boldsymbol{P}^B$ and obtains the new sequence $\widehat{\boldsymbol{P}}^B$ after inferring the **PO**. Since the original channel measurement sequence $\boldsymbol{P}^A$ was not made public, the permutation order $\mathbf{PO} = [k_1, k_2, \bullet\bullet\bullet, k_M]$ is a secret between Alice and Bob and is unknown to the potential attackers. Determining the permutation order is also equal to achieving a key agreement between Alice and Bob.

Then, Alice and Bob restore the vectors $\widehat{\boldsymbol{P}}^A$ and $\widehat{\boldsymbol{P}}^B$ to matrices $\widehat{\boldsymbol{Y}}^A$ and $\widehat{\boldsymbol{Y}}^B$ according to the initial segment length, where $\widehat{\boldsymbol{Y}}^A$ and $\widehat{\boldsymbol{Y}}^B$ can be seen as being randomly permutated. As the example shown in Figure 6, the channel measurement matrix after random permutation will have lower regularity and more complexity.

*3.3. Matching Algorithm.* In order to reduce the time cost of inferring the permutation order, we use the minimum weight bipartite graph matching to find the perfect match. Episodes in $\widehat{\boldsymbol{P}}^A$ and $\boldsymbol{P}^B$ are considered as vertices of a




(a) Original CSI

(b) CSI after PCA processing

(c) CSI after PCA and permutation

FIGURE 6: CSI after different preprocessing schemes.

weighted undirect graph $G$, and vertices are connected by edges. The edges only exist between the vertices of $\widehat{\boldsymbol{P}}^A$ and $\boldsymbol{P}^B$ in $G$ (i.e., no edge connects the vertices within $\widehat{\boldsymbol{P}}^A$ or $\boldsymbol{P}^B$ ). The weight of the edges can be denoted as $w_{A,B}(k_m, k) = \|\boldsymbol{p}^A_{k_m} - \boldsymbol{p}^B_k\|$, where $k_m, k \in [1, M]$ and $\|\bullet\|$ represents taking the absolute value. A perfect match in $G$ consists of a set of vertex-disjoint edges with every vertex of $G$. A perfect match can be always found in $G$ to satisfy the reciprocal mapping between the channel measurement matrices of Alice and Bob due to the channel reciprocity. The sum of weights of the match between Alice and Bob can be denoted as $W_{A,B} = \sum_{k_m,k}^{M} w_{A,B}(k_m, k)$, where $k_m, k \in [1, M]$. The minimum weight matching problem is to find the match with the smallest sum of weights. The minimum weight matching can be transformed into the maximum weight matching problem after converting the weight $w_{A,B}(k_m, k)$ to $\widehat{w}_{A,B}(k_m, k) = C - w_{A,B}(k_m, k)$, where $C \geq \max(w_{A,B}(k_m, k))$. We use the Kuhn-Munkres algorithm to solve the maximum weight matching problem. To minimize the summation of its associated weights, the following linear programming with integer constraints relaxation is formulated:

$$\begin{aligned} \min \quad & \sum_{k_m,k}^{M} (w_{A,B}(k_m, k) - l_A(k_m) - l_B(k)), \\ \text{s.t.} \quad & l_A(k_m) \geq 0, l_B(k) \geq 0, \\ & k_m, \quad k \in [1, M], \end{aligned} \tag{8}$$

where $l_A(k_m), l_B(k)$ are the feasible label with the value equal to the weight of the perfect match output by the algorithm as follows:

$$\begin{aligned} \max \quad & \sum_{k_m \in [1,M]} l_A(k_m) + \sum_{k \in [1,M]} l_B(k), \\ \text{s.t.} \quad & l_A(k_m) + l_B(k) \leq w_{A,B}(k_m, k), \quad \forall (k_m, k) \in E, \end{aligned} \tag{9}$$

where $E$ denotes all the edges in $G$. Any feasible prime label in a perfect match has a weight as large as the value of any

feasible dual-labeling. If $l_A(k_m) + l_B(k) = w_{A,B}(k_m, k)$, the edge $(k_m, k)$ is tight. A match is optimal if it only uses tight edges when given any dual feasible label.

To find the perfect match, a random feasible dual label $l$ is used to find a maximum-cardinality matching that uses tight edges. The process is over if the match is perfect. If not, the dual label is updated and the process continues until an optimal match is found. After the graph matching, Bob infers the permutation order $\mathbf{PO}' = [k_1', k_2', \bullet\bullet\bullet, k_M']$, where $\mathbf{PO} = \mathbf{PO}'$.

*3.4. Wireless Key Generation Based on Random Perturbation.* The wireless key generation process based on permutation and matching is divided into the following three steps: obtaining the initial secret key, generating and splicing the random perturbation sequence, and performing the XOR operation.

In this paper, after the preprocessing, different components of channel measurements have different SNRs, which can be expressed as

$$\mathrm{SNR}_i = \frac{\lambda_i^2}{\sigma_n^2}. \tag{10}$$

$\mathrm{SNR}_i$ represents the SNRs of different components. As the index of components increases, the SNR decreases. To make full use of the high SNR of dominant components, we employ flexible quantization levels in the quantization algorithm to quantify the initial keys.

The first step is to obtain the initial keys. Alice and Bob get their respective initial keys $\mathrm{IK}^A$ and $\mathrm{IK}^B$ after the quantization process on their channel measurement matrices $\widehat{\boldsymbol{Y}}^A$ and $\widehat{\boldsymbol{Y}}^B$. The length of the initial keys is $L_1$.

The second step is to generate the random perturbation sequence through the negotiated permutation order $\mathbf{PO} = [k_1, k_2, \bullet\bullet\bullet, k_M]$. First, convert **PO** into a binary bit sequence RS, and the length of the converted bit sequence RS is $L_2 = c \times M$, where $c$ is the length of each binary bit sequence converted by $k_m, k_m \in [1, M]$ in **PO**. RS then needs to be spliced into $RS'$. Repeatedly splicing the stochastic perturbation sequence until it is equal to the key length $L_1$, that is,

$$L_1 = k \times L_2 + k', \tag{11}$$

where $k$ is a positive integer and $0 < L_2 < L_1$.

The last step is to XOR the random perturbation sequence $RS'$ and the initial secret keys. Alice and Bob perform XOR operation between their initial key $\mathrm{IK}^A, \mathrm{IK}^B$ and the perturbation sequence $\mathrm{RS}'$ to get the perturbed key $\mathrm{PK}^A$ and $\mathrm{PK}^B$, which are given by

$$\begin{aligned} \mathrm{PK}^A &= \mathrm{IK}^A \oplus \mathrm{RS}', \\ \mathrm{PK}^B &= \mathrm{IK}^B \oplus \mathrm{RS}'. \end{aligned} \tag{12}$$

Compared with the key sequence before random perturbation, the number of secret keys does not increase. However, the method based on random perturbation reduces the correlation between the two adjacent sets of secret keys, so it can effectively increase the KGR.

Then Alice and Bob perform information reconciliation on their perturbed keys $\mathrm{PK}^A$ and $\mathrm{PK}^B$. The main purpose of information reconciliation is to correct the inconsistent bits in the key bit sequences without divulging the key information as much as possible. After information reconciliation, Alice and Bob will agree on an error-free secret key.

Table 2: Simulation parameters.

| Parameter | Value |
|---|---|
| Channel model | TGn |
| Scenario | NLOS |
| SNR | 40 dB |
| Bandwidth | 20 MHz |
| PSDU length | 20 bytes |
| Carrier number | 56 |
| RMS delay spread | 15 ns |
| Channel coding | BCC |
| Maximum delay | 80 ns |
| Sampling interval | 0.5 ms |

## 4. Performance Evaluation

To evaluate the performance of our proposed scheme, we conduct numerical simulations. We build the simulation model based on a Matlab implementation of the TGn multipath fading channel. The detailed parameters are summarized in Table 2. Alice and Bob are randomly distributed, and the distance between them is greater than or equal to five meters. We focus on the non-line-of-sight (NLOS) scenario. An OFDM model with 56 subcarriers is utilized. We sample 400 independent channel vectors to perform the key generation process.

In this section, we evaluate the performance of the RPP-WKG scheme and compare it with the wireless key generation without processing (named as "Initial"), and with the wireless key generation scheme based on PCA and without random permutation and perturbation (named as "PCA-WKG"). We evaluate the key performance from 4 aspects: the KGR, the KMR, the correlation between the secret keys, and the randomness of the keys.

*4.1. Key Generation Rate.* The KGR reflects the speed of the wireless key generation. The actual wireless key generation system has high requirements on the KGR. If the KGR is too low, the time cost required for wireless key generation will be too high, which is not suitable for practical applications. In this section, we test the KGR of the RPP-WKG scheme in the case of different SNRs. The test results are shown in Figure 7. As the SNR increases, the KGR gradually increases. The KGR can reach 480 bits/packet when the SNR is 45 dB. We also compare the KGR performance of the RPP-WKG scheme with the PCA scheme without random permutation and perturbation. The comparison results show

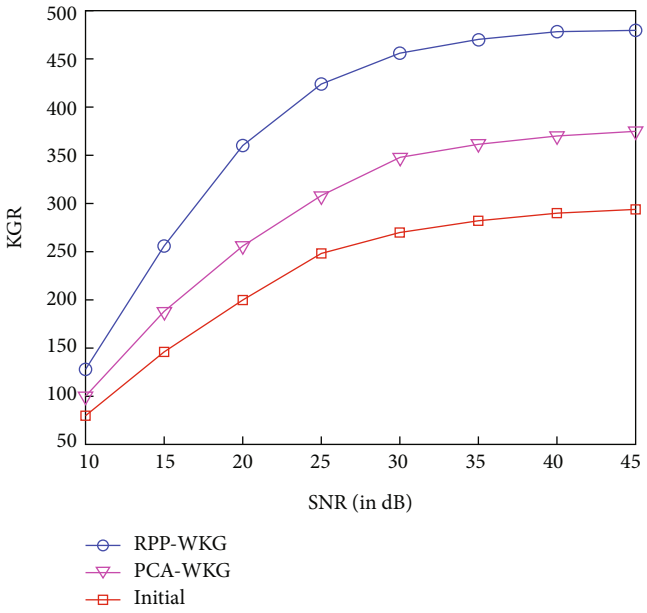



Figure 7: KGR performance under the impact of different SNRs.

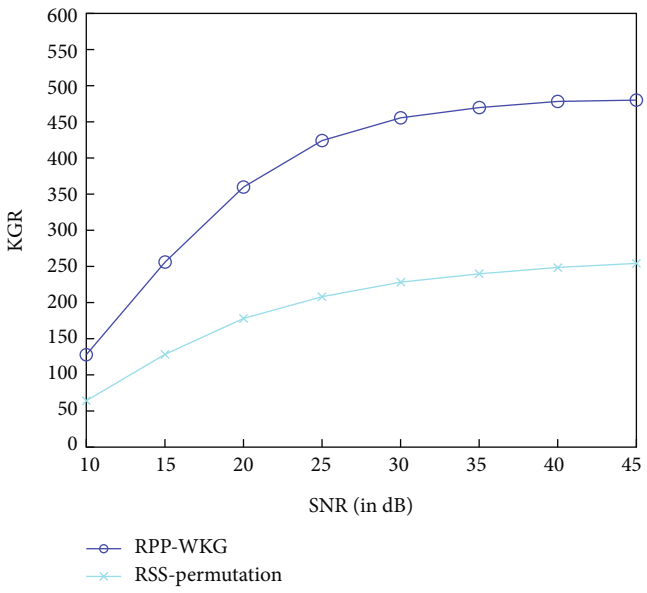



Figure 8: KGR performance for different channel measurements.

that the KGR can be further improved by the random permutation and perturbation scheme.

In addition, we compare the performance of our proposed RPP-WKG scheme with the scheme based on RSS permutation. Based on the comparison results shown in Figure 8, our RPP-WKG scheme will achieve a higher KGR by using CSI as the channel measurements than the traditional RSS permutation-based scheme.

*4.2. Key Mismatch Rate.* The KMR reflects the inconsistency rate of the secret keys quantified, respectively, by Alice and Bob. Due to the influence of ambient noise and other factors,

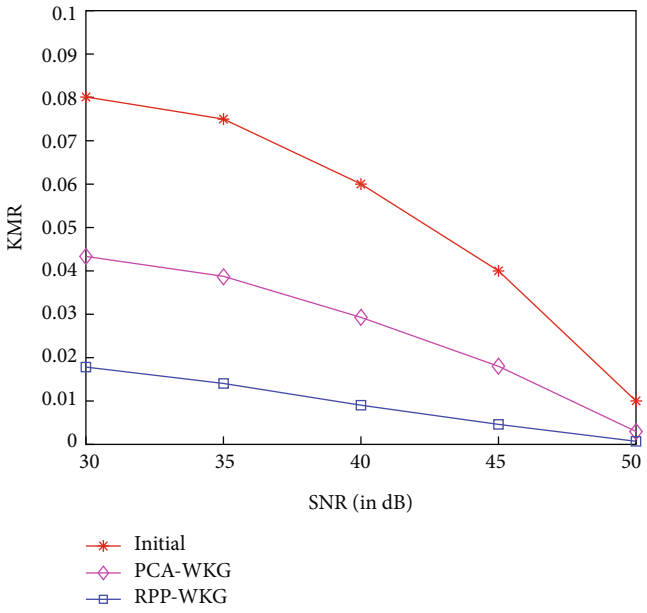



Figure 9: KMR performance under the impact of different SNRs.

there will be certain errors in the bit sequences quantized by Alice and Bob according to their respective CSI. Figure 9 shows the KMR performance of the initially generated secret keys and the secret keys after the RPP-WKG scheme. As the SNR increases, the KMR gradually decreases. According to the comparison results, the RPP-WKG scheme achieves a lower KMR. Figure 10 shows the KMR performance of secret keys generated by the RPP-WKG scheme under the impact of different episode numbers. As the number of permutation episodes increases, the KMR of the secret keys will increase significantly.

*4.3. Correlation between the Secret Keys.* The correlation between the secret keys represents the degree of linear correlation between adjacent sets of keys. We use the Pearson correlation coefficient [35–37] to calculate the correlation between keys. The Pearson correlation coefficient is defined as

$$\rho_{XY} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\sqrt{E(Y^2) - E^2(Y)}}, \quad (13)$$

where $X$ and $Y$ are the two sets of secret key sequences.

The correlation coefficient between two sets of secret keys is a value between -1 and 1. The stronger the correlation between the two sets of secret keys, the closer the absolute value of the correlation coefficient is to 1. If the correlation coefficient is equal to 0, it indicates that there is no linear correlation between the two sets of secret keys.

In this section, we first calculate the correlation coefficient between the secret keys of the three schemes. We display the calculation results in the form of heat maps. The horizontal and vertical coordinates represent the index number of the keys, and the colour of each dot represents the correlation between the secret keys. The yellower the colour, the higher the correlation between the secret keys. The bluer

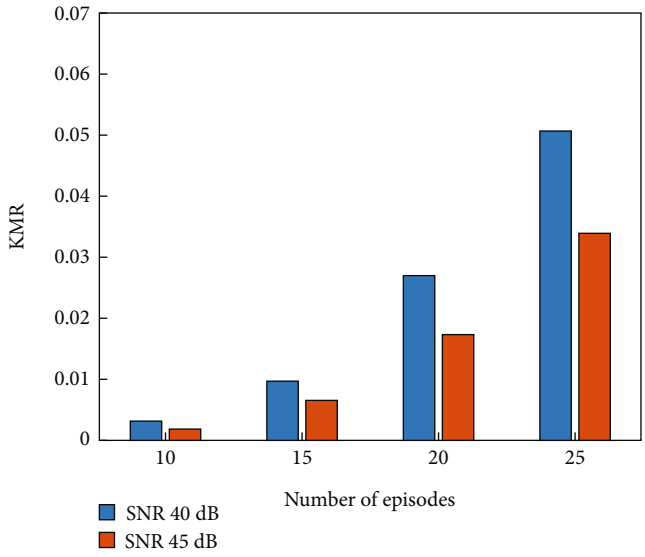



Figure 10: KMR performance of RPP-WKG scheme under the impact of different episode numbers.

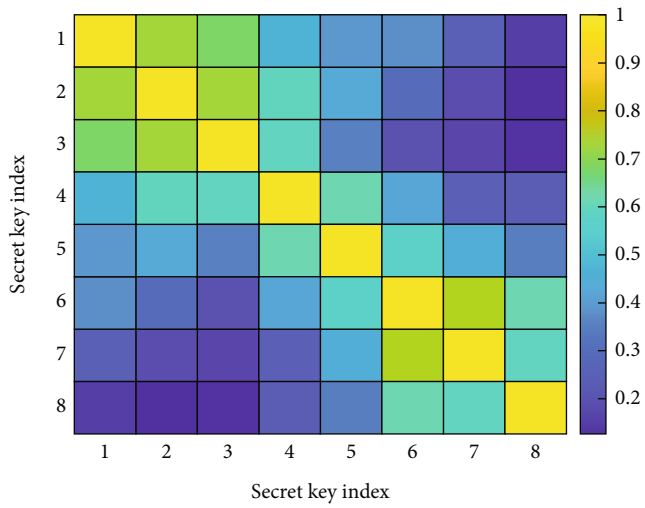



Figure 11: Correlation between initial secret keys.

the colour, the lower the correlation between the secret keys. We test the correlation between the initial secret keys, the secret keys after PCA processing, and the secret keys after random permutation and perturbation. According to the test results, the correlation between initial secret keys is the highest in Figure 11, and the correlation between secret keys after random permutation and perturbation is the lowest. Figure 12 shows that the secret keys quantized by the CSI after PCA processing can obtain a lower correlation than the initial secret keys. Figure 13 shows the advantage of the random permutation and perturbation scheme in the process of key generation. The test results reflect the RPP-WKG scheme has an obvious effect on reducing the correlation between secret keys.

We also calculate how the correlation coefficient changes as the number of episodes increases. As shown in Figure 14, once the CSI is randomly permuted and perturbed, the correlation coefficient between secret keys will drop significantly. As the number of permutation episodes increases, the correlation coefficient between secret keys will slowly decrease. When the number of permutation episodes is greater than 10, a good correlation reduction effect can be obtained. Considering that as the number of permutation episodes increases, the time cost to find the correct permutation order using the matching algorithm will also increase, and the number of permutation episodes should not be set too large.

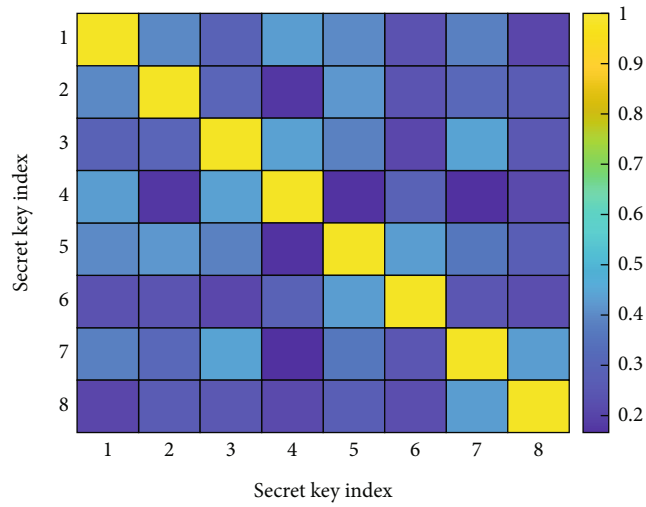



Figure 12: Correlation between secret keys after PCA processing.

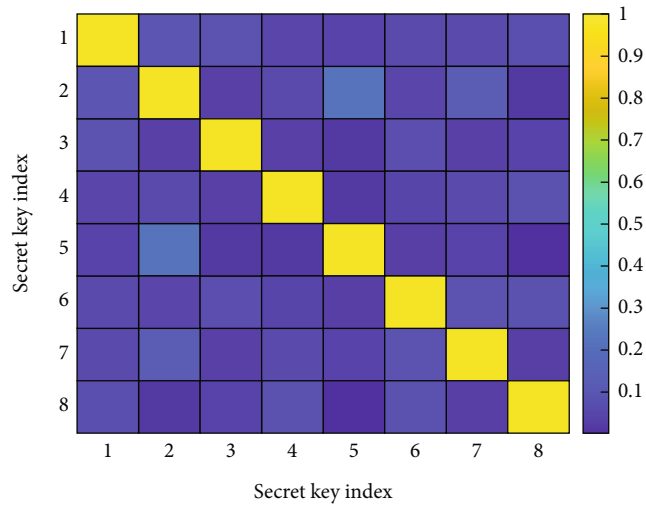



Figure 13: Correlation between secret keys of RPP-WKG scheme.

*4.4. Randomness of Secret Keys.* The randomness of the key is an important standard to measure the performance of the secret keys. The definition of key randomness is the uniformity of the distribution of 0 and 1 in the generated secret keys. The higher the randomness of the key, the more difficult it is for the eavesdropper to guess the key. To ensure that the secret keys generated are substantially random, the standard randomness test suite from NIST is employed to

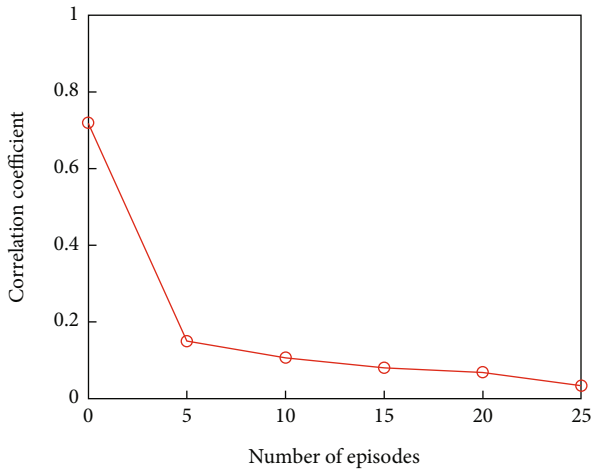



Figure 14: Correlation with the number of episodes.

Table 3: NIST randomness test of RPP-WKG scheme.

| Test | $p$ value |
|---|---|
| Frequency test | 0.729034 |
| Frequency test within a block | 0.731615 |
| Run test | 0.902544 |
| Longest run of ones in a block | 0.773102 |
| Discrete Fourier transform | 0.791081 |
| Nonoverlapping temple match | 0.999959 |
| Serial test | 0.561915 |
| Approximate entropy test | 1.0 |
| Cumulative sums (forward) test | 0.700062 |
| Cumulative sums (reverse) test | 0.407770 |

Table 4: NIST randomness test of different schemes.

| Test | Initial | PCA-WKG | RSS-permutation | RPP-WKG |
|---|---|---|---|---|
| 1 | 0.204023 | 0.448213 | 0.446671 | 0.729034 |
| 2 | 0.457833 | 0.385534 | 0.489325 | 0.731615 |
| 3 | 0.170472 | 0.395013 | 0.576431 | 0.902544 |
| 4 | 0.057768 | 0.731615 | 0.663982 | 0.773102 |
| 5 | 0.063689 | 0.426776 | 0.512378 | 0.791081 |
| 6 | 0.678439 | 0.827952 | 0.843564 | 0.999959 |
| 7 | 0.343128 | 0.498961 | 0.378615 | 0.561915 |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 |
| 9 | 0.211935 | 0.368282 | 0.397446 | 0.700062 |
| 10 | 0.166529 | 0.297799 | 0.337512 | 0.407770 |

verify the effectiveness of the secret keys extracted after the wireless key generation scheme based on permutation and perturbation [38, 39]. The output result of each test is an indicator called the $p$ value. A tested secret key sequence passes a test when the $p$ value is greater than the threshold, usually chosen as 0.01. We run 10 NIST tests on the secret keys generated on the RPP-WKG scheme, as listed in Table 3. All the results pass the tests, indicating the randomness of the generated secret keys is sufficient for practical key generation. In addition, in this section, we also compare the randomness of the initial secret keys, the secret keys quantified after PCA processing, the secret keys generated by RSS, and the secret keys generated by the RPP-WKG scheme. Table 4 shows the comparison results: the secret key generation scheme based on random permutation and perturbation has obvious advantages in the tests.

## 5. Conclusions

In this paper, we propose an efficient wireless key generation scheme based on random permutation and perturbation, which achieves high randomness and a high KGR between the legitimate users, Alice and Bob, in a quasistatic environment. In the proposed RPP-WKG scheme, we can mitigate the impact of the quasistatic channel and achieve secret keys with high randomness and low correlation. The efficient and secure permutation method allows legitimate users to perform the same random permutation on their respective CSI to acquire new random sources with random and great fluctuations. The minimum weight-based matching method helps legitimate users to obtain an agreement on the permutation order of CSI without revealing it. The random perturbation generation method based on the permutation order improves the randomness and reduces the correlation of secret keys. Simulation results show that the proposed RPP-WKG scheme can efficiently improve the randomness and KGR of the generated secret keys in a quasistatic environment.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the National Key R&D Program of China (2020YFE0200600).

## References

[1] B. Mao, Y. Kawamoto, and N. Kato, "Ai-based joint optimization of qos and security for 6G energy harvesting internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7032–7042, 2020.

[2] H. Fang, X. Wang, and L. Hanzo, "Learning-aided physical layer authentication as an intelligent process," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2260–2273, 2018.

[3] H. Fang, X. Wang, and S. Tomasin, "Machine learning for intelligent authentication in 5g and beyond wireless

networks," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 55–61, 2019.

[4] A. Bunin, Z. Goldfeld, H. H. Permuter, S. S. Shitz, P. Cuff, and P. Piantanida, "Key and message semantic-security over state-dependent channels," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1541–1556, 2018.

[5] H. Mack and T. Schroer, "Security midlife crisis: building security in a new world," *IEEE Security & Privacy*, vol. 18, no. 4, pp. 72–74, 2020.

[6] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 2011, https://www.amazon.com/Cryptography-Network-Security-Principles-Practice/dp/0133354695.

[7] K. Zeng, "Physical layer key generation in wireless networks: challenges and opportunities," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 33–39, 2015.

[8] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radiotelepathy: extracting a secret key from an unauthenticated wireless channel," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pp. 128–139, San Francisco, California USA, 2008.

[9] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 17–30, 2010.

[10] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, San Diego, CA, USA, 2010.

[11] J. Zhang, T. Q. Duong, A. Marshall, and R. Woods, "Key generation from wireless channels: a review," *IEEE Access*, vol. 4, pp. 614–626, 2016.

[12] A. Salam, M. C. Vuran, and S. Irmak, "A statistical impulse response model based on empirical characterization of wireless underground channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 5966–5981, 2020.

[13] W. Zhang, J. Chen, Y. Kuo, and Y. Zhou, "Transmit beamforming for layered physical layer security," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9747–9760, 2019.

[14] R. Chopra, C. R. Murthy, and R. Annavajjala, "Physical layer security in wireless sensor networks using distributed co-phasing," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2662–2675, 2019.

[15] Á. Vázquez-Castro and M. Hayashi, "Physical layer security for rf satellite channels in the finite-length regime," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 981–993, 2018.

[16] Z. Li, Q. Pei, I. Markwood, Y. Liu, and H. Zhu, "Secret key establishment via rss trajectory matching between wearable devices," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 802–817, 2017.

[17] H. Zhao, Y. Zhang, X. Huang, Y. Xiang, and C. Su, "A physical-layer key generation approach based on received signal strength in smart homes," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 4917–4927, 2021.

[18] G. Margelis, X. Fafoutis, G. Oikonomou, R. Piechocki, T. Tryfonas, and P. Thomas, "Physical layer secret-key generation with discreet cosine transform for the internet of things," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, 2017.

[19] H. Liu, Y. Wang, Y. Ren, and Y. Chen, "Bipartite graph matching based secret key generation," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10, Vancouver, BC, Canada, 2021.

[20] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, 2014.

[21] Y. Xi, A. Burr, J. Wei, and D. Grace, "A general upper bound to evaluate packet error rate over quasi-static fading channels," *IEEE Transactions on Wireless Communications*, vol. 10, no. 5, pp. 1373–1377, 2011.

[22] T. Lu, L. Chen, J. Zhang, K. Cao, and A. Hu, "Reconfigurable intelligent surface assisted secret key generation in quasi-static environments," *IEEE Communications Letters*, vol. 26, no. 2, pp. 244–248, 2021.

[23] M. He, J. Xu, W. Xu, H. Shen, N. Wang, and C. Zhao, "RIS-assisted quasi-static broad coverage for wideband mmwave massive MIMO systems," 2022, https://arxiv.org/abs/2203.00400.

[24] Y. Huang, L. Jin, H. Wei, Z. Zhong, and S. Zhang, "Fast secret key generation based on dynamic private pilot from static wireless channels," *China Communications*, vol. 15, no. 11, pp. 171–183, 2018.

[25] N. Aldaghri and H. Mahdavifar, "Physical layer secret key generation in static environments," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2692–2705, 2020.

[26] G. Li, A. Hu, J. Zhang, L. Peng, C. Sun, and D. Cao, "High-agreement uncorrelated secret key generation based on principal component analysis preprocessing," *IEEE Transactions on Communications*, vol. 66, no. 7, pp. 3022–3034, 2018.

[27] Y. Wei, K. Zeng, and P. Mohapatra, "Adaptive wireless channel probing for shared key generation based on pid controller," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1842–1852, 2012.

[28] Y. Peng, P. Wang, W. Xiang, and Y. Li, "Secret key generation based on estimated channel state information for tdd-ofdm systems over fading channels," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5176–5186, 2017.

[29] C. Chen and M. A. Jensen, "Secret key establishment using temporally and spatially correlated wireless channel coefficients," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 205–215, 2010.

[30] D. Chen, Z. Qin, X. Mao, P. Yang, Z. Qin, and R. Wang, "Smokegrenade: an efficient key generation protocol with artificial interference," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1731–1745, 2013.

[31] Y. Liu, S. C. Draper, and A. M. Sayeed, "Exploiting channel diversity in secret key generation from multipath fading randomness," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1484–1497, 2012.

[32] S. Wang and C. Li, "Discrete double-bit hashing," *IEEE Transactions on Big Data*, vol. 8, pp. 482–494, 2019.

[33] R.-C. Tu, X.-L. Mao, B. Ma et al., "Deep cross-modal hashing with hashing functions and unified hash codes jointly learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, pp. 560–572, 2020.

[34] Q. Wang, H. Su, K. Ren, and K. Kim, "Fast and scalable secret key generation exploiting channel phase randomness in wireless networks," in *2011 Proceedings IEEE INFOCOM*, pp. 1422–1430, Shanghai, China, 2011.

[35] T. Peng, W. Dai, and M. Z. Win, "Efficient and robust physical layer key generation," in *MILCOM 2019-2019 IEEE Military*

*Communications Conference (MILCOM)*, pp. 1–6, Norfolk, VA, USA, 2019.

[36] Z. Ji, Z. He, Y. Zhang, and X. Chen, "A two-step decorrelation method on time-frequency correlated channel for secret key generation," in *2018 IEEE wireless communications and networking conference (WCNC)*, pp. 1–6, Barcelona, Spain, 2018.

[37] F. Passerini and A. M. Tonello, "Secure phy layer key generation in the asymmetric power line communication channel," *Electronics*, vol. 9, no. 4, p. 605, 2020.

[38] H. Tan, D. Ostry, and S. Jha, "Exploiting multiple side channels for secret key agreement in wireless networks," in *Proceedings of the 19th International Conference on Distributed Computing and Networking*, pp. 1–10, Varanasi, India, 2018.

[39] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, Booz-Allen and Hamilton Inc Mclean Va, Tech. Rep, 2001.