Hindawi
Security and Communication Networks
Volume 2023, Article ID 4911338, 14 pages
https://doi.org/10.1155/2023/4911338

Research Article

A Physical Layer Key Generation Scheme Based on Deep Learning Compensation and Balanced Vector Quantization

Liquan Chen,[1,2] Zhaofa Chen,[1] Tianyu Lu,[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 16 August 2022; Revised 13 February 2023; Accepted 28 February 2023; Published 8 April 2023

Academic Editor: Je Sen Teh

Channel reciprocity is the foundation for physical layer key generation, which is influenced by noise, hardware impairments, and synchronization offsets. Weak channel reciprocity will result in a high key disagreement rate (KDR). The existing solutions for improving channel reciprocity cannot achieve satisfactory performance improvements. Furthermore, the existing quantization algorithms generally use one-dimensional channel features to quantize and generate secret keys, which cannot fully utilize channel information. The multidimensional vector quantization technique also needs to improve in terms of randomness and time complexity. This paper proposes a physical layer key generation scheme based on deep learning and balanced vector quantization. Specifically, we build a channel reciprocity compensation network (CRCNet) to learn the mapping relationship between Alice and Bob's channel measurements. Alice compensates for channel measurements via a trained CRCNet to reduce channel measurement errors between legitimate users and enhance channel reciprocity. We also propose a balanced vector quantization algorithm based on integer linear programming (ILP-BVQ). ILP-BVQ reduces the time complexity of quantization on the basis of ensuring key randomness and a low KDR. Simulation results showed that the proposed CRCNet performs better in terms of channel reciprocity and KDR, while the proposed ILP-BVQ algorithm improves time consumption and key randomness.

1. Introduction

With the development of wireless communication and the commercialization of 5G technology, more sensitive information (such as mobile banking, e-payments, and medical data) is transmitted via the wireless medium [1, 2]. Due to the broadcast nature of the wireless channel, anyone with the proper receiving equipment can monitor a wireless transmission line and eavesdrop maliciously. Generally, data is encrypted by various encryption algorithms. Traditional encryption algorithms are implemented based on computational complexity and usually require a key management center, which is challenging in terms of key distribution and management. With the advancement of quantum computing and IoT applications, traditional encryption algorithms have gradually become unsuitable. Recently, physical layer security schemes based on the characteristics of the wireless environment have attracted much attention, including physical layer key generation techniques [2–5], authentication schemes based on wireless environment characteristics [6, 7], etc. This paper works on physical layer key generation between peer-to-peer users. Two legitimate nodes use channel variations between nodes to generate shared keys without the involvement of third-party entities, potentially achieving information-theoretic security.

Physical layer key generation schemes operate under the assumptions of wireless channel reciprocity, time variability, and spatial decoupling [8]. Wireless channel reciprocity ensures that two communicating parties generate the same key through the quantization process, while time-varying and spatial decorrelation ensure the randomness of the shared key. Current physical layer key generation schemes are shown in Figure 1, which consist of four steps: channel measurement, quantization, message coordination, and privacy amplification [4]. The legitimate users, Alice and Bob, obtain channel state information (CSI) via channel

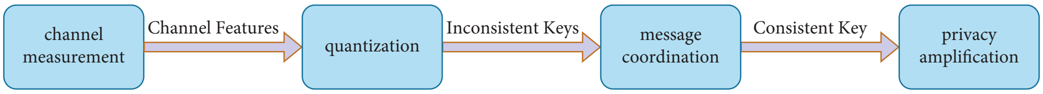


Figure 1: Flowchart of the classical physical layer key generation scheme.

measurement. In the quantization process, Alice or Bob uses quantization algorithms to map the channel measurements into a string of binary bits. The information reconciliation process further ensures the consistency of shared keys through error correction codes, while the privacy amplification process aims to increase key entropy.

The scheme shown in Figure 1 is the basic flow of existing physical layer key generation techniques, and there are still some shortcomings waiting for improvement in all four steps. During the real-world channel probing process, the reciprocity of channel measurements between two legitimate users is usually weak due to nonsimultaneous measures, channel noise, and hardware impairments [9]. This leads to a high KDR between the keys generated by each of the two legitimate users. Additionally, most of the physical layer key generation schemes use only a unique channel feature for quantization [10–12], resulting in incomplete channel information utilization and the inability to achieve optimal performance. Wireless key generation is enabled by joining multidimensional channel features, which in theory, can improve information utilization and increase the key generation rate (KGR). So, this paper focuses on the two problems of weak channel reciprocity in the channel measurement step and multidimensional vector quantization in the quantization step.

Firstly, to solve the problem of weak channel reciprocity, researchers adopt artificially designed reciprocity features or linear transformation methods to improve channel reciprocity, such as low-pass filter [13], DCT [14], and PCA [15]. These manually designed feature extraction methods are mainly based on personal observation or experience and were designed and implemented for a special channel model. Therefore, they cannot be flexibly applied to various models or practical environments and have significant limitations.

Deep learning can extract high-performance features without predefining channel model statistics. Compared with manually designed feature extraction-based algorithms, deep learning-based key generation methods are not limited by the channel model and can achieve superior performance. However, few studies have focused on applying deep learning to improve the correlation between channel measurements of weak reciprocal channels. Therefore, we intend to use deep learning's powerful learning capability to compensate for imperfect channels in real-world wireless systems.

Then, researchers adopt vector quantization to solve the problem of low information utilization in scalar quantization. However, the existing vector quantization algorithms, such as [16–18], have defects in key randomness, KDR, or time complexity. Therefore, we propose a new vector quantization algorithm in this paper to improve performance compared to the existing vector quantization algorithms.

1.1. Related Work

1.1.1. Review of Channel Reciprocity Improvement.

Researchers adopt manually designed feature extraction algorithms to extract reciprocal features from weak reciprocal channels. One direction of research in feature extraction methods is to use traditional linear transforms such as the discrete cosine transform (DCT) [14], principal component analysis (PCA) [15, 19], and wavelet transform [10, 20]. Another research direction is to extract nonlinear features from the original channel response, such as amplitude, phase, and power delay distribution [13, 21, 22]. These artificially designed feature extraction methods have shortcomings in performance enhancement and applicability.

In recent decades, deep learning has been gradually applied to wireless communications and networks [23, 24], including channel estimation [25], modulation classification [26], and resource allocation [27]. With its excellent performance, deep learning has also been applied to enhance network security such as the fields of authentication, physical layer key generation, etc. Fang et al. [6] proposed an adaptive authentication scheme based on time-varying environments and intelligent machine learning-assisted processes. Zhang et al. [28] use fully connected neural networks to implement the mapping of nonreciprocal uplink and downlink channels in FDD systems. Letafati et al. [29] utilized RNNs to compensate for discrepancies in observations from both sides caused by injected signals from man-in-the-middle attacks and flaws in legitimate transceivers. Han et al. [30] model communication as an end-to-end autoencoder to improve channel reciprocity and perform better than the original scheme. Letafati et al. [31] utilize echo-state networks to compensate for the observation mismatch between legitimate communicating parties caused by unbalanced hardware impairments. Guan et al. [32] propose a feed-forward neural network-based prediction (NNBP) algorithm to enhance key consistency.

1.1.2. Review of Vector Quantization.

Chen et al. [16] used the kmeans clustering algorithm for vector quantization, but the limitations of the kmeans algorithm will lead to weak randomness in shared keys. Hong et al. [17] used a balanced kmeans vector quantization algorithm that adds a limit of an equal number of samples in each cluster to the kmeans clustering algorithm to improve the randomness of the generated keys. However, the time consumption of this algorithm is too high, and the time complexity of the clustering process for sample assignment is $O(n^3)$. Han et al. [18] proposed a heuristic-based balanced vector quantization algorithm that assigns samples to clusters based on their distances to the corresponding cluster centers. However, instead of immediately assigning all samples to their nearest centroid, samples are assigned one by one while sequentially ignoring already full clusters. The vector quantization algorithm proposed in [18] can achieve the goal of balanced cluster quantization. The time complexity of this algorithm is $O(nk\log_2 n)$. However, the algorithm cannot guarantee that the clusters do not overlap, and one cluster may have a high dispersion and contain points from another cluster within its bounding volume. This leads to a high disagreement rate for the quantized shared keys.

In [33], a balanced clustering algorithm based on integer linear programming was proposed. The algorithm is similar to kmeans in that both consist of initialization and iteration phases, with assignment and update steps for each iteration. The iterative phase is implemented by formulating the sample assignment problem with constant size constraints as an integer linear program and solving it by a simplex algorithm. The time complexity of this algorithm is only $O(n^{1.7})$. In addition, the clustering algorithm minimizes cluster dispersion while satisfying the constant size constraint, providing compact and nonoverlapping clusters.

1.2. Main Contribution. Inspired by these works, this paper applies deep learning and balanced vector quantization to the physical layer key generation field. Firstly, we propose a method that uses deep learning to compensate for weak reciprocal channel measurements for efficient key generation. Specifically, we designed a channel reciprocal compensation network (CRCNet), a one-dimensional convolutional neural network (CNN) driven by the CSI of TDD orthogonal frequency division multiplexing (OFDM) systems. In the training phase, the model adaptively learns the mapping relationship between the channel measurements from two legitimate users by using the CSI data collected in the channel probing phase. Then, we extend the balanced clustering algorithm proposed in the paper [33] to the quantization process of physical layer key generation, called the balanced vector quantization algorithm based on integer linear programming (ILP-BVQ). ILP-BVQ adds a cluster balancing mechanism to guarantee the randomness of shared keys and solves the sample allocation problem in the balancing quantization process by integer linear programming. It greatly reduces the time complexity while ensuring key randomness and a low KDR. Our main contributions are presented as follows:

(1) We design a CRCNet for weak reciprocal channels. Without the need to know the statistical distribution of the channel response, the model trained with the collected CSI data can compensate for the channel measurements to obtain highly correlated channel measurements and enhance channel reciprocity.

(2) Based on existing balanced vector quantization algorithms, we propose a balanced vector quantization algorithm based on integer linear programming. The algorithm lowers the time complexity of the balanced vector quantization process from $O(n^3)$ to $O(n^{1.7})$ and ensures the randomness of the generated keys and the KDR.

(3) A practical key generation scheme based on the proposed CRCNet and ILP-BVQ algorithm is constructed. Simulation experiments verify the performance of the proposed scheme. Compared to the existing schemes, our method performs better regarding channel reciprocity, KDR, and time consumption.

The remaining sections of this paper are organized as follows. Section 2 illustrates the system model. Section 3 describes the proposed channel reciprocity compensation network. Section 4 describes the proposed balanced vector quantization algorithm. Then, the proposed scheme's performance is evaluated by simulation experiments in Section 5. Section 6 concludes the paper.

2. System Model

The system model consists of two legitimate users, Alice and Bob, and an eavesdropper, Eve, as shown in Figure 2. Each user is equipped with a single antenna. Alice and Bob aim to generate a consistently shared key using the unauthenticated wireless channel. Eve can eavesdrop on all communications between the communicating parties. We assume that Eve cannot interfere with the communications during the channel measurement process. Assume that all three parties in the model know the common pilot information. Let x be the common pilot signal and h_{ij} denote the channel vector from device i to device j, where $i, j \in \{A, B, E\}$. The signals received by Alice and Bob can be expressed as follows:

$$\begin{aligned} y_A &= h_{\mathrm{BA}}x + n_A, \\ y_B &= h_{\mathrm{AB}}x + n_B, \end{aligned} \tag{1}$$

where n_i is the additive Gaussian white noise, $n_i \sim \mathscr{CN}(0, \sigma_n^2)$.

Theoretically, the channels between legitimate users are reciprocal in coherent time, $h_{\mathrm{BA}} = h_{\mathrm{AB}}$, and two legitimate users obtain the same key by quantizing the reciprocal channel measurements. However, in the practical scenario, the channel estimation of legitimate users includes noise, device hardware impairment, and synchronization error. Alice and Bob's actual channel measurement can be expressed as follows:

$$\begin{aligned} \widehat{h}_{\mathrm{BA}} &= h_{\mathrm{BA}} + \epsilon_1, \\ \widehat{h}_{\mathrm{AB}} &= h_{\mathrm{AB}} + \epsilon_2, \end{aligned} \tag{2}$$

where ϵ_i denotes the complex Gaussian estimation error of zero mean with variance σ_i^2 at node i.

We verify the channel reciprocity between two legitimate users through preliminary experimental simulations. The experimental parameters and environmental settings are the same as in the results section. Figure 3(a) shows the comparison of the RSSI between legitimate users without any preprocessing operation. Figure 3(b) shows the comparison for the CSI between legitimate users without any preprocessing

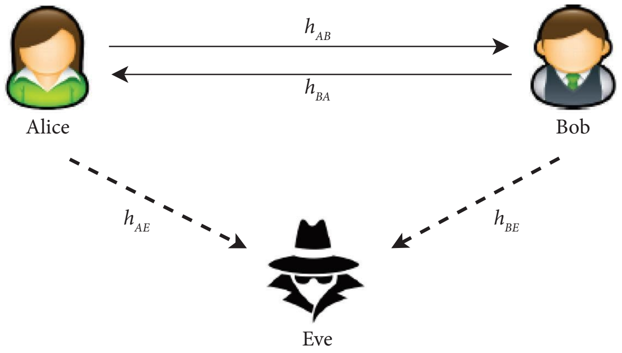


Figure 2: System model.

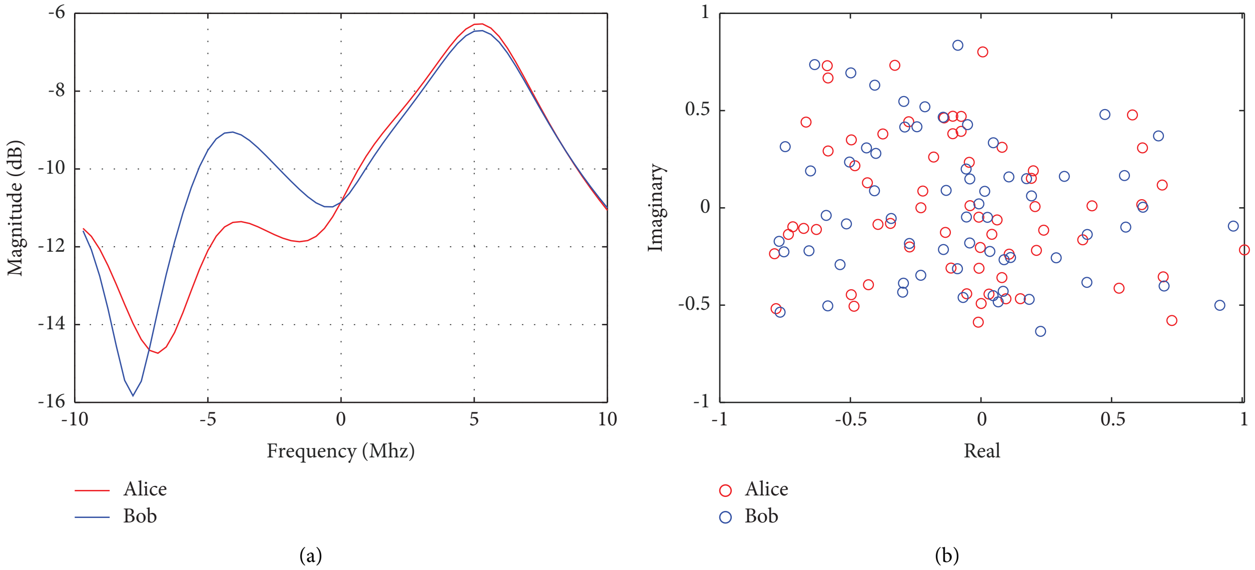


Figure 3: Channel measurements. (a) RSSI without DL. (b) CSI without DL.

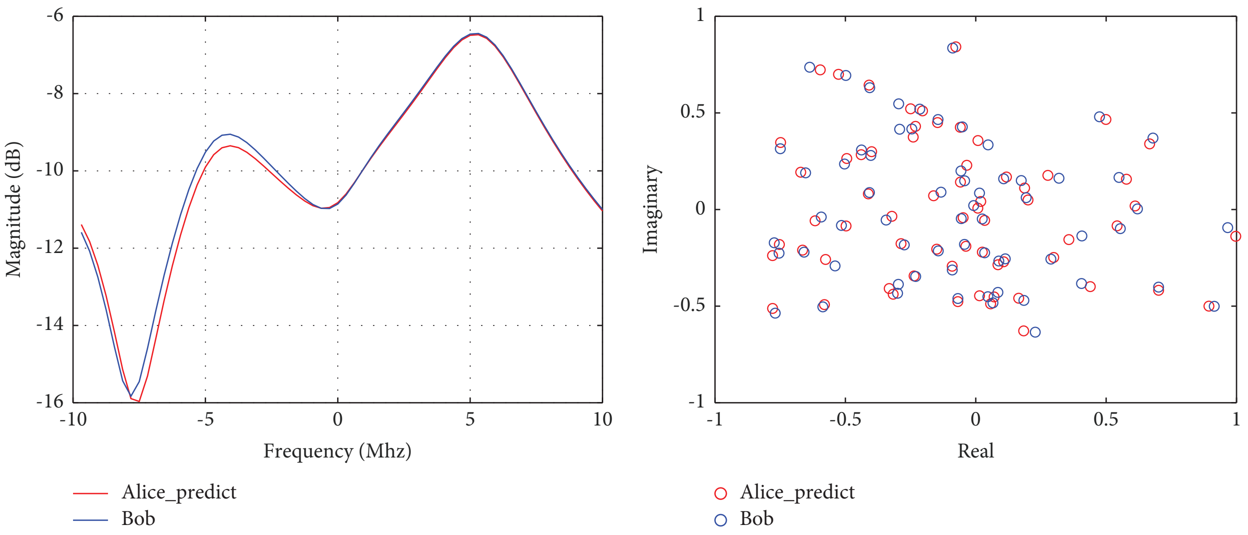
operation. The phase slope effect due to synchronization offset must be considered when using CSI as a quantization feature, unlike RSSI as a quantization feature. We can see that other factors apparently weaken that channel reciprocity. Therefore, we train a CRCNet at Alice to improve channel reciprocity by learning the correlation between channel measurement sequences between two legitimate users. The details of our proposed model will be discussed later.

In our system model, the eavesdropper speculates the shared secret keys between legitimate users by quantifying the eavesdropped channel vectors h_{AE} and h_{BE}. Assume that the eavesdropper is at least half a wavelength away from the legitimate user. Since the wireless channel gain is decorrelated beyond half a wavelength in a multipath environment [34], the channel of Eve is not correlated with the channel of legitimate users. Eve cannot infer the shared keys generated by legitimate users from the measurement of the eavesdropped channel.

3. Channel Reciprocal Compensation Network

This research aims to improve the reciprocity of channel measurements among legitimate users to generate as many consistent keys as possible. Deep learning can extract high-performance features without predefining channel model statistics. The latest research shows that deep learning can capture the correlation between two signals in complex ways. So we are motivated to use deep learning to design CRCNet. CRCNet captures correlation information between CSI sequences of legitimate users to enhance channel reciprocity among legitimate users. Figures 4(a) and 4(b) show the CSI and RSSI between legitimate users after CRCNet processing, and the channel reciprocity is significantly improved.

The structure of CRCNet is shown in Figure 5 and consists of four hidden layers and one output layer. The first hidden layer is a fully connected layer containing 256 neurons. The second hidden layer is a one-dimensional convolutional layer without padding, with a kernel size and step size of 1 × 2 and 2, respectively. The third layer is another fully-connected layer with 256 neurons. The last hidden layer is an unpadded 1D convolutional layer with kernel size and step size of 1 × 2 and 2, respectively. All four hidden layers use the Relu function as their activation function. The fully connected output layer contains 128 neurons.



(a)

(b)

Figure 4: Compensating channel reciprocity by deep learning. (a) RSSI with DL. (b) CSI with DL.

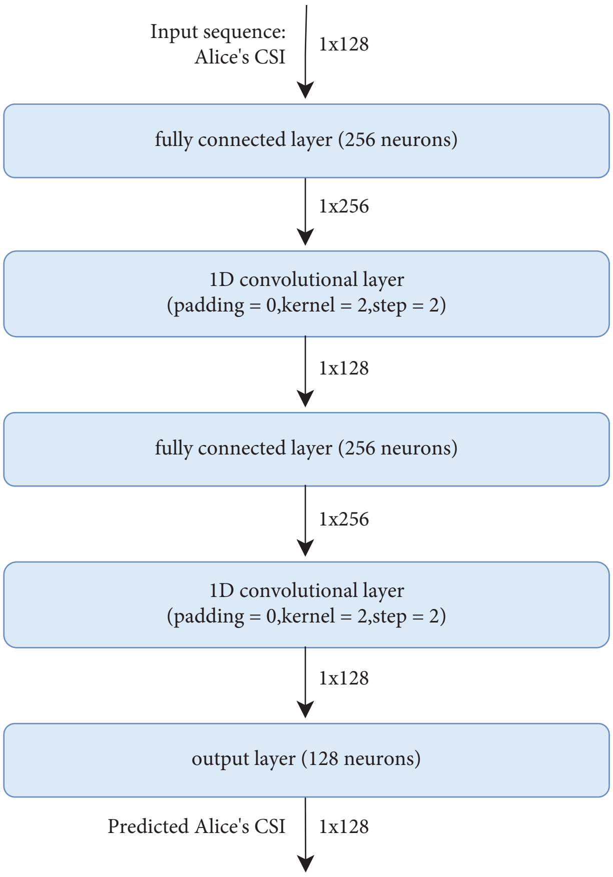

Input sequence:
Alice's CSI
1x128
fully connected layer (256 neurons)
1x256
1D convolutional layer
(padding = 0,kernel = 2,step = 2)
1x128
fully connected layer (256 neurons)
1x256
1D convolutional layer
(padding = 0,kernel = 2,step = 2)
1x128
output layer (128 neurons)
Predicted Alice's CSI
1x128

Figure 5: The architecture of CRCNet.

Since the neural network cannot handle complex numbers and channel state information is a complex-valued matrix, this requires preprocessing of the dataset. Firstly, the real and imaginary parts of the CSI matrix are stacked using the following equation:

$$H' \longrightarrow \left[\mathrm{Real}\left(H'\right), \mathrm{Imag}\left(H'\right)\right]. \qquad (3)$$

Then, stretch the stacked matrix into a one-dimensional vector of size 128. Because each dimension of the raw input has a different magnitude, we also need to normalize the dataset so that it ranges between 0 and 1.

The flow of physical layer key generation with CRCNet is shown in Figure 6 and consists of two phases: the training phase and the key generation phase.

In the training phase, we must first collect sufficient CSI data to serve as a dataset for the training model. Alice and Bob send common pilots to each other, and the channel response H is obtained through the least-squares estimation. After collecting sufficient CSI data at different coherence times, Bob sends the preprocessed CSI data stream C_B to Alice. Alice uses the preprocessed CSI data stream C_A as input and C_B as target output to train CRCNet. We use the mean square error (MSE) function as a loss function to minimize the error between the predicted and target outputs. The adaptive moment estimation optimizer is used to inversely learn and correct weight parameters layer-by-layer until the neural network converges.

In the key generation phase, the network parameters stay constant, and the trained network is deployed to Alice. Alice and Bob exchange common pilot signals for channel estimation and preprocess the measured CSI to obtain CSI streams C_A and C_B. Alice uses C_A as the input of the trained CRCNet to obtain the compensation CSI C_A^p. Alice uses the predicted output C_A^p as the input of the quantization algorithm, and Bob uses the C_B directly as the input of the quantization algorithm.

The main challenge in applying deep learning to physical layer key generation is the additional need for computational resources. Therefore, we choose to train the network on the base station side and deploy it to meet the demand of training and storing the network without additional consumption on the user side [28, 35]. Moreover, the network's training can be conducted in the cloud without consuming node resources. Given an environment to train a network, the network can be used for a long time as long as no large-scale changes occur in that environment. Therefore, it is more cost-effective to incur some computational costs in exchange for a lower KDR.

4. Balanced Vector Quantization Based on Integer Linear Programming

The reasons for using the ILP-BVQ algorithm in this paper are as follows: firstly, compared with the traditional scalar quantization algorithm, the vector quantization algorithm can improve information utilization and key generation rates by combining multidimensional channel features. Then, compared to [16], the ILP-BVQ algorithm with balancing constraints can increase the randomness of shared keys. Compared to the balanced vector quantization scheme in [17], the ILP-BVQ algorithm models the sample assignment task as an integer linear programming problem, reducing its time complexity from $O(n^3)$ to $O(n^{1.7})$. The algorithm in [18] can provide similar algorithmic time complexity and key randomness as ILP-BVQ. However, it cannot guarantee that the clusters will not overlap, which could lead to a higher KDR. Therefore, this paper uses the ILP-BVQ algorithm as the quantization algorithm for the physical layer key generation scheme.

Assume that the channel measurement samples used by Alice and Bob for quantization are CSI_A and CSI_B, and $\mathrm{CSI}_A = [c_{a1}, c_{a2}, c_{a3}, \ldots c_{\mathrm{an}}]$, $\mathrm{CSI}_B = [c_{b1}, c_{b2}, c_{b3}, \ldots c_{\mathrm{bn}}]$, where n is the CSI sequence length. To reduce the computational effort and KDR, we run the complete iterative clustering process at Alice. Bob only executes the sample assignment process once using Alice's clustering results. The detailed description of the ILP-BVQ algorithm is as follows.

Given the CSI sequences CSI_A and CSI_B, the quantization order Q, and the number of quantization regions $m = 2^Q$. The objective is to independently divide CSI_A and CSI_B into m clusters each, and each cluster should contains $\lfloor n/m \rfloor$ to $\lceil n/m \rceil$ sample points. The c_{ai} and c_{bi} with the same subscript index should fall in the same quantization region as much as possible to obtain the same binary code. The problem can be formulated as follows:

$$\begin{aligned} &\min \ \mathrm{MSE} = \left(\frac{1}{n}\right) \sum_{j=1}^{m} \sum_{c_{ai}\in\mu_j} \left\| c_{ai} - c_j \right\|^2, \\ &s.t. \left\lfloor \frac{n}{m} \right\rfloor \leq \left|\mu_j\right| \leq \left\lceil \frac{n}{m} \right\rceil, \end{aligned} \tag{4}$$

where c_{ai} is the i-th sample point, μ_j is the set of sample points in the j-th cluster, $|\mu_j|$ represents the number of samples in the j-th cluster, c_j represents the cluster center of the j-th cluster, and $\|c_{ai} - c_{aj}\|^2$ represents the square distance between c_{ai} and c_{aj}.

Let p denote the division matrix, $p_{i,j} = 1$ indicate that the sample point c_{ai} belongs to cluster j, and $p_{i,j} = 0$ indicate that the sample point c_{ai} does not belong to cluster j. Thus, the problem shown in equation (4) can be reformulated as follows:

$$\begin{aligned} &\min \ \mathrm{MSE} = \left(\frac{1}{n}\right) \sum_{j=1}^{m} \sum_{i=1}^{n} p_{i,j} \left\| c_{\mathrm{ai}} - c_{\mathrm{aj}} \right\|^2, \\ &\left\lfloor \frac{n}{m} \right\rfloor \leq \sum_{i=1}^{n} p_{i,j} \leq \left\lceil \frac{n}{m} \right\rceil, j \in [1, m], \\ &\sum_{i=1}^{n} p_{i,j} = 1, i \in [1, n]. \end{aligned} \tag{5}$$

The problem shown in equation (5) can be solved with the iterative method. Firstly, select m initial cluster centers by kmeans++ algorithm [36]. Then iteratively execute the sample assignment step and the cluster center update step.

In the sample assignment step, the cluster centers remain unchanged and samples are assigned to various clusters based on their distances from each cluster center. Minimize the MSE while satisfying the constraints. Thus, the problem of equation (5) can be reformulated as an ILP as shown in equation (6), where ε_{1j}, ε_{2j}, and ε_{3j} are the slack variables used to eliminate inequalities, they are integers, and $\varepsilon_{1j}, \varepsilon_{2j}, \varepsilon_{3j} \geq 0$. The paper [33] proved that the integer constraint can be removed from equation (6), which makes it a linear programming task that can be solved efficiently with the simplex algorithm [37].

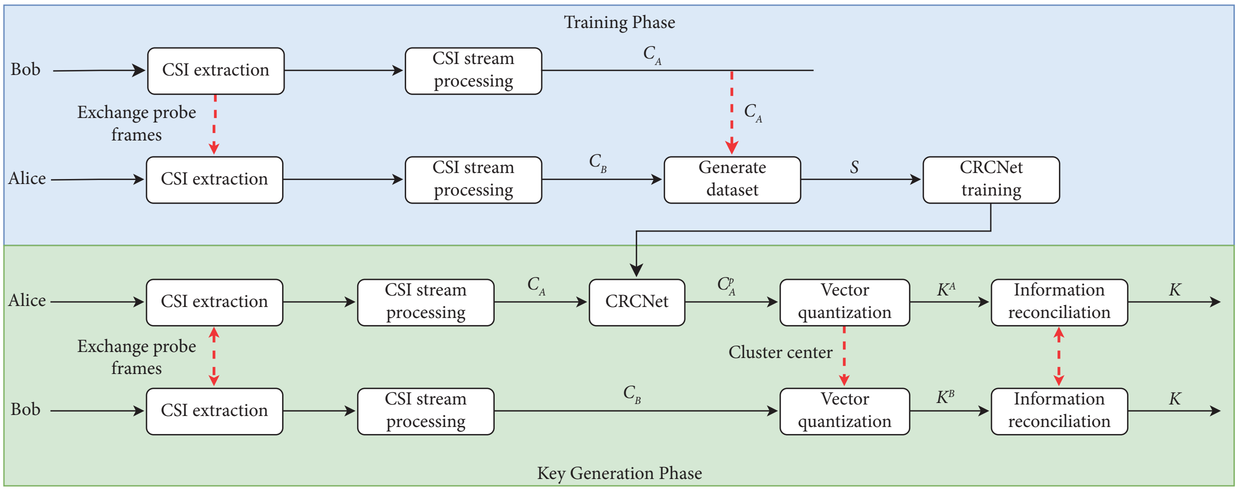


Figure 6: Physical layer key generation process based on CRCNet.

$$\begin{aligned} &\min\ \text{MSE} = \left(\frac{1}{n}\right)\sum_{j=1}^{m}\sum_{i=1}^{n} p_{i,j}\left\| c_{\text{ai}} - c_j \right\|^2, \\ &\text{s.t.} \sum_{i=1}^{n} p_{i,j} + \varepsilon_{1j} = \left\lceil \frac{n}{m} \right\rceil, j \in [1,m], \\ &\text{s.t.} \sum_{i=1}^{n} p_{i,j} - \varepsilon_{2j} = \left\lfloor \frac{n}{m} \right\rfloor, j \in [1,m], \\ &\sum_{j=1}^{k} p_{i,j} = 1, i \in [1,n], \\ &p_{ij} + \varepsilon_{3j} = 1, i \in [1,n], j \in [1,m]. \end{aligned} \tag{6}$$

In the cluster center update step, we minimize the MSE by updating the cluster center when all sample points are evenly assigned. Cluster centers are updated as follows:

$$c_j = \left(\frac{1}{\sum_{i=1}^{n} p_{i,j}}\right)\sum_{i=1}^{n} p_{i,j} c_{ai}, j \in [1,m]. \tag{7}$$

The specific process of ILP-BVQ is shown in Algorithm 1.

5. Results

5.1. Simulation Setup and Dataset Generation. Through simulation experiments, we evaluate the performance of CRCNet and the ILP-BVQ algorithm proposed in this paper. We use QUAsi deterministic RadIo channel GenerAtor (QuaDRiGa) to simulate multipath fading channels [38]. The QuaDRiGa simulation platform has detailed simulation scene geometry, continuous-time evolution, and spatially correlated large- and small-scale fading. Moreover, the QuaDRiGa platform employs a drift model to enable the smooth evolution of small-scale parameters such as the mobile terminal's power, delay, transmission angle, and arrival angle over short multipath intervals. In this work, the reason for using QuaDRiGa is to consider the generation of channel pulses close to real-world application scenarios.

The steps to simulate the channel through the QuaDRiGa platform are as follows: First, define the channel parameters, such as the channel bandwidth, center frequency, signal-to-noise ratio, etc. Then, define the network layout, including setting the transmitter position, receiver movement trajectory, and transmitter and receiver antenna properties. Next, select the channel model that best represents the conditions you want to simulate for the receiver's movement trajectory, such as an urban environment, a suburban environment, or a rural environment. The channel model is defined by the parameter file. Finally, use QuaDRiGa to generate a channel model based on the defined parameters and the selected channel model.

To evaluate the performance of the key generation scheme, we simulated an OFDM system with 64 subcarriers for our study. We design the codebook by generating a random multipath channel with $L = 8$ taps and a power delay distribution. Table 1 presents the channel simulation parameter settings. The channel model is an urban macrocellular scenario selected from the 3GPP report TR38.901 [39].

This paper uses the above QuaDRiGa platform and OFDM system to obtain CSI data. First, using the QuaDRiGa platform to build the channel model, we obtain the channel coefficients h that vary with time. Then, the channel coefficient h is used as the channel of the OFDM system, and the legitimate communication parties perform channel detection through the OFDM system. Finally, the receivers preprocess the received signals and perform channel estimation to obtain the CSI data for experiments.

Figure 7 shows the simulation experiment setup, where the base station Alice is at a fixed position at 25 m height and the user Bob at 1.5 m. Alice and Bob were both equipped with a vertically polarized short-dipole single antenna. The red line in the figure represents the channel probing process, and the black line represents the moving direction of Bob, who moves along a straight line at a speed of 0.5 m/s. Alice

```
Input:
  Alice's and Bob's CSI sequences $\mathrm{CSI}_A$ and $\mathrm{CSI}_B$.
  Number of quantified regions $m$.
Output:
  Quantification results (Alice and Bob's initial keys).
Alice uses the kmeans++ algorithm to get $m$ initial cluster centers: $c_1, c_2, c_3, \ldots c_m$;
repeat
Sample assignment: using the simplex algorithm to solve (6) to assign $\mathrm{CSI}_A$ evenly to the $m$ cluster centers;
Cluster center update: Alice updates the cluster center $c_1, c_2, c_3, \ldots c_m$ according to (7) and the sample assignment result;
until Alice's new cluster centers are the same as in previous iterations;
Alice sends the final clustering centers $c_1, c_2, c_3, \ldots c_m$ to Bob;
Bob assigns $\mathrm{CSI}_B$ evenly according to the received cluster centers by using the simplex algorithm to solve (6);
Alice and Bob get the region index of the sample points according to the final assignment matrix and then quantize the index to get
  the original key sequence.
```

Algorithm 1: Balanced vector quantization algorithm based on ILP.

Table 1: Simulation parameter settings.

Parameters	Values
Channel model	3GPP_38.901_UMa_NLOS
Duplex method	TDD
Carrier frequency	3.7 GHz
Bandwidth	20 MHz
Number of subcarriers	64
Antenna configuration of BS	1
Antenna configuration of UE	1
Channel estimation	Minimum mean square error

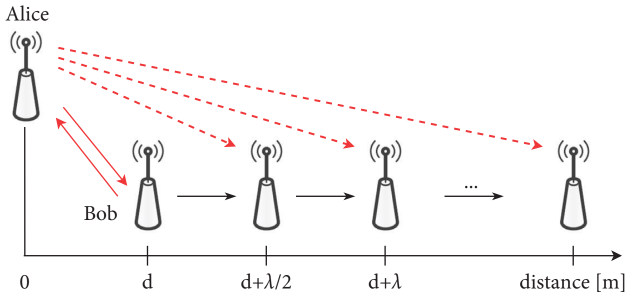


Figure 7: The simulation experiment setting.

and Bob perform channel probing at a fixed time interval. After obtaining sufficient channel data, Alice merges the CSI data collected by Bob, which is used to generate a dataset to train a deep learning model. Note that the physical environment and the spatial distribution of scattering clusters do not change with each sampling.

5.2. Evaluation Metrics. We use the following metrics for performance evaluation.

(i) Pearson correlation coefficient ρ is used to measure the correlation of the channel between two legitimate users which is defined as follows:

$$\rho = \frac{E\left[\left(C_A - \mu_{C_A}\right)\left(C_B - \mu_{C_B}\right)\right]}{\sigma_{C_A}\sigma_{C_B}}, \tag{8}$$

where μ and σ denote the mean and variance. $E\,[\bullet]$ denotes the expectation operation.

(ii) Mean Square Error (MSE) is used to evaluate the average error of the channel measurements between two legitimate users which is defined as follows:

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left\|C_A[i] - C_B[i]\right\|_2, \tag{9}$$

where m denotes the length of the CSI sequence.

(iii) Key disagreement rate (KDR) is the percentage of different bits between shared keys generated by two legitimate users which is defined as follows:

$$\mathrm{KDR} = \frac{\sum_{i=1}^{N}\left|K^A(i) - K^B(i)\right|}{N}, \tag{10}$$

where N represents the length of the key obtained by quantization.

(iv) Time consumption is used to reveal the time complexity of different quantization algorithms. We tested the time consumption by building a simulation system in Matlab.

(v) Randomness reveals the distribution of the shared keys. We used the National Institute of Standards and Technology (NIST) statistical test [40] to test the randomness of the generated keys.

(vi) Mutual information is a measure of the interdependence between variables. In this paper, we use it to measure the similarity between the eavesdropping key and the shared key which is defined as follows:

$$I(X;Y) = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}. \tag{11}$$

5.3. CRCNet Performance Evaluation. The function of CRCNet is to enhance the correlation of channel measurements between legitimate users to lower the KDR in the subsequent quantization process. We compare the performance of CRCNet with the following five schemes:

(1) PCA [19]: using principal component analysis algorithm to reduce hardware fingerprint variance during channel measurements.

(2) WAKG [20]: using wavelet analysis to preprocess the channel estimation to improve channel correlation.

(3) SGF [21]: the Savitzky Golay Filter (SGF) method is used to increase the correlation of RSS values between users and access points.

(4) KGNet [28]: a key generation neural network (KGNet) is proposed to generate reciprocal channel features in FDD communication systems based on band feature maps.

(5) AE [30]: the inverse features are extracted from the weakly correlated channel estimates with a trained autoencoder to generate features for quantization in various channel environments.

In Figure 8, the performance of the various channel reciprocity improvement schemes used to obtain channel correlation is plotted as a function of the SNR in dB. We can see that the correlation of all schemes tends to increase when the SNR increases. And our proposed CRCNet has the best performance in channel compensating, with at least a 1% performance improvement compared to the other five schemes. In Figure 9, the MSE performance of the legitimate interuser channel obtained by the various channel reciprocity improvement schemes is plotted as a function of SNR in dB. It can be seen that these six schemes perform similarly at low SNRs. As the SNR increases, the MSE reduces, and the performance advantage of CRCNet steadily increases compared to the other schemes.

Figure 10 compares the KDR for the six reciprocity improvement schemes. As expected, CRCNet achieves the lowest KDR. At SNR = 30 dB, the KDR of CRCNet reduces from 4.8% to 0.89% compared to WAKG, whereas the KDRs of the other four schemes are 3.9%, 2.8%, 1.5%, and 1.13%, respectively. Note that the six schemes here use the same quantization algorithm.

5.4. ILP-BVQ Performance Evaluation. In this paper, we compare the performance of the ILP-BVQ algorithm with the following four quantization algorithms:

(1) Scalar quantization [12]: a multi-bit adaptive quantization scheme that uses only magnitude for quantization.

(2) Kmeans vector quantization [16]: multidimensional quantization using the kmeans clustering algorithm with CSI.

(3) Balanced kmeans quantization [17]: add balanced constraints to the kmeans vector quantization to ensure the randomness of the key.

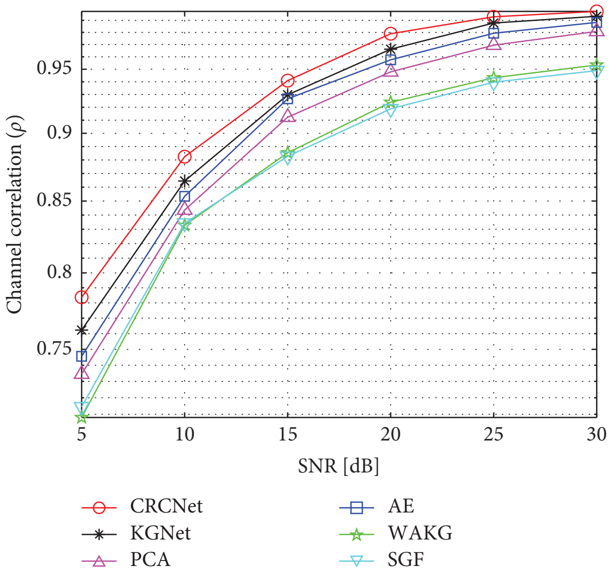


Figure 8: Comparison of the channel correlation of six channel reciprocity improvement schemes.

(4) BKQ-BM [18]: a variation of the kmeans algorithm ensures that each cluster has the same size, randomizing the shared key.

Figure 11 compares the KDR of the five quantization algorithms. We can see that the KDR of four vector quantization algorithms is significantly lower than scalar quantization. In addition, three balanced vector quantization algorithms have a lower KDR than the kmeans algorithm. This is because the balanced quantization algorithm adds the constraint of the same-sized number of samples per cluster, which is used to improve the secret key randomness with some performance sacrifice. ILP-BVQ and balanced kmeans perform the same, and they differ only in the sample allocation method, which does not affect the result of key generation. The BKQ-BM algorithm has the highest KDR because it cannot guarantee that the clustering results do not overlap. Figure 12 compares the KDR of the three quantization algorithms with and without using CRCNet for channel compensation. It shows that for all algorithms, using CRCNet can reduce the KDR as a result of the improved channel correlation between legitimate users through CRCNet channel compensation.

In Figure 13, the time consumption of Alice and Bob with four vector quantization algorithms is plotted as a function of SNR in dB. It is evident that the kmeans quantization algorithm, which does not require consideration of the balancing constraint, has the lowest time consumption. Among the three balanced vector quantization algorithms, ILP-BVQ and BKQ-BM have similar time consumption and are significantly lower than the balanced kmeans algorithm. This is because the time complexity of the sample allocation algorithm for each of these three algorithms is $O(n^{1.7})$, $O(nk\log_2 n)$, and $O(n^3)$, respectively. Figure 14 shows a comparison of the time consumption of Alice and Bob using the four vector quantization algorithms

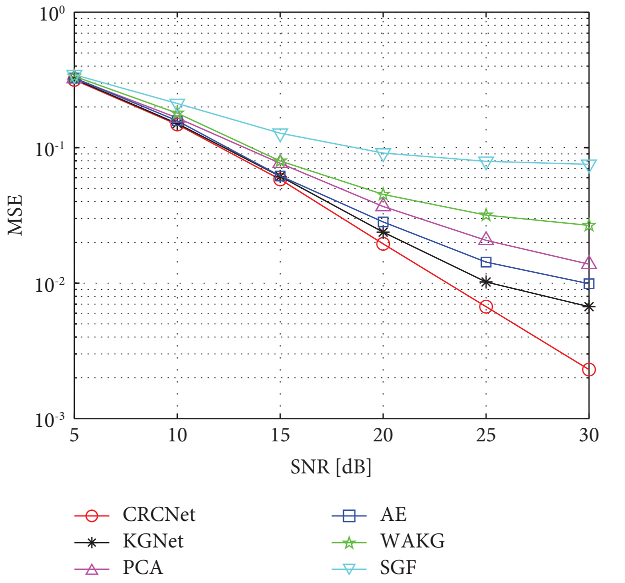


Figure 9: MSE comparison of six channel reciprocity improvement schemes.

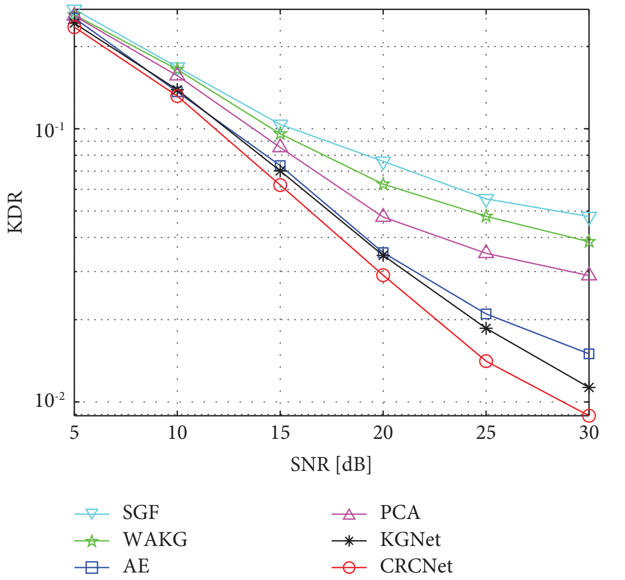


Figure 10: KDR comparison of six channel reciprocity improvement schemes.

in different numbers of quantization regions. The kmeans quantization algorithm has the lowest time consumption, followed by ILP-BVQ and BKQ-BM, while the balanced kmeans have the highest time consumption.

In this paper, we add a balancing constraint to the kmeans vector quantization algorithm to limit the number of samples in each cluster to enhance shared key randomness. We use the NIST test [40] to verify the randomness of generated keys of the proposed scheme. The output of each NIST test is the p value. The detected sequence passes the test when the p value exceeds a certain threshold (usually 0.01). Only test the original key generated by the quantized gray code. In this case, set the number of quantized regions to 16. This experiment tests four vector quantization algorithms. Each algorithm tests 100 sets of keys, and the length of each set of keys is 256.

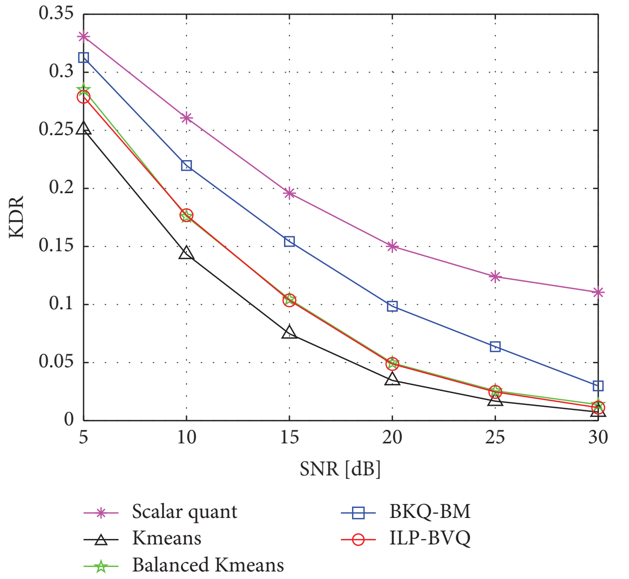


Figure 11: Comparison of the KDR of the five quantization algorithms.

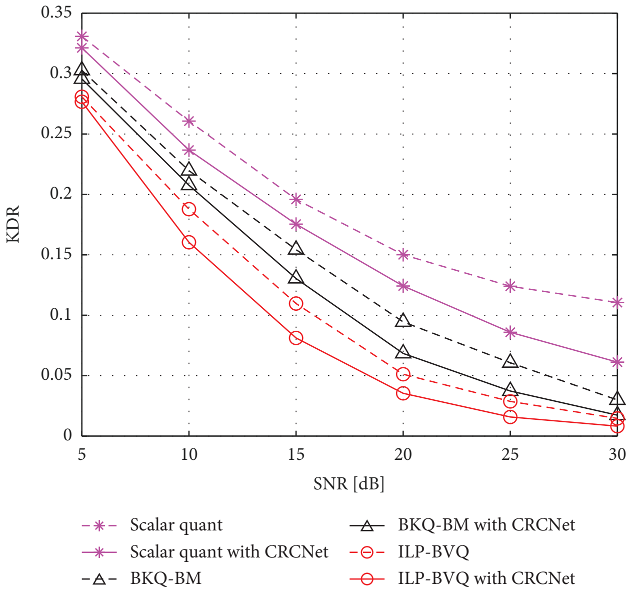


Figure 12: Comparison of the KDR of the three quantization algorithms with or without CRCNet.

The results of the NIST test are shown in Table 2. The tests of three balanced kmeans algorithms passed most of the test items with high probability. The kmeans algorithm has significantly lower test results than the three balanced kmeans algorithms, passing a minority of test items. This is because the kmeans quantization algorithm cannot ensure the same number of samples in each cluster. The number of samples in a specific cluster may be relatively high, and some bit sequences in the quantization results are large-scale repeated, resulting in lower key randomness.

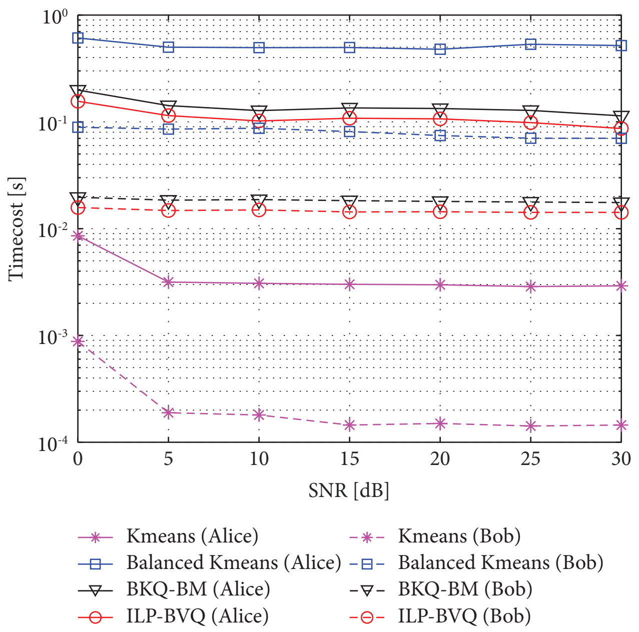


FIGURE 13: Comparison of Alice and Bob's time consumption using the four vector quantization algorithms.

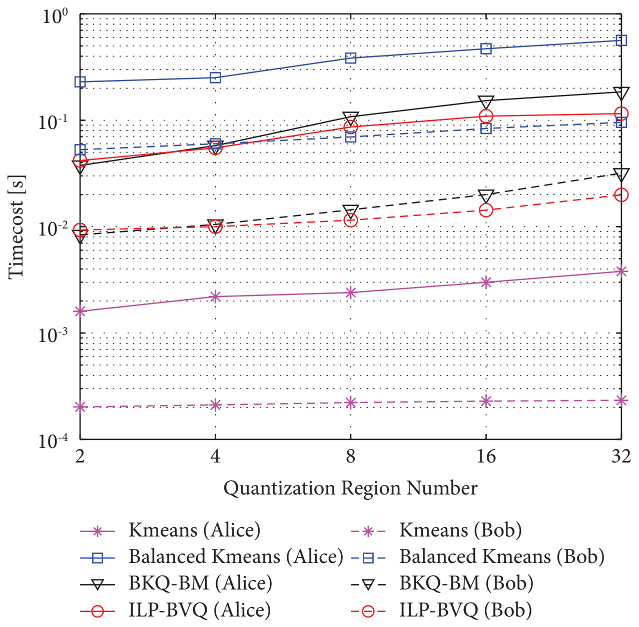


FIGURE 14: Comparison of time consumption in different quantization regions.

5.5. *Information Leakage.* Considering the existence of eavesdroppers, we analyze the relationship between eavesdropping channel correlation and information leakage by controlling the relative positions of the eavesdroppers and legitimate users. Figure 15 shows the setup of the experimental scenario in the presence of an eavesdropper. Fix the positions of base station Alice and legitimate user Bob, and control the eavesdropper Eve to move to both sides away from Alice and Bob, respectively. The blue line in the figure is the eavesdropper's moving direction. We analyze the effect of the eavesdropper being close to the base station or close to the user on key leakage. The differences between the three balanced quantization schemes, ILP-BVQ, BKQ-BM, and balanced kmeans, are only reflected in the quantization process on the Alice side of the base station. Therefore, only the scalar and kmeans quantization schemes are compared here.

TABLE 2: NIST test.

	Kmeans	Balanced kmeans	BKQ-BM	ILP-BVQ
Runs	0	0.949	0.99	1.0
Frequency	0	0.973	0.989	0.995
Dft	0.0005	0.75	0.58	0.907
Longest run	0.086	0.471	0.82	0.237
Non-overlapping	0.066	1.0	1.0	1.0
Approximate entropy	0	0.489	0.73	0.809
Serial	0	0.123	0.64	0.55
Cum.sums	0.0004	1.0	1.0	1.0

This paper uses mutual information to represent the information leakage rate. Before computing the mutual information, we need to build a dataset containing K_a, K_b, and K_e with a size of $3 \times L \times N$. Where K_a and K_b are the original keys quantified by Alice and Bob, K_e is the eavesdropping key quantified by Eve using the same quantization algorithm as the legitimate user, L is the key length, and N is the number of samples. Both the original key and the eavesdropping key are discrete random variables represented in binary with the same length.

This paper uses the Monte Carlo technique-based approximation technique to estimate the mutual information. At first, we randomly selected a large number of samples from the data set (each sample contains K_a, K_b, and K_e). Then, the mutual information of each sample is calculated using equation (11). Finally, the mutual information of all samples is averaged to obtain the approximate mutual information value.

The first scenario assumes that eve is near user Bob. The results of mutual information in the case with and without the CRCNet model using three quantization algorithms, scalar quantization, kmeans quantization, and ILP-BVQ, are shown in Figure 16. As the figure shows, the key leakage is minimal, only reaching about 10% at $\rho = 0.8$. This indicates that even if the eavesdropping channel is highly correlated with the legitimate channel, the eavesdropper can obtain only limited information about generated keys. Using CRCNet and the balanced vector quantization algorithm can also help reduce the rate of key leakage.

The second scenario is when Eve is close to the base station, Alice. This scenario is slightly different from the previous one. In this case, the data Alice uses for quantization is compensated, and the correlation of the channel data Eve uses for quantization is different from the actual one. Figure 17 shows the results of the mutual information calculation. Under the same eavesdropping channel correlation condition, the mutual information is significantly smaller when close to base station Alice than when close to

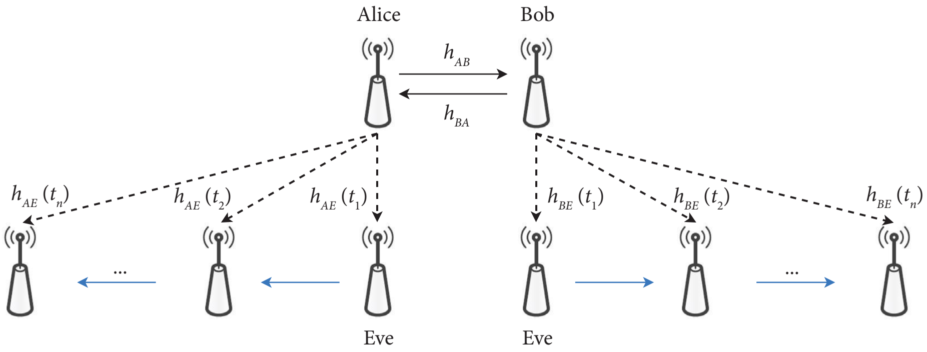


FIGURE 15: Eavesdropper position.

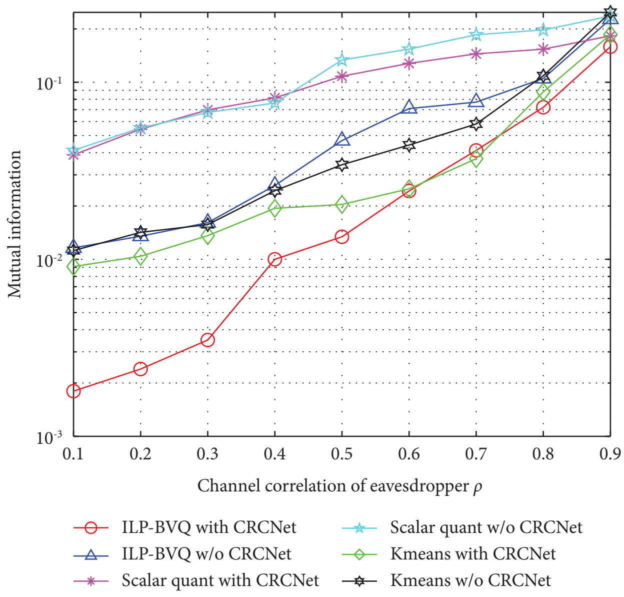


FIGURE 16: Mutual information between the legitimate user-generated key and the eavesdropper as he approaches Bob ($m = 16$, SNR = 30).

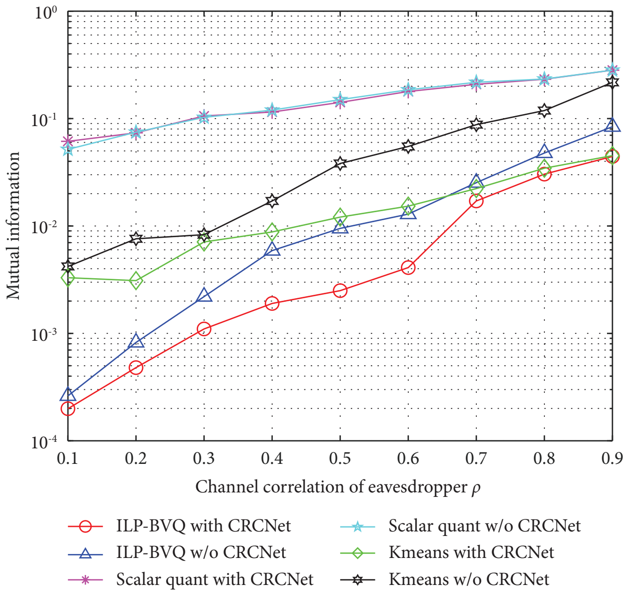


FIGURE 17: Mutual information between the legitimate user-generated key and the eavesdropper as he approaches Alice ($m = 16$, SNR = 30).

user Bob. This is because the eavesdropper has no information about the parameters of the CRCNet. Alice's data for quantization are equivalent to a transformation, which reduces the key leakage rate. The experimental results of the two scenarios above show that vector quantization outperforms scalar quantization in terms of preventing information leakage. Moreover, CRCNet can reduce the key leakage rate and improve key robustness.

6. Conclusions

In this paper, we propose a deep learning-based channel reciprocity compensation method and a balanced vector quantization algorithm for generating physical layer keys. Firstly, we build a channel reciprocity compensation network to describe the wireless environment and learn the mapping relationship of wireless channels among legitimate users. The trained CRCNet compensates Alice's channel measurements, and shared keys are generated based on quantizing the compensated channel measurements. This scheme achieves significant performance improvements over existing schemes in terms of channel reciprocity and KDR. In addition, to address the weak randomness of the kmeans algorithm and the high time complexity of the balanced kmeans algorithm in existing vector quantization algorithms. We propose an ILP-based balanced vector quantization algorithm. A cluster balancing mechanism is added to ensure the randomness of shared keys and solve the sample assignment problem in the balanced quantization process by integer linear programming. Because the ILP-BVQ algorithm's clustering results do not overlap, we can reduce the algorithm's time complexity from $O(n^3)$ to $O(n^{1.7})$ while maintaining a low KDR. Finally, we build a physical layer key generation system on Matlab by simulating the channel variations in a real environment with QuaDRiGa [38]. The comparable performance of our scheme has been verified.

6.1. Future Work. One of the future research works is to use migration learning and meta-learning models in the field of physical layer key generation, which can help to solve the inapplicability problem caused by environmental changes. Since most of the current physical layer key generation schemes are based on point-to-point schemes, the other future research work is to extend the research to the multiuser domain and group key generation application.

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 are no conflicts of interest.

Acknowledgments

This work is supported by the National Key R&D Program of China (grant no. 2020YFE0200600) and National Nature Science Foundation of China (grant no. U22B2026).

References

[1] N. Wang, P. Wang, A. Alipour-Fanid, L. Jiao, and K. Zeng, "Physical-layer security of 5G wireless networks for IoT: challenges and opportunities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8169–8181, 2019.

[2] J. Tang, H. Wen, K. Zeng, R. Liao, F. Pan, and L. Hu, "Lightweight physical layer enhanced security schemes for 5G wireless networks," *IEEE Network*, vol. 33, no. 5, pp. 126–133, 2019.

[3] J. D. V. Sánchez, L. Urquiza-Aguiar, M. C. P. Paredes, and D. P. M. Osorio, "Survey on physical layer security for 5G wireless networks," *Annals of Telecommunications*, vol. 76, no. 3-4, pp. 155–174, 2021.

[4] G. Li, C. Sun, J. Zhang, E. Jorswieck, B. Xiao, and A. Hu, "Physical layer key generation in 5G and beyond wireless communications: challenges and opportunities," *Entropy*, vol. 21, no. 5, p. 497, 2019.

[5] F. Irram, M. Ali, M. Naeem, and S. Mumtaz, "Physical layer security for beyond 5G/6G networks: emerging technologies and future directions," *Journal of Network and Computer Applications*, vol. 206, Article ID 103431, 2022.

[6] 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, March 2019.

[7] H. Fang, X. Wang, S. Tomasin, and N. Al-Dhahir, "Lightweight group authentication for decentralized edge collaboration," *IEEE Communications Magazine*, vol. 60, no. 12, pp. 124–129, December 2022.

[8] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 240–254, 2010.

[9] J. Zhang, R. Woods, T. Q. Duong, A. Marshall, and Y. Ding, "Experimental study on channel reciprocity in wireless key generation," in *Proceedings of the 2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, Edinburgh, UK, July 2016.

[10] L. Cheng, W. Li, D. Ma, L. Zhou, C. Zhu, and J. Wei, "Towards an effective secret key generation scheme for imperfect channel state information," in *Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 915–920, Tianjin, China, August 2016.

[11] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong, "Secure key generation from OFDM subcarriers' channel responses," in *Proceedings of the 2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 1302–1307, Austin, TX, USA, December 2014.

[12] 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.

[13] J. Zhang, R. Woods, A. Marshall, and T. Q. Duong, "An effective key generation system using improved channel reciprocity," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1727–1731, South Brisbane, Australia, April 2015.

[14] A. Goel and V. P. Vishwakarma, "Efficient feature extraction using DCT for gender classification," in *Proceedings of the 2016 IEEE International Conference on Recent Trends in*

Electronics, Information & Communication Technology (RTEICT), pp. 1925–1928, Bengaluru, India, May 2016.

[15] 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.

[16] Y. Chen, H. Wen, J. Wu et al., "Clustering based physical-layer authentication in edge computing systems with asymmetric resources," *Sensors*, vol. 19, no. 8, 2019.

[17] Y. W. P. Hong, L. M. Huang, and H. T. Li, "Vector quantization and clustered key mapping for channel-based secret key generation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1170–1181, 2017.

[18] Q. Han, J. Liu, Z. Shen, J. Liu, and F. Gong, "Vector partitioning quantization utilizing K-means clustering for physical layer secret key generation," *Information Sciences*, vol. 512, pp. 137–160, 2020.

[19] Z. Li and L. Peng, "Research on the design of highly random and consistent wireless key generation system," in *Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 1685–1690, Chongqing, China, December 2020.

[20] O. Alp Topal, Z. Liang, G. Ascheid, G. Dartmann, and G. Karabulut Kurt, "Using of wavelets for secret key generation: a measurement based study," in *Proceedings of the 2018 26th Telecommunications Forum (TELFOR)*, pp. 1–4, Belgrade, Serbia, November 2018.

[21] M. Yuliana and W. Wirawan, "Performance evaluation of savitzky golay filter method that implement within key generation," in *Proceedings of the 2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 343–348, Purwokerto, Indonesia, November 2021.

[22] M. Yuliana, E. Suwadi, and T. Suryani, "Enhancing channel reciprocity of secret key generation scheme by using modified polynomial regression method," in *Proceedings of the 2018 International Conference on Computer Engineering Network and Intelligent Multimedia (CENIM)*, pp. 35–40, Surabaya, Indonesia, November 2018.

[23] O. I. Abiodun, M. U. Kiru, A. Jantan et al., "Comprehensive review of artificial neural network applications to pattern recognition," *IEEE Access*, vol. 7, pp. 158820–158846, 2019.

[24] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: a survey," *Heliyon*, vol. 4, no. 11, Article ID e00938, 2018.

[25] W. Ma, C. Qi, Z. Zhang, and J. Cheng, "Sparse channel estimation and hybrid precoding using deep learning for millimeter wave massive MIMO," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2838–2849, 2020.

[26] Y. Wang, J. Yang, M. Liu, and G. Gui, "LightAMC: lightweight automatic modulation classification via deep learning and compressive sensing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3491–3495, 2020.

[27] P. Yu, F. Zhou, X. Zhang, X. Qiu, M. Kadoch, and M. Cheriet, "Deep learning-based resource allocation for 5G broadband TV service," *IEEE Transactions on Broadcasting*, vol. 66, no. 4, pp. 800–813, 2020.

[28] X. Zhang, G. Li, J. Zhang, A. Hu, Z. Hou, and B. Xiao, "Deep learning-based physical-layer secret key generation for FDD systems," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6081–6094, 2022.

[29] M. Letafati, H. Behroozi, B. H. Khalaj, and E. A. Jorswieck, "Deep learning for hardware-impaired wireless secret key generation with man-in-the-middle attacks," in *Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Madrid, Spain, December 2021.

[30] J. Han, X. Zeng, X. Xue, and J. Ma, "Physical layer secret key generation based on autoencoder for weakly correlated channels," in *Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1220–1225, Chongqing, China, August 2020.

[31] M. Letafati, H. Behroozi, B. H. Khalaj, and E. A. Jorswieck, "Wireless-powered cooperative key generation for e-health: a reservoir learning approach," in *Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–7, Helsinki, Finland, June 2022.

[32] X. Guan, N. Ding, Y. Cai, and W. Yang, "Wireless key generation from imperfect channel state information: performance analysis and improvements," in *Proceedings of the 2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, Shanghai, China, May 2019.

[33] W. Tang, Y. Yang, L. Zeng, and Y. Zhan, "Optimizing MSE for clustering with balanced size constraints," *Symmetry*, vol. 11, no. 3, p. 338, 2019.

[34] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: 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, CA, USA, October 2008.

[35] D. Han, A. Li, and J. Li, "DroneKey: a drone-aided group-key generation scheme for large-scale IoT networks," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1306–1319, New York, NY, USA, June 2021.

[36] D. Arthur and S. Vassilvitskii, *K-Means++: The Advantages of Careful Seeding*, Stanford University, Stanford, CA, USA, 2006.

[37] A. Koberstein, "Progress in the dual simplex algorithm for solving large scale LP problems: techniques for a fast and stable implementation," *Computational Optimization and Applications*, vol. 41, no. 2, pp. 185–204, 2008.

[38] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: a 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 3242–3256, 2014.

[39] Global Initiative Against Transnational Organized Crime, "Study on Channel Model for Frequencies from 0.5 to 100 GHz (Release 14)," Global Initiative Against Transnational Organized Crime, 3GPP TR 38.901, 2019.

[40] A. Rukhin, J. Sota, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," National Institute of Standards and Technology, Gaithersburg, MD, USA, Special Publication NIST 800-22, 2010.