Physical Layer Secret Key Generation in Static Environments

Nasser Aldaghri, *Student Member, IEEE*, and Hessam Mahdavifar, *Member, IEEE*

***Abstract*—Two legitimate parties, referred to as Alice and Bob, wish to generate secret keys from the wireless channel in the presence of an eavesdropper, referred to as Eve, in order to use such keys for encryption and decryption. In general, the secret key rate highly depends on the coherence time of the channel. In particular, a straightforward method of generating secret keys in static environments results in ultra-low rates. In order to resolve this problem, we introduce a low-complexity method called *induced randomness*. In this method, Alice and Bob independently generate local randomness to be used together with the uniqueness of the wireless channel coefficients in order to enable high-rate secret key generation. In this work, two scenarios are considered: first, when Alice and Bob share a direct communication channel, and second, when Alice and Bob do not have a direct link and communicate through an untrusted relay. After exchanging the induced randomness, post-processing is done by Alice and Bob to generate highly-correlated samples that are used for the key generation. Such samples are then converted into bits, disparities between the sequences generated by Alice and Bob are mitigated, and the resulting sequences are then hashed to compensate for the information leakage to the eavesdropper and to allow consistency checking of the generated key bit sequences. We utilize semantic security measures and information-theoretic inequalities to upper bound the probability of successful eavesdropping attack in terms of the mutual information measures that can be numerically computed. Given certain reasonable system parameters this bound is numerically evaluated to be 2^{-31} and $2^{-10.57}$ in the first and the second scenario, respectively.**

***Index Terms*—Information theoretic security, physical layer security, distributed wireless systems, secret key generation, semantic security, static environments.**

I. Introduction

Wireless networks are becoming increasingly distributed in future systems, e.g., the fifth generation of wireless networks (5G) and the Internet of Things (IoT), which, consequently, poses a higher risk of malicious attacks against message confidentiality in these systems. In general, communication devices secure messages using either symmetric-key encryption schemes such as Advanced Encryption Standard (AES) [2], or asymmetric-key encryption schemes such as Rivest–Shamir–Adleman (RSA) [3]. Asymmetric-key schemes are not preferred for devices with limited resources, e.g., as in IoT networks, due to their complex mathematical operations. Instead, symmetric-key cryptographic schemes are desired in IoT networks due to their low-complexity implementations [4]. Such schemes require the secret keys for the encryption and decryption to be distributed beforehand between the legitimate parties. To complement the symmetric-key cryptographic schemes, physical layer security methods can be deployed to exchange secret keys between the nodes in order to be used in the encryption and the decryption algorithms [5].

The fundamental works of [6], [7] established an information-theoretic framework to study the use of common randomness for secret key generation. In practice, characteristics of wireless links are shown to provide a great source for the common randomness to be used for secret key generation, which have recently received significant attention [8], [9]. More specifically, the wireless channel has two main features that are essential for secret key generation, namely, reciprocity and randomness. The wireless channel is reciprocal over each single coherence time interval [10], and it has inherent randomness due to the variation of the channel coefficients between different coherence time slots [8]. Note that the former requires an underlying synchronization mechanism while the latter assumes a dynamic environment. These features are often assumed to be available to the wireless nodes, i.e., the legitimate parties, which can then be utilized in low-complexity secret key generation protocols at the physical layer. The setup for the key generation protocols is as follows: the legitimate parties Alice and Bob share a common wireless channel, either directly or indirectly through a relay node. They communicate through this channel with the goal of generating a common secret key bit sequence, while keeping a passive eavesdropper Eve oblivious about the generated key. Such protocols often include the following steps [9]:

1) Randomness sharing: In this step, the legitimate parties observe correlated samples from a common source of randomness, e.g., wireless channel coefficients.
2) Quantization: This is the process of converting such correlated samples, which are often real-valued, into binary bits.
3) Reconciliation: In general, there is a mismatch between the binary sequences observed and quantized by Alice and Bob. Reconciliation is the process of mitigating such mismatch between Alice's and Bob's bit sequences using methods such as cosets of binary linear codes.
4) Privacy amplification: This is the process of compensating for the information leakage to the eavesdropper Eve during the aforementioned steps.

A. Related Work

This section provides an overview of related work on secret key generation using characteristics of wireless channel under two main scenarios; the first scenario where the legitimate parties Alice and Bob have a direct communication channel as the

The material in this paper was presented in part at the IEEE Global Communications Conference in December 2018 [1]. This work was supported in part by the National Science Foundation under grants CCF–1763348, and CCF–1909771.

N. Aldaghri and H. Mahdavifar are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 (email: aldaghri@umich.edu and hessam@umich.edu).

only means of communication, and the second scenario where their communication is helped by a relay node.

1) Secret key generation over direct communication channels: In this case, different characteristics of the wireless channel can be utilized as the source of common randomness in secret key generation protocols. This includes the channel state information (CSI), the received signal strength (RSS), and the channel phase, just to name a few [11]. As mentioned earlier, there are two main underlying assumptions in such protocols. First, the assumption on reciprocity of the wireless channel guarantees the reliability of such protocols. Second, the randomness of the key is guaranteed by the assumptions on temporal decorrelation [12]. The resulting secret key generation protocols, e.g., [8], [10], [13]–[17], where orthogonal frequency division multiplexing (OFDM) is utilized to increase the key rate in [17], often require dynamic environments in order to satisfy the second assumption and to enable secret key generation at non-zero rates. It is worth noting that some imperfections of the channel measurements may occur due to mismatched hardware and synchronization errors [18].

Wireless channels can be naturally assumed to be dynamic assuming a certain level of mobility by users and/or in the surrounding environment. However, such assumptions do not hold in static environments such as indoor IoT networks. Consequently, the aforementioned protocols result in ultra-low/zero secret key rates in such environments. This issue has been studied in the literature and various solutions have been proposed. Solutions include utilizing multiple-input-multiple-output (MIMO) antennas systems [19]–[21], beamforming [22], deploying friendly jamming [23] where the users act as jammers to confuse the eavesdropper, and using artificial noise to confuse the eavesdropper [24]. In another line of work, some user-introduced randomness is utilized for various purposes [18], [25]–[28]. For instance, the user's randomness is used to counter certain types of attacks by the eavesdropper in [18]. However, the use of induced randomness for key generation, and more specifically to increase the key rate in static scenarios, is not discussed in [18]. In general, prior schemes that use some user-introduced randomness require complex underlying architectures, e.g., MIMO transceivers, or unconstrained sources of randomness, i.e., continuous sources whose Shannon entropy is infinity, which are expensive to implement [29]. This, in turn, makes them unappealing for applications where nodes experience a static environment and have limited resources, e.g., IoT networks, sensor networks, etc. Also, solutions based on utilizing coupled dynamics existing in synchronization mechanisms [30], [31] and based on full-duplex communications [32] are proposed for low-complexity IoT networks, which are often limited to very-short-range communications due to power constraints for implementing coupled dynamics in practice. Moreover, several prior works have considered utilizing relays for secret key generation, as discussed next.

2) Secret key generation with the help of a relay: In this case, there exists a relay node that assists Alice and Bob to generate the shared secret keys. The wireless characteristics used for the randomness sharing in the first scenario, e.g., CSI and RSSI, can be similarly applicable here. Various methods have been proposed in the literature to utilize relays in order to improve the key generation rate when Alice and Bob have a direct communication link as well [33], [34]. The use of relays in generating secret keys when Alice and Bob do not have a direct communication link is studied in [35]–[37]. A major arguable assumption in these related works is that the relay nodes are trusted. However, the wireless nodes, especially when they are considered low-complex and low-cost as in IoT networks, are susceptible to hacking, even after the key generation process is done. Hence, it is highly desirable to ensure that limited information about the generated secret key is leaked to the relay throughout the process. This is the motivation behind several other related works which assumed the relay nodes are untrusted. For instance, a method to accommodate this case by utilizing friendly jamming is introduced in [38]. Another method that requires a moving relay to generate secret bits is proposed in [39]. Also, a novel method to resolve the issue of untrusted relays using a MIMO architecture is suggested in [40]. Such methods, however, require dynamic environments. As mentioned before, these protocols are not appealing for applications where nodes are resource-constrained and the environment is static.

B. Our Contributions

Our main contribution in this work is a solution, based on low-complexity methods, for resolving the issue of low/zero rate secret key generation between two legitimate nodes in static environments. In the proposed solution, we utilize induced randomness generated by the legitimate parties and exchanged between them. More specifically, Alice and Bob independently generate a certain number of random bits. Then, they map these bits to quadrature amplitude modulation (QAM) symbols which they exchange using the direct communication channel (the first considered scenario) or through an untrusted relay (the second considered scenario). After the exchange of the generated randomness, Alice and Bob process their received sequences, which are the generated randomness by the other party and passed through the channel, using their own random sequences. The reciprocity of the channel/channels ensures that they obtain highly correlated sequences. Such common noisy randomness is then used to extract shared secret keys by following quantization, reconciliation, and privacy amplification steps. The reliability and the security of the proposed protocols are analyzed by upper bounding the probability of falsely accepting a mismatched secret key and the probability of a successful eavesdropping attack by Eve, respectively. While most of prior works on designing physical layer secret key generation protocols rely on spatial decorrelation assumptions to guarantees the security of the key, we provide, to the best of our knowledge, the first rigorous result on upper bounding the probability of successful eavesdropping attack in such protocols. It is worth noting that although the motivation behind the design of the proposed protocols is to resolve the issue of environment immobility, they work in dynamic environments as well assuming that the wireless channel does not change during each session of randomness exchanges.

The proposed protocols are considered under two major scenarios. In the first scenario, secret key generation over a direct communication channel is considered, which was presented in part in [1]. In the second scenario, secret key generation with the help of an untrusted relay is considered assuming that there is no direct communication link between Alice and Bob. In the proposed protocols, a communication scheme based on OFDM is assumed to increase the secret key rate as in [17]. Furthermore, we utilize secure sketch [41] and universal hash functions (UHF) [42] to ensure reliability and security of the generated keys while enhancing the randomness of the key bit sequences. Numerical results are provided for the proposed protocol assuming reasonable parameters in the communication setup. These parameters include the modulation order, the number of OFDM subcarriers, the signal-to-noise ratio (SNR), and the quantization resolution. Then, various fundamental metrics are characterized including the bit generation rate (BGR), the bit mismatch rate (BMR), the bit error rate (BER), and the randomness of the key generated using the National Institute of Science and Technology (NIST) randomness tests [43]. In addition, a setup in which realistic channel coefficients for 5G millimeter wave (mmWave) channels are generated by the NYUSIM Channel Simulator [44] is considered, assuming the first scenario, in order to evaluate the protocol in a realistic environment. Furthermore, we introduce a new efficiency measure for protocols that utilize induced randomness. This parameter, called *randomness efficiency*, measures what percentage of the induced randomness is utilized in the generated common random sequence. The randomness efficiency in the first scenario is 50 %, while it is 33 % in the second scenario.

The rest of this paper is organized as follows. The system models for the two considered scenarios are discussed in Section II. In Section III, the proposed protocols for generating secret keys are discussed. In Section IV, the security of the proposed protocols is analyzed. Numerical results are provided in Section V. Finally, the paper is concluded in Section VI.

II. System Model

Secret key generation protocols consist of two legitimate parties Alice and Bob who aim to generate a common, random, and secure bit sequence using an authenticated shared wireless channel between them. In addition to Alice and Bob, there is an authenticated relay node named Carol, who is honest but curious, and is able to help Alice and Bob generate such keys by relaying their signals when no direct channel exists between them. As the legitimate parties execute the secret key generation protocol, a passive eavesdropper Eve is observing all communications between Alice, Bob, and Carol, and tries to learn as much information as possible about the secret key being generated and shared between Alice and Bob.

A. Direct Secret Key Generation

The channel between Alice and Bob is assumed to be an authenticated wireless channel, but it is not secure. The eavesdropper Eve is assumed to be a passive eavesdropper. The setup of the considered secret key generation (SKG) system is shown in Figure 1. The wireless channel considered in this work is assumed to be a fading channel. Suppose that Alice transmits a signal $x_{\text{Alice}}(t)$ to Bob, he receives

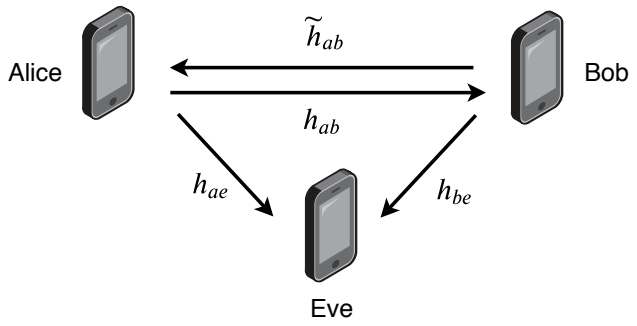

Figure 1. System model for direct secret key generation.

$$y_{\text{Bob}}(t) = x_{\text{Alice}}(t) \circledast h_{ab}(t) + n_b(t), \tag{1}$$

where t denotes the time, $\circledast$ denotes the convolution operator, $h_{ab}(t)$ denotes the circularly-symmetric Gaussian-distributed channel response with mean 0 and variance $\sigma_h^2/2$ in each dimension, and $n_b(t)$ denotes the circularly-symmetric Gaussian-distributed additive noise component with mean 0 and variance $\sigma_n^2/2$ in each dimension. In the case of flat fading channels, the convolution converts to multiplication and the channel response is the Rayleigh-distributed fading gain coefficient with parameter σ, i.e., $|h_{ab}| \sim \text{Rayleigh}(\sigma)$, and the phase is uniformly distributed, i.e., $\phi(h_{ab}) \sim U[-\pi, \pi]$. The same applies when Bob transmits $x_{\text{Bob}}(t)$ to Alice, she receives

$$y_{\text{Alice}}(t) = x_{\text{Bob}}(t) \circledast \widetilde{h}_{ab}(t) + n_a(t). \tag{2}$$

The distribution of the channel coefficients h_{ab} will be slightly different in Section V-A2 for the numerical evaluation of the protocol assuming realistic 5G mmWave coefficients. More specifically, samples of Rayleigh distribution are replaced with realistic 5G mmWave channel coefficients considered in [44].

Wireless channels are essentially reciprocal [8], meaning that the CSI observed at Bob's end from Alice is the same as Alice's end from Bob assuming an underlying synchronization mechanism. The reciprocity property, i.e., $h_{ab} \approx \widetilde{h}_{ab}$, is the key to most of the secret key generation protocols that utilize characteristics of the physical layer channel. Also, Alice and Bob are assumed to use OFDM. Suppose that Alice and Bob transmit the j-th element of the vectors $\mathbf{x}_{\text{Alice}}(t)$ and $\mathbf{x}_{\text{Bob}}(t)$, respectively, over the j-th OFDM subcarrier. The received signals are expressed as follows:

$$\mathbf{y}_{\text{Alice}}(t) = \mathbf{x}_{\text{Bob}}(t) \circ \widetilde{\mathbf{h}}_{ab}(t) + \mathbf{n}_a(t), \tag{3}$$

$$\mathbf{y}_{\text{Bob}}(t) = \mathbf{x}_{\text{Alice}}(t) \circ \mathbf{h}_{ab}(t) + \mathbf{n}_b(t), \tag{4}$$

where $\circ$ denotes the Hadamard product, i.e., the element-wise product. By using the received signals at Alice and Bob together with the uniqueness of wireless channel coefficients between them they aim at extracting a shared secret key. Note that in addition to the wireless channel, Alice and Bob are assumed to share a noiseless public channel that Eve has access to. This

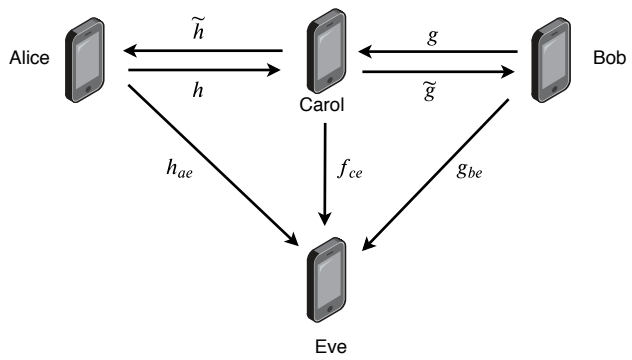

Figure 2. System model for relay-based secret key generation.

channel can be realized by using appropriate off-the-shelf modulation and channel coding schemes.

B. Secret Key Generation Using a Relay

In this case, Alice and Bob do not have access to a direct wireless channel. Instead, there is an intermediate party, also referred to as Carol, operating as a relay node with whom Alice and Bob share authenticated wireless channels which are not secure. The relay is considered to be amplify-and-forward and is assumed to be compliant with the protocol, i.e., it amplifies and forwards the signals without tampering with their contents. However, the relay is considered to be untrusted. This is because it might be susceptible to hacking attacks by an adversary or might be simply curious to learn the contents communicated between Alice and Bob. The eavesdropper Eve is considered to be a passive eavesdropper. The system model is shown in Figure 2. Similar to the model discussed in Section II-A, the channel between each two entities is modeled as a wireless fading channel. Alice, Bob, and Carol utilize OFDM in their transmissions. Alice wishes to transmit a signal $\mathbf{x}_{\text{Alice}}(t)$ to Bob through the relay node Carol. First, Alice transmit $\mathbf{x}_{\text{Alice}}(t)$ to Carol, who receives

$$\mathbf{y}_{\text{Carol}}(t) = \mathbf{x}_{\text{Alice}}(t) \circ \mathbf{h}(t) + \mathbf{n}_r(t). \tag{5}$$

Next, the relay amplifies the signal using amplification factor α and forwards the amplified signal to Bob. Bob receives

$$\begin{aligned}\mathbf{y}_{\text{Bob}}(t) &= \alpha \circ \mathbf{y}_{\text{Carol}}(t) \circ \widetilde{\mathbf{g}}(t) + \mathbf{n}_b(t) &(6)\\ &= \alpha \circ (\mathbf{x}_{\text{Alice}}(t) \circ \mathbf{h}(t) + \mathbf{n}_r(t)) \circ \widetilde{\mathbf{g}}(t) + \mathbf{n}_b(t), &(7)\end{aligned}$$

which holds due to the use of OFDM in transmissions between Alice, Carol, and Bob. The same applies when Bob transmits $\mathbf{x}_{\text{Bob}}(t)$ to Alice through Carol. Alice receives

$$\mathbf{y}_{\text{Alice}}(t) = \alpha \circ (\mathbf{x}_{\text{Bob}}(t) \circ \mathbf{g}(t) + \mathbf{n}_r(t)) \circ \widetilde{\mathbf{h}}(t) + \mathbf{n}_a(t), \tag{8}$$

where the j-th elements of the vectors $\mathbf{h}(t)$, $\widetilde{\mathbf{h}}(t)$ are both circularly-symmetric Gaussian-distributed with mean 0 and dimension-variance $\sigma_h^2/2$, and $\mathbf{g}(t)$, $\widetilde{\mathbf{g}}(t)$ are also circularly-symmetric Gaussian-distributed with mean 0 and dimension-variance $\sigma_g^2/2$. On the other hand, the j-th elements of the vectors $\mathbf{n}_r(t)$, $\mathbf{n}_a(t)$, and $\mathbf{n}_b(t)$ are independent and circularly-symmetric Gaussian-distributed with mean 0 and dimension-variance $\sigma_{n_R}^2/2$, $\sigma_{n_A}^2/2$, and $\sigma_{n_B}^2/2$, respectively.

Finally, Alice and Bob use their received signals and utilize the uniqueness of the wireless channel coefficients between them and Carol to extract a secret key. As in the direct secret key generation scenario, a noiseless public channel is available between Alice and Bob through Carol which Eve has access to. Such a channel can be realized by using appropriate off-the-shelf modulation and coding schemes from Alice to Carol, from Carol to Bob, and vice versa.

Remark: The relay employs an amplify-and-forward (AF) function with amplification factor α. The relay node is placed such that α can be selected according to a certain desired criterion such as maintaining the average transmitted power at the relay, or maintaining the average SNR at the receiver, see, e.g., [45] for a detailed discussion. To implement the protocols proposed in this paper, a similar criterion can be adopted since the aim is to create highly correlated sequences which depends on the average received SNR at Alice and Bob. Also, for simplicity it is assumed that the amplification factor is the same for the transmissions to Alice and Bob; however, it can be different for each of them to achieve some specific metric such as the received SNR. The reciprocity property of the indirect channel between Alice and Bob holds in this scenario, since the reciprocity of the individual channels between Alice and Carol, and Carol and Bob still holds.

C. Evaluation Metrics for SKG Protocols

Metrics that are often used to evaluate the performance of secret key generation protocols are as follows [14]:

1) Bit Generation Rate (BGR): This measures the number of bits per packet in the quantized sequences generated by Alice and Bob, denoted by $\mathbf{q}_a$ and $\mathbf{q}_b$, respectively.
2) Bit Mismatch Rate (BMR): This measures the ratio of the number of bits that are mismatched between $\mathbf{q}_a$ and $\mathbf{q}_b$. This quantity can be also measured at Eve's side. Note that the BMR at Eve should be higher than the BMR measured between Alice and Bob; otherwise, no secret key can be generated.
3) Bit Error Rate (BER): This measures the ratio of the number of bits that do not match in the final key generated by Alice and Bob as the output of the protocol. This quantity can be also measured at Eve's side, which, ideally, should be close to 50%.
4) Randomness: This indicates whether the final key bit sequence generated by the protocol, denoted by $\mathbf{K}_{ab}$, is indistinguishable from a random binary bit sequence. This is often tested using the NIST statistical test suite [43].

In addition to the aforementioned metrics, we introduce a new parameter, referred to as *randomness efficiency*, to measure the length of the shared sequence normalized by the total amount of randomness available to Alice and Bob. Let R_Q denote the total number of shared random bits after quantization. The randomness efficiency, denoted by E_R, is defined as

$$E_R \stackrel{\text{def}}{=} \frac{R_Q}{H(S) + H(V)}, \tag{9}$$

where $H(S)$ and $H(V)$ are the entropy of Alice's and Bob's sources of randomness, respectively.

Table I
NOTATION SUMMARY FOR THE i-TH SKG SESSION

Symbol	Description
$\mathbf{p}$	Known probing vector
$\mathbf{s}_i$	Alice's local randomness
$\mathbf{v}_i$	Bob's local randomness
$\mathbf{h}_{i,ab}$	Channel coefficients from Alice and Bob
$\widetilde{\mathbf{h}}_{i,ab}$	Channel coefficients from Bob to Alice
$\mathbf{h}_i$	Channel coefficients between Alice and the relay
$\widetilde{\mathbf{h}}_i$	Channel coefficients between the relay and Alice
$\mathbf{g}_i$	Channel coefficients between Bob and the relay
$\widetilde{\mathbf{g}}_i$	Channel coefficients between the relay and Bob
$\mathbf{h}_{i,ae}$	Channel coefficients between Alice and Eve
$\mathbf{h}_{i,be}$	Channel coefficients between Bob and Eve
$\mathbf{w}_{i,ab}$	Alice's samples used for quantization
$\widetilde{\mathbf{w}}_{i,ab}$	Bob's samples used for quantization
$\mathbf{q}_{i,a}$	Alice's quantized version of $\mathbf{w}_{i,ab}$
$\mathbf{q}_{i,b}$	Bob's quantized version of $\widetilde{\mathbf{w}}_{i,ab}$
$\mathbf{K}_{i,ab}$	Alice's key bits
$\widetilde{\mathbf{K}}_{i,ab}$	Bob's key bits
$\mathbf{C}_{i,ab}$	Alice's check sequence bits
$\widetilde{\mathbf{C}}_{i,ab}$	Bob's check sequence bits

III. PROPOSED PROTOCOLS

The proposed protocols for both scenarios, i.e., secret key generation using a direct channel and relay-based secret key generation, can be partitioned into four stages: induced randomness exchange, quantization, reconciliation, and privacy amplification together with consistency checking. The first stage, i.e., induced randomness exchange, is done differently in the two considered scenarios, while the remaining stages are similar.

In the first stage, the randomness is induced by Alice and Bob at each of the N OFDM subcarriers, provided that each two-way exchange is done within the same coherence time interval. After the exchange of induced randomness, Alice and Bob process what they receive by performing quantization followed by reconciliation to correct the disparities between their bit sequences. As a result, they obtain, with high probability, identical bit sequences. Then, they use privacy amplification to improve the security of the generated bit sequences. Finally, they check whether their keys are consistent or not. If the keys are not consistent, they re-initiate a new session. The notations for various vectors in the protocol are summarized in Table I. Also, Figure 3 shows an overview of a single session of the key generation protocol for the scenario involving a direct channel, and Figure 4 shows a single session of the relay-based secret key generation protocol. For ease of notation, we remove the time index t from the functions while keeping in mind that the exchanges are done within the same coherence time. Next, detailed descriptions of various stages of the proposed protocols are discussed.

A. Induced Randomness Exchange

In this stage we aim at creating highly correlated yet random observations at Alice and Bob. We discuss this stage separately for the two considered scenarios as follows:

1) Direct Induced Randomness Exchange: In this stage, Alice and Bob exchange randomly generated symbols with each other. In the i-th session, Alice chooses a vector $\mathbf{s}_i$ of length N and Bob also chooses a vector $\mathbf{v}_i$ of length N. Each element of the vectors $\mathbf{s}_i$ and $\mathbf{v}_i$ is chosen independently and uniformly at random from a set of M symbols in a M-QAM constellation. Then, the symbols are multiplied by a pulse/carrier signal for transmission. The reason behind choosing the symbols from M-QAM constellation is that the hardware for transmitting and receiving QAM symbols is readily available in many wireless devices. After the exchange of random symbols, Alice and Bob multiply what they sent with what they received. This results in random sequences $\mathbf{w}_{i,ab}$ and $\widetilde{\mathbf{w}}_{i,ab}$ available at Alice and Bob, respectively, as follows:

$$\mathbf{w}_{i,ab} = \mathbf{s}_i \circ \mathbf{v}_i \circ \widetilde{\mathbf{h}}_{i,ab} + \mathbf{s}_i \circ \mathbf{n}_{i,a}, \tag{10}$$

$$\widetilde{\mathbf{w}}_{i,ab} = \mathbf{s}_i \circ \mathbf{v}_i \circ \mathbf{h}_{i,ab} + \mathbf{v}_i \circ \mathbf{n}_{i,b}. \tag{11}$$

These two vectors are random and highly correlated, as will be shown, which makes them suitable for extracting shared secret keys between Alice and Bob.

2) Relay-Based Induced Randomness Exchange: First, the relay transmits a known probing vector $\mathbf{p}$ to Alice and Bob, who receive $\mathbf{y}_{i,1,a}$ and $\mathbf{y}_{i,1,b}$, respectively, specified as follows:

$$\mathbf{y}_{i,1,a} = \mathbf{p} \circ \widetilde{\mathbf{h}}_i + \mathbf{n}_{i,1,a}, \tag{12}$$

$$\mathbf{y}_{i,1,b} = \mathbf{p} \circ \widetilde{\mathbf{g}}_i + \mathbf{n}_{i,1,b}. \tag{13}$$

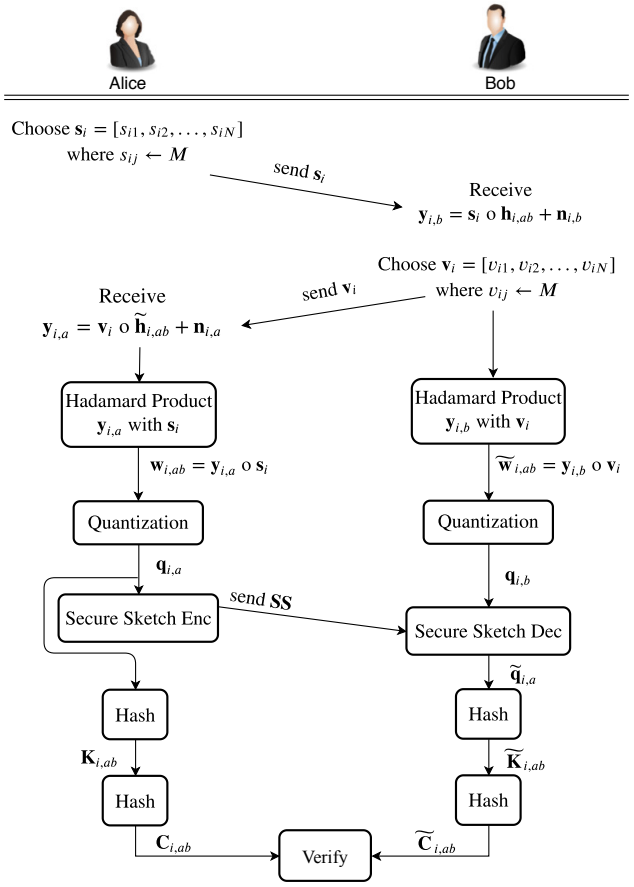

Figure 3. Direct secret key generation protocol overview of a single session.

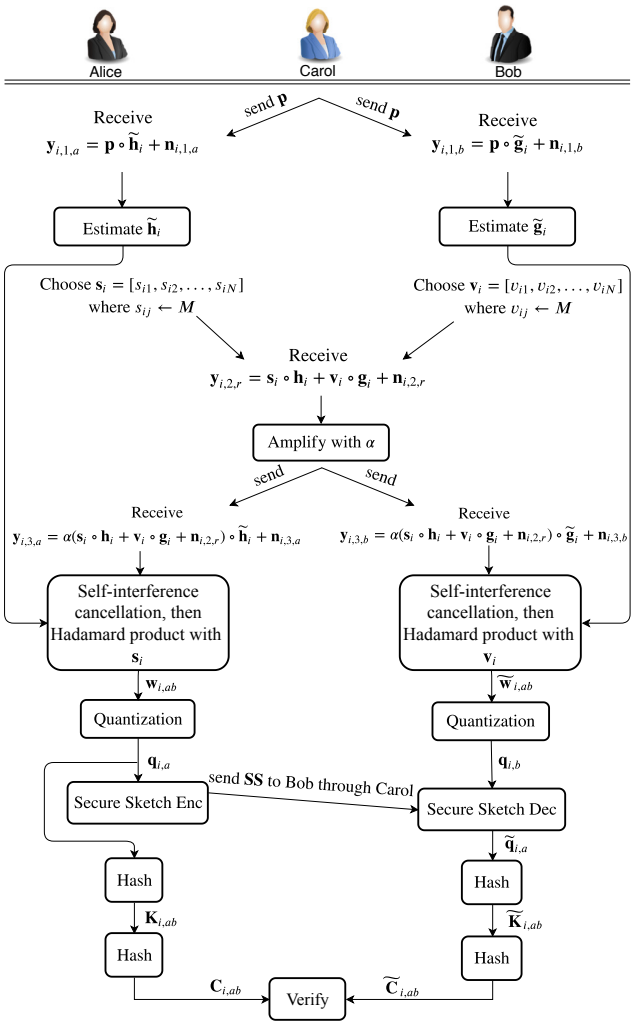

Figure 4. Relay-based secret key generation protocol overview of a single session.

Alice and Bob then estimate the channels between themselves and the relay, i.e., $\widetilde{\mathbf{h}}_i$ and $\widetilde{\mathbf{g}}_i$, respectively, using their observations. Their estimates are denoted by $\widehat{\mathbf{h}}_i$ and $\widehat{\mathbf{g}}_i$ with estimation errors defined as $\mathbf{z}_{i,a} = (\mathbf{h}_i \circ \widetilde{\mathbf{h}}_i - \widehat{\mathbf{h}}_i^{\circ 2})$ and $\mathbf{z}_{i,b} = (\mathbf{g}_i \circ \widetilde{\mathbf{g}}_i - \widehat{\mathbf{g}}_i^{\circ 2})$, respectively, where $(.)^{\circ 2}$ denotes the element-wise square operation. Alice and Bob utilize their respective channel estimates together with their respective local randomness to eliminate the self-interference terms and to generate the correlated samples, to be described next.

Alice and Bob generate, independently and uniformly at random, vectors of length N consisting of M-QAM symbols. Let $\mathbf{s}_i$ and $\mathbf{v}_i$ denote Alice's and Bob's vectors, respectively. They use the probing vector $\mathbf{p}$ also for synchronization and, simultaneously, transmit their vectors to the relay in such a way that the received SNRs at the relay with respect to the received sequences from Alice and Bob are the same, and equal to a predetermined value. The relay receives

$$\mathbf{y}_{i,2,r} = \mathbf{s}_i \circ \mathbf{h}_i + \mathbf{v}_i \circ \mathbf{g}_i + \mathbf{n}_{i,2,r}. \tag{14}$$

Then, it amplifies $\mathbf{y}_{i,2,r}$ with an amplification factor α, which is chosen to meet a specific SNR at Alice and Bob, and forwards

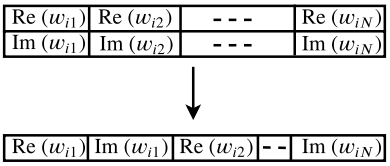

Figure 5. Sorting $\mathbf{w}_{i,ab}$ values before feeding them to the quantizer.

the amplified signal to Alice and Bob who receive $\mathbf{y}_{i,3,a}$ and $\mathbf{y}_{i,3,b}$, respectively, as follows:

$$\mathbf{y}_{i,3,a} = \alpha(\mathbf{s}_i \circ \mathbf{h}_i + \mathbf{v}_i \circ \mathbf{g}_i + \mathbf{n}_{i,2,r}) \circ \widetilde{\mathbf{h}}_i + \mathbf{n}_{i,3,a}, \tag{15}$$
$$\mathbf{y}_{i,3,b} = \alpha(\mathbf{s}_i \circ \mathbf{h}_i + \mathbf{v}_i \circ \mathbf{g}_i + \mathbf{n}_{i,2,r}) \circ \widetilde{\mathbf{g}}_i + \mathbf{n}_{i,3,b}. \tag{16}$$

The value of the amplification factor α is assumed to be publicly known. Alice and Bob utilize what they receive from the relay together with their locally generated vectors, their channel estimates, and α in order to construct highly correlated samples. More specifically, the self-interference terms $\alpha \mathbf{s}_i \circ \mathbf{h}_i \circ \widetilde{\mathbf{h}}_i$ and $\alpha \mathbf{v}_i \circ \mathbf{g}_i \circ \widetilde{\mathbf{g}}_i$ are cancelled at Alice and Bob, respectively, using their local randomness and the channel estimates. The results are normalized by α and then multiplied by the local randomness, which results in $\mathbf{w}_{i,ab}$ and $\widetilde{\mathbf{w}}_{i,ab}$ at Alice and Bob, respectively, as follows:

$$\mathbf{w}_{i,ab} = \mathbf{s}_i \circ \mathbf{v}_i \circ \mathbf{g}_i \circ \widetilde{\mathbf{h}}_i + \widehat{\mathbf{n}}_{i,3,a}, \tag{17}$$
$$\widetilde{\mathbf{w}}_{i,ab} = \mathbf{s}_i \circ \mathbf{v}_i \circ \widetilde{\mathbf{g}}_i \circ \mathbf{h}_i + \widehat{\mathbf{n}}_{i,3,b}, \tag{18}$$

where

$$\widehat{\mathbf{n}}_{i,3,a} = \mathbf{s}_i^{\circ 2} \circ \mathbf{z}_{i,a} + \mathbf{s}_i \circ \mathbf{n}_{i,2,r} \circ \widetilde{\mathbf{h}}_i + \mathbf{s}_i \circ \mathbf{n}_{i,3,a}/\alpha, \tag{19}$$
$$\widehat{\mathbf{n}}_{i,3,b} = \mathbf{v}_i^{\circ 2} \circ \mathbf{z}_{i,b} + \mathbf{v}_i \circ \mathbf{n}_{i,2,r} \circ \widetilde{\mathbf{g}}_i + \mathbf{v}_i \circ \mathbf{n}_{i,3,b}/\alpha, \tag{20}$$

are the noise terms. The two vectors $\mathbf{w}_{i,ab}$ and $\widetilde{\mathbf{w}}_{i,ab}$ observed by Alice and Bob are highly correlated and random at each session, which makes them suitable for extracting secret keys.

B. Quantization

In this stage, the complex-valued shared sequences $\mathbf{w}_{i,ab}$ and $\widetilde{\mathbf{w}}_{i,ab}$ are turned into bit streams. We use a similar quantization method as suggested in [13]. A brief description of the quantization scheme is included next. After collecting the complex-valued measurements $\mathbf{w}_{i,ab}$ and $\widetilde{\mathbf{w}}_{i,ab}$, they are sorted as shown in Figure 5. Then, Alice and Bob find the range of sorted data, which is defined as the difference between the maximum value and the minimum value of the sorted vectors. Then, using the range and the quantization resolution δ, they identify $\Delta = 2^{\delta}$ uniform quantization intervals, and assign a Gray-code sequence to each interval. Finally, they map each sample to its quantized bit sequence based on the interval it belongs to. The resulting bit sequences for Alice and Bob are denoted by $\mathbf{q}_{i,a}$ and $\mathbf{q}_{i,b}$, respectively.

C. Reconciliation

The aim of this stage is to mitigate disagreements between Alice's and Bob's quantized bit sequences. To this end, various methods, such as error-correcting codes, can be used.

In our protocols we use error-correcting code-based secure sketch [41], while picking a convolutional code as the underlying code. The reason to pick convolutional codes is due to the simplicity of the encoding process using shift registers and the decoding process using Viterbi decoders [46]. A formal definition of a general secure sketch scheme is as follows:

Definition 1: [47] An $(\mathcal{M}, m_1, m_2, t)$-secure sketch scheme consists of a sketch function, and a recovery function such that the following properties hold:

1) The sketch function takes an input $w \in \mathcal{M}$ and returns a randomized $SS(w) \in \{0,1\}^*$.
2) The recovery function takes $SS(w)$ and $\tilde{w} \in \mathcal{M}$, and returns w with probability one as long as the distance between w and $\tilde{w}$ is less than a certain threshold t.
3) For any random variable W over $\mathcal{M}$ with min-entropy m_1, an adversary observing $SS(W)$ has an average min-entropy of W conditioned on $SS(W)$ as $\tilde{H}_\infty(W|SS(W)) \geqslant m_2$.

Note that the min-entropy function of a random variable X is computed as $H_\infty(X) = -\log_2(\max_x(\Pr(\mathrm{X} = \mathrm{x})))$ and the average min-entropy function of X conditioned on Y is computed as $\tilde{H}_\infty(X|Y) = -\log_2\big(\mathop{\mathbb{E}}_{y \leftarrow Y}[2^{-H_\infty(X|Y=y)}]\big)$.

Next, we describe a construction known as the code-offset secure sketch [41]. The encoder is chosen in such a way that the length of its output is equal to the length of $\mathbf{q}_{i,a}$. Once the quantized sequences $\mathbf{q}_{i,a}$ and $\mathbf{q}_{i,b}$ are available, Alice chooses a bit string $\mathbf{r}$ uniformly at random and encodes it using the convolutional encoder to get $\mathrm{Enc}(\mathbf{r})$, which is of the same length as $\mathbf{q}_{i,a}$. Then, she computes

$$\mathbf{SS} = \mathbf{q}_{i,a} \oplus \mathrm{Enc}(\mathbf{r}), \tag{21}$$

where $\oplus$ is the addition modulo 2, and transmits the resulting sequence over the noiseless public channel, either directly as in the first scenario or through the relay as in the second scenario, to Bob. Then, Bob takes the addition modulo 2 of $\mathbf{SS}$ and $\mathbf{q}_{i,b}$, feeds it to the Viterbi decoder to get $\widetilde{\mathbf{r}}$, and re-encodes $\widetilde{\mathbf{r}}$ to get $\mathrm{Enc}(\widetilde{\mathbf{r}})$. He computes the final sequence as

$$\begin{aligned}\widetilde{\mathbf{q}}_{i,a} &= \mathbf{SS} \oplus \mathrm{Enc}(\mathrm{Dec}(\mathbf{SS} \oplus \mathbf{q}_{i,b})) \\ &= \mathbf{SS} \oplus \mathrm{Enc}(\widetilde{\mathbf{r}}).\end{aligned} \tag{22}$$

Remark: A binary linear code of length n and dimension k with minimum distance $2t+1$ can be used to build an $(\mathcal{F}^n, m_1, m_1 - (n-k), t)$-secure sketch scheme, where $\mathcal{F} = \{0,1\}$ for binary codes [47]. The error correction capability of the linear code is related to the underlying rate of the code. This introduces a trade-off between the error correction capability and the security, as higher rates provide better security but can correct less errors, and vice versa. Alice and Bob should start with an initial high rate code and then reduce it accordingly if they observe several consecutive unsuccessful attempts of the protocol.

D. Privacy Amplification Consistency Checking

Since some information about the shared key is leaked to Eve during the exchange of random symbols and the reconciliation stages, we exploit universal hash functions (UHF) to increase the level of security. In general, UHFs are desired in such scenarios due to their resilience against collisions.

Definition 2: [42] A family of hash functions H that maps a set of inputs U, e.g., binary vectors of length n, to a value in the hash table of size t is called universal if for any two inputs $x, y \in U$ with $x \neq y$, we have

$$\Pr_{h \leftarrow H}(h(x) = h(y) | x \neq y) \leqslant \frac{1}{t}. \tag{23}$$

We also use UHFs to check consistency between keys generated by Alice and Bob, without leaking any information to Eve, as suggested in [15].

Given that h should be chosen randomly from H, the question is how do we ensure that Alice and Bob agree on the same h? We propose a method that guarantees the same choice of h at Alice and Bob if inputs to the UHF are consistent. Suppose we have a random binary sequences $\mathbf{q}_i$ of length n (This is $\mathbf{q}_{i,a}$ for Alice and $\widetilde{\mathbf{q}}_{i,a}$ for Bob). For simplicity, we assume that n is an even multiple of some integer $m \geqslant 1$. We divide $\mathbf{q}_i$ into two sequences of equal length $\mathbf{q}_i = \mathbf{q}_{i,1} \| \mathbf{q}_{i,2}$ each of length $n/2$, which is an integer since n is even. Then, $\mathbf{q}_{i,1}$ is used to choose h from H, and $\mathbf{q}_{i,2}$ is used as the input to the hash function h. Next, a well-known construction of UHF is described that we use in our protocol [42]. First, the largest prime p with $2^{m-1} < p < 2^m$, i.e., its binary representation consists of m bits, is chosen, where m is the length of the output bit sequence (such a prime number always exist for $m > 1$ by Bertrand's postulate). Then, for $i = 1, 2$, we divide $\mathbf{q}_{i,1}$ and $\mathbf{q}_{i,2}$ into l parts $q_{i,1,k}$ and $q_{i,2,k}$ for $k = 1, 2, \ldots, l$, where the length of each part is less than or equal to m bits. For ease of notation, let $q_{i,j,k}$ also denote the number with the binary representation $q_{i,j,k}$. Finally, the following summation is computed:

$$h_{\mathbf{q}_{i,1}}(\mathbf{q}_{i,2}) = \sum_{k=1}^{l} q_{i,1,k} q_{i,2,k} \bmod p. \tag{24}$$

Next, the randomness of $\mathbf{q}_i$ is discussed. In our protocol, $\mathbf{s}_i$ and $\mathbf{v}_i$ are chosen uniformly at random for each key generation session. Hence, the value of $\mathbf{q}_i$ is also random. Therefore, the hash function is randomized during each session, which will be verified in the numerical results section. The output of the aforementioned described hash function is the key bit sequences $\mathbf{K}_{i,ab}$ for Alice and $\widetilde{\mathbf{K}}_{i,ab}$ for Bob, which are matched with high probability given the reconciliation step. Before Alice and Bob are able to use the key sequences for encryption and decryption, they need to verify the consistency of their keys. To this end, Alice and Bob hash their key sequences $\mathbf{K}_{i,ab}$ and $\widetilde{\mathbf{K}}_{i,ab}$ again similar to the previously described process. The output of this step is their respective check sequences $\mathbf{C}_{i,ab}$ and $\widetilde{\mathbf{C}}_{i,ab}$, which they use to verify whether or not their keys are consistent. It is worth noting that, in our protocol, the length of the check sequences, $\mathbf{C}_{i,ab}$ and $\widetilde{\mathbf{C}}_{i,ab}$, is half the length of the key.

Theorem 1: The probability of accepting a mismatched key as consistent by the described protocol with hash table size p for the check sequence is upper bounded as follows:

$$\Pr(\mathbf{C}_{i,ab} = \widetilde{\mathbf{C}}_{i,ab} | \mathbf{K}_{i,ab} \neq \widetilde{\mathbf{K}}_{i,ab}) \leqslant \frac{1}{p}. \tag{25}$$

Proof: This follows directly from the definition of universal hash functions, specified in (23), where the output hash table size is p. ■

IV. Attacker Model and the Resilience of Proposed Protocols

In this section we discuss eavesdropping strategies by the passive eavesdropper Eve in both scenarios, i.e., whether the communication is through a direct communication channel or through a relay, and provide an upper bound on the probability of a successful eavesdropping attack.

A. Direct Secret Key Generation

In this scenario, Eve's best strategy is to acquire $\mathbf{s}_i, \mathbf{v}_i$ and $\mathbf{h}_{i,ab}$. When Alice and Bob exchange signals, Eve receives

$$\mathbf{e}_{i,1} = \mathbf{s}_i \circ \mathbf{h}_{i,ae} + \mathbf{n}_{i,e_1}, \tag{26}$$

$$\mathbf{e}_{i,2} = \mathbf{v}_i \circ \mathbf{h}_{i,be} + \mathbf{n}_{i,e_2}. \tag{27}$$

If Eve is able to estimate both $\mathbf{s}_i$ and $\mathbf{v}_i$ from her observations in (26) and (27) perfectly, she can create samples of the following form:

$$\mathbf{w}_{i,e_d,1} = \mathbf{s}_i \circ \mathbf{v}_i \circ \mathbf{h}_{i,ae} + \mathbf{n}_{i,e_3}, \tag{28}$$

$$\mathbf{w}_{i,e_d,2} = \mathbf{s}_i \circ \mathbf{v}_i \circ \mathbf{h}_{i,be} + \mathbf{n}_{i,e_4}. \tag{29}$$

Note that she still needs to know $\mathbf{h}_{i,ab}$ at all different subcarriers in order to obtain $\mathbf{w}_{i,ab}$ and/or $\widetilde{\mathbf{w}}_{i,ab}$, as described in (10) and (11). Luckily, this is, almost, not possible for Eve as discussed next.

In general, the Pearson correlation coefficient ρ of the channel fading coefficients at locations separated by distance d can be computed as follows [12]:

$$\rho = [J_0(kd)]^2, \tag{30}$$

where $J_0(.)$ is the Bessel function of first kind, and k is the wavenumber. Therefore, if the distance between Alice/Bob and Eve is larger than half of the wavelength $\lambda/2$, e.g., 5 cm in 3GHz band, they will experience almost uncorrelated fading channels. Therefore, the leaked information about the generated secret key to Eve is small and is often assumed to be negligible in the literature. However, it is fundamentally important to quantitatively measure the security level. An information-theoretic measure of security is the mutual information between the shared random sequence, from which the secure key will be generated, and what Eve observes. If we assume that the effect of quantization is negligible and also assume that Eve can perfectly recover $\mathbf{s}_i$ and $\mathbf{v}_i$, this mutual information is equal to the mutual information between $\mathbf{h}_{i,ab}$ and the pair $(\mathbf{h}_{i,ae}, \mathbf{h}_{i,be})$. One can assume that Eve is closer to Bob than Alice and hence, only consider the mutual information between $\mathbf{h}_{i,ab}$ and $\mathbf{h}_{i,ae}$ as the dominating term. This can be calculated in each subcarrier as stated in the next lemma.

Lemma 2: Let h_{bk} and h_{ek} denote the fading coefficients of Bob's and Eve's channels at the k-th subcarrier. Also, let ρ denote the correlation coefficient between h_{bk} and h_{ek}, specified in (30). Then, the mutual information between h_{bk} and h_{ek} is given by

$$I(h_{bk}; h_{ek}) = -\log(1-\rho^2) \text{ bits}. \tag{31}$$

Proof: We have $h_{bk} = h_{bk,I} + jh_{bk,Q}$, and $h_{ek} = h_{ek,I} + jh_{ek,Q}$. $h_{bk,I}, h_{bk,Q}$ are independent and identically distributed as $\mathcal{N}(0, \sigma_b^2/2)$ and $h_{ek,I}, h_{ek,Q}$ are independent and identically distributed as $\mathcal{N}(0, \sigma_e^2/2)$. The real parts of Bob's and Eve's channel coefficients are correlated with the parameter ρ, and the imaginary parts are also correlated with ρ. Then, we have the following covariance matrices:

$$\Sigma_1 = \begin{bmatrix} \sigma_b^2/2 & 0 \\ 0 & \sigma_b^2/2 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} \sigma_e^2/2 & 0 \\ 0 & \sigma_e^2/2 \end{bmatrix}, \tag{32}$$

$$\Sigma_3 = \begin{bmatrix} \sigma_b^2/2 & 0 & \frac{\rho\sigma_b\sigma_e}{2} & 0 \\ 0 & \sigma_b^2/2 & 0 & \frac{\rho\sigma_b\sigma_e}{2} \\ \frac{\rho\sigma_b\sigma_e}{2} & 0 & \sigma_e^2/2 & 0 \\ 0 & \frac{\rho\sigma_b\sigma_e}{2} & 0 & \sigma_e^2/2 \end{bmatrix}. \tag{33}$$

Then, the following series of equalities holds:

$$\begin{aligned} I(h_{bk}; h_{ek}) &= I(h_{bk,I} + jh_{bk,Q}; h_{ek,I} + jh_{ek,Q}) \\ &\overset{(a)}{=} I(h_{bk,I}, h_{bk,Q}; h_{ek,I}, h_{ek,Q}) \\ &\overset{(b)}{=} H_d(h_{bk,I}, h_{bk,Q}) + H_d(h_{ek,I}, h_{ek,Q}) \\ &\quad - H_d(h_{bk,I}, h_{bk,Q}, h_{ek,I}, h_{ek,Q}) \\ &\overset{(c)}{=} \frac{1}{2}\log(\det(2\pi e\Sigma_1)) + \frac{1}{2}\log(\det(2\pi e\Sigma_2)) \\ &\quad - \frac{1}{2}\log(\det(2\pi e\Sigma_3)) \\ &\overset{(d)}{=} \log(\pi e\sigma_b^2) + \log(\pi e\sigma_e^2) \\ &\quad - \log((\pi e\sigma_b\sigma_e)^2(1-\rho^2)) \\ &\overset{(e)}{=} -\log(1-\rho^2), \end{aligned} \tag{34}$$

where:
(a) holds due to having a one-to-one mapping;
(b) is the expansion of the mutual information expression in terms of differential entropy;
(c) holds by using the well-known expression that the differential entropy of multivariate Gaussian random variables $\mathbf{X}^n = (X_1, X_2, ..., X_n)$ with covariance matrix Σ_i is $H_d(\mathbf{X}^n) = \frac{1}{2}\log(\det(2\pi e\Sigma_i))$;
and (d) and (e) are simplification steps. ■

Note that as ρ goes to zero, the mutual information, given by Lemma 2, also goes to zero.

The next question, which also applies to any physical layer security scheme that utilizes information-theoretic measures of security, is how to quantitatively characterize the chances of a successful eavesdropping attack by Eve, i.e., guessing the key, given the leaked information? The latter is often measured in terms of semantic security, which is a classical notion of security in cryptosystems [48]. Direct connections between metrics for the information-theoretic security, based on the mutual information, and cryptographic measures of security, including semantic security, are provided in [49]. We use these connections to arrive at the following theorem which characterizes the

security of the proposed protocol from the aforementioned perspective:

Theorem 3: Let N denote the number of subcarriers used in the proposed protocol and δ denote the quantization resolution. Then, the probability of a successful eavesdropping attack by Eve is upper bounded as follows:

$$\Pr(\text{Successful attack}) < \big(2^{-2\delta} + \sqrt{2I(h_b;h_e)}\,\big)^N + 2^{-\delta N}, \tag{35}$$

where h_b and h_e denote the fading coefficients of Bob's and Eve's channels at a subcarrier.

Proof: [49, Theorem 5] relates the mutual information between Bob's and Eve's observations to the increase in the probability of a successful eavesdropping attack by Eve given her observations. More specifically, the increase in the latter probability is quantified in terms of the mutual information between Bob's and Eve's observations [49, Theorem 5]. Note that the probability that Eve successfully guesses the bits, with no observations, at a single subcarrier is $2^{-2\delta}$. In addition to that, by [49, Theorem 5], the probability that Eve can guess the shared random bits in a single subcarrier, given her observations in this subcarrier, is increased by at most $\sqrt{2I(h_b;h_e)}$ compared to the case where she does not have any observation. Therefore, Eve's probability of successfully guessing these quantized key bits is upper bounded by $2^{-2\delta} + \sqrt{2I(h_b;h_e)}$. The probability that Eve can recover the shared randomness over all subcarriers is then given by $\big(2^{-2\delta} + \sqrt{2I(h_b;h_e)}\big)^N$. Note that $I(h_b;h_e)$ is the same across all the subcarriers and is actually computed in terms of ρ in Lemma 2. If Eve cannot recover all the shared randomness, the probability that she can guess the secret key correctly, by the property of hash functions in the privacy amplification part of our protocol, is at most $2^{-\delta N}$, when using a key sequence of half the quantized bit sequence length. Utilizing these together with the union bound completes the proof. ∎

Note that Theorem 3 together with Lemma 2 can be used to provide a numerical upper bound on the probability of a successful eavesdropping attack given a lower bound on the distance between Eve and both Alice and Bob. For instance, suppose that the distance between Eve and Bob is at least half of a wavelength, i.e., $d = \lambda/2$ and is less than the distance between Eve and Alice. Then, the correlation coefficient ρ is at most 0.09 and by Lemma 2 the resulting mutual information $I(h_b;h_e)$ is at most 0.01 bits at any of the subcarriers. Suppose that $N = 16$ and $\delta = 2$, which are also used in the numerical results provided in the next section. Then, by Theorem 3, the probability of a successful attack by Eve given such parameters is at most $2^{-37} + 2^{-32} < 2^{-31}$.

B. Relay-based Secret Key Generation

In this scenario, Eve tries to use her observations and the messages transmitted over the public channel to guess $\mathbf{w}_{i,ab}$ and/or $\widetilde{\mathbf{w}}_{i,ab}$, as described in (17) and (18). Her best strategy is to find $\mathbf{s}_i$, $\mathbf{v}_i$, $\mathbf{g}_i$ and $\mathbf{h}_i$. When Alice and Bob transmit their induced randomness, Eve receives

$$\mathbf{w}_{i,e,1} = \mathbf{s}_i \circ \mathbf{h}_{i,ae} + \mathbf{v}_i \circ \mathbf{g}_{i,be} + \mathbf{n}_{i,e_5}. \tag{36}$$

However, when Carol, the relay, amplifies and forwards the signal from Alice and Bob, Eve receives

$$\mathbf{e}_{i,3} = \alpha(\mathbf{s}_i \circ \mathbf{h}_i + \mathbf{v}_i \circ \mathbf{g}_i + \mathbf{n}_{i,2,r}) \circ \mathbf{f}_{i,ce} + \mathbf{n}_{i,e_6}. \tag{37}$$

Since Eve can estimate the channel coefficients $\mathbf{f}_{i,ce}$ from the relay's transmission when it transmits the known probing vector and, also, she knows the value of α from the messages over the public channel, she can successfully estimate

$$\mathbf{w}_{i,e,2} = \mathbf{s}_i \circ \mathbf{h}_i + \mathbf{v}_i \circ \mathbf{g}_i + \mathbf{n}_{i,2,r}. \tag{38}$$

In a worst-case scenario from the legitimate parties' perspective, Eve has as much information as the relay has, in addition to her own observations. Note that this coincides with the problem of securing the shared key against the untrusted relay Carol when Eve is at Carol's location. In the remaining of this section, we analyze the probability of a successful eavesdropping attack assuming that the eavesdropper Eve has all the information available to Carol, in addition to her own observations.

Note that the computations involving the spatial correlation parameter of the wireless channels do not help in ensuring the security in this scenario as they do in the first scenario with a direct communication channel. Also, the mutual information between $\mathbf{w}_{i,ab}$, as described in (17), and the pair $(\mathbf{w}_{i,e,1}, \mathbf{w}_{i,e,2})$, as described in (36) and (38), respectively, is expected not to be very small as it was in the first scenario. For instance, if this mutual information is greater than 0.5, then using [49, Theorem 5], same as in the proof of Theorem 3, does not yield a non-trivial upper bound on the probability of a successful eavesdropping attack. Hence, instead of utilizing semantic security, we need to use an alternative approach to relate $I(\mathbf{w}_{i,ab}; \mathbf{w}_{i,e,1}, \mathbf{w}_{i,e,2})$ to the probability of a successful eavesdropping attack. To this end, we use Fano's inequality to bound the probability of successful estimation of the quantized bits $\mathbf{q}_{i,a}$ by the eavesdropper in terms of the conditional entropy of the quantized bits $\mathbf{q}_{i,a}$ given the eavesdropper's observations $(\mathbf{w}_{i,e,1}, \mathbf{w}_{i,e,2})$. Note that the latter can be bounded in terms of $I(\mathbf{w}_{i,ab}; \mathbf{w}_{i,e,1}, \mathbf{w}_{i,e,2})$. The details of this analysis are given next in the proof of Theorem 4.

To simplify the expressions in the next theorem, let us consider an arbitrary subcarrier and denote the corresponding entries of the vectors $\mathbf{w}_{i,ab}$, $\mathbf{w}_{i,e,1}$, $\mathbf{w}_{i,e,2}$, and $\mathbf{q}_{i,a}$ as w_{ab}, $w_{e,1}$, $w_{e,2}$, and q_a, respectively. Note that the result of Theorem 4 does not depend on the choice of the subcarrier.

Theorem 4: Let N denote the number of subcarriers used in the proposed protocol and δ denote the quantization resolution. Then, the probability of a successful eavesdropping attack by Eve is upper bounded as follows:

$$\Pr(\text{Successful attack}) < \Big(1 - \frac{H(q_a) - I_{ab,e} - 1}{\log_2(|\mathcal{Q}_A|)}\Big)^N + 2^{-\delta N}, \tag{39}$$

where $I_{ab,e}$ denotes $I(w_{ab}; w_{e,1}, w_{e,2})$, $\mathcal{Q}_A$ denotes the support of q_a, and $|\mathcal{Q}_A|$ denotes its cardinality.

Proof: Let C denote the event of correct estimation of q_a and E denote the event of erroneous estimation of q_a by the

eavesdropper. Then, we have the following

$$\Pr(\mathrm{C}) = 1 - \Pr(\mathrm{E}) \tag{40}$$

$$\overset{(a)}{\leqslant} 1 - \frac{H(q_a|w_{e,1}, w_{e,2}) - 1}{\log_2(|\mathcal{Q}_A|)} \tag{41}$$

$$\overset{(b)}{=} 1 - \frac{H(q_a) - I(q_a; w_{e,1}, w_{e,2}) - 1}{\log_2(|\mathcal{Q}_A|)} \tag{42}$$

$$\overset{(c)}{\leqslant} 1 - \frac{H(q_a) - I(w_{ab}; w_{e,1}, w_{e,2}) - 1}{\log_2(|\mathcal{Q}_A|)} \tag{43}$$

$$\overset{(d)}{=} 1 - \frac{H(q_a) - I_{ab,e} - 1}{\log_2(|\mathcal{Q}_A|)}, \tag{44}$$

where:
(a) holds by Fano's inequality [50];
(b) is the expansion of conditional entropy;
(c) follows from the data processing inequality because q_a is a deterministic function of w_{ab} and hence, $(w_{e,1}, w_{e,2})$, w_{ab}, and q_a form a Markov chain;
(d) is a change of the notation of $I(w_{ab}; w_{e,1}, w_{e,2})$ to $I_{ab,e}$.

Note that the probability of correctly estimating every bit of $\mathbf{q}_a$, denoted by $\Pr(\mathrm{C_N})$, is equal to the probability of correctly estimating q_a over all the N subcarriers, since the computation of mutual information is the same over all subcarriers. Hence, by using the independence of such events across the N subcarriers, we have

$$\Pr(\mathrm{C_N}) \leqslant \Big(1 - \frac{H(q_a) - I_{ab,e} - 1}{\log_2(|\mathcal{Q}_A|)}\Big)^N. \tag{45}$$

If Eve cannot recover all the shared randomness bits in a single session, the probability that she correctly guesses the secret key, by the property of hash functions in the privacy amplification part of our protocol, is at most $2^{-\delta N}$. This, together with (45), and using the union bound complete the proof. ∎

Next, we illustrate how Theorem 4 can be used in a numerical setup to upper bound the probability of a successful eavesdropping attack by Eve. Suppose that Alice and Bob use 64-QAM constellation points to transmit their induced randomness, the received SNR is 23 dB at Alice and Bob in (15), the quantization parameter δ is 2, and the number of subcarriers N is 16. Eve is located close to Carol, but at least $\lambda/2$ away from her. Given these parameters the mutual information $I_{ab,e} = I(w_{ab}; w_{e,1}, w_{e,2})$ is numerically estimated as $I_{ab,e} \approx 1.39$ bits, and the entropy of the generated key bits is numerically estimated as $H(q_a) \approx 3.86$ bits. Then, by Theorem 4, the probability of successful eavesdropping attack is upper bounded, approximately, by $2^{-10.57}$.

V. Numerical Results

In this section we consider numerical setups with reasonable parameters and evaluate the proposed protocols for the two described scenarios, i.e., when a direct channel exists, and when a relay is used for the key generation, using the metrics described in Section II. Also, in order to provide numerical results in a realistic environment, a certain number of channel coefficients is generated by the NYUSIM Channel Simulator [44]. The simulator is used to generate channel coefficients for realistic 5G mmWave channels from measurement-based models. A description of the three setups is discussed next, followed by numerical results shown for all the considered setups.

A. Setup

1) Direct Secret Key Generation: In this scenario, it is assumed that Alice and Bob communicate over a direct and reciprocal wireless channel. The constellation size for each subcarrier is $M = 16$, i.e., the set of 16-QAM symbols are used as the set from which local randomness is chosen and transmitted by Alice and Bob. Also, $N = 16$ OFDM subcarriers are assumed to be available in the channel between Alice and Bob. The quantization is done with $\delta = 2$, i.e., the real and imaginary parts of the received symbol in each subcarrier are quantized into one of the four possibilities as discussed in Section III-B. Finally, the remaining steps including secure sketch, hashing, and consistency checking are performed as discussed in Section III.

2) NYUSIM-Based Secret Key Generation: In this scenario, it is assumed that Alice and Bob have a direct reciprocal wireless channel where the coefficients are generated by the NYUSIM Channel Simulator [44]. They operate in a non-line-of-sight (NLOS) urban micro-cellular environment at 20°C, the operating frequency is 28 GHz, and the distance between Alice and Bob and Alice and Eve is 10 meters. The path contains 1 meter of foliage, and there is an outdoor-to-indoor low loss. Channel coefficients between Alice and Bob and channel coefficients between Alice and Eve are generated by the NYUSIM Channel Simulator over $N = 16$ subcarriers. Alice and Bob choose their induced randomness from the set of 16-QAM symbols, and the quantization is done with $\delta = 2$. The remaining steps follow as in the first scenario.

3) Relay-Based Secret Key Generation: In this scenario, it is assumed that Alice and Bob have direct and reciprocal wireless channels with the relay, which can be perfectly estimated. Also, a scenario is considered for eavesdropping, as discussed in Section IV-B, where Eve uses the relay's observations. Alice and Bob choose their induced randomness from the set of 64-QAM symbols, and set their power levels in such a way that the average received SNRs at the relay from both Alice and Bob are equal. Then, the amplification vector α is chosen such that the average SNR, which is considered in the results, of Alice and Bob's correlated observations, (17) and (18), respectively, is the same. The remaining parameters and steps are similar to the previous scenarios.

B. Results

1) Bit Generation Rate: For all the setups described above, Alice and Bob exchange their induced randomness over $N = 16$ subcarriers, with quantization resolution $\delta = 2$ for the real and imaginary parts separately. Note that $16 \times 2 \times 2 = 64$ bits are generated by Alice and Bob during each session of the protocol. Hence, the bit generation rate (BGR) is 64 bits/packet. The length of the final secret key is $64/2 = 32$ bits. In order to increase Eve's bit error rate, we assume that four blocks of keys, generated during four separate successful sessions, are added together modulo 2 to obtain one final key of length 32 per each four sessions. Such BGR is considered high compared to protocols designed for static channels setups, which have BGR of $\frac{1}{4}$ to $\frac{1}{2}$ bits/packet as in [25], or 8 bits/packet as in [26], and

it is comparable with protocols designed for dynamic environments, such as [14] whose BGR is $60 - 90$ bit/packet.

2) Bit Mismatch Rate and Bit Error Rate: The bit mismatch rate (BMR) and bit error rate (BER) between Alice and Bob, and Alice and Eve are shown in Figure 6 and Figure 7, respectively, for the three described setups. For the bit mismatch rate, in the direct and NYUSIM-based SKG setups we compare Alice's and Bob's quantized sequences of (10) and (11), respectively, and Alice's and Eve's quantized sequences of (10) and (28), respectively. Also, for the relay-based SKG setup, we compare Alice's and Bob's quantized sequences of (17) and (18), respectively, and Alice's and Eve's quantized sequences of (17) and (38), respectively. It is worth noting that as the average SNR increases in the NYUSIM-based SKG setup, Eve's BMR decreases but the rate of decrease slows down. It can be observed that an increase of around 3 dB of the average SNR is required in the relay-based SKG setup to achieve a BMR similar to the first two setups. In comparison with other protocols for static environments at 20 dB, they have BMR of around 1% as in [25], 4% as in [26], 4% and 13% for the direct and relay-based setups as in [39].

As for the BER, we compare Alice's and Bob's final key sequences and Alice's and Eve's final key sequences. It can be observed that the BER at Bob is extremely low due to the requirement of the consistency checking step in the protocol, which only allows keys whose consistency is verified with high probability to be accepted. Note that the main reason for the average BER at Eve being around 50% is the privacy amplification step of the protocol. In addition to that, the cumulative distribution function (CDF) of the BER at Eve's final key at 20 dB average SNR for both the direct and NYUSIM-based SKG setups, and 23 dB average SNR for the relay-based SKG setup is shown in Figure 8. Note that the curves for all the setups are similar because these curves compare the keys which are the addition modulo 2 of four separate outputs of the hash functions at Alice and Eve. The universal hash function generates a uniformly random output resulting in the similarity of the curves. Also, it is observed that the probability of accepting a mismatched key for the aforementioned average SNRs in the direct and relay-based SKG setups is around 0.0015%, and for the NYUSIM-based SKG setup is around 0.00152%, which are less than 0.00153% as predicted by Theorem 1. The aforementioned probability is considered to be very low. In comparison, it is far less than the probability of generating mismatched keys of the protocol proposed for direct SKG in static environments in [28], which is at least 3%. In addition to that, as discussed in the security evaluation of the protocol, the probability of acquiring the key perfectly by Eve is, at most, 2^{-31} and $2^{-10.57}$, for the direct and relay-based SKG, respectively. In comparison, the protocol proposed for direct SKG in static environments in [28] has the probability of acquiring the key by Eve in the range $0.09\% - 0.47\%$.

3) Randomness: The randomness of the generated final key sequence is examined using the NIST statistical test suite [43]. The suite consists of 15 tests and generates a probability value, also referred to as p-value, for each individual test. For each test, a sequence is considered random with 99% confidence if the corresponding p-value is greater than 0.01. We run

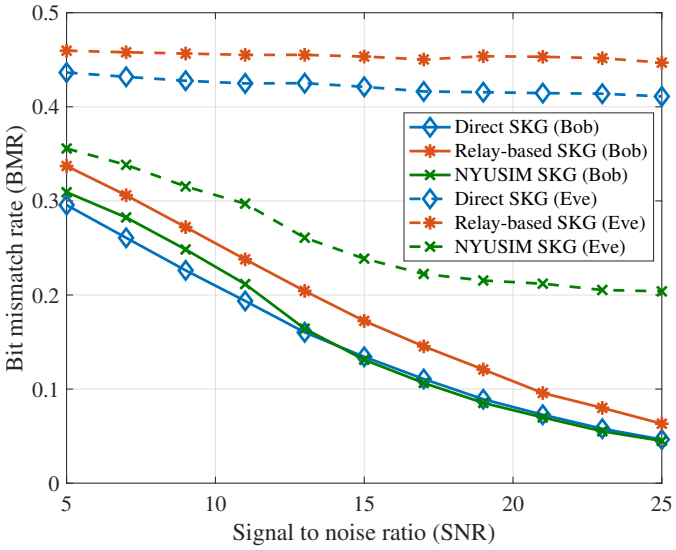

Figure 6. The bit mismatch rate (BMR) between Alice's sequence and Bob's and Eve's sequences versus the signal to noise ratio.

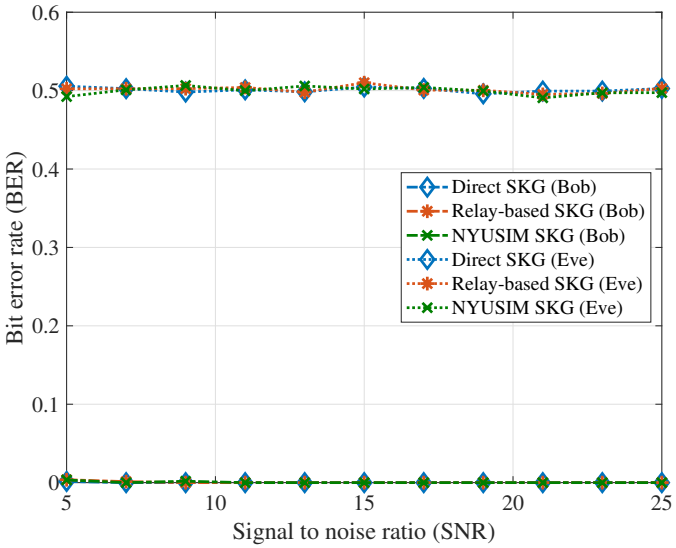

Figure 7. The bit error rate (BER) between Alice's sequence and Bob's and Eve's sequences versus the signal to noise ratio.

the protocol using constant channel coefficients at 20 dB average SNRs for the direct and NYUSIM-based SKG setups, and 23 dB average SNR for (17) and (18) in the relay-based SKG setup to generate a sequence of length 2^{20} bits and feed it to the test suite. Since the sequences pass all the tests as shown in Table II, they are considered random with 99% confidence.

4) Randomness Efficiency: This is computed according to (9). For the direct and NYUSIM-based SKG setups, Alice and Bob randomly choose induced randomness bit sequences of length 64, and therefore, $H(S) = H(V) = 64$. Note that the length of the quantized bit sequence is 64, therefore, $R_Q = 64$. This implies that the randomness efficiency is 50%. On the other hand, for the relay-based SKG setup, Alice and Bob separately induce 96 random bits during each round, resulting in $H(S) = H(V) = 96$, while the length of the quantized bit se-

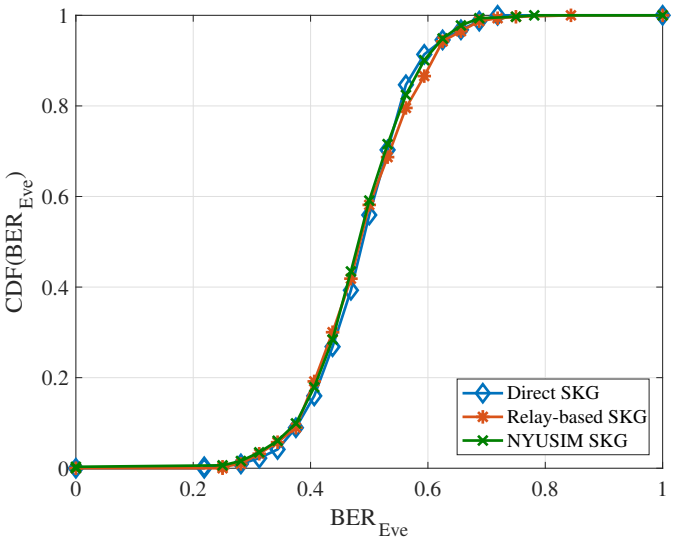

Figure 8. The cumulative distribution function of the BER at Eve for the direct, relay-based, and NYUSIM-based SKG setups. The compared sequences are the modulo 2 addition of the outputs of four successful key generation sessions. The universal hash function is supposed to generate a uniformly random output, hence the similarity of the curves.

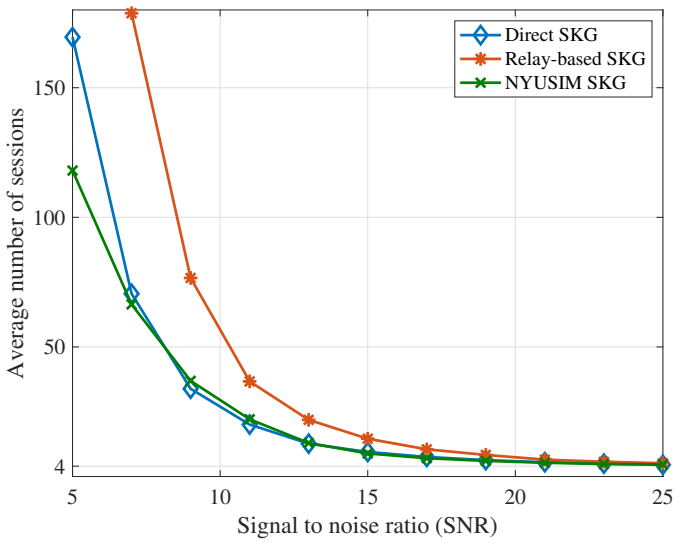

Figure 9. The average number of sessions until key agreement versus the signal to noise ratio.

Table II
NIST STATISTICAL TEST RESULTS

Test	Direct	Relay	NYUSIM
Monobit	0.8712	0.9968	0.2829
Frequency Block	0.3529	0.7458	0.3172
Runs	0.5347	0.9781	0.4516
Longest Run of Ones	0.7696	0.3552	0.1692
Binary Matrix Rank	0.9263	0.9974	0.3125
DFT	0.6413	0.3469	0.0121
Non-Overlapping Template Matching	1	1	1
Overlapping Template Matching	0.2830	0.3501	0.4043
Maurer's Universal Statistical	0.9991	1	0.9993
Linear Complexity	0.9909	0.5323	0.0227
Serial	0.2989	0.5852	0.7236
Approximate Entropy	0.4808	0.9160	0.7529
Cumulative Sums	0.7825	0.8392	0.1833
Random Excursion	0.0179	0.2924	0.0925
Random Excursion Variant Test	0.0434	0.0154	0.0615

quence is $R_Q = 64$. The resulting randomness efficiency of the relay-based SKG setup is 33%. Roughly speaking, the remaining part of the available randomness is used to provide security. The exact trade-off between randomness efficiency and security is an interesting problem.

5) Average Number of Sessions Required to Generate Keys: In this part, the average number of sessions Alice and Bob need to generate their final secret key given different values of SNR is compared for all three setups. Note that the length of final secret key is 32, which is obtained by adding modulo 2 the outputs of the protocol in four successful sessions. Hence, the average number of sessions required to generate a key approaches 4 as SNR grows large. The number of required sessions for the relay-based scenario is higher due to a more severe effect of the noise on the shared randomness. This, consequently, affects how often Alice and Bob obtain the same key sequence resulting in a successful session of the protocol. Figure 9 shows the average number of sessions for all considered setups.

6) Impact of Non-reciprocity: The perfect channel reciprocity feature is assumed to hold throughout the paper; however, in some practical scenarios, different factors such as mismatched hardware and synchronization errors may cause the channel coefficients experienced at Alice and Bob to not being perfectly reciprocal [14], [18], [51]. Such imperfections can be taken into account using the Pearson correlation coefficient, denoted by ζ, between such channel coefficients explained as follows. In general, under perfect channel reciprocity conditions, we have $\zeta = 1$, while imperfections reduce the value of ζ. As suggested in [51], a model to describe the relation between the channel coefficients at a subcarrier during session i observed at Alice, i.e., $\widetilde{h}_{i,ab}$, and Bob, i.e., $h_{i,ab}$, when they observe the same SNR is as follows:

$$h_{i,ab} = \zeta \widetilde{h}_{i,ab} + \sqrt{1-|\zeta|^2}\frac{\sigma_{h_i}}{\sqrt{2}} n_i, \tag{46}$$

where ζ is the correlation coefficient, $\sigma^2_{h_i}/2$ is the dimension variance of $h_{i,ab}$ and $\widetilde{h}_{i,ab}$, and n_i denotes the circularly-symmetric Gaussian-distributed independent noise component with mean 0 and unit dimension variance. In order to illustrate the effect of imperfect reciprocity in the direct SKG setup, the bit mismatch rate for different values of the correlation coefficient ζ is shown in Figure 10. It can be observed that as the correlation coefficient between the channel coefficients decreases, the BMR between Alice's and Bob's quantized sequences increases causing the protocol to experience higher number of unsuccessful sessions. For instance, to achieve a BMR around 22%, the required SNR is 9 dB for $\zeta = 1$, whereas it is 15 dB for $\zeta = 0.9$. On the other hand, when comparing the average number of sessions required to agree on a key at 15 dB, it is around 9 sessions for $\zeta = 1$, while it is around 37 sessions for $\zeta = 0.9$. Depending on the severity of the imperfections, the protocol's parameters would require certain adjustments to overcome such degradation. For example, the legitimate parties can decrease the bit generation rate

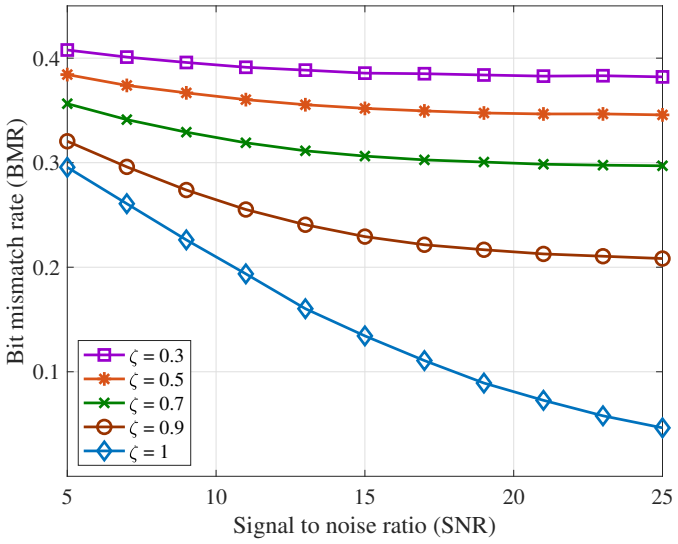

Figure 10. The bit mismatch rate (BMR) in the direct SKG setup between Alice's and Bob's sequences versus the signal to noise ratio for different values of the correlation coefficient ζ of the channels experienced at Alice and Bob.

by using a lower quantization resolution δ, or decrease the rate of the error-correcting code used for reconciliation which results in an increase of the amount of information leaked to the eavesdropper.

VI. Conclusion

In this paper, we propose a new low-complexity approach to generate secret keys in static environments at high rates using induced randomness. We utilize a low-complexity method where legitimate parties induce locally-generated randomness into the channel such that high-rate common randomness can be generated. More specifically, two main scenarios are considered for the proposed protocols taking into account whether a direct wireless channel is available between legitimate parties or no such channel is available and the transmissions occur through an intermediate relay. We evaluate the reliability and security of the proposed protocols using information theoretic measures. The protocols are also evaluated using metrics including BGR, BMR, BER, and the newly introduced randomness efficiency. Furthermore, numerical results are also shown for a realistic 5G mmWave setup, where channel coefficients are generated by the measurement-based NYUSIM Channel Simulator [44]. To ensure that the keys generated by this protocol are random, the generated keys are tested using the NIST statistical test suite. The low-complexity nature of the various steps of the proposed protocols make them appealing for applications concerning resource-constrained devices, e.g., IoT networks, where low complexity methods for generating distributed secret keys are highly desirable.

There are several possible directions for future work. It is interesting to extend the setups considered in this paper to multi-user scenarios where multiple users wish to generate shared secret keys with the help of multiple intermediate relays. From an information-theoretic perspective, this relates to the problem of distributed secret sharing in multi-user scenarios [52]. Also, investigating scenarios where the passive eavesdropper has further capabilities than what is considered in this paper, e.g., being able to deploy multiple antennas in the surrounding environment, is another interesting direction. Moreover, studying the resilience of the proposed protocols in the presence of an active eavesdropper who can act as a jammer with the aim of partially crippling the key generation process by sending intentional interference during the randomness exchange is another interesting direction for future work.

References

[1] N. Aldaghri and H. Mahdavifar, "Fast secret key generation in static environments using induced randomness," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[2] N. F. Pub, "197: Advanced encryption standard (AES)," *Federal information processing standards publication*, vol. 197, no. 441, p. 0311, 2001.

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[4] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: a survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[5] M. Bloch and J. Barros, *Physical-layer security: from information theory to security engineering*. Cambridge University Press, 2011.

[6] R. Ahlswede and I. Csiszár, "Common randomness in information theory and cryptography. i. secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, 1993.

[7] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE transactions on information theory*, vol. 39, no. 3, pp. 733–742, 1993.

[8] 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*. ACM, 2008, pp. 128–139.

[9] 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.

[10] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 364–375, 2007.

[11] K. Ren, H. Su, and Q. Wang, "Secret key generation exploiting channel characteristics in wireless communications," *IEEE Wireless Communications*, vol. 18, no. 4, 2011.

[12] A. F. Molisch, *Wireless communications*. John Wiley & Sons, 2012, vol. 34.

[13] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "Secret key extraction from wireless signal strength in real environments," *IEEE Transactions on mobile Computing*, vol. 12, no. 5, pp. 917–930, 2013.

[14] H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 3048–3056.

[15] W. Xi, X.-Y. Li, C. Qian, J. Han, S. Tang, J. Zhao, and K. Zhao, "KEEP: Fast secret key extraction protocol for D2D communication," in *Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of*. IEEE, 2014, pp. 350–359.

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

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

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

[19] M. Zorgui, Z. Rezki, B. Alomair, E. A. Jorswieck, and M.-S. Alouini, "On the ergodic secret-key agreement over spatially correlated multiple-antenna channels with public discussion," *IEEE Transactions on Signal Processing*, vol. 64, no. 2, pp. 495–510, 2016.

[20] E. A. Jorswieck, A. Wolf, and S. Engelmann, "Secret key generation from reciprocal spatially correlated MIMO channels," in *Globecom Workshops (GC Wkshps), 2013 IEEE*. IEEE, 2013, pp. 1245–1250.

[21] L. Jiao, N. Wang, and K. Zeng, "Secret Beam: Robust secret key agreement for mmWave massive MIMO 5G communication," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[22] M. G. Madiseh, S. W. Neville, and M. L. McGuire, "Applying beamforming to address temporal correlation in wireless channel characterization-based secret key generation," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1278–1287, 2012.

[23] S. Gollakota and D. Katabi, "Physical layer wireless security made fast and channel independent," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1125–1133.

[24] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE transactions on wireless communications*, vol. 7, no. 6, 2008.

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

[26] P. Huang and X. Wang, "Fast secret key generation in static wireless networks: A virtual channel approach," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2292–2300.

[27] G. Li, A. Hu, J. Zhang, and B. Xiao, "Security analysis of a novel artificial randomness approach for fast key generation," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[28] S. Fang, I. Markwood, and Y. Liu, "Manipulatable wireless key establishment," in *2017 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2017, pp. 1–9.

[29] B. Sunar, W. J. Martin, and D. R. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Transactions on computers*, vol. 56, no. 1, 2007.

[30] N. Ebrahimi, H. Mahdavifar, and E. Afshari, "A novel approach to secure communication in physical layer via coupled dynamical systems," *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2018.

[31] H. Mahdavifar and N. Ebrahimi, "Secret key generation via pulse-coupled synchronization," *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pp. 3037–3041, 2019.

[32] N. Ebrahimi, B. Yektakhah, K. Sarabandi, H.-S. Kim, D. D. Wentzloff, and D. Blaauw, "A novel physical layer security technique using master-slave full duplex communication," *Proceedings of IEEE/MTT-S International Microwave Symposium (IMS)*, pp. 1096–1099, 2019.

[33] L. Lai, Y. Liang, and W. Du, "Cooperative key generation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 8, pp. 1578–1588, 2012.

[34] Q. Wang, K. Xu, and K. Ren, "Cooperative secret key generation from phase estimation in narrowband fading channels," *IEEE Journal on selected areas in communications*, vol. 30, no. 9, pp. 1666–1674, 2012.

[35] L. Dong, Z. Han, A. P. Petropulu, and H. V. Poor, "Improving wireless physical layer security via cooperating relays," *IEEE transactions on signal processing*, vol. 58, no. 3, pp. 1875–1888, 2010.

[36] T. Shimizu, H. Iwai, and H. Sasaoka, "Physical-layer secret key agreement in two-way wireless relaying systems," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 650–660, 2011.

[37] H. Zhou, L. M. Huie, and L. Lai, "Secret key generation in the two-way relay channel with active attackers." *IEEE Trans. Information Forensics and Security*, vol. 9, no. 3, pp. 476–488, 2014.

[38] R. Zhang, L. Song, Z. Han, and B. Jiao, "Physical layer security for two-way untrusted relaying with friendly jammers," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3693–3704, 2012.

[39] R. Guillaume, S. Ludwig, A. Müller, and A. Czylwik, "Secret key generation from static channels with untrusted relays," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on*. IEEE, 2015, pp. 635–642.

[40] C. D. T. Thai, J. Lee, and T. Q. Quek, "Physical-layer secret key generation with colluding untrusted relays," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1517–1530, 2016.

[41] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM conference on Computer and communications security*. ACM, 1999, pp. 28–36.

[42] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.

[43] 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.

[44] S. Sun, G. R. MacCartney, and T. S. Rappaport, "A novel millimeter-wave channel simulator and applications for 5G wireless communications," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7.

[45] M. O. Hasna and M.-S. Alouini, "End-to-end performance of transmission systems with relays over rayleigh-fading channels," *IEEE transactions on Wireless Communications*, vol. 2, no. 6, pp. 1126–1131, 2003.

[46] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[47] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 523–540.

[48] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of computer and system sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[49] M. Bellare, S. Tessaro, and A. Vardy, "Semantic security for the wiretap channel," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 294–311.

[50] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[51] S. Primak, K. Liu, and X. Wang, "Secret key generation using physical channels with imperfect CSI," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. IEEE, 2014, pp. 1–5.

[52] M. Soleymani and H. Mahdavifar, "Distributed multi-user secret sharing," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 1141–1145.

Nasser Aldaghri (S'18) received the B.S. degree in electrical engineering from King Saud University, Riyadh, Saudi Arabia, in 2014, and the M.S. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2017, where he is currently pursuing his Ph.D. degree. His research interests include information theoretic security, physical layer security, and coding theory.

Hessam Mahdavifar (S'10, M'12) is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan Ann Arbor. He received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 2007, and the M.Sc. and the Ph.D. degrees from the University of California San Diego (UCSD), La Jolla, in 2009, and 2012, respectively, all in electrical engineering. He was with the Samsung US R&D between 2012 and 2016, in San Diego, US, as a staff research engineer.

He received the NSF career award in 2020. He also received Best Paper Award in 2015 IEEE International Conference on RFID, and the 2013 Samsung Best Paper Award. He also received two Silver Medals at International Mathematical Olympiad in 2002 and 2003, and two Gold Medals at Iran National Mathematical Olympiad in 2001 and 2002. His main area of research is coding and information theory with applications to wireless communications, storage systems, security, and privacy.