Bipartite Graph Matching Based Secret Key Generation

Hongbo Liu*, Yan Wang†, Yanzhi Ren*, Yingying Chen‡
*Dept. of CS, UESTC †Dept. of CIS, Temple University ‡WINLAB, Rutgers University
hongbo.liu, renyanzhi05@uestc.edu.cn *y.wang@temple.edu* *yingche@scarletmail.rutgers.edu*

IEEE INFOCOM 2021 - IEEE Conference on Computer Communications | 978-1-6654-0325-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOM42981.2021.9488848

***Abstract*—The physical layer secret key generation exploiting wireless channel reciprocity has attracted considerable attention in the past two decades. On-going research have demonstrated its viability in various radio frequency (RF) systems. Most of existing work rely on quantization technique to convert channel measurements into digital binaries that are suitable for secret key generation. However, non-simultaneous packet exchanges in time division duplex systems and noise effects in practice usually create random channel measurements between two users, leading to inconsistent quantization results and mismatched secret bits. While significant efforts were spent in recent research to mitigate such non-reciprocity, no efficient method has been found yet. Unlike existing quantization-based approaches, we take a different viewpoint and perform the secret key agreement by solving a bipartite graph matching problem. Specifically, an efficient dual-permutation secret key generation method, DP-SKG, is developed to match the randomly permuted channel measurements between a pair of users by minimizing their discrepancy holistically. DP-SKG allows two users to generate the same secret key based on the permutation order of channel measurements despite the non-reciprocity over wireless channels. Extensive experimental results show that DP-SKG could achieve error-free key agreement on received signal strength (RSS) with a low cost under various scenarios.**

I. Introduction

As the expansion of wireless communications, establishing cryptographically secure communication links between a pair of transceivers becomes an urgent need nowadays. Traditional symmetric or asymmetrical cryptographic algorithms [1] mainly rely on generating secret keys based on pre-agreed information to protect users' information from adversarial access. However, many of these algorithms are not applicable to device-to-device communications in practice due to the lack of key management infrastructures and limited resources of mobile devices. To enable practical secure communication, researchers have proposed to exploit the inherent physical properties of wireless channels to complement or enhance the traditional cryptographic mechanisms [2]. Along this direction, secret key generation leveraging wireless channel reciprocity becomes a promising option, which extracts secret bits from a shared random channel between a pair of wireless transceivers [3]–[10].

The vast majority of existing work for physical layer secret key generation involves four stages: channel probing, quantization, information reconciliation, and privacy amplification. Among all these stages, quantization is considered to be the core function that ensures different users achieve the secret key agreement in a reciprocal channel. During the quantization process, channel measurements are converted into binary vectors based on pre-defined quantitative rules to serve as preliminary secret key bits. However, the non-simultaneous channel probing in time division duplex system plus noise effects [4]–[6] would harm the channel reciprocity, causing inconsistent quantization results between two users and mismatched secret bits. For example, Radiotelepathy [3] first extracts secret keys using the channel impulse response (CIR) in the wireless channel, but its key generation rate is low without presenting statistical key mismatch rate. Patwari et al. [4] and Zeng et al. [5] apply multi-level adaptive quantization and multi-antenna diversity respectively to achieve higher key generation rate, but the mismatch rate is not satisfied in many scenarios. Liu et al. [6] proposes to use fine-grained Channel State Information for key generation and develops a channel gain complement mechanism to reduce the impact of non-reciprocity on quantization. Li et al. [9] proposes a mean-value quantization scheme to parse the RSS sequences for better key generation performance. Margelis et al. [10] even adopts Discrete Cosine Transform (DCT) on channel observations to reduce the mismatches caused by quantization. As shown above, huge efforts have been spent to mitigate such non-reciprocity over a wireless channel but with limited improvement, especially when dealing with steady channel measurement sequences. Therefore, an efficient and robust solution is required to achieve error-free secret key generation to facilitate fast secret key generation using physical layer information.

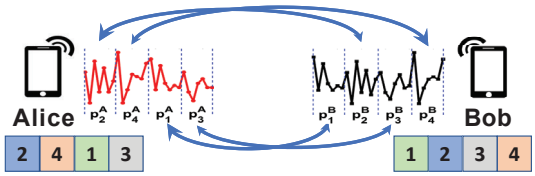


Fig. 1. Basic idea of DP-SKG.

We find that the traditional quantization process is the bottleneck of physical layer secret key generation because it is prone to unstable wireless channels and dynamic noises, resulting in high bit mismatching. To avoid the quantization process while maintaining the reciprocal secrecy between legitimate users, we take a different viewpoint to propose a dual-permutation secret key generation method based on minimum weight bipartite graph matching, DP-SKG, which provides high accuracy and robustness for secret key agreement. Based on DP-SKG, we develop a secret key generation method that

aims at extracting secret bits from RSS at low mismatch rate and high speed. We chose to use RSS because it is readily available in all wireless devices at a low cost. Although existing work has shown that channel state information (CSI) could provide more channel information and high secret bit generation rate than RSS does, we show that our method could also achieve high bit generation rate and error-free key agreement by using RSS. Our method could significantly improve the practicality of the physical layer secret key generation in general. Specifically, suppose a pair of users, say Alice and Bob, seeking for shared key establishment, Bob follows Alice to apply the same random permutation (1st permutation) to their sequences of channel measurements without revealing the values of channel measurements. The 1st permutation does not destroy the channel reciprocity but increases the complexity of the channel measurement sequence, which helps mitigate the ambiguous matching caused by ambient noise. Next, both Alice and Bob segment their permuted channel sequence into multiple episodes. In the meanwhile, Alice applies the 2nd random permutation to these channel episodes to construct a new sequence, which then will be sent to Bob. By minimizing the discrepancy between Alice's permuted episodes and Bob's episodes via bipartite graph matching, Bob can infer the episode permutation order (i.e., 2nd permutation) and use it to generate secret bits. Figure 1 illustrates the basic idea of our approach. Since it has been proved that the channel measurements are secret information unknown to attackers who are at a reasonable distance to the users (i.e., over half a wavelength), the random permutation order of the channel sequences is also secure and unknown to the attackers.

Realizing the proposed method faces many challenges. For instance, when using the minimum weight bipartite graph matching, it is critical to choose appropriate weight metric and efficient searching algorithm to ensure robust and fast matching between the RSS sequences. Although our DP-SKG method can achieve highly accurate key generation performance, it can not completely avoid mismatched episodes. Therefore, an efficient information reconciliation scheme is also required with a limited number of challenge-response exchange to accurately identify the mismatched episodes. Besides reducing the mismatch errors, the proposed DP-SKG method will also integrate a channel reuse mechanism leveraging the different permutations of channel measurements to boost the key generation rate.

The main contributions of this work are listed as follows:

- We take an unconventional approach to achieve significantly low bit mismatch rate and high bit generation rate by modeling the secret key agreement process as a bipartite graph matching problem instead of the quantization process.
- We propose a new and practical dual-permeation secret key generation method, DP-SKG, based on minimum weight bipartite graph matching that would greatly mitigate the impact of ambient noise and achieve highly accurate key agreement.
- We develop an efficient information reconciliation scheme and channel reuse mechanism to enhance the proposed DP-SKG method and ensure fast and error-free key agreement with high entropy.
- It is demonstrated with extensive experiments on real dataset that the proposed DP-SKG can achieve secret key agreement in a timely manner under various practical scenarios.

The rest of the paper is organized as follows: We introduce the related work on existing secret key generation techniques in Section II. In Section III, we present our problem formulation, preliminary study, and attack model. Next, we describe the detailed algorithm procedures and security analysis in Section IV. The performance is extensively evaluated through real experimental dataset in Section V. Finally, we conclude our work in Section VI.

II. Related Work

There have been ongoing studies on secret key generation leveraging the reciprocity property in a wireless channel. Among all these existing studies, quantization usually acts as the core function to achieve the secret key agreement between different users. Various physical layer characteristics over the wireless channel have been exploited to facilitate the quantization. RSS measurements, which are readily available in existing wireless infrastructures, have been widely exploited to generate secret bits. For instance, some existing RSS based methods utilize temporal and spatial variations of radio channel [3], [4], [11] and antenna diversity [5] for secret bit generation. Li et al. [9] propose a mean-value quantization scheme to parse the RSS sequences for better key generation performance. Margelis et al. [10] even adopt the Discrete Cosine Transform (DCT) on channel observations to reduce the mismatches caused by quantization. These methods have limited key generation rates and agreement rates, especially in static environments.

Meanwhile, phase information of the channel response [12]–[14] has been used as an alternative characteristic to facilitate secret key generation. However, the channel phase may not always be reciprocal since the hardware characteristics vary among different devices, resulting in a high disagreement rate. Furthermore, Tope *et. al.* [15] utilize the randomness of the received signal's envelope to share the secrecy between two parties. Similarly, secret bits can also be extracted from the deep fades of channel gain caused by multipath [16]. But these schemes are based on theoretical analysis, and only simulation results are provided.

Recent studies have provided a richer source of secret information by utilizing the Channel State Information (CSI). Liu *et al.* [17] show the feasibility of using CSI for secret key extraction. A fast and practical secret key extraction scheme [6] is also proposed by developing channel gain complement mechanism to reduce the impact of non-reciprocity on quantization, but the key agreement rate is still limited in some scenarios. Moreover, Xi *et al.* [18], [19] propose a validation-recombination mechanism by combining information of all sub-carriers to be resilient to key cracking attacks, and Zhang *et al.* [20] leverage the Orthogonal Frequency-Division Multiplexing Access (OFDMA) to extend key gener-

ation to multiple users. However, CSI is not available in many wireless devices without equipping dedicated hardware, thus restricting it from widely adopted for key generation. Unlike the aforementioned work, our work takes a different viewpoint to perform highly accurate and robust secret key agreement by solving a bipartite graph matching problem instead of the quantization based approaches. As such, our approach could enable low-cost, fast secret key generation in mobile devices without dedicated hardware.

III. Feasibility Study and Attack Model

A. Problem Formulation

Suppose that a probing signal, S, is transmitted from user Alice (i.e., A) to Bob (i.e., B), who are both located within a common communication environment, a sequence of channel measurements received by A is defined:

$$V^A = H^{BA} * S + \Pi^A \tag{1}$$

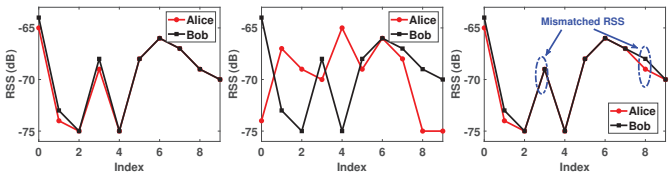
where $V^A = \left[v_1^A, v_2^A, \cdots, v_N^A\right]$, H^{BA} and Π^A are the channel response and the ambient noise observed by Alice, respectively. Similarly, the sequence of received channel measurement observed by Bob, V^B, is defined as:

$$V^B = H^{AB} * S + \Pi^B \tag{2}$$

where H^{AB} and Π^B are the channel response and ambient noise observed by Bob.

According to the principle of channel reciprocity, when Alice and Bob probe the channel between them within the channel's coherence time, the channel response H^{BA} and H^{AB} should be highly correlated in practice. Since the ambient noises Π^A and Π^B are usually considered to follow a zero-mean Gaussian distribution, the received channel measurements V^A and V^B should also be highly correlated. Based on the above theories, traditional secret key generation methods allow Alice and Bob to extract the same secret bit sequence by quantizing each channel measurement in V^A and V^B, respectively. However, V^A and V^B could be easily affected by non-simultaneous channel probing and random ambient noise [4], resulting in inconsistent quantization results and mismatched secret bit sequences between two users.

To address the secret bit mismatching issue, we propose to extract secret bits based on the permutation order of reciprocal channel measurements. Assuming Alice discloses a random permutation of her channel measurements as $\hat{V}^A = \left[v_{k_1}^A, v_{k_2}^A, \cdots, v_{k_N}^A\right]$, where $v_{k_n}^A \in V^A$, and $k_n \in [1, N]$ is the original index of channel measurement in V^A. Due to the channel reciprocity, the reciprocal matching on the channel measurements between Alice and Bob still exists even the order of channel measurements is disrupted. If the permutation order of V^A can be derived in an efficient way by comparing his channel measurements V^B with $\hat{V}^A$, it can serve as a secrecy between Alice and Bob for secret key generation even though $\hat{V}^A$ has been made public.



(a) Original RSS (b) RSS after permutation (c) RSS after matching

Fig. 2. Feasibility of using graph-based matching to determine the permuted order of RSS measurements.

B. Preliminaries and Challenges

To verify the aforementioned speculation, we further conduct a feasibility study to examine the performance of secret key generation using the permutation order of reciprocal channel measurements. Specifically, we deploy two laptops equipped with Intel 5300 NIC card operating in the 802.11n 2.4GHz channel as a pair of legitimate users, Alice and Bob, who wish to establish secret key agreement. Then two RSS sequences, each consisting of 10 RSS measurements, are collected from the probe packets exchanged between Alice and Bob as shown in Figure 2(a). Next, the RSS sequence of Alice is permuted and disclosed to the public as shown in Figure 2(b). At Bob's side, Bob keeps permuting the order of his RSS sequence and conduct exhaustive search for a complete graph mapping with minimum distance between the RSS sequences of Alice and Bob. Figure 2(c) shows the matching results, where most of the RSS sequence finds the corresponding right match except two RSS measurements as highlighted. The remaining questions are how to efficiently derive the permutation order and guarantee its uniqueness. An intuitive way is to search for a complete graph mapping between V^B and $\hat{V}^A$, if the measurements in both channel measurement sequences are considered as the vertices in a graph, with minimum discrepancy (e.g., minimum Euclidean distance). The inherent reciprocity and randomness of wireless channels, to some certain extent, would guarantee the uniqueness of such mapping.

Overall, the above preliminary study provides encouraging results on the feasibility of the proposed idea to facilitate key generation. However, realizing the proposed method in practice has some challenges that we need to address:

Robust Matching Algorithm. Ambient noise is the main cause of the mismatch between reciprocal channel measurements. Meanwhile, a steady channel measurement sequence with few sharp changes would amplify the impact of ambient noise, resulting in severe degradation of the key agreement performance. Furthermore, the instability of using a single channel measurement value for matching is also a problem as shown in Figure 2(b). Thus, a robust matching algorithm with sophisticated design should be developed to suppress the impact of ambient noise.

Efficient Searching Algorithm. Exhaustive searching with high computational complexity is not an optimal solution to determine permutation order. An efficient searching algorithm is required to achieve fast and practical matching between reciprocal channel measurements.

Effective Reconciliation Method. Robust matching algo-

rithm could minimize the errors caused by ambient noise, but it cannot guarantee error-free key generation due to occasional non-reciprocity over wireless channel. Therefore an effective information reconciliation method is a must to achieve error-free key generation.

C. Attack Model

We further consider two types of attacks that have been identified harmful to secret key generation in real environments.

Predictable Channel Attack [11]**:** When both Alice and Bob are stationary, the wireless channel between them is relatively stable. The attacker can use planned movements to cause desired and predictable changes in the channel measurements between Alice and Bob, referred as predictable channel attack. For example, it can be easily inferred that when the Line-of-Sight (LoS) between Alice and Bob is blocked (e.g., intentionally crossing the wireless links between Alice and Bob), the transmitted signal may suffer sharp attenuation.

Stalking Attack [21]**:** The stalker follows the trajectory of either Alice or Bob during the secret key establishment and eavesdrops all the legitimate communication between them. The Stalker measures the wireless channels between itself to Alice or Bob when Alice and Bob are exchanging probe packets. Moreover, Stalker also has the knowledge to perform secret key generation as Alice and Bob. We assume Stalker cannot be too close to either Alice or Bob, otherwise it increases the chances to expose himself to be detected.

IV. Algorithm Design

A. Basic Idea

The basic idea of our DP-SKG algorithm is to match the permuted channel measurements between a pair of reciprocal users, and perform key generation based on the agreed permutation order without involving error-prone quantization. As shown in Figure 3, assuming Alice and Bob has collected their respective channel measurement sequences, V^A and V^B of length N, during the *Channel sampling* stage. Alice comes up with a permutation order $[k_1, k_2, \cdots, k_N]$, where $k_n \in [1, N]$, and informs it to Bob during *Entropy-based Permutation*. Both Alice and Bob apply the same permeation order to their channel measurements and obtain two new sequences with high entropy as $\hat{V}^A = [v^A_{k_1}, v^A_{k_2}, \cdots, v^A_{k_N}]$ and $\hat{V}_B = [v^B_{k_1}, v^B_{k_2}, \cdots, v^B_{k_N}]$, respectively. $\hat{V}^A$ and $\hat{V}^B$ are next segmented into M episodes, $P^A = [p^A_1, p^A_2, \cdots, p^A_M]$ and $P^B = [p^B_1, p^B_2, \cdots, p^B_M]$, where p^A_m and p^B_m are m^{th} episode with length $L = N/M$. Next, Alice applies the *Random Episode Permutation* to its episodes P^A and constructs a new channel measurement sequence $\hat{P}^A = [p^A_{\kappa_1}, p^A_{\kappa_2}, \cdots, p^A_{\kappa_M}]$, where $\kappa_m \in [1, M]$ denotes the original index κ_m of episode $p^A_{\kappa_m}$ in P^A. $\hat{P}^A$ is then sent to Bob via a public channel without revealing the permutation order. By finding the episode having the most similar pattern between $\hat{P}^A$ and P^B through the *Bipartite Graph Matching*, Bob can infer the permutation order $PO = [\kappa_1, \kappa_2, \cdots, \kappa_M]$ of $\hat{P}^A$, which is unique and secret between Alice and Bob. Last, Alice and Bob perform the *Information Reconciliation and Privacy Amplification* on PO to further eliminate occasional errors and generate a secret key with high randomness. The details of these components are elaborated as follows.

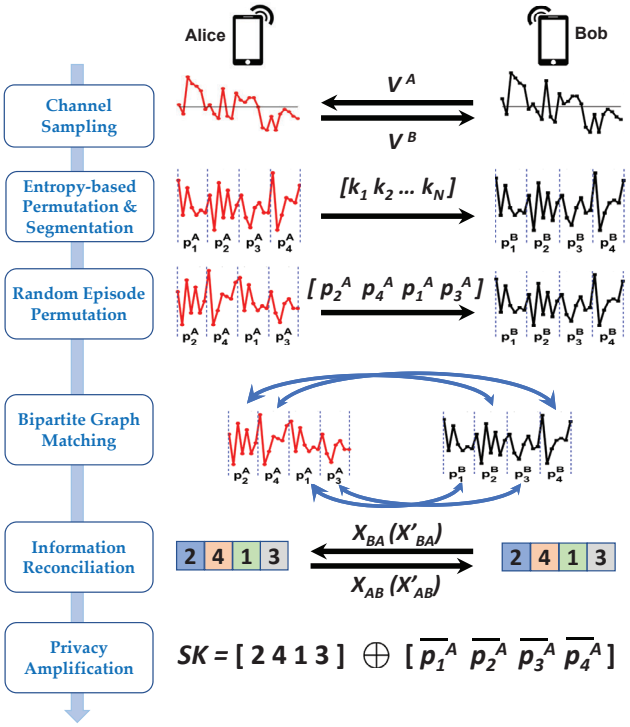


Fig. 3. DP-SKG algorithm flow.

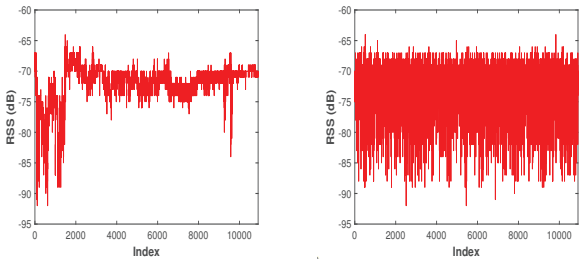
B. Channel Sampling

To initialize the secret key generation, Alice and Bob will perform channel sampling through multiple rounds of probe packet exchange, and each round is controlled within the coherence time to ensure channel reciprocity. After the probe packets are received at each end, the channel measurements will be extracted from the probe packets to construct reciprocal channel sequences V^A and V^B for Alice and Bob, respectively. Once a certain number of probe packets are collected, the channel sampling process is completed.

C. Entropy-based Permutation & Segmentation

In practice, the non-simultaneous channel sampling in time division duplex system plus ambient noise harm the channel reciprocity, which is the theoretical foundation of secret key generation. Particularly for the channel measurement sequences involving steady episodes as the example shown in Figure 4(a), the non-reciprocity in the channel measurement sequences could be further amplified by ambient noise, adversely affecting the subsequent matching operation between Alice and Bob. To mitigate such impact, we perform an entropy-based permutation to increase the complexity of the collected channel measurements and force the abrupt changes evenly distributed in the sequence.

To evaluate the complexity of the permuted channel measurement sequence, we adopt Sample Entropy (SE) [22] to provide insights into the complexity of fluctuations of data



(a) Original sequence (b) Permuted sequence

Fig. 4. Example of an RSS sequence after the entropy-based permutation with evenly distributed channel interruption.

sequences. SE is a negative logarithm of the conditional probability of the sequences of a data vector. If a vector of length τ has repeated itself in tolerance γ for t points, it will also do so for $t+1$ points. The conditional probability means the ratio of counts of repeated time of $t+1$ points to that of t points. So higher SE means lower regularity and more complexity in the channel measurement sequence. Due to the space limit, we refer to [22] for the derivation of SE.

We repeatedly apply random permutation to the channel measurement sequence of Alice V^A until its SE beyond a predefined threshold (i.e., 3 in our work). The permuted sequence $\hat{V}^A$ will involve a few steady episodes as the example shown in Figure 4(b). Alice then informs Bob about its permutation order $[k_1, k_2, \cdots, k_N]$ without revealing their actual channel measurement values, where $k_n \in [1, N]$ denotes the original index of channel measurement $v^A_{k_n}$ in V^A. Bob applies the same permutation order as Alice to V^B and obtains $\hat{V}^B$, so the channel reciprocity maintains and also ensures the $SE(\hat{V}^B)$ beyond the predefined threshold.

Next, both Alice and Bob segment their respective permuted sequence $\hat{V}^A$ and $\hat{V}^B$ into M episodes of the same length, where p^A_m and p^B_m denotes the mth episode in P^A and P^B, respectively. The episode length should be carefully chosen to balance the bit generation rate and mismatch rate. A long episode could ensure robust matching between P^B and a permuted P^A in the later stage, thereby achieving a low bit mismatch rate. However, it also takes more channel measurements to represent a single secret bit, lowering the key generation rate. Without stating elsewhere, we choose the default episode length as 7 in this work.

D. Random Episode Permutation

Given the segmented channel measurements P^A, Alice performs the random episode permutation to create a new channel measurement sequence $\hat{P}^A = \{p^A_{\kappa_1}, p^A_{\kappa_2}, \cdots, p^A_{\kappa_M}\}$, where $\kappa_m \in [1, M]$ is the original index of the episode $p^A_{\kappa_m}$ in P^A. Then $\hat{P}^A$ is broadcasted the public channel, where both Bob and potential attackers listen to. Since the original channel measurement sequence was not made public, the episode permutation order $PO = [\kappa_1, \kappa_2, \cdots, \kappa_M]$ is unknown to the attackers and can serve as the secrecy between Alice and Bob.

One concern is whether the attacker can infer the permutation order from the public $\hat{P}^A$ based on the correlation between its episodes. Since the *Entropy-based Permutation* has increased the entropy of $\hat{V}^A$ and the segmented $\hat{P}^A$, the correlation between any pair of episodes in $\hat{P}^A$ is largely reduced, leaving less chance for the attacker to infer the permutation order.

E. Bipartite Graph Matching

The channel reciprocity ensures that each episode $p^A_{\kappa_m}$ in $\hat{P}^A$, even permuted, can always find a reciprocal episode p^B_{κ} in P^B with a similar pattern. Since we propose to use the permutation order PO as the secret key, achieving key agreement between Alice and Bob is equivalent to determine a perfect matching between the episodes in $\hat{P}^A$ and P^B with the minimum discrepancy.

To facilitate the matching process, we formulate a minimum weight bipartite graph matching problem. Specifically, episodes in $\hat{P}^A$ and P^B are considered as vertices of a weighted undirect graph G, and the edges only exist between the episodes in $\hat{P}^A$ and P^B in G (i.e., no edge connects the vertices within $\hat{P}^A$ or P^B). We choose Manhattan distance as the weight of edges (κ_m, κ), $w_{A,B}(\kappa_m, \kappa) = \|p^A_{\kappa_m} - p^B_{\kappa}\|_1$, where $\kappa_m, \kappa \in [1, M]$. Manhattan distance is based on absolute value distance that usually gives more robust results than generic Euclidean distance, which is usually influenced by abnormal values. We seek for a perfect matching in G, which consists of a set of vertex-disjoint edges with every vertex of G. Due to the channel reciprocity, there is always a perfect matching in G to fulfill the reciprocal mapping between the channel measurement sequences of Alice and Bob. Specifically, we formulate the following Linear Programming with integer constraints relaxation aiming to minimize the summation of its associated edge weights:

$$\begin{aligned} &min \sum_{\kappa_m,\kappa} w_{A,B}(\kappa_m,\kappa)\cdot x_{A,B}(\kappa_m,\kappa) \\ &s.t. \sum_{\kappa_m} x_{A,B}(\kappa_m,\kappa) = 1, \quad \sum_{\kappa} x_{A,B}(\kappa_m,\kappa) = 1, \\ &x_{A,B}(\kappa_m,\kappa) > 0, \\ &\forall \kappa_m \in [\kappa_1, \kappa_2, \cdots, \kappa_M], \forall \kappa \in [1, 2, \cdots, M]. \end{aligned} \tag{3}$$

To complement the relaxation condition $x_{A,B}(\kappa_m, \kappa) > 0$, we also construct a feasible solution to the dual of Equation 3 with a value equal to the weight of the perfect matching output by the algorithm as follows:

$$\begin{aligned} &max \sum_{a\in\{\kappa_1,\kappa_2,\cdots,\kappa_M\}} y(a) + \sum_{b\in\{1,2,\cdots,M\}} y(b), \\ &s.t. y(a) + y(b) \le w_{A,B}(\kappa_m,\kappa), \forall (a,b) \in E, \end{aligned} \tag{4}$$

where E denotes all the edges in G. Equation 4 indicates that any feasible primal solution in Equation 3 (a perfect matching Λ) has the weight at least as large as the value of any feasible dual solution. That is, given any dual feasible solution y, if we can find a perfect matching Λ only using tight edges (i.e., an edge (a, b) is tight if $y(a) + y(b) = w(a, b)$), Λ must be optimal.

To solve the above LP problem, we maintain a feasible dual y and attempt to find a close-to-primal-feasible solution

(i.e., matching Λ) that satisfies complementary relaxation. Specifically, we begin with an arbitrary feasible dual solution y, and find a maximum-cardinality matching Λ' that uses only tight edges. If Λ is perfect, we are done; if not, we update our dual solution. This process continues until an optimal solution is found. The overall complexity is bounded by $O((|P^A| + |P^B|)^2)$ [23], so the total execution overhead will be small for various mobile platforms for efficient bipartite graph matching. After the graph matching, the permutation order inferred by Bob is defined as $PO' = [\kappa'_1, \kappa'_2, \cdots, \kappa'_M]$.

F. Information Reconciliation & Privacy Amplification

Bipartite graph matching can achieve highly accurate key agreement, but it is still essential to integrate information reconciliation to achieve error-free key agreement. Instead of adding redundant bits to the secret key as conventional methods [2], we propose to identify the mismatched episodes with challenge-response message exchange between Alice and Bob as follows:

$$\begin{aligned} &Alice \rightarrow Bob : X_{AB} = R \oplus PO \\ &Bob \rightarrow Alice : X_{BA} = X_{AB} \oplus PO' \oplus \left[\overline{p^A_{\kappa'_1}}, \cdots, \overline{p^A_{\kappa'_M}}\right] \end{aligned} \tag{5}$$

where R is a random vector of length M, $\overline{p^A_{\kappa'_m}}$ is the average value of κ'_mth episode in P^A, and $\oplus$ denotes the logical exclusive or operation.

Alice next identifies the indices of unequal elements between X_{BA} and $X_A = R \oplus \left[\overline{p^A_{\kappa_1}}, \cdots, \overline{p^A_{\kappa_M}}\right]$ as $\Phi = [\phi_1, \phi_2, \cdots, \phi_S]$, where $\overline{p^A_{\kappa_m}}$ is the average value of κ_mth episode in P^A and $\phi_s \in [1, M]$. But the actual mismatched ones are only a subset of Φ due to the difference between $\overline{p^A_{\kappa_m}}$ and $\overline{p^A_{\kappa'_m}}$ for some matched indices. To identify the actual mismatched ones, Alice randomly chooses S matched indices between X_{BA} and X_A as $\Psi = [\psi_1, \psi_2, \cdots, \psi_S]$, where $\Psi \cap \Phi = \emptyset$. Next Alice will inform Bob both Φ and Ψ, and send the challenge message X'_{AB}:

$$\begin{aligned} Alice \rightarrow Bob : X'_{AB} =& R' \oplus \left[\overline{p^A_{\kappa_{\psi_1}}}, \cdots, \overline{p^A_{\kappa_{\psi_S}}}\right] \\ Bob \rightarrow Alice : X'_{BA} =& X'_{AB} \oplus \left[\kappa'_{\phi_1}, \cdots, \kappa'_{\phi_S}\right] \\ & \oplus \left[\overline{p^A_{\kappa'_{\psi_1}}}, \cdots, \overline{p^A_{\kappa'_{\psi_S}}}\right] \end{aligned} \tag{6}$$

where R' is a random vector of length S, $\overline{p^A_{\kappa_{\psi_s}}}$ and $\overline{p^A_{\kappa'_{\psi_s}}}$ denote the average value of κ_{ψ_s}th and κ'_{ψ_s}th episode in P^A, respectively. The actual mismatched indices will be determined by identifying the unequal elements between X'_{BA} and $X'_A = R' \oplus [\kappa_{\phi_1}, \kappa_{\phi_2}, \cdots, \kappa_{\phi_S}]$. Since the mismatched indices usually appear in pairs, Alice only needs to swap the order of the mismatched indices, which correspond to the same weight value during bipartite graph matching, and the permutation order between Alice and Bob will be agreed (i.e., $PO = PO'$).

After information reconciliation, Alice and Bob will agree on an error-free secret key. But as a secret key, it also needs to fulfill a certain degree of randomness. Since P^A can be easily recovered from $\hat{P}^A$ with derived permutation order PO' by Bob, so Alice and Bob will obtain the agreed secret key as $SK = PO \oplus \left[\overline{p^A_1}, \overline{p^A_2}, \cdots, \overline{p^A_M}\right] = PO' \oplus \left[\overline{p^A_1}, \overline{p^A_2}, \cdots, \overline{p^A_M}\right]$ through privacy ampliation with exclusive or operation, where $\overline{p^A_m}$ is the average value of mth episode in P^A. Our experimental results show that SK achieves high randomness, which will be presented in experimental results.

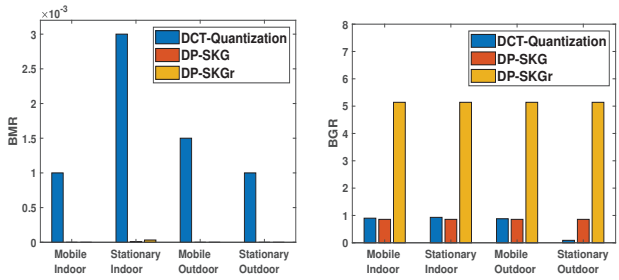
G. Boosting Key Generation Rate with Channel Reuse

A pair of reciprocal channel measurement sequence can only be used to generate one secret key between Alice and Bob, limiting the bit generation rate. We propose a channel reuse mechanism to boost the bit generation rate. Before each time of channel reuse, we reconstruct a new sequence by re-permuting the channel measurement sequence $\hat{V}^A$, even though the measurements in the sequence are the same, and then feed it into our DK-SKG method for another round of key generation. Specifically, given the secret key SK^i derived from last round of key generation between Alice and Bob, we exploit SK^i as a seed value of Fisher-Yates shuffle algorithm [24] to control the random permutation of $\hat{V}^A$. Since SK^i is an agreed secret key between Alice and Bob, Bob can apply the same permutation to $\hat{V}^B$ with the same seed SK^i without talking to Alice. Moreover, SK^i is unknown to the attackers, and there is no information exchange about the new permutation order between Alice and bob, so the new permutation order is also unavailable to the attacker. Once the new permutations of $\hat{V}^A$ and $\hat{V}^B$ are completed, the resulted sequences (i.e., $\tilde{V}^A$ and $\tilde{V}^B$) are used as the input to start the next round of DP-SKG for generating SK^{i+1}. In addition, if the complexity of $\tilde{V}^A$ and $\tilde{V}^B$ are sufficiently high, we can skip the entropy-based permutation and start from bipartite graph matching for key generation.

Given a channel measurement sequence of length N, theoretically, we can use the same channel measurements for $N!$ times with different permutation orders. However, many of the possible permutations have overlapped episodes, which may be exploited by attackers to infer the generated keys. Therefore, for a channel measurement sequence consisting of M episodes, we empirically choose to reuse the sequence for at most $log_2(M)$ times in avoid of potential security risks. For example, 64 episodes are used to generate 6 different secret keys. The in-depth study on security guarantee and efficiency of channel reuse will be left to our future work due to the limited space.

H. Security Analysis

Given M RSS episodes involved in the key generation process, each corresponds to an integer index within $[1, M]$, and there are in total $M!$ possible permutations for the generated secret key, which imply that the encryption strength will be $M!$. For example, $M = 64$ has the encryption strength much larger a 256-bit key (i.e., $64! > 2^{256}$). If each RSS episode is encrypted as a $log_2 64 = 6$-bit long binaries, and each episode includes 7 samples, the bit generation rate will be $6/7 \approx 0.857 bit/sample(bps)$. Similarly, $M = 128$ has the encryption strength greater than a 512-bit key, and each episode is encoded as a $log_2 128 = 7$-bit long binaries. So the bit generation rate will be $7/7 = 1 bps$.



(a) BMR. (b) BGR.

Fig. 5. Secret key generation performance comparison between the DCT-based method (i.e., DCT-Quantization) and our methods (i.e., DP-SKG and DP-SKGr).

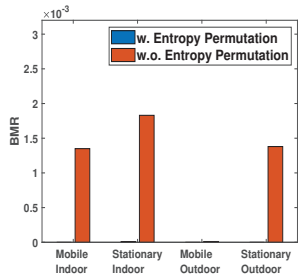


Fig. 6. Performance with and without the entropy-based permutation.

During the key generation process, the information exchange between Alice and Bob does not directly or indirectly disclose any information related to the secret key (i.e., permutation order) to the attackers. Moreover, the privacy amplification process further enhances the randomness of the secret key, which is validated by our experimental study V. Therefore, the secret key agreement process based on our proposed method is secure from external attackers. To crack the key, an attacker has to perform random guess on the secret bits, which is usually not affordable to many attackers.

V. Performance Evaluation

A. Experimental Setup

To validate the performance of the proposed method, we conduct experiments based on a public dataset built by G.Margelis et al. in their RSS-based key generation work [10]. The data are collected with an open-source operating system, Contiki, on IoT devices. Both indoor and outdoor environments with the devices in both stationary and dynamic positions are examined. Specifically, for the indoor environment (i.e., office space) CC2650 radio [25] operating at 2.4 GHz is used, while for the outdoor environment (i.e., close to a busy road next to University) the CC1310 radio [25] at 868 MHz is adopted. In both cases, three devices, acting as Alice, Bob, and Attacker, are employed, collecting at least 10000 RSS measurements, and their equipped radios operate at half-duplex mode without affecting the channel reciprocity.

B. Metrics

The experimental results are presented with the following metrics, and all results are the average value based on at least 1000 RSS sequences.

Bit Generation Rate (BGR): It is defined as the number of secure bits per channel measurement for key agreement.

Bit Mismatch Rate (BMR): It is the number of mismatched bits over the total number of generated secret bits. Note that the bit mismatch rate is measured before information reconciliation.

Randomness: It is used to evaluate the quality of keys. We measure the randomness of the keys with standard NIST test.

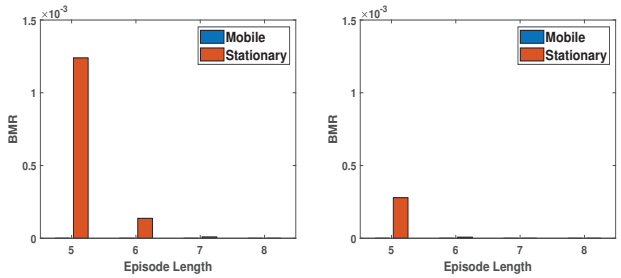
C. Performance Evaluation

Secret Key Generation Performance. We first evaluate the performance of our secret key generation method by comparing our method with the most recent quantization-based key generation method, DCT-based method [10] (i.e., DCT-Quantization), under different user motion status (i.e., mobile and stationary) and different environments (i.e., indoor and outdoor). As shown in Figure 5(a), given the fixed number (i.e., 64) of episodes involved for key generation and episode length (i.e., 7), DP-SKG achieves BMR of 0's except the scenario with stationary users in indoor environments (i.e., BMR = 0.0000096), while DCT-Quantization always has BMR over 0.001 for all scenarios. As indicated before, 64 RSS episodes in our method can achieve an encryption strength of over 256 bits. If the index of each episode is encoded as a 6-bit binary vector. As such, with this evaluation setup, DP-SKG's BGR can achieve $6/7 \approx 0.857$ bits per sample (bps). Overall we can clearly observe that DP-SKG consistently outperforms DCT-quantization method on BMR while has comparable BGR as DCT-Quantization in all scenarios.

We also evaluate the performance of our DP-SKG method with channel reuse (i.e., DP-SKGr) for the RSS sequence consisting of 64 episodes, which can be reused $log_2 64 = 6$ times for key generation. Similarly, we observe that DP-SKGr has BMRs close to 0 across different scenarios (e.g., BMR = 0.000033 for the scenario with stationary users in indoor environments), which are much lower than those of DCT-Quantization. We also observe that BGR of DP-SKGr goes as high as $5.14bps$. As the number of episodes increases, BGR will go even higher. Since DP-SKGr has a comparable BMR as DP-SKG, so we only present the BMR for DP-SKG in the later part.

Impact of Entropy-based Permutation. We examine the impact of the entropy-based permutation to DP-SKG by examining BMR of DP-SKG with and without the entropy-based permutation. As shown in Figure 6, the entropy-based permutation could increase the complexity of generated keys through disrupting the original RSS sequence order and decrease BMR from over 0.001 to almost 0 for the cases of stationary indoor, mobile indoor, and stationary outdoor. Due to fewer impacts of ambient noise in outdoor environments, BMR decreases from 0.0000112 to 0 with the entropy-based permutation. Overall, the aforementioned results demonstrate the effectiveness of the entropy-based permutation.

Impact of Episode Length. Next, we study how the RSS episode length affects key generation performance. Specifically, we fix the number of RSS episodes used for the key generation as 64 and vary the RSS episode length from 5 to 8. The RSS episode length affects the weight used for bipartite



(a) Indoor. (b) Outdoor.

Fig. 7. Performance under the impact of different episode lengths.

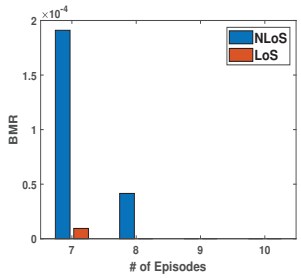


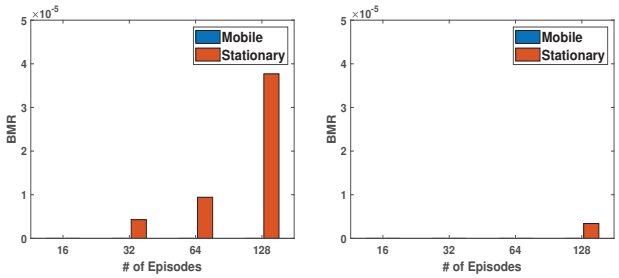
Fig. 8. Performance under the impact of LoS and NLoS channels.

graph matching. Short episodes increase the probability of forming the same weights when pairing other RSS episodes, leading to the ambiguity in the matching process. In contrast, long episodes decrease such likelihood and thereby the bit mismatch rate. The results in Figure 7 shows that BMR for stationary users decreases from 0.0012 and 0.00028 to 0 for both indoor and outdoor environments, respectively.

Impact of Line-of-Sight Channel. To further inspect the environmental impact, we also study the performance with Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) channels by the dataset provided by [10]. NLoS channel induces lower power levels at the receiver side, leading to lower SNR, so it is easy to be affected by ambient noise and thereby BMR. As shown in Figure 8, BMR of our method decreases from 0.00019 to 0 for NLoS channels when the episode length increases from 7 to 10, while BMR maintains close to 0 for LoS channels. This is because long episode provides more robust weight value for bipartite graph matching. The above observation shows that the impact of NLoS channels is suppressed as the episode length increases

Impact of the Number of RSS Episodes. Using more RSS episodes may increase the probability that more pairs of RSS episodes have the same weight during the matching process. We fix the RSS episode length as 7 and examine the bit mismatch rate by changing the number of episodes from 16 to 128, which are convenient for binary encoding. The results are presented in Figure 9. We find that BMRs for mobile users are always 0. We also find that BMRs for stationary users are close to zero, with small increases from 0 to 0.000038 and 0.000034 for indoor and outdoor environments. Therefore, to achieve an error-free key agreement, it is essential to choose a smaller number of RSS episodes to ensure a low BMR, considering an acceptable BGR.

Randomness Test with NIST. To ensure that the secret key generated is substantial random, the standard randomness test



(a) Indoor. (b) Outdoor.

Fig. 9. Performance under the impact of different key lengths.

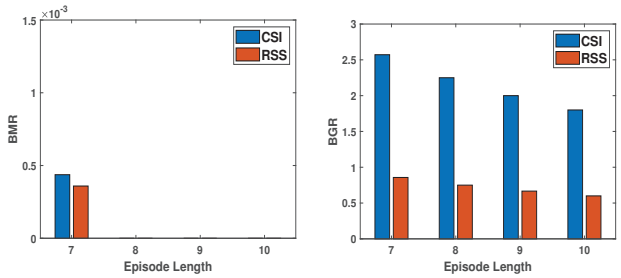
suite from NIST [26] is employed to verify the effectiveness of the secret bits extracted after secret key reconciliation and privacy amplification. We run 9 NIST tests on the secret keys generated from the data of four different scenarios, as listed in Table I. All the results pass the tests, indicating the randomness of the generated secret is sufficient for practical key generation.

Extension to CSI Measurements. DP-SKG can be easily extended to using CSI measurements for secret key generation. We conduct real experiments by employing two laptops equipped with COTS Intel 5300 wireless NIC, which operates in the 802.11n 2.4GHz channel. We collect RSS and CSI measurements in an office environment. We compare our method's performance of using CSI and RSS measurements to generate secrete keys in Figure 10. Due to the correlation between adjacent subcarriers of CSI measurements, we pick 3 subcarriers to construct CSI sequences for key generation. With a fixed episode length of 7, we vary the number of RSS episodes involved for key generation from 16 to 128. We find that the key generation with CSI measurements also has a low BMR and high BGR as using RSS measurements.

Time Cost. To study the complexity of DP-SKG, we evaluate the time cost for generating a secret key, which is dominated by the bipartite graph matching. We vary the number of RSS episodes from 16 to 128, which corresponding to the key length from 112 to 896 bits, and present the time cost in Figure 11. We find that the time cost of our method generating a secret key increases from 0.01 seconds to 1.7 seconds. Particularly, the time cost with respect to 64 episodes is around 250 $msec$. Intuitively, when more RSS episodes are involved in key generation, it will construct a larger bipartite graph for matching, resulting in higher time cost. The time cost significantly increases after the number of RSS episodes is changed to larger than 64.

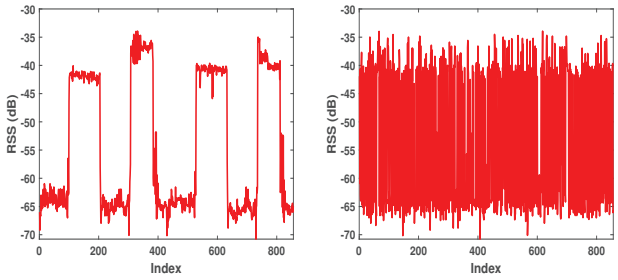
TABLE I
RANDOMNESS

Test	Static Indoor	Mobile Outdoor	Static Indoor	Mobile Outdoor
Freq.	0.451	0.970	0.705	0.878
Block Freq.	0.309	0.743	0.609	0.767
Cumsum (Fwd).	0.822	0.821	0.609	0.742
Cumsum (Rev).	0.515	0.787	0.749	0.599
Runs.	0.897	0.089	0.492	0.092
Longest Run of 1's.	0.055	0.349	0.569	0.776
Approx. Entropy.	0.999	0.999	0.999	1.000
FFT.	0.240	0.163	0.615	0.699
Serial.	0.766	0.357	0.014	0.645



(a) BMR. (b) BGR.

Fig. 10. Secret key generation performance with CSI measurements.



(a) Original RSS sequence. (b) Permuted RSS sequence.

Fig. 12. Copping with predictable channel attack.

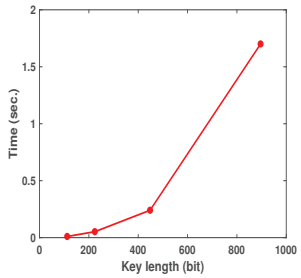


Fig. 11. Time cost for secret key generation.

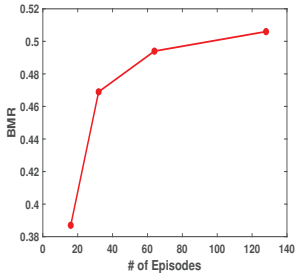


Fig. 13. BMR for stalking attacker.

D. Security Performance

Coping with Predictable Channel Attack. RSS measurements are usually dominated by LOS signals. If the attacker moves along a planned trajectory to block the LOS between Alice and Bob, the RSS measurements will become predictable when both Alice and Bob are stationary, making generated keys easy to predict. We evaluate our method under such attack by experimenting with two stationary laptops acting as Alice and Bob. A volunteer is asked to periodically block LOS of Alice and Bob for 60 seconds. As shown in Figure 12 (a), the RSS measurements observed by Alice have an obvious pattern corresponding to the blockage of LoS. Figure 12 (b) shows that after permutation, the RSS measurements do not have any observable pattern, indicating that our method does not reveal the original RSS sequence. Even if the attacker can infer the time periods when specific RSS measurements are possibly collected, it is still difficult to recover the RSS sequence with correct permutation order. Therefore, our proposed method can successfully defeat predictable channel attacks.

Coping with Stalking Attack. Intuitively, if an attacker follows either Alice or Bob's trajectory, the sequence of RSS measurements collected by the attacker would be similar to those obtained by Alice or Bob. However, such an attacker can not be very close to either Alice or Bob (at least half of the wavelength away), otherwise the attacker risks exposing himself. As such, the attacker observes a wireless channel independent from Alice and Bob's channel even though the attacker follows the user's trajectory. Accordingly, the attacker cannot obtain the same sequence of RSS measurements as Alice or Bob does. Furthermore, the permutation in our method disrupts the order of the collected RSS sequence, which amplifies the difference between the RSS sequence obtained by the attacker and the RSS sequence received by the user. We conduct an experiment by asking a volunteer to follow Alice from a short distance (i.e., around 30cm) and collect RSS measurements from Bob when Alice and Bob are exchanging probe packets. We try to use the RSS measurements collected by the attacker to generate secret keys and match them with the keys generated by Bob. As shown in Figure 13, BMR for the stalker increases to over 50% as the number of episodes increases to 64, indicating that the stalker's generated key is roughly a random guess.

VI. Conclusions

In this work, we propose an efficient dual-permutation secret key generation method, DP-SKG, which achieves the key agreement between two users by matching the randomly permuted channel measurements (e.g., RSS) sequences. Instead of using the error-prone quantization that is popularly used by conventional methods, DP-SKG formulates the secret key agreement as a bipartite graph matching problem and determines the secret key by minimizing the discrepancy between two permuted RSS sequences in a holistic way. The generated secret key is derived based on permutation order with bit mismatch rates as low as zero. Furthermore, new information reconciliation and channel reuse mechanisms are integrated to ensure error-free key agreement with boosted bit generation rate and high randomness. Security analysis and performance evaluation based on extensive experiments under different scenarios demonstrate the effectiveness and efficiency of DP-SKG. Results show that it takes around 250 $msec$ to achieve over 256-bit-equivalent cryptographic strength.

VII. Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 62020106013, 61802051, and 61772121, Sichuan Science and Technology Program under Grants 2020JDTD0007 and 2020YFG0298.

REFERENCES

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 6th ed. USA: Prentice Hall Press, 2013.

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

[3] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *ACM MobiCom*, 2008.

[4] N. Patwari, J. Croft, S. Jana, and S. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Transactions on Mobile Computing*, 2009.

[5] K. Zeng and et. al., "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *IEEE INFOCOM*, 2010.

[6] H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response," in *Proceedings IEEE INFOCOM*, 2013.

[7] W. Xi, C. Qian, J. Han, K. Zhao, S. Zhong, X.-Y. Li, and J. Zhao, "Instant and robust authentication and key agreement among mobile devices," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[8] X. Zhu, F. Xu, E. Novak, C. C. Tan, Q. Li, and G. Chen, "Using wireless link dynamics to extract a secret key in vehicular scenarios," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 2065–2078, 2017.

[9] Z. Li, Q. Pei, I. Markwood, Y. Liu, and H. Zhu, "Secret key establishment via rss trajectory matching between wearable devices," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 802–817, 2018.

[10] G. Margelis, X. Fafoutis, G. Oikonomou, R. Piechocki, T. Tryfonas, and P. Thomas, "Efficient dct-based secret key generation for the internet of things," *Ad Hoc Networks*, vol. 92, p. 101744, 2019.

[11] S. Jana and et. al., "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *ACM MobiCom*, 2009.

[12] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *IEEE ICASSP*, 2008.

[13] Q. Wang, H. Su, K. Ren, and K. Kim, "Fast and scalable secret key generation exploiting channel phase randomness in wireless networks," in *IEEE INFOCOM*, 2011.

[14] R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Transactions on Information Forensics and Security*, 2007.

[15] M. Tope and J. McEachen, "Unconditionally secure communications over fading channels," in *IEEE MILCOM*, 2001.

[16] B. Azimi-Sadjadi and et. al., "Robust key generation from signal envelopes in wireless networks," in *ACM CCS*, 2007.

[17] Y. Liu, S. C. Draper, and A. M. Sayeed, "Exploiting channel diversity in secret key generation from multipath fading randomness," *IEEE Transactions on Information Forensics and Security*, 2012.

[18] Wei Xi, Xiang-Yang Li, Chen Qian, Jinsong Han, Shaojie Tang, Jizhong Zhao, and Kun Zhao, "Keep: Fast secret key extraction protocol for d2d communication," in *IEEE 22nd International Symposium of Quality of Service (IWQoS)*, 2014.

[19] W. Xi, M. Duan, X. Bai, K. Zhao, L. Mo, and J. Zhao, "Keep: Secure and efficient communication for distributed iot devices," *IEEE Internet of Things Journal*, 2020.

[20] J. Zhang, M. Ding, D. Lopez-Perez, A. Marshall, and L. Hanzo, "Design of an efficient ofdma-based multi-user key generation protocol," *IEEE Transactions on Vehicular Technology*, 2019.

[21] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *IEEE INFOCOM*, 2012.

[22] C. Hansen, Q. Wei, J.-S. Shieh, P. Fourcade, B. Isableu, and L. Majed, "Sample entropy, univariate, and multivariate multi-scale entropy in comparison with classical postural sway parameters in young healthy adults," *Frontiers in human neuroscience*, vol. 11, p. 206, 2017.

[23] A. Schrijver, "Combinatorial optimization: Polyhedra and efficiency," *Discrete Applied Mathematics*, vol. 146, pp. 120–122, 2005.

[24] B. G. Staff, L. Books, and B. Group, *Permutations: Shuffling, Permutation, Parity of a Permutation, Transposition Cipher, Cayley's Theorem, Stirling Number, Landau's Function*. General Books, 2010.

[25] "Texas Instruments CC13x0, CC26x0 SimpleLink Wireless MCU," Technical Reference Manual at https://www.tij.co.jp/lit/ug/swcu117i/swcu117i.pdf, 2020.

[26] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Booz-allen and hamilton inc mclean va, Tech. Rep., 2001.