Fast Secret Key Generation in Static Environments Using Induced Randomness

Nasser Aldaghri, and Hessam Mahdavifar
Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109, USA
Email: aldaghri@umich.edu, hessam@umich.edu

Abstract*—Secret key agreement in distributed low-power networks, such as Internet of Things (IoT) networks, is a major requirement for deploying cryptographic protocols to protect the security of sensitive data. However, due to the distributed nature of such networks, the devices need to be able to generate secret keys locally from some common source of randomness. The randomness in the characteristics of the physical layer channel provides such sources, however, this can be quite limited if the devices operate in a static environment and experience static or very slow fading channel. Therefore, fast secret key generation in such environments while keeping a low complexity architecture for the network nodes, such as IoT devices, remains a challenging task. We design a low-complexity protocol for fast secret key generation in static environments. To this end, we propose to use a limited number of random bits independently generated by the legitimate parties, referred to as Alice and Bob, in combination with the fading parameter to create a common source of randomness. In the proposed protocol, Alice and Bob share their random bits over the public channel, assumed to be a fading channel, and then construct a common random sequence. Then they perform several steps for recovery from errors in the shared sequence, privacy amplification to limit the chances of a successful attack, and consistency checking by exploiting universal hash functions. We characterize the reliability of the proposed protocol and provide an upper bound on the probability of accepting a mismatched key by Alice and Bob. The eavesdropper Eve is assumed to be passive and a successful attack by her is the event of guessing the key right based on her observations. We provide an analytical upper bound, that can be numerically evaluated, on the probability of a successful attack by Eve using the cryptographic notion of semantic security. In the simulations, the proposed protocol achieves a bit generation rate of 64-96 bits/packet, bit mismatch rate of 11-24%, bit error rate of 0.005%, 50% randomness efficiency, the probability of successful attack of at most 2^{-31}, and the probability of consistency checking failure of at most 2^{-16}.

I. Introduction

Security of communication systems is becoming more of a requirement rather than an accessory especially between devices that transmit users' sensitive information, i.e., Internet of Things (IoT) devices, medical devices, etc. To ensure security of communication, a wide range of cryptographic schemes have been introduced in the literature throughout the years [1].

A common feature of many cryptographic protocols is a secure key known to legitimate parties that is used in the encryption and decryption schemes. A major challenge is then the process of key agreement between the legitimate parties or devices. Such process is either computationally expensive, e.g., Diffie-Hellman key exchange [2], or requires a secure channel, e.g., through a central entity, to securely share the key. Networks of massive connections, such as IoT networks, can be very resource-constrained which makes public key cryptography a very expensive option. Furthermore, many emerging communications and networking technologies are based on distributed protocols that do not require a central entity, thereby making the second option difficult to implement. Therefore, implementing secure key agreement protocols without the help of a central entity is a major requirement in future communication networks. In general, a key generation protocol between legitimate parties, often referred to as Alice and Bob, requires a common source of randomness which is readily available in the physical layer [3]. A fundamental feature of the physical wireless channel is its reciprocity between Alice and Bob during any coherence time [4]. Exploiting this characteristic of the physical layer is the key in many recently introduced key generation protocols [3], [5]–[7].

The randomness of the wireless channel is due to the temporal and spatial decorrelation in the environment [8]. Some specific characteristics of the wireless channel that can be used as the source of randomness are channel state information (CSI) [4], received signal strength indicator (RSSI) [3], etc. Most of secret key generation (SKG) protocols that exploit the randomness in physical layer require dynamic environments or channels in order to enable high rate key generation. In other words, the bit generation rate in such protocols depends on the movements of the surrounding environment resulting in channel fluctuation [5], [6], [9], [10]. Therefore, these protocols are not appealing in situations where the channels are static, e.g., indoor IoT networks. Various solutions have been proposed to overcome this challenge, such as exploiting beamforming in multiple-input and multiple-output (MIMO) antennas systems [11], friendly jamming by legitimate parties [7], and using artificial random noise to confuse eavesdropper [12], [13]. However, such schemes require relatively complex hardware architectures, e.g., MIMO transceivers, or assume an unconstrained source of artificial random noise at the devices, something that is rather expensive to process in order to get true random numbers [14].

In this work we aim at resolving the issue of SKG in static channels or when the channel experiences slow fading. In particular, we focus on a low-complexity solution with simple architectures that is feasible to implement in resource-constrained devices. To this end, we utilize orthogonal frequency division multiplexing (OFDM) to exploit the diversity of the physical channel across different subcarriers. Since the channel is static, a straightforward method to generate keys from the physical channel results in highly correlated keys across time. To resolve this issue, we use randomness induction by legitimate parties Alice and Bob allowing them to generate high rate secret bits if required. Instead of assuming an unconstrained source of randomness, we assume that Alice and Bob can generate a certain number of random bits that they will map to quadrature amplitude modulation (QAM) symbols and share over the public physical channel. They will combine the random symbols they have generated with the noisy symbols received from the other side to construct a shared secret. In order to implement the proposed SKG protocol we utilize universal hash functions (UHF) [15] to amplify the randomness of generated keys, and to check the consistency between Alice and Bob as described in [10]. We characterize the reliability of the proposed protocol, by providing an upper bound on the probability of accepting a

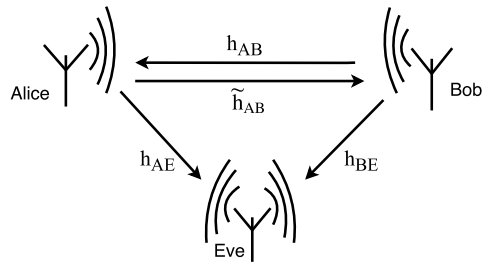

Figure 1. System model for secret key generation.

mismatched key, and the security of the protocol, by providing an upper bound on the probability of a successful attack by Eve, i.e., guessing the secret key right based on her observations, using the cryptographic notion of semantic security. To the best of our knowledge, this is the first analytical approach to quantify the resilience of a physical-layer based SKG against an eavesdropper's attack using cryptographic measures of security. A new notion to measure the *randomness efficiency* of the SKG protocol is introduced. To this end, the length of shared random sequence in each session is normalized by the total amount of randomness available to Alice and Bob. The latter can be measured in terms of number of bits or source entropy, that the legitimate parties have access to or have generated during the considered session. We argue that this is an important parameter to maximize in situations where shared randomness is not readily available to the legitimate parties, such as when the physical channel is static. The protocol is simulated and numerical results are provided for the bit generation rate (BGR), bit mismatched rate (BMR), and the key randomness based on the National Institute of Standards and Technology (NIST) test suite. Note that the proposed protocol can be also used in the scenarios with dynamic channels as long as the channel does not change during one session.

The rest of paper is organized as follows. In Section II we discuss the system model for the OFDM-based secret key generation over wireless fading channel. We explain our proposed protocol in detail in Section III and provides a bound on the reliability of the shared secret key. Then we discuss the protocol's resilience against attacks in Section IV. In Section V we evaluate our protocol and provide numerical results. Finally, the paper is concluded in Section VI.

II. System Overview

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

$$y_{\text{Bob}}(t) = x_{\text{Alice}}(t) \circledast h_{AB}(t) + n_B(t) \tag{1}$$

where $\circledast$ denotes convolution, $h_{AB}(t)$ denotes the circularly-symmetric Gaussian-distributed channel response with mean 0 and variance $\sigma_c^2/2$ 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 becomes the Rayleigh-distributed fading gain coefficient with parameter δ, i.e., $|h_{AB}| \sim \text{Rayleigh}(\delta)$, and the phase is uniformly distributed, i.e., $\text{phase}(h_{AB}) \sim U[-\pi, \pi]$. The same applies when Bob transmits $x_{\text{Bob}}(t)$ to Alice and she receives

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

Wireless channels have the property of reciprocity [16], meaning the CSI observed at Bob's end from Alice is the same as Alice's end from Bob. The reciprocity property, i.e., $h_{AB} \approx \widetilde{h}_{AB}$ is the key to many of the physical layer secret key generation protocols. Furthermore, wireless fading channels have the property of spatial and temporal decorrelation, i.e., the fading coefficients experienced at two different locations or coherence time intervals are uncorrelated [8]. These two properties are exploited to ensure reliability and security in many secret key generation protocols, including this proposed protocol. Also, Alice and Bob use OFDM and due to having different fading in each narrowband subcarrier, the rich CSI information available at each subcarrier will be utilized to generate more secret bits. Alice and Bob transmit the j-th element of the vectors $\mathbf{x}_{\text{Alice}}(t)$ and $\mathbf{x}_{\text{Bob}}(t)$ over the j-th subcarrier, receptively. The received signals at Alice and Bob can be 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., element-wise product. Using the received signals at Alice and Bob and the uniqueness of wireless channel between them they aim to construct the shared secret key.

Metrics that are often used to evaluate secret key generation protocols are: Bit generation rate (BGR), and bit mismatch rate (BMR) [6]. These metrics are defined as the number of secret bits generated per packet and the number of bits that are mismatched after quantization between Alice and Bob's sequences, denoted by $\mathbf{q}_{A,i}$ and $\mathbf{q}_{B,i}$ in Figure 2, respectively. The third metric is the randomness of the generated key sequence, denoted by $\mathbf{K}_{AB,i}$ in Figure 2, which is tested using NIST statistical test suite [17]. Also, we introduce a new parameter, referred to as *randomness efficiency* to measure the length of shared sequence normalized by the total amount of randomness available to Alice and Bob. Let R_Q denote the total amount of shared random bits. Then randomness efficiency, denoted by E_R, is defined as

$$E_R \stackrel{\text{def}}{=} \frac{R_Q}{H(S) + H(V)}, \tag{5}$$

where $H(S)$ is the entropy of Alice's source of randomness and $H(V)$ is the entropy of Bob's source of randomness.

III. Proposed Protocol

Our proposed protocol can be partitioned into four stages: induced randomness exchange, quantization, reconciliation, and privacy amplification together with consistency checking. The randomness is induced by Alice and Bob and is done at each of the N subcarriers, given 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: they perform quantization followed by reconciliation to recover from the disparities between their bit sequences. Then they have similar bit strings, with high probability. Then they use privacy amplification to increase the security of the generated streams. Finally, they check whether their keys are consistent or not. If the keys are not consistent, they re-initiate a new session. An overall description of a session of the proposed protocol

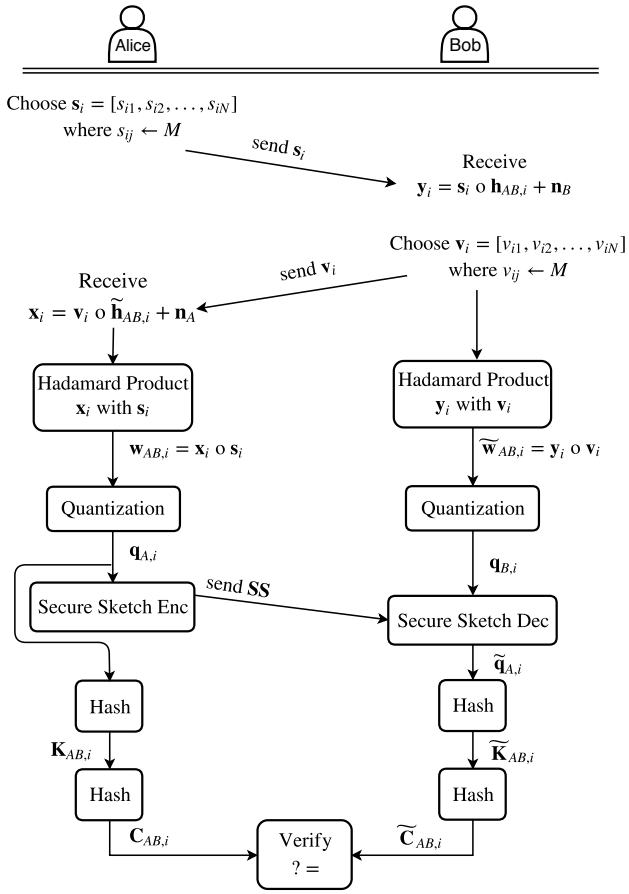

Figure 2. Protocol overview for a single session.

is shown in Figure 2. 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, we describe different stages of the protocol in detail.

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. Note that each element of such vectors is chosen uniformly at random from a set of M symbols in a M-QAM constellation. Then the symbols are multiplied by a pulse signal $p(t)$ before 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, thereby making the protocol appealing for resource-constrained devices. After the exchange of random symbols, Alice and Bob multiply what they sent with what they received. This results in a shared and secure, as it will be shown, random sequences between them that is

$$\mathbf{w}_{AB,i} = \mathbf{v}_i \circ \mathbf{s}_i \circ \mathbf{h}_{AB,i} + \text{noise} \tag{6}$$

$$\widetilde{\mathbf{w}}_{AB,i} = \mathbf{v}_i \circ \mathbf{s}_i \circ \widetilde{\mathbf{h}}_{AB,i} + \text{noise} \tag{7}$$

Quantization: In this stage, the real-valued shared sequences $\mathbf{w}_{AB,i}$ and $\widetilde{\mathbf{w}}_{AB,i}$ are turned into bit streams. We use a similar quantization method as suggested in [9]. A brief description of this quantization scheme is included next. After collecting the complex-valued measurements $\mathbf{w}_{AB}$, they are sorted as shown in Figure 3. Then Alice and Bob find the range of sorted data,

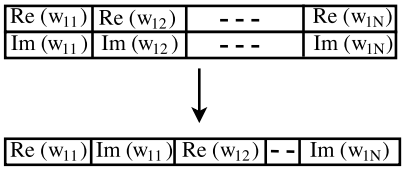

Figure 3. Sorting $\mathbf{w}_{AB}$ values before feeding them to quantizer.

which is defined as $\text{range}(\mathbf{w}_{AB}) = \max(\mathbf{w}_{AB}) - \min(\mathbf{w}_{AB})$. They then find the quantization resolution δ, and use it to compute the number of uniform intervals $\Delta = 2^{\delta}$, then assign a Gray-code sequences to each interval. Finally, they match each sample to its bit sequence based on the interval it lies in.

Reconciliation: The aim of this stage is to mitigate disagreements between Alice and Bob's bit strings after quantization. To this end, various methods, such as error-correcting codes, can be used. In our protocol, we use secure sketch, introduced by [18], with convolutional codes. The reason to pick convolutional codes is due to the simplicity of the encoding process using shift registers and the decoding process using Viterbi decoder [19]. The output of the encoder has the same length as $\mathbf{q}_A$. Once the quantized strings $\mathbf{q}_A$ and $\widetilde{\mathbf{q}}_B$ are available, Alice chooses a bit string $\mathbf{r}$ uniformly at random and encodes it using convolutional encoder to get $\text{Enc}(\mathbf{r})$, which is of the same length as $\mathbf{q}_A$. Then she computes

$$\mathbf{SS} = \mathbf{q}_A \oplus \text{Enc}(\mathbf{r}) \tag{8}$$

and transmits this reliably, using another appropriate level of channel coding and modulation scheme, to Bob. We exclude the effect of physical layer on this transmission and assume that Bob successfully receives **SS**. Then, he takes the XOR of **SS** and $\mathbf{q}_B$, feeds it to the Viterbi decoder to get $\widetilde{\mathbf{r}}$, then re-encodes $\widetilde{\mathbf{r}}$ to get $\text{Enc}(\widetilde{\mathbf{r}})$. Bob gets the final string as

$$\begin{aligned} \widetilde{\mathbf{q}}_A &= \mathbf{SS} \oplus \text{Enc}(\text{Dec}(\mathbf{SS} \oplus \mathbf{q}_B)) \\ &= \mathbf{SS} \oplus \text{Enc}(\widetilde{\mathbf{r}}). \end{aligned} \tag{9}$$

Remark: The error correction capability of the convolutional code depends on the underlying rate of the code. This rate is selected according to channel conditions when implementing the protocol. If Alice and Bob do not have access to estimation of CSI, they can start from some initial code rate and then reduce it accordingly if they observe several consecutive unsuccessful attempts of the protocol.

Privacy Amplification and Consistency Checking: Since some information about the shared key is leaked to Eve during the exchange of random symbols, 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 1: [15] The 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 2^m is called universal if for any two inputs $x, y \in U$ such that $x \neq y$, we have

$$\Pr_{h \leftarrow H}(h(x) = h(y) | x \neq y) \leqslant \frac{1}{2^m}. \tag{10}$$

We also use UHFs to check consistency between keys generated by Alice and Bob, without leaking any information to Eve, as suggested in [10].

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 string $\mathbf{q}$ of length n (This is $\mathbf{q}_A$ for Alice and $\widetilde{\mathbf{q}}_A$ for Bob). For simplicity we assume that n is an even multiple of m. We divide $\mathbf{q}$ into two strings of equal length $\mathbf{q} = \mathbf{q}_1 \parallel \mathbf{q}_2$. Then $\mathbf{q}_1$ is used to choose h from H, and $\mathbf{q}_2$ is used as the input to the hash function. Next, a well-known construction of UHF is described that we use in our protocol [15]. First, the largest prime p of m bits, i.e. $2^{m-1} < p < 2^m$, is chosen where m is the length of the output bit sequence (such primes always exist by Bertrand's postulate). Then for $i = 1, 2$, we divide $\mathbf{q}_i$ into l parts q_{ij}, for $j = 1, 2, \ldots, l$, where the length of q_{ij} is less than or equal m. For ease of notation, let q_{ij} also denote the number with the binary representation q_{ij}. Finally, we evaluate the following summation:

$$h_{\mathbf{q}_1}(\mathbf{q}_2) = \sum_{j=1}^{l} q_{1j} q_{2j} \bmod p \tag{11}$$

Note that in our protocol, $\mathbf{q}$ is random, and will be different at each session of the protocol due to the randomness of $\mathbf{s}_i$ and $\mathbf{v}_i$. The output of this step are the keys $\mathbf{K}_{AB}$ for Alice and $\widetilde{\mathbf{K}}_{AB}$ for Bob. Since we have done the reconciliation step, the two keys will match with high probability. Note that the probability of two keys not matching depends on the error correction capability of the underlying code in the secure sketch step. It is very crucial that Alice and Bob verify whether their keys match or not. To this end, they hash $\mathbf{K}_{AB}$ and $\widetilde{\mathbf{K}}_{AB}$ again similar to the described process above by splitting them into two parts and using the first part as the hash function and so on. Then they get check sequences $\mathbf{C}_A$ and $\mathbf{C}_B$ and check whether they match or not. It is worth noting that the length of the check sequences, $\mathbf{C}_A$ and $\mathbf{C}_B$, is half of the length of the key.

Theorem 1: The probability of accepting a mismatched key of length m bits as consistent by the described protocol is upper bounded as follows:

$$\Pr(\mathbf{C}_A = \mathbf{C}_B | \mathbf{K}_{AB} \neq \widetilde{\mathbf{K}}_{AB}) \leqslant \frac{1}{2^{m/2}} \tag{12}$$

Proof: This follows directly from the definition of universal hash functions, using equation (10) where the output hash table size is $2^{m/2}$. □

IV. Attacker Model and Resilience of Proposed Protocol

In this section, we discuss attack strategies that Eve can carry against the proposed scheme and the security of the protocol against such attacks. Note that Eve is assumed to be a passive attacker. Eve's best strategy is to acquire $\mathbf{s}_i, \mathbf{v}_i$ and $\mathbf{h}_{AB,i}$. When Alice and Bob exchange signals, Eve receives

$$\mathbf{e}_{1i} = \mathbf{s}_i \circ \mathbf{h}_{AE,i} + \mathbf{n}_{E1} \tag{13}$$

$$\mathbf{e}_{2i} = \mathbf{v}_i \circ \mathbf{h}_{BE,i} + \mathbf{n}_{E2} \tag{14}$$

Even if Eve is able to estimate both $\mathbf{v}_i$ and $\mathbf{s}_i$ from her observations $\mathbf{e}_{1i}$ and $\mathbf{e}_{1i}$, perfectly, she still needs to know $\mathbf{h}_{AB,i}$ at all different subcarriers in order to get $\mathbf{w}_{AB}$. Luckily, this is not possible for Eve due to the spatial decorrelation. As shown in [8], the correlation coefficient ρ of the channel fading coefficients at locations separated by distance d is calculated as follows,

$$\rho = [J_0(kd)]^2 \tag{15}$$

where $J_0(.)$ is the Bessel function of first kind, and k is the wavenumber. Therefore, if the distance between Alice or Bob and Eve is larger than half of a 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 $\mathbf{v}_i$ and $\mathbf{s}_i$, this mutual information is equal to the mutual information between $\mathbf{h}_{AB,i}$ and $(\mathbf{h}_{AE,i}, \mathbf{h}_{BE,i})$. One can assume that Eve is closer to Bob than Alice, and hence only considers the mutual information between $\mathbf{h}_{AB,i}$ and $\mathbf{h}_{AE,i}$ as the dominating term. This can be calculated in each subcarrier as stated in the next theorem.

Theorem 2: Let h_{Bk} and h_{Ek} denote the fading coefficients of Bob's and Eve's channel at k-th subcarrier. Also, let ρ denote the the correlation coefficient between h_{Bk} and h_{Ek}, as calculated in (15). Then the mutual information between h_{Bk} and h_{Ek} is given by

$$I(h_{Bk}; h_{Ek}) = -\frac{1}{2} \log(1 - 2\rho^2 + \rho^4) \text{ bits} \tag{16}$$

Proof: We have $h_{Bk} = h_{Bk,I} + j h_{Bk,Q}$, and $h_{Ek} = h_{Ek,I} + j h_{Ek,Q}$. $h_{Bk,I}, h_{Bk,Q}$ 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 are correlated with ρ, and the imaginary parts are also correlated random variables with ρ. We have the following covariance matrices:

$$\Sigma_1 = \begin{bmatrix} \sigma_B^2/2 & 0 \\ 0 & \sigma_B^2/2 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} \sigma_E^2/2 & 0 \\ 0 & \sigma_E^2/2 \end{bmatrix} \tag{17}$$

$$\Sigma_3 = \begin{bmatrix} \sigma_B^2/2 & 0 & \frac{\rho\sigma_B\sigma_E}{2} & 0 \\ 0 & \sigma_B^2/2 & 0 & \frac{\rho\sigma_B\sigma_E}{2} \\ \frac{\rho\sigma_B\sigma_E}{2} & 0 & \sigma_E^2/2 & 0 \\ 0 & \frac{\rho\sigma_B\sigma_E}{2} & 0 & \sigma_E^2/2 \end{bmatrix} \tag{18}$$

Then we have the following series of equalities:

$$\begin{aligned} I(h_{Bk}; h_{Ek}) &= I(h_{Bk,I} + j h_{Bk,Q}; h_{Ek,I} + j h_{Ek,Q}) \\ &\overset{(a)}{=} I(h_{Bk,I}, h_{Bk,Q}; h_{Ek,I}, h_{Ek,Q}) \\ &\overset{(b)}{=} H_d(h_{Bk,I}, h_{Bk,Q}) + H_d(h_{Ek,I}, h_{Ek,Q}) \\ &\quad - H_d(h_{Bk,I}, h_{Bk,Q}, h_{Ek,I}, h_{Ek,Q}) \\ &\overset{(c)}{=} \frac{1}{2} \log(\det(2\pi e \Sigma_1)) + \frac{1}{2} \log(\det(2\pi e \Sigma_2)) \\ &\quad - \frac{1}{2} \log(\det(2\pi e \Sigma_3)) \\ &\overset{(d)}{=} \log(\pi e \sigma_B^2) + \log(\pi e \sigma_E^2) \\ &\quad - \log((\pi e \sigma_B \sigma_E)^2 \sqrt{1 - 2\rho^2 + \rho^4}) \\ &\overset{(e)}{=} -\frac{1}{2} \log(1 - 2\rho^2 + \rho^4) \end{aligned} \tag{19}$$

where:
(a) holds because it is a one-to-one mapping;
(b) is the expansion of the mutual information expression in terms of differential entropy;
(c) holds by the differential entropy of multivariate Gaussian-distributed 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), (e) are simplification steps. □

Clearly, as ρ goes to zero, the mutual information also goes to zero. For example, if the distance between Bob and Eve is $\lambda/2$, then the correlation coefficient is $\rho = 0.09$ and the resulting mutual information is $I(h_{Bk}, h_{Ek}) = 0.01$ bits.

The next question, which applies to any work on physical layer security that uses information-theoretic measures of security, is how to quantitatively characterize the chances of a successful 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 [20]. 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 [21]. In particular, by [21, Theorem 5] and assuming $I(h_{Bk}, h_{Ek}) = 0.01$ bits, as calculated above, the probability that Eve can guess the four shared random bits in subcarrier k, given her observations in this subcarrier, is increased by at most $\sqrt{2 \times I(h_{Bk}, h_{Ek})} \approx 0.14$ comparing to the case where she does not have any observation. Therefore, her chances of successfully guessing these four bits is roughly $\frac{1}{2^4} + 0.14 \approx 0.2$. The probability that Eve can recover all the unique shared randomness over all subcarriers is then roughly $0.2^{16} \approx 2^{-37}$, assuming that we have 16 subcarriers. If Eve can not recover all the shared randomness, the probability that she can guess the secret key right, by the property of hash functions in the privacy amplification part of our protocol, is at most $\frac{1}{2^{n/2}}$, where n is the length of shared randomness. In our numerical evaluations in the next section, $n = 64$. Therefore, by union bound, the overall probability of a successful attack by Eve given such parameters is at most $2^{-32} + 2^{-37} < 2^{-31}$.

V. Numerical Results

In this section the proposed protocol is evaluated in detail using the metrics discussed in Section II. In our simulation we assume that Eve is close to Bob, but at least more than $\lambda/2$ away from Bob in order to observe an almost uncorrelated fading channel. Signal-to-noise ratio (SNR) at Alice, Bob, and Eve is around 17 dB. Also, $M = 16$ and 16-QAM symbols are transmitted. We do the randomness induction between Alice and Bob over 16 OFDM subcarriers. The quantization is done with $\delta = 2$, i.e., 2 bits per sample. Finally, we do the secure sketch followed by hashing and then by consistency checking, as discussed in Section III. To increase the bit error rate at Eve, we XOR the 4 generated blocks of 32 secret bits together to obtain the final key. It is shown next that Eve will have an average bit error rate (BER) around 50%, while the BER between Alice and Bob is negligible.

Bit Generation Rate: For static channels, the rate of the bit generation is often very low due to the constant channel state information over a long period of time. This problem is resolved in our protocol using the randomness induction. In the simulations we get a rate of 64 bits/packet and 96 bits/packet for 2-bit and 3-bit quantization, respectively. This is considered a high rate and is close to one of the latest SKG protocols for dynamic channels described in [6]. In [6] secret rates of 60-90 bits/packet are obtained. Note that this is derived using 30 subcarriers, while we use 16 subcarriers.

Bit Mismatch Rate: When using 2-bit quantization level a BMR of 11% and 24% is observed in static and dynamic channels, respectively. Comparing this to the BMR between Alice and Bob in [6] of around 5% for both static and dynamic environments. The appropriate choice of convolution codes according to the BMR helps mitigate such errors. For Eve, the BMR is 53% and 65% in static and dynamic channels, respectively. In the simulation, 2^{20} bits are generated, and it is observed that the average bit error rate (BER), which is defined as the error rate between Alice and Bob at the final agreed upon key, is about 5×10^{-5} in both static and dynamic channels. It is also worth noting that the probability of failure in our simulation for a session of the protocol is 1.3×10^{-5}, which is less than the upper bound provided in theorem 1, that is 2^{-16}. On the other hand, as Figure 4 shows the BER at Eve is between 30% and 70% and the average BER is 50%. The CDF curves for both static and dynamic channels are matched. Table I shows a summary of the results.

Table I
Summary of BMR and BER in Static and Dynamic Channels

Channel	BMR_{AB}	BMR_{AE}	BER_{AB}	BER_{AE}
Static	11%	53%	0.005%	50%
Dynamic	24%	65%	0.005%	50%

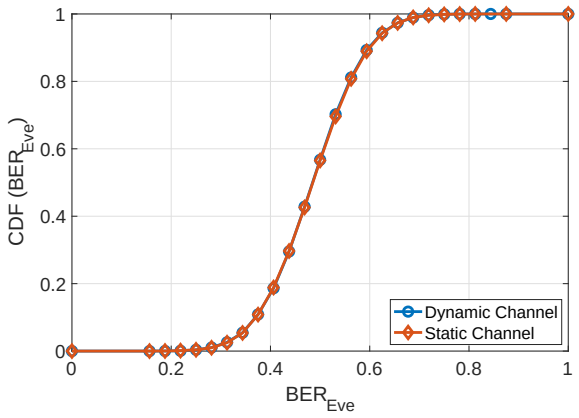

Figure 4. Cumulative distribution function of the BER at Eve.

Randomness: The randomness of the generated sequence is examined using the NIST statistical test suite [17]. The suite consists of 15 tests and generates a *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 test the randomness of keys generated by the proposed protocol in two cases: (A) static channel and (B) dynamic channel. For the static channels the channel gain is considered constant, while in dynamic channels it is considered random according to complex Gaussian distribution. We run the protocol to generate a sequence of length 2^{20} bits and feed it to the test suite. Since the sequence passes all the tests as shown in Table II, the sequence that was generated by the proposed protocol is considered random with 99% confidence.

Remark: In our simulation, the number of sessions or trials, i.e., the number of protocol runs required to get final key agreement between Alice and Bob's XORed strings, was on average 8 sessions in static channels and 10 sessions in dynamic channels, i.e., 2 and 2.5 sessions for each 32-bit block in static and dynamic channels respectively. Figure 5 shows the cumulative distribution function (CDF) of the number of sessions in both static and dynamic channels. Figure 6 shows the average number of sessions versus the signal-to-noise ratio.

Randomness Efficiency: This is computed according to (5). Alice and Bob randomly choose a bit string of length 64, therefore, $H(S) = H(V) = 64$. Note that the length of the bit string after reconciliation is 64, which is considered random due to the statistical test results shown above. Therefore, we have $R_Q = 64$. This implies that $E_R = 50\%$. Roughly speaking, the

Table II
NIST STATISTICAL TEST RESULTS

Test	A	B
Monobit	0.8746	0.7300
Frequency Block	0.7594	0.5811
Runs	0.9559	0.0103
Longest Run of Ones	0.3405	0.3535
Binary Matrix Rank	0.6703	0.9600
DFT	0.7776	0.8365
Non-Overlapping Template Matching	1	1
Overlapping Template Matching	0.9800	0.2158
Maurer's "Universal Statistical"	0.9994	0.9988
Linear Complexity	0.7629	0.2360
Serial	0.2080	0.1793
Approximate Entropy	0.5803	0.1772
Cumulative Sums	0.9670	0.5908
Random Excursion	0.0122	0.1969
Random Excursion Variant	0.0752	0.0142

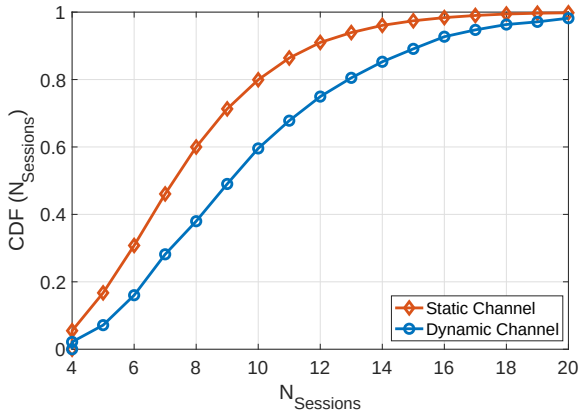

Figure 5. Cumulative distribution function of the number of sessions for 17 dB SNR.

other 50% of available randomness is used to provide security. The exact trade-off between randomness efficiency and security is an interesting problem.

VI. CONCLUSION AND FUTURE WORK

We proposed a low-complexity solution for high rate SKG over static channels. A method based on induced randomness is introduced to increase the number of random samples, which are then used to generate secret keys. Also, numerical results for the performance of the proposed protocol are shown. It is shown that the proposed protocol has a superior performance while keeping the complexity of the different stages of the protocol low. It provides a high rate compared to CSI-based SKG protocols, even those designed for dynamic channels, while keeping the failure rate negligible. Furthermore, analytical bounds for the reliability and security of the proposed protocol are provided. In the numerical results, we show that the sequences generated by the protocol pass the randomness tests provided by NIST.

A main feature of the physical channel exploited in this work and many other related works on physical layer secret key generation is the spatial decorrelation of the wireless fading channel. A direction for future work is to design protocols that are resilient to attacks by the eavesdropper even if she can obtain partial or full information about the fading coefficients of the channel between Alice and Bob. This becomes relevant especially when the eavesdropper is close to one of the legitimate parties at low frequency bands.

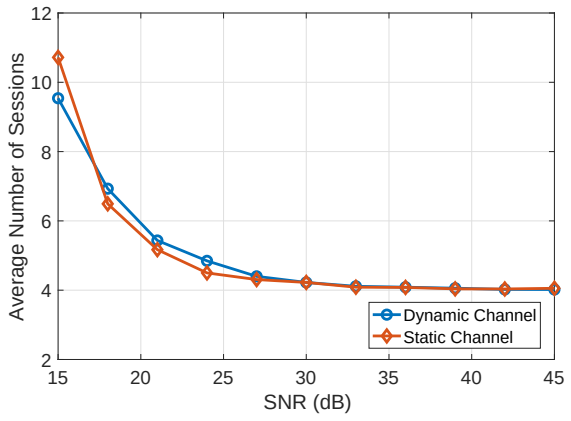

Figure 6. Average number of sessions vs SNR.

REFERENCES

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

[2] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[3] 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.

[4] 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.

[5] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3776–3784, 2005.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

[16] H. Koorapaty, A. A. Hassan, and S. Chennakeshu, "Secure information transmission for mobile radio," *IEEE Communications Letters*, vol. 4, no. 2, pp. 52–55, 2000.

[17] 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.

[18] 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.

[19] 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.

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

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