Hindawi
Security and Communication Networks
Volume 2022, Article ID 1844345, 11 pages
https://doi.org/10.1155/2022/1844345

Research Article

Deep Learning-Based Channel Reciprocity Learning for Physical Layer Secret Key Generation

Haoyu He,[1] **Yanru Chen**,[1] **Xinmao Huang**,[2] **Minghai Xing**,[3] **Yang Li**,[4] **Bin Xing**,[5] **and Liangyin Chen**[1,6]

[1]*School of Computer Science & School of Software Engineering, Sichuan University, Chengdu 610065, China*
[2]*Sichuan GreatWall Computer System Co., Ltd, Luzhou 646000, China*
[3]*CEC Jiutian Intelligent Technology Co., Ltd, Shuangliu District, Chengdu, Sichuan 610299, China*
[4]*Science and Technology on Security Communication Laboratory, Institute of Southwestern Communication, Chengdu 610041, China*
[5]*Chongqing Innovation Center of Industrial Big-Data Co., Ltd, Chongqing 400707, China*
[6]*Institute for Industrial Internet Research, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Bin Xing; xingbin@casic.com and Liangyin Chen; chenliangyin@scu.edu.cn

Received 4 November 2021; Revised 11 January 2022; Accepted 1 March 2022; Published 19 March 2022

Academic Editor: Yanhui Guo

Using the physical layer channel information of wireless devices to establish the highly consistent secret keys is a promising technology for improving the security of wireless networks. Nevertheless, in the time division duplex system, the reciprocity of the wireless channel that is the basic principle of key generation is impaired by nonsimultaneous sampling and noise factors. Existing physical layer key generation approaches rely on hand-crafted feature extraction algorithms, which have high overhead or security issues and are impractical in real-world situations. This paper presents a novel physical layer key generation method to extract highly consistent keys from imperfect channel responses, which exploits channel reciprocity through deep learning. Specifically, we first design the Channel Reciprocity Learning Net (CRLNet), a neural network for efficiently learning channel reciprocity features from the wireless channel in TDD OFDM systems. Later, a new key generation scheme based on CRLNet is developed that can achieve a high key agreement rate. Experiments indicate that the CRLNet-based key generation scheme performs excellently in terms of key generation rate, key error rate, and randomness, confirming that our method has better performance and lower overhead than existing methods.

1. Introduction

Physical layer key generation is a promising encryption technique in wireless systems, which has recently received much attention [1–3]. In contrast to its traditional counterpart, physical layer key generation uses the inherent unpredictability of channel changes between valid users to establish information-theoretic security without the assistance of a third party. This channel-based scheme achieves stronger security, higher key generation rate, and lower costs than the conventional ways [4–6]. With the development of emerging wireless systems, e.g., IoT, Internet of Vehicles (IoV), and Unmanned Aerial Vehicles (UAV), the physical layer key generation becomes popular in recent years [7–9]. The basic concept behind physical layer key generation is that two legitimate devices named Alice and Bob exchange information publicly to acquire channel responses, which are then utilized to extract symmetric keys. The viability of this procedure is based on three principles: temporal variation, channel reciprocity, and spatial decorrelation [2]. Among them, channel reciprocity requires that the channel features collected by Alice and Bob within the coherence time be highly correlated. This is fundamental for legal devices to produce matching keys from channel features separately. However, key generation is most commonly used in time-division duplex (TDD) systems, since

nonsimultaneous sampling, channel noise, and hardware diversity significantly weaken the correlation of channel measurements between two authorized users [10]. Thus, it is crucial to construct reciprocal channel features in a nonideal wireless channel.

Intentionally designed feature extraction methods can capture reciprocal component within channel measurements. Some researchers use conventional linear transforms to extract characteristics, such as principal component analysis (PCA) [10, 11], discrete cosine transform (DCT) [12], and wavelet transform (WT) [13]. Several applied nonlinear feature extraction approaches outperform the linear transform, such as [14, 15]. In simulation experiments, these approaches provide good key agreement and a high generation rate, but they lack robustness in practical wireless situations. There are some preprocessing algorithms that are specifically developed to exploit reciprocity from channel measurements [16–18]. On the one hand, these methods are computationally expensive or have security vulnerabilities. On the other hand, these human-crafted feature extraction approaches are based on personal observations and are inflexible in diverse real-world environments.

Deep learning is a powerful features extraction technology, which does not need a predefined statistical characteristic of the channel model. In the field of wireless communication and networking, various deep learning applications have emerged in recent decades, including resource allocation [19], channel estimation [20], modulation classification [21], and low rate CSI feedback [22]. However, few studies have focused on applying deep learning to capture reciprocity in the field of physical layer key generation. Existing key generation methods depend on the prior knowledge of wireless channel and cannot work properly in the situations that do not follow the statistical distribution assumption of the channel response. Therefore, we intend to leverage powerful feature learning capability of DL to adaptively construct consistent secret keys from the imperfect channel in real-world wireless systems.

In this paper, by employing deep learning, we propose a novel method for capturing reciprocal channel characteristics to efficiently generate secret keys. Specifically, we design the Channel Reciprocity Learning Net (CRLNet), a multibranch autoencoder-based neural network based on the prior knowledge of the channel measurement model. The proposed model is driven by the channel state information (CSI) of the TDD Orthogonal Frequency-Division Multiplexing (OFDM) system, which can achieve a higher key generation rate than RSS [2]. The CRLNet can be trained to adaptively learn the reciprocity components in weakly correlated channels using the collected CSI data. Furthermore, a complete key generation scheme is designed based on the CRLNet. To demonstrate the validity and effectiveness of our algorithm, we conduct comprehensive testing in various real-world wireless environments. The following are our primary contributions in detail.

(1) We design a novel multibranch autoencoder-based neural network, named CRLNet, and a special hybrid loss function to train the network according to the channel measurement model. Without any knowledge of the statistical distribution of channel responses, the model trained utilizing the CSI data from the commercial WiFi devices can construct highly correlated channel features that can be quantized into high-agreement secret keys.

(2) Based on our deep learning model, a practical secret key generation mechanism is developed. In contrast to the existing method, the proposed scheme achieves higher performance without high computing overhead or security risks.

(3) Extensive testing results conducted under static and mobile environments in both indoor and outdoor scenarios show that the proposed method is feasible and effective. A superior key generation rate, key error rate, and randomness are all achieved as compared to previous methods.

The rest of this paper is arranged as follows. Related Works shows relevant researches. The standard flow for secret key generation and the reciprocal feature learning algorithm developed by us are shown in Materials and Methods. The designed deep learning based key generation scheme is also included in this section. The experiments used to evaluate the proposed method's performance are presented in the Results and Discussion, which is preceded by the Conclusions.

2. Related Works

Physical layer key generation was first studied in the mid-1990s. It has been demonstrated that extracting secret keys from wireless channel characteristics can achieve reliable information-theoretic security [23]. In [24], the authors collect the sender's signal in the IEEE 802.11 wireless network and extract the RSS estimation in the channel to quantify it into secret keys. The authors in [25] analyze the impact of changes in the environment on the performance of the RSS-based method and discover that the key generation rate was higher in a dynamic environment. Due to the limited key generation rate and insufficient randomness of RSS-based methods, [16, 26, 27] exploit fine-grained channel state information (CSI) to achieve better performance. In addition, [16] proves that CSI-based methods are immune to attacks that RSS-based methods are vulnerable to, such as predicted channel attack and stalker attack. However, since the majority of these researches are limited to theoretical analysis and simulation studies, it is difficult to demonstrate their feasibility and generality in the real wireless environment.

Several deep learning based methods have recently emerged for extracting meaningful features from physical layer channel responses [28–32]. O'Shea and Hoydis [28]

show several potential applications of deep learning in the physical layer. The authors model the wireless communication as the end-to-end autoencoder and achieve better performance than conventional methods. Abyaneh et al. [29] use convolutional neural networks to extract features from CSI, which improves the accuracy of the physical layer authentication system. Liu et al. [30] propose a self-supervised learning framework for IoT applications to learn the underlying physical features of sensing signals. Huang et al. [31] design a deep neural network for channel calibration in massive MIMO systems, which can construct a nonlinear mapping between DL and UL channels. Zhang et al. [32] use a fully connected neural network to learn the mapping function between CSI of different frequency bands in an FDD system. Inspired by these works, we apply deep learning to learn reciprocal features from the imperfect channel to generate consistent secret keys in complex wireless communication.

3. Materials and Methods

3.1. Secret Key Generation Flow. Generally, key generation based on physical channel characteristics includes five steps [2].

(1) Channel probing: legitimate devices named Alice and Bob periodically exchange probing packets to facilitate channel estimation in receiving end. Assuming the wireless channel response recorded at Alice and Bob are H_a' and H_b', respectively, as expressed below,

$$H_u' = H_u + N_u, \tag{1}$$

where $u = \{a, b\}$ represent channel response of Alice and Bob, H_u is the wireless channel response of the perfect channel, and N_u is the independent nonreciprocal components in the both ends of wireless communication.

(2) Reciprocity feature extraction: as mentioned above, reciprocity is greatly weakened by nonsimultaneous measurements due to the TDD system and separate noise residing in various devices, as present in Figure 1. Because the reciprocal components of the channel are mixed with variable nonreciprocal components in the environment, it is difficult to directly extract matching key pairs from the results of channel estimation. The reciprocity feature extraction method is therefore in charge of extracting the reciprocity feature from the original channel response.

(3) Quantization: this stage's goal is to convert the channel measurements into a bit sequence. Depending on the wireless communication environment, different quantization levels should be specified, resulting in a compromise among the key generation rate and the key error rate [6].

(4) Information reconciliation: the initial generated keys are not all exactly the same. Reconciliation is used to correct the mismatched bits in the key. The main methods include Cascade [33], error correcting code (ECC) [34], BCH code [35], and Golay code [36].

(5) Privacy amplifying: during the information reconciliation stage, Alice and Bob transmit some information that eavesdroppers may hear. To ensure the security of the generated key, the hash function is generally used to convert the corrected initial key to fixed-length final keys that can be used directly in cryptographic techniques [37].

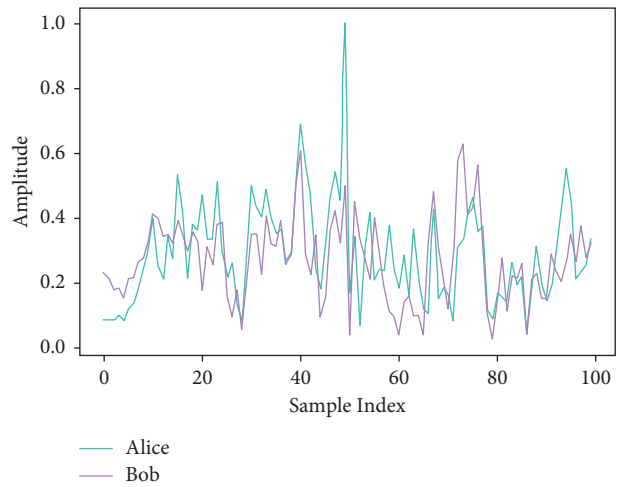

Figure 1: The first dimension of normalized CSI value's real part recorded by Alice and Bob. It can be observed that the reciprocity of the channel is weakened by other factors.

Reciprocity feature learning is the most essential phase in physical layer key generation, with a significant influence on key error rate, key generation rate, and randomness of the generated key. The initial key with a lower key error rate can facilitate the subsequent stages. Therefore, we build a neural network model capable of learning the reciprocity component from the channel response.

3.2. Reciprocity Learning Design. The major focus of this research is to extract the reciprocity component from the channel response in order to generate extremely consistent keys. We present the Channel Reciprocity Learning Net (CRLNet) to efficiently learn the reciprocal component of the original channel response.

The design of CRLNet is based on formula (1) and its structure is displayed in Figure 2. To eliminate N_u from channel response as much as possible, we designed a nonreciprocity learning module (NRLM), which consists of three hidden linear layers whose numbers of neurons are 512, 256, and 256, its input is the origin channel response, and the output is nonreciprocal component expressed as N_u. The multibranch autoencoder part is a symmetrical structure, which consists of two encoders (Encoder_a, Encoder_b) and a shared weight decoder (Decoder). The whole CRLNet is composed of a multibranch autoencoder and two NRLMs.

During the training phase, the input of the neural network is paired CSI (H_a', H_b') records by Alice and Bob,

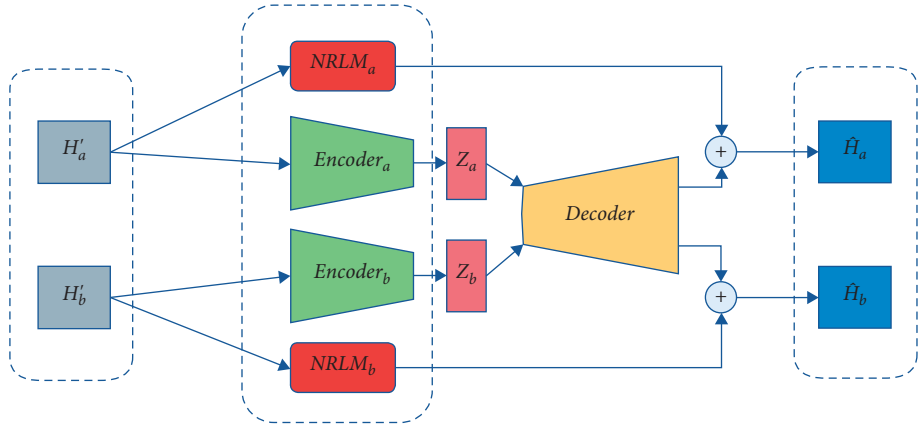

Figure 2: CRLNet structure design.

and the output is $\widehat{H_a}$ and $\widehat{H_b}$, which suppress the nonreciprocal part and are highly correlated. The encoded primary features Z_a of channel response are learned by Encoder$_a$ from H'_a, whereas NRLM$_a$ is employed to distinguish the nonreciprocal component. Encoder$_b$ and NRLM$_b$ do the same operation on H'_b as Encoder$_a$ and NRLM$_a$. The shared weight decoder is utilized to reconstruct $\widehat{H_a}$ and $\widehat{H_b}$ from strongly reciprocal encoded features Z_a, Z_b. Finally, the desired reciprocal compressed features Z_a and Z_b can directly be used for quantization. The reason why we choose Z_a, Z_b instead of $\widehat{H_a}$ and $\widehat{H_b}$ that is also highly correlated is that they eliminate redundancy from adjacent subcarriers that may mitigate randomness [26].

We propose a hybrid loss function, which consists of three parts:

(1) We adopt the mean squared error loss for Z_a and Z_b to enforce the proposed neural network to learn the reciprocity between paired channel responses, as shown below:

$$L_1 = \frac{1}{m}\sum_{i=1}^{m}\left\|Z_a^i - Z_b^i\right\|_2, \tag{2}$$

where Z_a, Z_b represent output of Encoder$_a$ and Encoder$_b$ respectively, and m is the number of training samples.

(2) The second loss function is based on formula (1). It is introduced to minimize the difference between the channel response recovered by the decoder plus the nonreciprocal part extracted by NRLM and the original input of Alice, as shown below:

$$L_2 = \frac{1}{m}\sum_{i=1}^{m}\left\|\widehat{H_a^i} + N_a - H_a^{i'}\right\|_2, \tag{3}$$

where $\widehat{H_a^i}$ is the output of Decoder for Z_a^i, N_a is the noise residing in Alice side, and $H_a^{i'}$ is imperfect CSI input from Alice. The result of loss restricts the difference between the result of reconstruction plus the nonreciprocal part extracted by NLRM and the original input to be small enough to ensure that the encoder has really learned the main channel features.

(3) The third loss function is similar to the second, but it is for Bob's channel response, as shown below:

$$L_3 = \frac{1}{m}\sum_{i=1}^{m}\left\|\widehat{H_b^i} + N_b - H_b^{i'}\right\|_2, \tag{4}$$

where $\widehat{H_b^i}$ is the output of Decoder for Z_b^i, N_b is the noise residing in Bob side, and $H_b^{i'}$ is imperfect CSI input from Bob.

The proposed final loss function is expressed as below:

$$\text{Loss} = L_1 + L_2 + L_3. \tag{5}$$

3.3. Data Collection and Preprocessing of Dataset. Our data is collected in a variety of scenarios, including mobile and static, and in both indoor and outdoor situations. Using two Lenovo X220 laptops equipped with the Intel WiFi Link 5300 wireless card, we acquired 100,000 pairs of CSI data in each scenario.

The signal from two transmitting antennas and three receiving antennas are recorded and parsed to CSI values by the Linux 802.11n CSI Tool [38]. By utilizing the antenna diversity [16, 26], the estimated CSI has better randomness than a Single-Input and Single-Output (SISO) system. Each packet's CSI value is a complex matrix with the shape of $30 \times 2 \times 3$, which extracts from 30 subcarriers. Since neural networks cannot deal with complex numbers, the first step in preprocessing the datasets is to stack the real and imaginary parts of CSI matrix H'_u, which can be defined as

$$H'_u \longrightarrow \left(\text{Real}\left(H'_u\right), \text{Imag}\left(H'_u\right)\right), \tag{6}$$

where $\text{Real}(\cdot), \text{Imag}(\cdot)$ denote the real and imaginary parts of the CSI matrix. The stacked matrix is then flattened to a one-dimensional vector with a size of 360. Because each dimension of the raw input regular has a distinctive magnitude, we normalize the datasets so that their range is between 0 and 1. The process for normalizing is as follows:

$$H'^{l}_{u_{\text{norm}}} = \frac{H'^{l}_{u} - H'^{l}_{u_{\min}}}{H'^{l}_{u_{\max}} - H'^{l}_{u_{\min}}}, \tag{7}$$

where $H'^{l}_{u_{\max}}$ and $H'^{l}_{u_{\min}}$ are the max value and minimum value of l th dimension of H'_u, H'^{l}_{u} is the l th element of the H'_u, and $H'^{l}_{u_{\text{norm}}}$ is the normalized l th element of H'_u.

3.4. Secret Key Generation Scheme Based on Reciprocity Learning. We design a complete key generation scheme based on our proposed reciprocity learning algorithm, which is present in Figure 3.

In the training phase, we need to collect enough CSI data to serve as the training dataset of our model. In particular, Alice sends a probing packet to Bob at a rate of 10 packets per second during the channel probing stage. When Bob receives the packet, he replies an ACK packet to Alice immediately. We guarantee that the time interval between receiving the corresponding packet at both ends is less than 10 ms, which is much less than the coherence time, so the channel can be regarded constant within this time interval. This process is repeated until we collect enough channel responses. After preprocessing the data according to the above method, we randomly shuffle the dataset, selecting 80% of it for training the proposed deep learning model and the rest 20% for testing. The PyTorch framework is used to implement the proposed neural network, which is trained for 50 epochs using the Adam algorithm. The batch size is set to 128 and the learning rate is set at $1e-3$.

In the key generation phase, we fix the network parameters of the model, and then equip Encoder_a on Alice and Encoder_b on Bob, respectively. The Decoder and NRLM are only used to ensure that the model can suppress nonreciprocal noise and help encoders learn the main features of channel response during training. This overhead does not exist during the operational phase.

We can intuitively observe whether CRLNet has extracted the reciprocity feature of the channel. The dispersion of channel characteristics throughout 30 subcarriers is illustrated in Figure 4. Figure 5 shows the channel feature processed by CRLNet, which has a high degree of reciprocity.

The collected channel features are converted to a binary key using the uniform quantization approach [2]. Each bit of the key sequence Q can be calculated as

$$Q^i = \begin{cases} 1, & x^i \geq q^+, \\ 0, & x^i \leq q^-, \\ -1, & \text{else}, \end{cases} \tag{8}$$

where Q^i is the i th bit of generated key and x^i is the i th element of obtained reciprocal feature. q^+ and q^- are defined as

$$\begin{aligned} q^+ &= F^{-1}(0.5 + \varepsilon) \\ q^- &= F^{-1}(0.5 - \varepsilon), \end{aligned} \tag{9}$$

where F^{-1} is the inverse of the cumulative distribution function (CDF) of Gaussian distribution $N(\mu, \sigma^2)$ and μ, σ are the mean and standard deviation value of the produced feature. ε is a quantization factor corresponding to the environment. The values which are quantified to -1 are deleted from the initial key from both Alice and Bob.

For information reconciliation, we can adopt Cascade [33] or BCH code [35] protocol. With regard to privacy amplification, hash functions [37] can be used to convert the key to a fixed length secret key, which can be used for encryption directly. However, only the performance of the initial secret key is evaluated without information reconciliation and privacy amplification to ensure that the comparison is fair.

4. Results and Discussion

4.1. Performance of Deep Learning Model. We compare the performance of CRLNet with the following three benchmark models in the four diverse environments (indoor static, indoor mobile, outdoor static, and outdoor mobile):

(1) AE [28] is a normal single-branch autoencoder model, whose encoder part and decoder part have the same structure as the CRLNet.

(2) FNN [31]: the FNN is a model for UL/DL channel calibration in generic massive MIMO systems. It is a multilayer perceptron with three hidden layers.

(3) KGNet [32]: the KGNet is proposed for band feature mapping function for key generation in FDD systems.

Because these comparison models have only single input and single output, these networks function as the mapping between CSI of Alice and CSI of Bob. The network's input is CSI of Alice, and mean square error is used to minimize the difference between the network output and CSI of Bob. The network's input and output are a pair of reciprocal features for these benchmark models. The mean square error (MSE) between the reciprocal features Z_a, Z_b is utilized to compare the performance of the neural networks, which is described as

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} \left\| Z_a^i - Z_b^i \right\|_2. \tag{10}$$

The MSE indicates the ability of the model to learn the channel reciprocity. Figure 6 shows the MSE comparison results in all testing scenarios. We observe that our model performs better than all other benchmark models in these scenarios, while the performance of AE is worse than that of CRLNet, which means that the NRLM we designed really learned to eliminate the nonreciprocal part from channel response.

To prove the efficiency of the proposed hybrid loss function, we compare the performance of trained model with L_1 and without L_1 loss function. The existence of reconstruction loss L_2 and L_3 loss functions is necessary, because it is used to ensure that the model has really learned the main features of the channel response. As shown in Figure 7, the CRLNet with the L_1 loss function performs excellently, while the model without the L_1 loss function fails

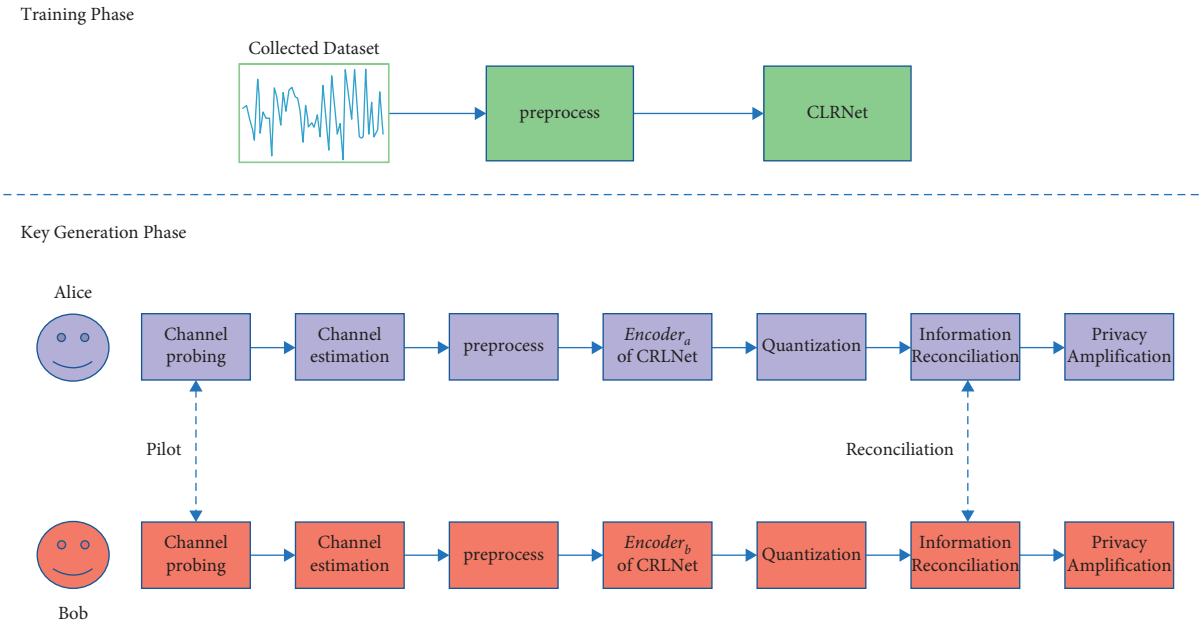

Figure 3: The proposed key generation procedure based on CRLNet.

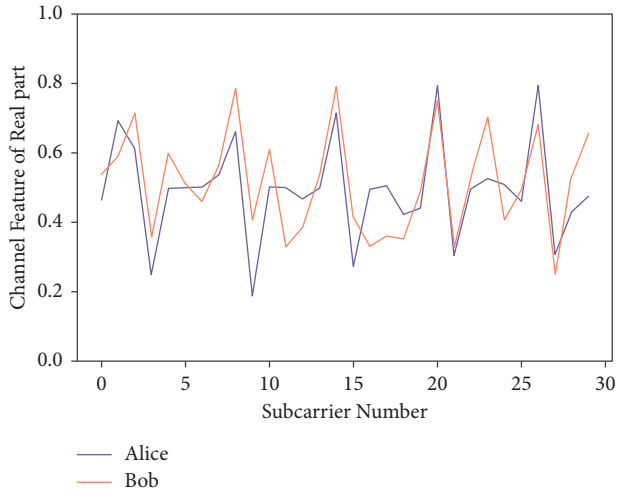

Figure 4: Channel response of Alice and Bob that contains nonreciprocal part before using CRLNet.

to achieve an acceptable MSE. The comparison results verify the irreplaceable role of L_1 for learning channel reciprocity.

The task of NRLM is to separate the nonreciprocal part from the input channel response. To study the influence of different network design of NRLM on the resulting reciprocal feature extraction performance, the following experiment was conducted. As shown in Figure 8, we compared original CRLNet and three modified CRLNet equipped with distinct NRLM modules in terms of MSE. The NRLM_1 is the NRLM we proposed above, the NRLM_2 adds an extra hidden layer on the original NRLM, the NRLM_3 increases the number of neurons in each hidden layer of original NRLM, and the NRLM_4 is composed of three 1D convolutional layers. The specific parameters of these modules are given in Table 1. We can see that increasing the hidden layer's depth or width does not bring a meaningful improvement. Using a 1D convolutional layer to replace the fully connected layer can get lower MSE, but it will significantly increase overhead in training phase. Therefore, the original NRLM has a good tradeoff between performance and computational cost.

4.2. Performance of Key Generation. We employ different models in the reciprocity feature extraction step to compare the performance of the key generation; the other steps remain the same. We use the following metrics to verify the effectiveness of the initial generated key:

(1) Key Error Rate (KER) is defined as the number of conflict bits in the initial keys generated by two devices divided by the total number of the generated bits.

(2) Key Generation Rate (KGR) is defined as the number of bits generated by each probing packet.

(3) Randomness: the standard NIST test suite [39] is used to measure the randomness of the initial key.

We first compare average KER of initial key of the four deep learning models under testing dataset in different environments. As shown in Figure 9, CRLNet outperforms other benchmark models in terms of KER. When the KER of other models is too high to be used for matched key establishment in practice, a suitably low KER is achieved in all test cases.

Our model and the benchmark model produce reciprocity feature with different sizes; therefore the KGR cannot be directly compared. For a fair comparison, we first use the PCA method to reduce the output features of the three comparison models to the same dimension as the CRLNet and then quantize the features to the initial keys. Figure 10 indicates that CRLNet has the highest KGR under all experimental scenarios.

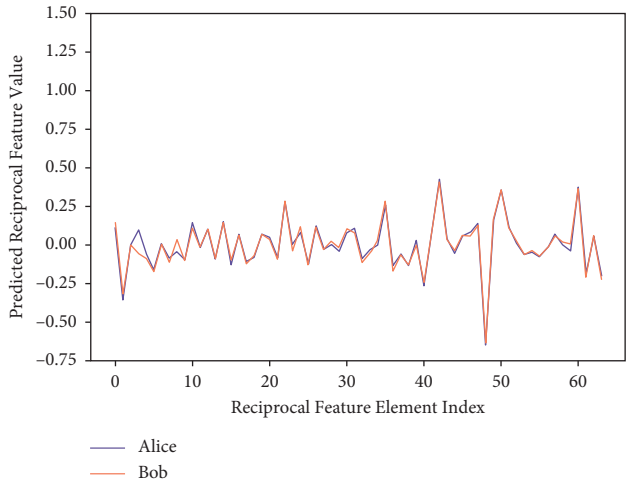

Figure 5: Channel response of Alice and Bob after using CRLNet, which is highly reciprocal.

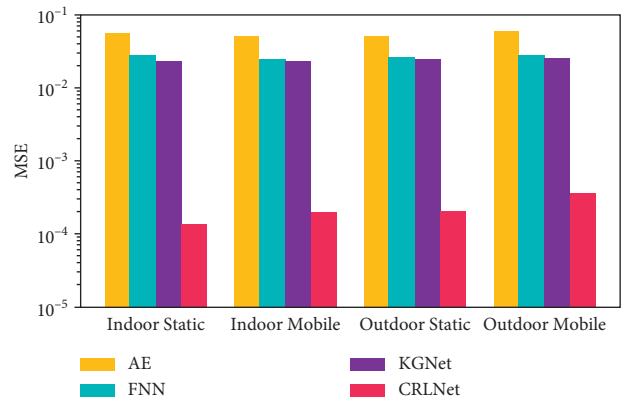

Figure 6: The comparison results of CRLNet and benchmark models in terms of MSE in different testing cases. It can be observed that CRLNet outperforms other neural networks.

In Figures 11 and 12, we compare KER and KGR with different quantization factors ε of 0.01, 0.05, 0.1, and 0.15 in all scenarios. The result demonstrates that as the quantization factor increases, both the KER and KGR decrease, implying a performance tradeoff. To acquire the best performance in the real world, we can adjust the quantization factor based on the signal-to-noise ratio (SNR) of the environment.

To verify the sufficient randomness of the generated keys, we perform NIST statistical tests on testing datasets. Table 2 shows the test results in four environments with the quantization factor of 0.01. All the cases pass the test and have the p-value much larger than 0.01, which is the threshold to pass the test.

4.3. Impact of the SNR. In order to prove the generalization performance of the proposed key generation method in complex scenarios, we added artificial Gaussian white noise, which caused the SNR to change from 0 dB to 20 dB. We compared our method with three benchmark methods

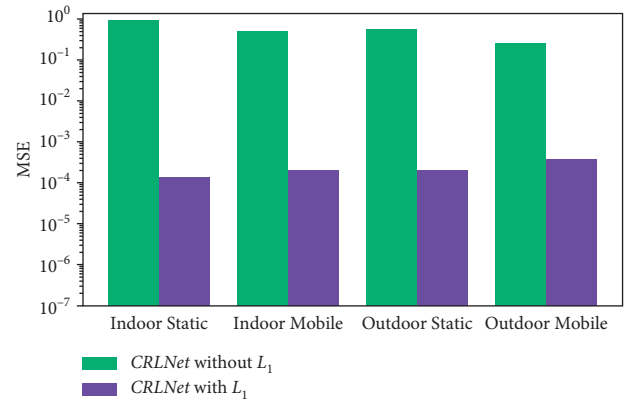

Figure 7: The MSE of CRLNet trained with L_1 and without L_1 loss function in different testing cases. This highlights how important L_1 loss is for CRLNet.

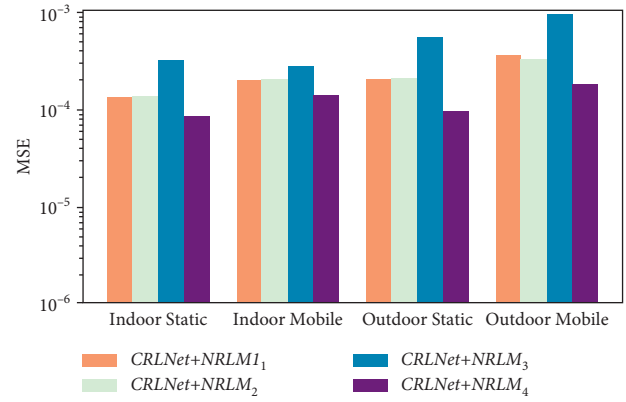

Figure 8: Compare CRLNet performance with the original and modified NRLM in terms of MSE.

[28, 31, 32], in terms of KER versus signal-to-noise ratio. As shown in Figure 13, the CRLNet based key generation scheme outperforms the counterparts in respect of KER. This demonstrates that the benefit of our method is that it is independent of the channel's statistical properties and can generate a consistent key based on the training dataset in an adaptable and effective manner.

4.4. Computational Complexity. The difference in computational complexity between our method and existing physical layer key generation methods is in the reciprocity feature extraction stage. In our method, there is no information sharing across legitimate nodes during the key generation process. In addition, our feature extraction only requires a single encoder of the CRLNet, which retains a low overhead.

In Figure 14, we compared the average execution time of the feature extraction stage for single packet between the proposed method and the above benchmark methods. The hardware environments we use are as follows: AMD Ryzen 5600X CPU, 16 GB RAM, Windows 10 Home 64-bit operating system. For all comparison methods, we use it to process 20,000 packets and calculate the average execution

Table 1: Parameters of original and modified NRLM.

Name	Hidden layers parameters
$NRLM_1$	(512, 256, 256)
$NRLM_2$	(512, 256, 256, 128)
$NRLM_3$	(1024, 512, 512)
$NRLM_4$	$conv_{1\times3}$, relu, $conv_{1\times3}$, relu, $conv_{1\times3}$

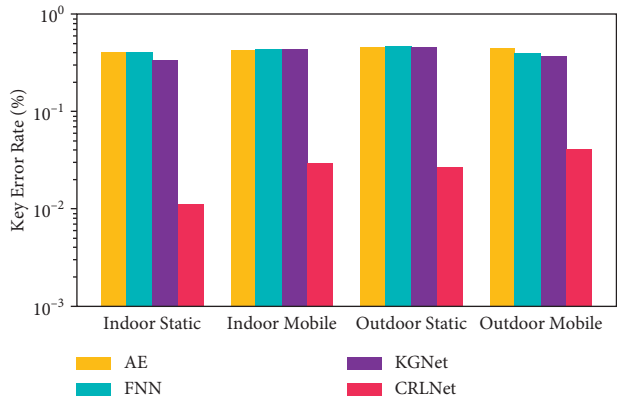

Figure 9: The KER of comparison model in all testing cases with the quantization factor ε is set to 0.01. The CRLNet achieve much lower KER than other models.

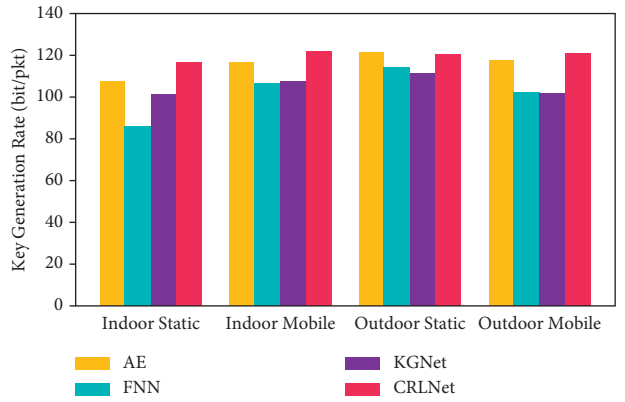

Figure 10: The KGR of comparison model in all testing cases with the quantization factor ε is set to 0.01. The CRLNet achieve the highest KGR.

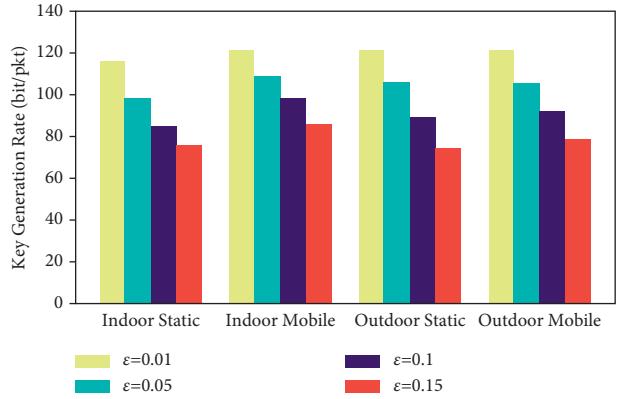

Figure 11: The KGR of CRLNet using different quantization factor in all testing cases.

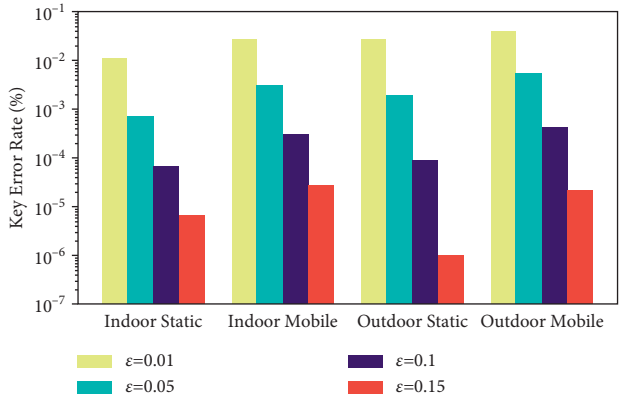

Figure 12: The KER of CRLNet using different quantization factor in all testing cases.

Table 2: NIST statistical test suite results in different scenarios with the quantization factor of 0.01.

Scenario	Indoor static	Indoor mobile	Outdoor static	Outdoor mobile
Approximate entropy	0.798	0.815	0.861	0.854
Cumulative sums	0.497	0.862	0.747	0.903
DFT	0.466	0.524	0.610	0.583
Frequency	0.451	0.779	0.605	0.822
Ranking	0.375	0.416	0.379	0.471
Runs	0.837	0.605	0.481	0.524
Serial	0.316	0.523	0.877	0.901
	0.452	0.506	0.708	0.924

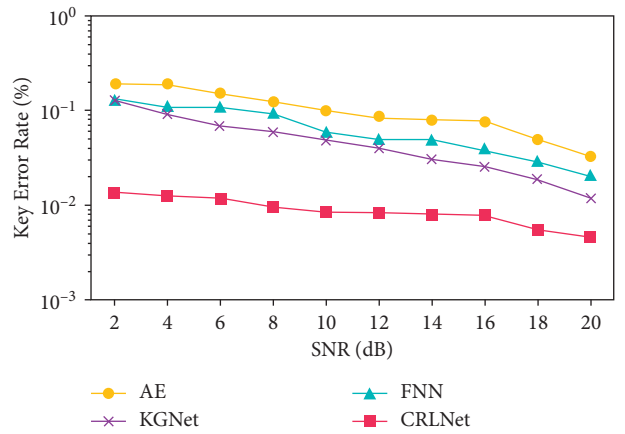

Figure 13: The KER of the four methods versus SNR of dataset.

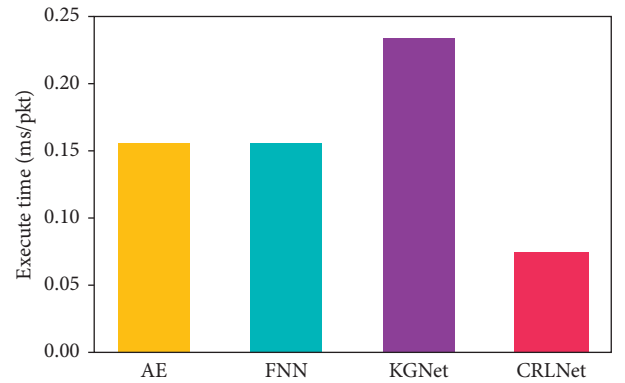

Figure 14: The execution time of channel feature extraction.

time for each packet. This process is repeated 20 times, and the average value is taken as the final result. As can be seen, our proposed method has the shortest execution time because it only relies on the lightweight encoder network module to extract reciprocal features.

5. Conclusions

In this paper, we propose a physical layer key generation scheme that can generate consistent keys from imperfect wireless channels, by designing deep neural networks to extract channel reciprocity features. The developed CRLNet can efficiently learn the reciprocity component of channel state information (CSI) in TDD OFDM systems. Based on the CRLNet, we design a complete key generation scheme that performs excellently on commercial WiFi devices. Extensive experiments are conducted under static and mobile environments in both indoor and outdoor scenarios. The results confirm that CRLNet can extract reciprocal channel features more efficiently than the benchmark neural networks in terms of MSE. Furthermore, the CRLNet-based key generation scheme achieves higher KGR, lower KER, and sufficient randomness compared to the existing methods in all testing scenarios.

Data Availability

The experimental CSI data used to support the findings of this study have been deposited in the GitHub repository (https://github.com/hehaoyulkeke/csi-data).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Haoyu He and Yanru Chen contributed equally to this work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (grant no. 62072319), the Science and Technology on Communication Security Laboratory (grant no. 6142103190415), the Luzhou Science and Technology Innovation R&D Program (grant no. 2021CDLZ-11), Chengdu Science and Technology R&D Project (grant no. 22ZDYF3672), and Chengdu Science and Technology R&D Project (grant no. 21ZDYF0393).

References

[1] A. Sayeed and A. Perrig, "Secure wireless communications: secret keys through multipath," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3013–3016, Las Vegas, NV, USA, April, 2008.

[2] 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.

[3] J. Zhang, G. Li, A. Marshall, A. Hu, and L. Hanzo, "A new Frontier for IoT security emerging from three decades of key generation relying on wireless channels," *IEEE Access*, vol. 8, Article ID 138406, 2020.

[4] Y. M. Al-Moliki, M. T. Alresheedi, Y. Al-Harthi, and A. H. Alqahtani, "Robust lightweight channel-independent OFDM-based encryption method for VLC-IoT networks," *IEEE Internet of Things Journal*, vol. 4662, no. c, p. 1, 2021.

[5] Y. M. Al-Moliki, M. T. Alresheedi, and Y. Al-Harthi, "Improving availability and confidentiality via hyperchaotic baseband frequency hopping based on optical OFDM in VLC networks," *IEEE Access*, vol. 8, pp. 125013–125028, 2020.

[6] C. Zenger, J. Zimmer, and C. Paar, "Security Analysis of Quantization Schemes for Channel-Based Key Extraction," *Security and Safety*, vol. 2, 2015.

[7] Y. Sun, Z. Tian, M. Li, C. Zhu, and N. Guizani, "Automated attack and defense framework toward 5G security," *IEEE Network*, vol. 34, no. 5, pp. 247–253, 2020.

[8] Y. Sun, Z. Tian, M. Li, S. Su, X. Du, and M. Guizani, "Honeypot identification in softwarized industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5542–5551, 2021.

[9] G. Li, Z. Zhang, J. Zhang, and A. Hu, "Encrypting wireless communications on the fly using one-time pad and key generation," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 357–369, 2021.

[10] 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.

[11] 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.

[12] A. Goel and V. P. Vishwakarma, "Efficient feature extraction using DCT for gender classification," in *Proceedings of the IEEE International Conference on Recent Trends in Electronics*, Bangalore, India, May, 2017.

[13] L. Cheng, L. Wei, D. Ma, Z. Li, and J. Wei, "Towards an effective secret key generation scheme for imperfect channel state information," in *Proceedings of the 2016 IEEE TrustCom*, Tianjin, China, August, 2016.

[14] W. J. Wang, H. Y. Jiang, X. G. Xia, and M. Q. Yin, "A wireless secret key generation method based on Chinese remainder theorem in FDD systems," *Science China Information Sciences*, vol. 55, 2012.

[15] X. Wu, Y. Peng, C. Hu, Z. Hui, and S. Lei, "A Secret Key Generation Method Based on CSI in OFDM-FDD System," in *Proceedings of the Globecom Workshops*, Atlanta, GA, USA, December, 2014.

[16] C. Point, "Fast and Practical Secret Key Extraction by Exploiting Channel Response," in *Proceedings of the 2013 Proceedings IEEE INFOCOM*, pp. 3048–3056, Turin, Italy, April, 2013.

[17] M. Zoli, A. N. Barreto, S. Köpsell, P. Sen, and G. Fettweis, "Physical-Layer-Security Box: a concept for time-frequency channel-reciprocity key generation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, pp. 1–24, 2020.

[18] H. Liu, C. Zhang, H. Fei, W. Hu, and D. Guo, "Feedback-based channel gain complement and cluster-based quantization for physical layer key generation," in *Proceedings of the*

2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), pp. 1–6, Kolkata, India, December, 2020.

[19] 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.

[20] 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.

[21] 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.

[22] H. Ye, F. Gao, J. Qian, H. Wang, and G. Y. Li, "Deep learning-based denoise network for CSI feedback in FDD massive MIMO systems," *IEEE Communications Letters*, vol. 24, no. 8, pp. 1742–1746, 2020.

[23] J. E. Hershey, A. A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *IEEE Transactions on Communications*, vol. 43, no. 1, pp. 3–6, 1995.

[24] 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, California, USA, September, 2008.

[25] S. N. Premnath, S. Jana, and J. Croft, "Secret key extraction from wireless signal strength in real environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 917–930, 2012.

[26] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong, "Efficient key generation by exploiting randomness from channel responses of individual OFDM subcarriers," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2578–2588, 2016.

[27] J. Zhao, W. Xi, and J. Han, "Efficient and Secure Key Extraction Using CSI without Chasing Down Errors," 2012, https://arxiv.org/abs/1208.0688.

[28] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[29] A. Y. Abyaneh, A. H. G. Foumani, and V. Pourahmadi, "Deep Neural Networks Meet CSI-Based Authentication," 2018, https://arxiv.org/abs/1812.04715.

[30] D. Liu, T. Wang, and S. Liu, "Contrastive self-supervised representation learning for sensing signals from the time-frequency perspective," in *Proceedings of the International Conference on Computer Communications and Networks, ICCCN*, Athens, Greece, 2021 July.

[31] C. Huang, G. C. Alexandropoulos, A. Zappone, C. Yuen, and M. Debbah, "Deep learning for UL/DL channel calibration in generic massive MIMO systems," in *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May, 2019.

[32] 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*, p. 1, 2021.

[33] X. Zhu, F. Xu, and E. Novak, "Extracting secret key from wireless link dynamics in vehicular environments," in *Proceedings of the IEEE INFOCOM*, pp. 2283–2291, Turin, Italy, April, 2013.

[34] M. Bloch, J. Barros, M. R. D. Rodrigues, and S. W. McLaughlin, "Wireless information-theoretic security," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2515–2534, 2008.

[35] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 523–540, Interlaken, Switzerland, May, 2004.

[36] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *Proceedings of the IEEE INFOCOM*, pp. 927–935, Orlando, FL, USA, March, 2012.

[37] S. Jana, S. N. Premnath, and M. Clark, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, pp. 321–332, Beijing, China, September, 2009.

[38] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release," *ACM SIGCOMM-Computer Communication Review*, vol. 41, no. 1, p. 53, 2011.

[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*, Createspace Independent Pub, Scotts Valley, California, US, 2001.