# How to Test the Randomness from the Wireless Channel for Security?

Zhe Qu, *Student Member, IEEE,* Shangqing Zhao, *Student Member, IEEE,* Jie Xu, *Member, IEEE,* Zhuo Lu, *Senior Member, IEEE,* and Yao Liu, *Senior Member, IEEE*

***Abstract*—We revisit the traditional framework of wireless secret key generation, where two parties leverage the wireless channel randomness to establish a secret key. The essence in the framework is to quantify channel randomness into bit sequences for key generation. Conducting randomness tests on such bit sequences has been a common practice to provide the confidence to validate whether they are random. Interestingly, despite different settings in the tests, existing studies interpret the results the same: passing tests means that the bit sequences are indeed random.**

**In this paper, we investigate how to properly test the wireless channel randomness to ensure enough security strength and key generation efficiency. In particular, we define an adversary model that leverages the imperfect randomness of the wireless channel to search the generated key, and create a guideline to set up randomness testing and privacy amplification to eliminate security loss and achieve efficient key generation rate. We use theoretical analysis and comprehensive experiments to reveal that common practice misuses randomness testing and privacy amplification: (i) no security insurance of key strength, (ii) low efficiency of key generation rate. After revision by our guideline, security loss can be eliminated and key generation rate can be increased significantly.**

***Index Terms*—Wireless key generation; Randomness test; Security; Maximum likelihood tree search.**

## I. Introduction

Leveraging the wireless channel randomness has become one of the fundamental approaches to build low-cost security designs for emerging wireless applications, such as radio frequency identification (RFID) [1] and Internet of Things (IoT) [2]. In particular, two communication parties, Alice and Bob, can use the random yet reciprocal wireless channel measurements, such as received signal strength information (RSSI) [3]–[5], channel state information (CSI) [6] and phase shifts [7], [8], to generate a common secret sequence to build a security design, such as secret key generation [2]–[14], secure communication [15], [16] and user authentication [17]. Then, Alice and Bob can enter the cryptographic domain [3]–[5] and use information reconciliation [18] and privacy amplification [19], [20] to compress their respective bit sequences. The framework is considered as secure enough because a third-party eavesdropper Eve is expected to gain little information about the shared secret if she is more than half the wavelength away from them because of wireless fading [9].

To evaluate the security of such a design, existing studies [2]–[14] adopt the NIST randomness tests [21] to evaluate whether the underlying secret bit sequences generated from the wireless channel exhibit randomness properties, which has become the common practice. However, many of them [3], [4], [6], [9], [11] simply adopt the default NIST choices to set up a randomness test and consider successfully passing the test as a demonstration of security strength. Although passing a randomness test may hint that there is no major flaw in the design, it may not provide a guaranteed level of security stength. It becomes necessary to quantitatively understand the security impact of setting up randomness testing for the designs extracting random secrets from the wireless channel.

At the same time, efficiency is always another important design aspect. In secret key generation [3], [4], [11], efficiency refers to the key generation rate, which depends on the strictness of the randomness test and the sequence compression rate for privacy amplification. However, there is no theoretical analysis in the literature on how to guarantee efficient secret key generation. Therefore, it is necessary to provide a design guideline for secrete key generation to ensure both security and efficiency.

In this paper, we ask a fundamental question: *How to properly set up statistical randomness tests for testing the wireless channel for both security and efficiency?*

In current study, the mismatched assumption between practical channel coherence and theoretical memoryless assumption leads to a gray area in realistic wireless key establishment applications. Each test in NIST assumes that the bit sequence is IID, but in practical wireless communication framework, the RSSI or CSI cannot be considered as IID [22], [23]. Despite the correlation, it can also pass the NIST tests by choosing $\alpha = 0.01$ [3], [4], [9] and $0.05$ [6], [11].

Before we verify the security level of any secret key generation design, the first step is to clearly define an adversary model. A formal adversary model will enable thorough security analysis and evaluation, but it has not been systematically studied in the literature. To this end, we study how an adversary can defeat a security design by taking advantage of a potential defect of the wireless channel, which is unpredictable with unknown ground truth. In particular, it is never known whether a channel can indeed yield independently random sequences for secret key generation. Therefore, under

Zhe Qu, Shangqing Zhao and Zhuo Lu are with the Department of Electrical Engineering, University of South Florida, Tampa, FL, 33620, USA.
Emails: {zhequ, shangqing, zhuolu}@usf.edu

Jie Xu is with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL, 33146, USA.
Email: jiexu@miami.edu

Yao Liu is with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL, 33620, USA.
Email: yliu@cse.usf.edu

the assumption that the generated secret sequence is indeed statistically correlated, an adversary can search for it from the most-likely sequence candidate to the least-likely one, as opposed to random guessing. This strategy, called Maximum Likelihood Tree Search (MLTS), is considered for the security evaluation of a wireless channel randomness based design.

With the understanding of the adversary's capability, we propose a new design guideline for randomness testing, which involves solving an optimization problem that maximizes key generation efficiency under guaranteed security. In particular, we derive a mathematical formula for choosing the proper P-value threshold of nine different NIST tests. To our best knowledge, our design guideline is the first theoretical framework for wireless channel randomness based secret key generation, where the randomness test parameters are not empirically set. We note that rather than designing a new secret key generation method from the wireless channel, our focus is on how to properly setup the randomness tests. We conduct real-world experiments to validate the analysis, and incorporate our design guideline into seven popular key generation methods and compare the difference. The results show that (i) using our design guideline, these methods achieve zero security loss in various experiment scenarios; (ii) the key generation efficiency can be significantly improved; (iii) Our design guideline is more adaptive for generating different bit length of key sequences. In summary, the main contributions of this paper are as follows:

1) We introduce the MLTS strategy to formalize the security analysis and evaluation of wireless channel randomness based designs.
2) We propose a new design guideline on how to properly setup the randomness test parameter to eliminate security loss and achieve high efficiency.
3) We conduct the experiments in practical environments to show the improvement by our design guideline compared with existing secret key generation studies.

## II. Background and Preliminaries

In this section, we briefly introduce the background of extracting secret from the wireless channel, and then formalize the framework of secret key generation. To this end, we discuss the scenario and assumptions in this paper.

### A. Extracting Random Secrets from Wireless Channels

Traditional cryptographic mechanisms (e.g., Diffie-Hellman and RSA [24]) reply on establishing computational difficulties for an adversary to achieve the goal of security. In wireless, mobile or IoT domains [2], [6], many wireless security designs have been proposed to leverage the reciprocal and random properties of wireless channel measurements (e.g., RSSI and phase shifts) to generate a common secret sequence between Alice and Bob. In many studies [2]–[5], [7]–[14], [25]–[29], such a sequence is directly used as the secret key for the secure communication between Alice and Bob. In this paper, we use secret key generation as our main application scenario to study how to test the wireless channel randomness for security, since it is the most representative study of security designs leveraging wireless channel randomness.

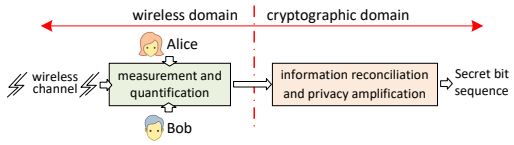


Fig. 1: Typical framework of random secret key generation from wireless channel between Alice and Bob.

Figure 1 shows that a typical framework for Alice and Bob extracting a common secret sequence from the wireless channel. The framework consists of design components in both wireless domain and cryptographic domain. In the wireless domain (shown on the left-hand side of Figure 1), Alice and Bob keep measuring the wireless channel response, such as measuring the RSSI, CSI or phase shifts between them, and then quantify the measurements into bits [2]–[4], [8], [9], [25]. Because of the reciprocal property of the wireless channel, their measurements are likely to be the same from the channel between them, and accordingly their quantified bits sequences should also be likely the same.

Then, Alice and Bob can enter the cryptographic domain [3], [4], [6] as shown on the right-hand side of Figure 1 and use information reconciliation [18] and privacy amplification [19], [20] to compress their respective bit sequences (with low per-bit entropy) into the same short sequence (with per-bit entropy expected to be near 1).

### B. Formalizing the Framework of Secret Key Generation

For the framework to extract random bit sequences from the wireless channel, there are two major components between Alice and Bob: the wireless domain design and the cryptographic domain design, which are formally modeled in the following.

**Definition 1** (Secret Bit Sequence Extraction Models)**.**

- *The wireless domain design is a function*

$$f_w : \Omega_D \rightarrow \{0,1\}^L \tag{1}$$

*mapping a random channel property (e.g., RSSI or phase shifts) in the continuous domain $\Omega_D$ during a time duration $D$ to a binary bit sequence in $\{0,1\}^L$, which denotes the set of all bit sequences with length $L$.*

- *The cryptographic domain design is a function*

$$f_c : \{0,1\}^L \rightarrow \{0,1\}^M \tag{2}$$

*mapping a binary bit sequence with length $L$ to a new sequence with shorter length $M \leq L$, in which the correlation among bits is minimized close to 0 by privacy amplification. When $M = L$, there is no cryptographic domain in a secret key generation design [9], [14], we simply set function $f_c$ as $f_c(x) = x$ for any input $x$.*

- *A statistical randomness test is a function*

$$T : \{0,1\}^* \rightarrow \{\{Accept\ H_0\}, \{Accept\ H_1\}\}, \tag{3}$$

*where $\{0,1\}^*$ denotes the set of bit sequences with any length (e.g., length of $L$ or $M$), $H_0$ and $H_1$ are null*

*and alternative hypotheses denoting the events that the randomness test succeeds and fails, respectively. The objective of Alice and Bob is to leverage the random channel property between them, denoted by $\omega_D \in \Omega_D$[1] during time period $D$, to compute a bit sequence*

$$K_D = f_c(f_w(\omega_D)) \in \{0,1\}^M \tag{4}$$

*for their security design purpose.*

In the extraction models, there is no evaluation that the bit sequence $K_D$ is sufficient for the security purpose. Therefore, security evaluation is another critical component for any wireless channel randomness based security design. To this end, NIST statistical randomness test suite [21] is widely adopted as a common practice in the literature [3], [8], [9] to test whether the generated bit sequence is random for the security purpose.

### C. Testing Randomness from Wireless Channels

The procedure of a randomness test in the NIST test suite [21] to test the bit sequences extracted from the wireless channel is straightforward: for a bit sequence $X$ quantified from the wireless channel, compute the statistics of $X$ based on a particular test, called P-value, and compare this P-value with a threshold $\alpha$. The test succeeds if the P-value is greater than $\alpha$, and fails otherwise.

For Alice and Bob, failing the NIST tests indicates that the wireless channel measurement does not have enough randomness [3]. They have to wait for a better channel condition or adjust their design parameters and then test again. Thus, randomness tests serve a critical role in evaluating the security of a design leveraging wireless channel randomness.

At first glance, it seems perfectly fine to use randomness tests for security evaluation because they are recommended for cryptographic use. But the key question here is not why, but how to use them to test the wireless channel randomness for security? We notice that existing studies adopt statistical randomness tests in different ways for security evaluations. Particularly, two major discrepancies exist in the literature.

1) Where to set the randomness test? There indeed exists a discrepancy in the literature to place the randomness test: a large number of designs [3]–[5], [12] choose to test the bit sequences at Position 2, and other designs test at Position 1 [9], [11], [14].
2) How to choose a critical parameter, the P-value threshold $\alpha$, in randomness tests? The value of $\alpha$ represents the confidence level of the test output. We notice that $\alpha = 1\%$ is dominantly adopted in existing studies [3]–[5], [9], [14] according to the NIST test suite [21]. However, some studies also choose $\alpha = 5\%$ for the tests, for example, [6] tests 200-bit sequences generated from the wireless channel between mobile devices.

[1]Alice and Bob may not observe exactly the same $\omega_D$ in practice because of noise and interference. In this regard, denote by $\omega_D^A$ and $\omega_D^B$ Alice's and Bob's observations respectively. Robust wireless domain design aims to achieve $f_w(\omega_D^A) = f_w(\omega_D^B)$ and information reconciliation also ensures $f_c(f_w(\omega_D^A)) = f_c(f_w(\omega_D^B))$. So $\omega_D^A \neq \omega_T^B$ does not affect our security analysis. For the sake of simple notation, we let $\omega_T = \omega_T^A = \omega_T^B$.

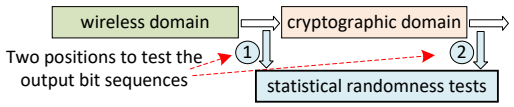


Fig. 2: The use of statistical testing at different positions in secret key generation from the wireless channel.

These observations in fact reveal that despite the advance on efficiently quantifying the wireless channel randomness (e.g., RSSI or CSI) into bit sequences, the common practice of using randomness testing exhibits discrepant setups for security evaluation. Are these setups equally secure, or secure enough for a particular application? As a result, the focus of this paper is to design a rigorous mechanism to understand how to properly use randomness tests for security evaluation of these security designs quantifying wireless channel randomness.

## III. Problem Formulation and Research Statement

We first formalize the models in extracting secret bit sequences from the wireless channel, then identify research challenges and propose the guideline for setting up statistical randomness tests.

### A. Formalizing the Role of Statistical Randomness Testing for Security

It is clear in Definition 1 that extracting a secret bit sequence from (4) does not rely on the statistical randomness test $T$ in (3). The role of $T$ is to ensure *security by denial*: if the bit sequence from the channel fails the test $T$, then the channel randomness is not sufficient for the security purpose.

However, a randomness test can be set up in many different ways (e.g., varying setups observed in the literature: [3], [4], [9] vs [6], [11]). Due to the fact that a randomness test only considers some specific parts to evaluate the degree of randomness for a bit sequence, the bit sequence can always pass a randomness test as long as its construction is biased toward the test. Blindly setting up the randomness tests provides no guarantee of security. We have to rethink about how to quantify the extent to which the security by denial via a randomness test meets the security goal of Alice and Bob, i.e., obtaining a secret bit sequence $K_D$ of length $M$ in (4).

To this end, we need to design $T$ in two steps: (1) Designing a technical adversary model Eve against Alice and Bob. However, there is no such model proposed in the literature, which makes formal analysis for wireless randomness based security difficult. (2) Defining Eve's attack success probability as a function of the randomness test. In this way, we can quantitatively measure the benefit of security by denial via a randomness test and properly set the test.

If the two steps are in place, we are able to evaluate whether a randomness test is properly set up for protecting the system security. Specifically, we aim to compare Eve's strategy under the randomness test with the benchmark random guess (RG) strategy in terms of the success probability, and set up the randomness test such that

$$\mathbb{P}(\text{Eve succeeds}) \leq \mathbb{P}(\text{RG succeeds}). \tag{5}$$

In other words, we must set up the randomness test such that Eve's attack is no better than RG. We also define the security loss due to the randomness testing as

$$L_{\text{security}} = \log_2(\mathbb{P}(\text{Eve succeeds})/\mathbb{P}(\text{RG succeeds})). \quad (6)$$

For example, if $\mathbb{P}(\text{Eve succeeds}) = 2^{-80}$ and $\mathbb{P}(\text{RG succeeds}) = 2^{-120}$, the security loss $L_{\text{security}}$ is computed as 40, which is the difference between the exponents in the two probabilities. In general, the security loss $L_{\text{security}}$ can have a negative value for some naive attack strategy (e.g., Eve always tries a fixed guess every time, which is even worse than RG). In this paper, we only consider non-trivial cases with $L_{\text{security}} \geq 0$ (i.e., a security loss is non-negative).

### B. Formalizing the Role of Statistical Randomness Testing for Efficiency

In Section III-A, we formalize the role of statistical randomness testing from the security perspective. It is also worth noting that efficiency for secret key generation is another important factor to consider. Note that the efficiency comes from two aspects: randomness test $T$ (i.e., the probability of bit sequence can be accepted or rejected) and privacy amplification rate $R_{\text{privacy}}$ (i.e., the compressed rate $M/L$). If statistical randomness test $T$ is set too strict, it will be difficult to generate wireless secrets during a short time period because $T$ rejects the channel samples too many times. On the other hand, privacy amplification allows the input of a correlated bit sequence and compresses it into a short sequence with higher per-bit entropy. A higher $R_{\text{privacy}}$ can loose the design of the test $T$, but at the same time reduce the efficiency because more bits are compressed, which indicates more bits extracted from wireless channel are discarded. In this paper, we only aim to achieve high efficiency by controlling $T$ and $R_{\text{privacy}}$. Thus, we consider the efficiency $E$ as our evaluation and formally define it as

$$E = \mathbb{P}(T \text{ accepts } H_0) \cdot R_{\text{privacy}}, \quad (7)$$

and the efficiency loss is defined as $L_{\text{efficiency}} = 1 - E$. Note that there are two situations of placing randomness test $T$ in the literature: if we place randomness test $T$ at position 1, $\mathbb{P}(T \text{ accepts } H_0) = \mathbb{P}(T(f_w(\omega_D)) \text{ accepts } H_0)$. If $T$ is at position 2, $\mathbb{P}(T \text{ accepts } H_0) = \mathbb{P}(T(K_D) \text{ accepts } H_0)$. For simplicity, we generalize these two situations into one formula $\mathbb{P}(T \text{ accepts } H_0)$. With security loss $L_{\text{security}}$ and efficiency loss $L_{\text{efficiency}}$, we can quantitatively evaluate a wireless secret bit generation design in terms of its security and efficiency. To properly set up the randomness testing, we must ensure that its security loss $L_{\text{security}}$ is zero under an adversary model and at the same time we also need to maximize its efficiency $E$ or, equivalently, minimize the efficiency loss $L_{\text{efficiency}}$.

## IV. Formal Adversary Model

With the definition of security loss and efficiency, our next goal is to define a formal adversary model to measure the security loss. It is well known that the wireless channel response is statistically correlated over time, but two channel response samples with interval larger than the coherence time are approximately independent [30]. This approximate independence assumption has been widely used for wireless communication performance analysis and evaluation. However, whether this assumption is able to serve a basis for security design is not fully investigated. We aim at defining an adversary model that takes advantage of imperfect channel independence to launch attacks targeting the secret random bit generation framework.

We first present the scenarios and assumptions, then model the secret bit extraction from the wireless channel, and finally propose and analyze the adversary model.

### A. Scenarios and Assumptions

We consider a wireless channel randomness based design scenario shown in Figure 2. We assume that all design specifications and parameters in the wireless domain (e.g., bandwidth, carrier frequency and quantization methods) and the cryptographic domain (e.g., cryptographic methods) in Figure 2 are known to the public. An adversary Eve can hear all communications between Alice and Bob, but cannot access Alice's or Bob's antenna. Therefore, she cannot directly measure the accurate channel response between Alice and Bob. We assume that Eve can neither actively affect the wireless channel between Alice and Bob, nor modify the content of any communication.

We also assume that Eve has a powerful yet finite computational capability. This enables Eve to perform intensive computations, leveraging the imperfect randomness of the wireless channel measurements, to search for the secret between Alice and Bob. Such an assumption of Eve's practical computational capability helps offer intuitive measurements of security degradation to indicate the importance of correctly setting up statistical testing for wireless security. For example, the 2017 SHA-1 collision attack has an estimated computational effort equivalent to $2^{63.1}$ SHA-1 calls [31]. We define Eve's capability and objective as follows.

**Definition 2** (Eve's Capability and Objective)**.** *Given Definition 1, Eve aims to develop a key search strategy to maximize her success probability by performing $N$ searches for the secret $K_D$, and $N$ is called Eve's capability.*

### B. Analyzing Secrets Generated from Wireless Channel Randomness

Given the fact that all models in Definition 1 are publicly known, the objective of Eve is to develop a strategy to efficiently search for $K_D$ without exact knowledge of Alice and Bob's channel response $\omega_D$. Existing studies have well explored the building of functions $f_w$ and $f_c$ to obtain a key from (4), but never fully investigated Eve's strategy. Suppose Eve has no smarter strategy but RG, if we assume that she has a maximum capability $2^{64}$, the probability is $2^{64}/2^{128} = 2^{-64}$ to obtain a 128-bit key generated by Alice and Bob.

Is there a smarter strategy for Eve to do better? We first analyze how the wireless channel generates secret bit sequences. [9] proposed the basic idea of the level-crossing algorithm: the channel information (e.g., RSSI or CSI) is

estimated over a time interval (TI) larger than the coherence time, set two thresholds $q_+$ and $q_-$ by calculating magnitude or phase; the measured channel information will be quantified to 1, if it is greater than a threshold $q_+$ or 0 less than $q_-$. It is widely suggested in existing works [2], [3], [9] that the measurement interval should at least equal the channel coherence time such that the measured samples are considered approximately independent.

Nonetheless, this approximate independence assumption creates a very vague boundary from the security perspective: the measurements from wireless channel are correlated [2], [3], [9], [32]; although the correlation becomes weaker and could be considered approximately independent for traditional performance analysis when the measurement interval increases, it still makes sense for security analysis to assume that the output bits from the wireless domain are statistically correlated rather than approximately independent. In other words, the input of function $f_w$ in (1) is regarded as a correlated signal, leading to a correlated output model of $f_w$.

**Definition 3** (Wireless Bit Generation Model)**.** *Given a channel measurement and quantification period $D$, the output from the wireless domain, denoted as bit sequence $X = f_w(\omega_D) = [x_1, x_2, \cdots, x_L]$, is modeled as a binary correlated sequence with correlation coefficient $\rho \in [-1, 1]$ for consecutive bits $x_i$ and $x_{i+1}$ for $i \in [1, L-1]$, which is written as*

$$\rho = \frac{\text{cov}(x_i, x_{i+1})}{\sigma(x_i)\sigma(x_{i+1})}, \tag{8}$$

*where $cov(x_i, x_{i+1}) = \mathbb{E}\left((x_i - \mathbb{E}(x_i))(x_{i+1} - \mathbb{E}(x_{x+1}))\right)$ is the covariance between $x_i$ and $x_{i+1}$, and $\sigma(x_i)^2 = \mathbb{E}((x_i - \mathbb{E}(x_i))^2)$ is the standard deviation of $x_i$.*

Definition 3 offers a more practical and generic model compared with the traditional one that assumes that channel samples are approximately independent with the sampling duration larger than the coherence time used in the literature. Apparently, we can set $\rho = 0$ to obtain from the correlated model to the traditional one. Moreover, from a security perspective, we should always assume a defective (rather than perfect) randomness model for security design. A good wireless domain design should generate a bit sequence with a correlation coefficient $\rho$ close to 0. But we can never know a design indeed achieves 0 in practice. Thus, it is always good to assume $|\rho| > 0$ even it is a very small value. Accordingly, Eve can leverage such a model to construct her attack strategy, which in turn facilitates formal security analysis for statistically testing wireless channel randomness.

### C. Eve's Strategy

Given the bit generation model from the wireless domain, let us look at the secret bit generation from Eve's perspective, shown in Figure 3. As Eve knows $K_D = f_c(f_w(\omega_D))$ from (4), there are three straightforward strategies.

1) Search for the secret $K_D$ in the $\{0,1\}^M$ space. There is no evident strategy better than RG because the last step of $f_c$ is privacy amplification.

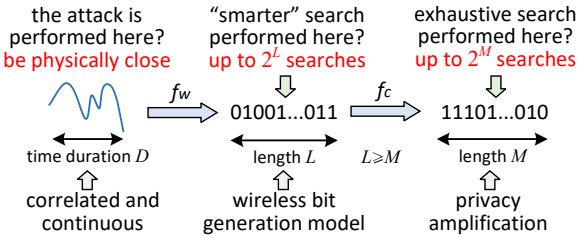


Fig. 3: Eve's perspective on the secret key generation.

2) Search for the wireless domain output $X = f_w(\omega_D)$ in the $\{0,1\}^L$ space. Then, compute $K_D = f_c(X)$ because $f_c$ is public. Note that $L \geq M$, so is it really worth searching in a potentially larger space? We find that leveraging the bit correlation to search for $X$ in $\{0,1\}^L$ can result in a better success probability than RG in $\{0,1\}^M$.
3) Search for $\omega_D$, then compute $K_D = f_c(f_w(\omega_D))$. We note that this is possible only if Eve can physically access Alice's or Bob's antenna. We assume that Eve has no such access in this paper.

In the three strategies, we show that the second one for Eve (i.e., searching for $X = f_w(\omega_D)$ then computing $K_D = f_c(X)$) can generate higher success probability if Eve leverages the correlation in the wireless bit generation model in Definition 3. Figure 4 illustrates an example of how the first 4 bits $x_1$, $x_2$, $x_3$ and $x_4$ from $X$ are generated: the wireless channel is slowly varying and the wireless domain design samples and quantifies the channel response into bits in a sequential way. The first two bits $x_1$ and $x_2$ are 1 and the channel changes so the last two bits $x_3$ and $x_4$ are 0.

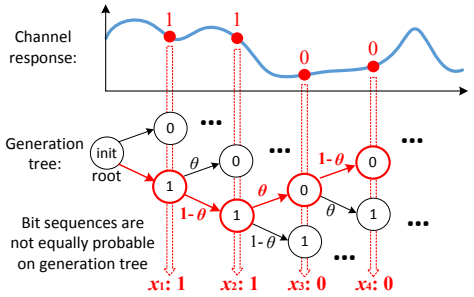


Fig. 4: Bit generations from the wireless domain forms a generation tree.

According to Definition 3, we map the wireless-domain generation into an abstract process in a generation tree as shown in Figure 4, which enumerates all possible bit values generated sequentially. A path from the root to a leaf can represent a generated bit sequence. For example, Figure 4 shows the path that generates 1100. We denote by $\theta$ the value transition probability in the tree (i.e., the probability that the values of two consecutive bits are different). When the correlation coefficient is larger than 0, the correlation among bits in fact means that a generated bit is more likely to have the same value of the previous generated bit, i.e., $\theta$ should be smaller than 0.5. If the correlation coefficient is smaller than 0, $\theta$ should be larger than 0.5, respectively. As a consequence,

all paths from the root to leaves in the tree exhibit different probabilities. This helps Eve because she can search for $X$ by starting from the most likely bit sequence towards the least likely bit sequence in the tree. We call such a strategy maximum-likelihood tree search (MLTS).

MLTS maximizes Eve's success probability if bits in $X$ are statistically correlated (i.e. $|\rho| > 0$), and has equal performance to RG otherwise (e.g., $\rho = 0$). In the following, we show the attack performance of MLTS.

**Theorem 1** (Maximum-Likelihood Tree Search)**.** *For the sake of simple notation, we let Eve's computational capability $N$ in Definition 2 satisfy $N = \sum_{i=0}^{n/2} \binom{L}{i} + \sum_{j=0}^{n/2} \binom{L}{j}$, where $0 \le n \le L$. Then, given the fact that a secret $K_D$ has been established, the attack success probability of* MLTS *is*

$$\begin{aligned} &\mathbb{P}(\text{MLTS succeeds} \,|\, K_D \text{ established}) \qquad (9)\\ &= I_{\frac{1-\rho}{2}}(L - \frac{n}{2}, \frac{n}{2} + 1) + I_{\frac{1+\rho}{2}}(L - \frac{n}{2}, \frac{n}{2} + 1) = I_{MLTS}, \end{aligned}$$

*where $I_x(a, b)$ is the regularized incomplete beta function*

$$I_x(a, b) = \frac{B(x; a, b)}{B(a, b)} \qquad (10)$$

*with incomplete beta function $B(x; a, b) = \int_0^x t^{a-1}(1-t)^{b-1}dt$ and complete beta function $B(a, b) = B(1; a, b)$.*

*Proof:* To facilitate smooth presentation of our design and results, we defer the proof to Appendix A for details. □

The advantage of MLTS is that it does not need to know the value of $\rho$ and the transition probability $\theta$ to work. Eve should always try to use MLTS in practice to search for $X$ then compute $K_D$. It is worth noting that we use the number of searches as an indicator of computational complexity. We consider Eve performs one search on a sequence if Eve spends some computations on the sequence. Due to the use of randomness testing and the use of hundreds of bits as a key in today's practice, Eve cannot easily exclude a wide range of sequences of hundreds of bits (or easily prone a large branch of the search tree) during searching for the correct key. Eve has to test sequences one by one. Even for a bit sequence that fails the randomness test, she still has to test it (thereby spending some computational time) before knowing that it cannot be used as the key. Or Eve can skip the test and directly spend computations on verifying if a key candidate is correct. These computations on the bit sequence constitute one search regardless of the fact that the bit sequence is test-compliant or not. Therefore, there is no straightforward way to skip all the sequences that are not test-compliant. Knowing the fact that $K_D$ is established does not reduce the number of searches to be performed by Eve. Note that we do not consider trivial cases here (e.g., Eve can simply exclude all 0 or 1 bits).

In order to improve the efficiency of key generation, multi-level quantization methods have been developed in [33], [34]. The core of MLTS, which is to search from the most likely sequence to the least likely sequence, can also been applied to multiple level quantization. Fig. 5 shows an example of 4-level quantization of the channel response. The quantization includes 4 states (i.e., 11, 10, 01, 00). As the figure shows, the system first quantifies the channel response to state 4 (11),

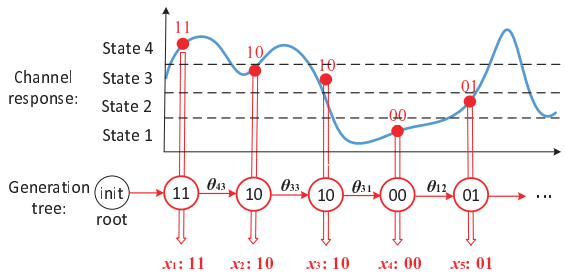


Fig. 5: Eve's perspective on the secret key generation of 4-level quantization.

then state 3 (10), state 3 (10), state 1 (00), and state 2 (01), which leads to the sequence of 1110100001. The correlation of two states is stronger if they are closer. As a result, Eve's MLTS can be executed from the most likely sequence to the least likely sequence on a tree with multiple-bit states (instead of single-bit ones) as individual nodes. In this paper, we will focus on the single-bit quantization as it has been widely used in existing studies [3]–[6], [9]–[11]. We provide basic performance analysis of MLTS for multiple-level quantization in Appendix B.

## V. Guidelines for Statistical Randomness Test Settings

With the clearly defined MLTS-based attack model for Eve, we are ready to address how Alice and Bob should test the wireless channel randomness for security. Alice and Bob must make sure that they will not create a common secret from the wireless channel with a high correlation over time. On the other hand, they never know the exact value of channel correlation in practice. Then, it seems natural for them to test the channel first, and then make a binary decision, which formalizes the role of the randomness test $T$ in Definition 1.

In this section, we analyze Eve's success probability as a function of randomness testing, then propose the guideline for Alice and Bob to properly set up the randomness testing for security and efficiency.

### A. Eve's Success Probability as A Function of Randomness Testing

Randomness testing aims to eliminate the security loss defined in (6) by denial and we should make sure that Eve's success probability is no better than the RG's success probability (i.e., no security loss). Based on Definition 1, we define $\mathbb{P}(\text{Eve succeeds})$ as

$$\begin{aligned} \mathbb{P}(\text{Eve succeeds}) &= \mathbb{P}(T \text{ Accept } H_0)\mathbb{P}(\text{Eve succeeds}|H_0) \\ &\le \mathbb{P}(\text{RG succeeds}), \qquad (11) \end{aligned}$$

where

- $\mathbb{P}(\text{Accept } H_0)$ is the probability that randomness test $T$ passes, which depends on the settings of $T$.
- $\mathbb{P}(\text{RG succeeds}) = N2^{-M}$ with Eve's capability $N$ and key length $M$.

- $\mathbb{P}(\text{Eve succeeds}|H_0) \approx I_{\text{MLTS}}$ obtained in (9), denoting the probability that Eve obtains the key $\Omega_D$ conditioned on randomness test $T$ passes[2].

According to our analysis, Eve should always use MLTS in practice and hope for a large $|\rho|$. As a result, Eve's success probability can be written as

$$\begin{aligned}&\mathbb{P}(\text{Eve succeeds}) = \\ &\mathbb{P}(\text{MLTS succeeds} \,|\, K_D \text{ established})\mathbb{P}(K_D \text{ established}) \\ &= I_{\text{MLTS}} \cdot \mathbb{P}(T \text{ accepts } H_0) \leq \mathbb{P}(\text{RG succeeds}), \end{aligned} \tag{12}$$

where the last equality follows from Theorem 1. Due to the fact that the binary sequence can be considered random enough after privacy amplification, MLTS only can focus on the sequence in the wireless domain. The more random (randomness test can reject large $|\rho|$ scenarios) and longer binary sequence ($L \geq M$) could eliminate the security loss.

### *B. Observations and Design Guideline*

From (12), we can only guarantee that there is no security loss but the test $T$ might be set too strict to cause a lower efficiency $E$. Hence, efficiency $E$ is also important and should be considered at the same time of the design guideline. As a result, the design guideline is proposed to find the settings for test $T$ and the privacy amplification rate $R_{\text{privacy}}$ to maximize efficiency under the constraint of no security loss, which is written as follows

$$\max \;\; E = \mathbb{P}(T \text{ accepts } H_0) \cdot R_{\text{privacy}} \tag{13a}$$

$$\text{s.t.} \quad I_{\text{MLTS}} \cdot \mathbb{P}(T \text{ accepts } H_0) \leq \mathbb{P}(\text{RG succeeds}). \tag{13b}$$

The design guideline is proposed to find the settings for test $T$ and the privacy amplification rate $R_{\text{privacy}}$ to maximize efficiency under the constraint of no security loss. (13b) ensures that $L_{\text{security}} = 0$ by selecting the proper P-value threshold $\alpha$ and $R_{\text{privacy}}$. We provide the theoretical analysis of how to calculate $\mathbb{P}(T \text{ accepts } H_0)$ for different $\alpha$ values under different randomness tests in Appendix C. Although we have both theoretical results of $I_{\text{MLTS}}$ and $\mathbb{P}(T \text{ accepts } H_0)$, there is no straightforward convex or concave property (i.e., increasing $\alpha$ and decreasing $R_{\text{privacy}}$ may both satisfy (13b)). In practical systems, $\alpha$ and $R_{\text{privacy}}$ have typical value ranges and we select $\alpha \in [0.0001, 0.3]$ and $R_{\text{privacy}} \in [0.1, 1]$ in this paper. Within the ranges, we use greedy search with small granularity to find the best pair that maximizes (13a). From the design guideline (13), we can answer the questions in Section II-C.

1) The cryptographic domain function $f_c$ is based on privacy amplification, however, over-estimating the entropy cannot be avoided, since it is generally difficult to accurately estimate the per-bit entropy of a physical source [36]. Consequently, if the randomness test $T$ is set in the cryptographic domain, it is equivalent to testing the output of a sufficiently random sequence, and always passing the test $T$. Therefore, it is reasonable to test the wireless domain output $X = f_w(\omega_D)$ when extracting randomness from the wireless channel.
2) Based on (13), we can solve the optimization function to find the sufficient P-value threshold $\alpha$ and $R_{\text{privacy}}$, instead of manually setting the parameters, to ensure no additional loss in security and achieve high efficiency.

Randomness test $T$ is an important part in the guideline (13). Rather than designing a new randomness test, we focus on NIST randomness tests as they have been well structured and widely adopted [2]–[14]. In order to configure the NIST randomness tests, we need to analyze the relationship between $\mathbb{P}(T \text{ accepts } H_0)$ and $(\rho, \alpha)$ for a specific test. Hence, in the following, we present how to bridge $\mathbb{P}(T \text{ accepts } H_0)$ to $h(\rho, \alpha)$, where $h(\cdot)$ represents the probability function of $(\rho, \alpha)$ for a specific NIST test.

In many scenarios, multiple randomness tests can be used together for testing. This combination can enhance security and indeed loosen the $R_{\text{privacy}}$ setting. However, the setup for a single test in existing studies is not loosened even when multiple tests are used. This has been common in existing studies for wireless key generation [2]–[14], hardware security [37], cryptography and software security [38], [39]. This is due to two major reasons: i) it can be mathematically intractable to analyze the joint correlation among multiple tests. The NIST guideline [21] performed such a correlation study and only shows that empirically the correction among NIST tests is very small. As a result, it can be difficult to show how much the setup for each test can be loosened analytically. ii) Using single-test setup can ensure the worst-case security guarantee even when multiple tests are used. We also adopt this practice in the paper and recommend the use of the single-test setup for multiple-test scenarios.

### *C. Analysis of NIST Randomness Tests*

There are fifteen tests provided in the NIST test suite [21], and we choose nine of them which are commonly used in the existing literature [2]–[14]. The nine tests in our study are frequency test, frequency test within a block, runs test, test for the longest run of ones in a block, discrete fourier transform test, non-overlapping template matching test, approximate entropy test and serial test (serial test has two orders). Based on the P-value computation formula, these tests can be categorized into two classes: Gaussian and chi-square distribution. In the following, we use the most common frequency test $T_{\text{freq}}$ as the representative to show the procedure of the relationship between $\mathbb{P}(T_{\text{freq}} \text{ accepts } H_0)$ and $(\rho, \alpha)$. The results of other tests are provided in Appendix C. We use the function $h_{\text{freq}}(\rho, \alpha)$ to represent frequency test in NIST. Given a bit sequence $X = [x_1, x_2, \ldots x_L]$ from Definition 3,

$$\begin{aligned} h_{\text{freq}}(\rho, \alpha) &= \mathbb{P}(T_{\text{freq}} \text{ accepts } H_0) \\ &= \mathbb{P}(|S_{\text{freq}}(X)| < \sqrt{2}\text{erfc}^{-1}(\alpha)), \end{aligned} \tag{14}$$

[2]Eve aims to search the wireless domain output $X$ yielding the key $K_D$. However, due to hash collision in privacy amplification, there exists a probability $\mathbb{P}$(collision) that Eve finds another bit sequence $X' \neq X$ satisfying $K_D = f_c(X) = f_c(X')$. As a result, $\mathbb{P}(\text{Eve succeeds}) = I_{\text{MLTS}} + \mathbb{P}(\text{collision})$, where $\mathbb{P}$(collision) can be approximated as $1 - (1 - 2^{-M})^{\dot{L}}$ in [35]. Since privacy amplification $f_c$ is always designed to make the collision probability $\mathbb{P}$(collision) negligible, for example, $L = M = 32$ and $N = 16$, $I_{\text{MLTS}} \geq \mathbb{P}(\text{RG}) = 1.53 \times 10^{-5}$ and $\mathbb{P}(\text{collision}) = 7.45 \times 10^{-9}$ such that $I_{\text{MLTS}} \gg \mathbb{P}(\text{collision})$. Therefore, we approximate that $\mathbb{P}(\text{Eve succeeds}) \approx I_{\text{MLTS}}$ for a sufficiently large capability $N$ for Eve in this paper.

where $S_{\text{freq}}(X) = \frac{1}{\sqrt{L}} \sum_{l=1}^{L} (2x_l - 1)$ is the statistics definition of frequency test. Since the correlated sequence $X$ can be considered as generating from a uniformly ergodic Markov chain [40], $S_{\text{freq}}(X)$ can be derived by the Markov central limit theorem, only if we know the mean and variance. $|S_{\text{freq}}(X)| \sim \mathcal{N}\left(0, \frac{1+|\rho|}{1-|\rho|}\right)$ is followed by Gaussian distribution. Thus, we have the $h_{\text{freq}}(\rho, \alpha)$ as follows

$$h_{\text{freq}}(\rho, \alpha) = \text{erf}\left(\text{erfc}^{-1}(\alpha)\sqrt{\frac{1-\rho}{1+\rho}}\right), \tag{15}$$

where erf and $\text{erfc}^{-1}$ are error function and inverse complementary error function. Based on the analysis of randomness test $T$, we can choose the proper pairs of P-value threshold $\alpha$ and privacy amplification rate $R_{\text{privacy}}$.

## VI. Experimental Evaluation

In this section, we obtain the security loss, efficiency loss and bits mismatch rate of the secret key generation by the wireless channel response. In the following, we first introduce the experimental setup. Then, we compare the performance of existing secret key generation methods before and after being revised by our design guideline in (13) under different experimental environments, different randomness tests and different length of keys.

### A. Experimental Setup

**Channel Response Measurements:** The first step towards analyzing the randomness of the secret key generation in the wireless domain is to collect a large number of channel information (RSSI, CSI and Phase shifts) in realistic environments. We use two USRP X310s, acting as a transmitter and a receiver respectively, to build our experimental platform, where each device is equipped with a UBX-160 daughterboard and a VERT 2450 antenna. The software toolkit is GNURadio. We implement a typical training data-aided time and frequency synchronization scheme based on [41] for channel probing whose procedure follows [9]. For the equalization, we adopt a frequency-domain OFDM equalizer with the aid of pilot tones [42]. To measure the channel information, the transmitter consistently sends training sequences (as known as the preamble in wireless standards) to the receiver with fixed transmit power. On this experiment platform, we collect more than 1 billion channel information in total spanning over 24 hours in different environments with 2.4GHz carrier frequency and 1.0MHz bandwidth.

TABLE I: Parameters setting of existing methods.

| Examples | Test setting domain | $\alpha$ value | $R_{\text{Privacy}}$ | Source |
|---|---|---|---|---|
| RT | wireless | 0.01 | 1 | RSSI |
| ZR | wireless | 0.01 | 1 | RSSI |
| TSCC | wireless | 0.05 | 0.5 | Phase |
| TDS | cryptographic | 0.05 | 0.32 | CSI |
| ASBG | cryptographic | 0.01 | 0.125 | RSSI |
| RSKE | cryptographic | 0.01 | 0.24 | RSSI |

**Secret Key Generation Model:** We compare the performance of 6 existing secret key generation models: Radio-telepathy (RT) [9], Zero Reconciliation (ZR) [14], Temporally and Spatially Correlated Coefficients (TSCC) [11], The Dancing Signals (TDS) [6], Adaptive Secret Bit Generation (ASBG) [3] and Robust Secret Key extraction (RSKE) [5], where the P-value threshold $\alpha$, $R_{\text{privacy}}$, different NIST statistical randomness test setting position and source of secret key generation are shown in Table I.

**Eve's Capability:** The attacker Eve aims to obtain the secret key $K_D$ through MLTS without knowing any channel information. Although a realistically powerful capability for Eve is $2^{63.1}$ [31], it is still statistically insignificant in the experimentation in order to crack a long key, for example, the attack success probability of cracking a 128-bit key is about $2^{-64}$. Thus, we consider a more powerful capability of Eve with $N = 2^{96}$ such that attack success probability increases from $2^{-64}$ to $2^{-32}$, where is observable in our experiments.

**Evaluation Metrics:** The evaluation metrics used in our experiments are security loss $L_{\text{security}}$ and efficiency loss $L_{\text{efficiency}}$ defined in Section III-B as well as the bits mismatch rate $R_{\text{mismatch}}$, which is the ratio of the number of bits that do not match between Alice and Bob to the number of bits extracted from channel. All the experimental results are the average value from at least 30 independent experiments.

### B. Evaluation Results

**Evaluation of Existing Secret Key Generation Methods:** We aim to show the performance (i.e., $L_{\text{security}}$ and $L_{\text{efficiency}}$ for 128-bit secret key) of existing methods with the P-value threshold $\alpha$ and $R_{\text{privacy}}$ settings in the literature in comparison with the new values for these parameters based on our design guideline. We collect the channel information under 5 meters laboratory environment and using frequency test.

Figure 6 shows that the security and efficiency losses of these seven different secret key generation methods before and after new parameter settings based on our guideline. We can see that RT, ZR and TSCC, which set the tests in wireless domain, have higher $L_{\text{security}}$ than others in the cryptographic domain, e.g., $L_{\text{security}}$ of RT is $4$. Although setting test $T$ in the cryptographic domain can eliminate security loss, it leads to high efficiency loss, e.g, ASBG has $L_{\text{efficiency}} = 0.883$. After the new settings based on our design guideline, i) all of these methods have no $L_{\text{security}}$; ii) $L_{\text{efficiency}}$ significantly decreases. For example, ASBG changes from $0.883$ to $0.487$ in terms of efficiency loss. Through the revision of parameters based on our guideline, $R_{\text{mismatch}}$ of all methods can be reduced, as shown in Figure 7. Figures 8 and 9 show the P-value threshold $\alpha$ and $R_{\text{privacy}}$ are calculated by our design guideline, which indicate that different secret key generation methods need to calculate different $\alpha$ and $R_{\text{privacy}}$ to eliminate security loss and achieve high efficiency. On the other hand, Figures 8 and 9 also illustrate that if we choose a higher P-value threshold, we also need a higher $R_{\text{privacy}}$ to eliminate security loss.

**Evaluation of Different Experimental Environments:** The secret key generation method may extract the secret keys with different randomness degrees under varying wireless communication environments. We will show $L_{\text{security}}$ and $L_{\text{efficiency}}$ with or without our design guideline in different practical

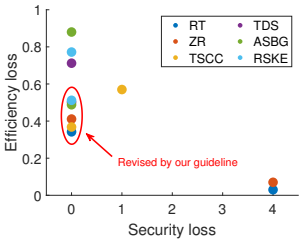


Fig. 6: Security loss vs Efficiency loss

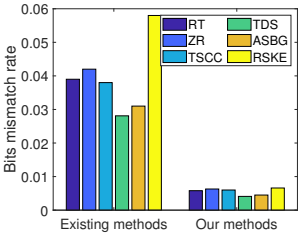


Fig. 7: Bits mismatch rate

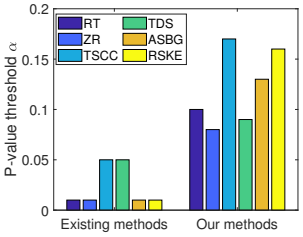


Fig. 8: P-value threshold $\alpha$

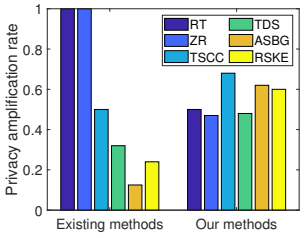


Fig. 9: Privacy amplification rate $R_{\text{Privacy}}$

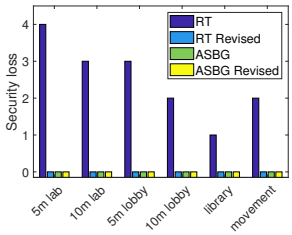


Fig. 10: $L_{\text{security}}$ in different experimental environments.

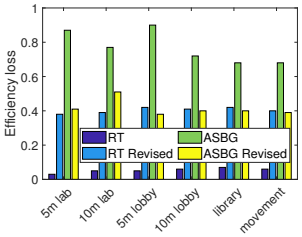


Fig. 11: $L_{\text{efficiency}}$ in different experimental environments.

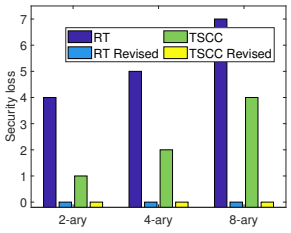


Fig. 12: $L_{\text{security}}$ under multiple level quantizations.

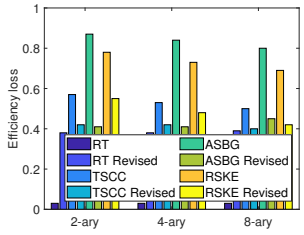


Fig. 13: $L_{\text{efficiency}}$ under multiple level quantizations.

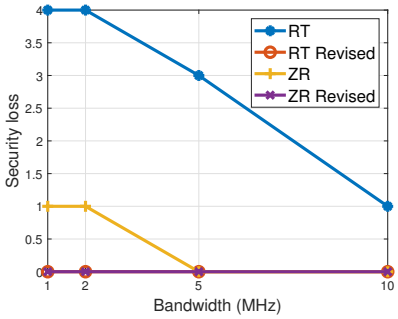


Fig. 14: $L_{\text{security}}$ under different bandwidths.

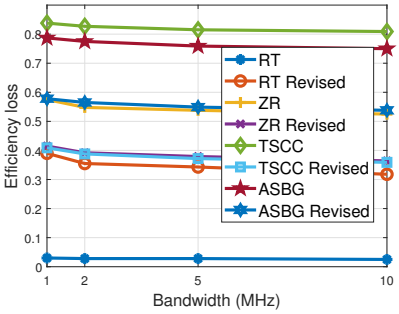


Fig. 15: $L_{\text{efficiency}}$ under different bandwidths.

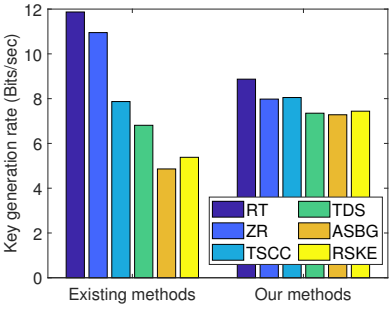


Fig. 16: Key generation rate (Bit/sec) of different methods.

scenarios. Among the 6 secret key generation methods, we select RT and ASBG, since they set the NIST test at the different positions. We conduct experiments to collect RSSI, CSI or phase shift in different indoor environments. For the static USRPs setting, we set the distance of two USRPs to be 5 and 10 meters in 40m$^2$ laboratory, 10 and 25 meters in 350m$^2$ public lobby, 25 meters in 450m$^2$ campus library. For the movement setting, we fix the position of one USRP, and randomly move another one in laboratory. We consider the laboratory environment line-of-sight condition, and lobby and library environments non-line-of-sight condition.

Figure 10 shows $L_{\text{security}}$ under different experimental environments, and that RT performs the worst for each environment. After revision by our guideline, we eliminate all security loss (i.e., $L_{\text{security}} = 0$). From Figure 11, we increase the $L_{\text{efficiency}}$ from $0.05$ to around $0.4$ of RT under each communication environment. In ASBG, since they set a small original $R_{\text{privacy}}$ value (i.e., $0.125$ in Table I), it incurs high $L_{\text{efficiency}} = \{0.85, 0.74, 0.87, 0.71, 0.68, 0.69\}$ under different environments. After revision by our guideline, we properly set the new $R_{\text{privacy}}$ so that ASBG performs more efficiently such that $L_{\text{efficiency}} = \{0.41, 0.49, 0.37, 0.39, 0.39, 0.38\}$.

**Evaluation of Multiple Level Quantizations:** Figures 12 and 13 show $L_{\text{security}}$ and $L_{\text{efficiency}}$ under $m$-ary quantization methods ($m \in \{2, 4, 8\}$) with 4 methods: RT, TSCC, ASBG and RSKE.

Due to the fact that ASBG and RSKE with original settings have no security loss, we do not show them in Figure 12 for clearer illustration. In Figure 12, we can see that a larger value of $m$ incurs a higher security loss (e.g., for RT, $L_{\text{security}} = 7$ when $m = 8$). After revision based on our guideline, we ensure $L_{\text{security}} = 0$ for RT and TSCC. In Figure 13, it is observed that when we ensure $L_{\text{security}} = 0$ by setting up randomness tests for different values of $m$, the efficiency $L_{\text{security}}$ approximately remains the same. This indicates that simply adopting a multiple level quantization method does not necessarily mean a faster key generation rate with security guarantee.

**Evaluation of Different Bandwidths:** Figures 14 and 15 show $L_{\text{security}}$ and $L_{\text{efficiency}}$ under different bandwidth set-

TABLE II: P-value threshold $\alpha$ and $R_{\text{privacy}}$ setting for different quantization methods.

| | RT | | ZR | | TSCC | | TDS | | ASBG | | RSKE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $R_{\text{privacy}}$ | $\alpha$ | $R_{\text{privacy}}$ | $\alpha$ | $R_{\text{privacy}}$ | $\alpha$ | $R_{\text{privacy}}$ | $\alpha$ | $R_{\text{privacy}}$ | $\alpha$ | $R_{\text{privacy}}$ |
| Frequency | 0.11 | 0.50 | 0.08 | 0.47 | 0.17 | 0.68 | 0.09 | 0.48 | 0.13 | 0.62 | 0.16 | 0.60 |
| Block of Frequency | 0.21 | 0.67 | 0.18 | 0.59 | 0.27 | 0.79 | 0.19 | 0.61 | 0.25 | 0.72 | 0.20 | 0.63 |
| Run | 0.14 | 0.51 | 0.09 | 0.51 | 0.22 | 0.70 | 0.11 | 0.52 | 0.14 | 0.63 | 0.14 | 0.57 |
| Longest Run of ones | 0.13 | 0.48 | 0.18 | 0.57 | 0.21 | 0.66 | 0.10 | 0.47 | 0.17 | 0.61 | 0.17 | 0.66 |
| DFT | 0.07 | 0.61 | 0.14 | 0.62 | 0.14 | 0.71 | 0.09 | 0.54 | 0.13 | 0.62 | 0.11 | 0.68 |
| Non-overlap | 0.08 | 0.46 | 0.20 | 0.55 | 0.17 | 0.61 | 0.17 | 0.50 | 0.22 | 0.66 | 0.21 | 0.63 |
| Approximate entropy | 0.06 | 0.41 | 0.18 | 0.53 | 0.14 | 0.62 | 0.15 | 0.51 | 0.19 | 0.62 | 0.15 | 0.61 |
| First order serial | 0.06 | 0.44 | 0.11 | 0.41 | 0.15 | 0.61 | 0.09 | 0.44 | 0.11 | 0.57 | 0.08 | 0.59 |
| Second order serial | 0.04 | 0.40 | 0.09 | 0.42 | 0.14 | 0.60 | 0.11 | 0.47 | 0.08 | 0.51 | 0.09 | 0.61 |

TABLE III: $L_{\text{efficiency}}$ of different bit sequence length before and revised by our guideline.

| | 128-bit | | 256-bit | | 512-bit | |
|---|---|---|---|---|---|---|
| $L_{\text{efficiency}}$ | Original | Revised | Original | Revised | Original | Revised |
| RT | 0.06 | 0.33 | 0.03 | 0.35 | 0.08 | 0.31 |
| ZR | 0.09 | 0.39 | 0.07 | 0.33 | 0.05 | 0.25 |
| TSCC | 0.58 | 0.37 | 0.57 | 0.31 | 0.55 | 0.24 |
| TDS | 0.71 | 0.57 | 0.71 | 0.48 | 0.73 | 0.30 |
| ASBG | 0.86 | 0.42 | 0.88 | 0.36 | 0.85 | 0.29 |
| RSKE | 0.73 | 0.57 | 0.70 | 0.44 | 0.68 | 0.21 |

tings $\{1, 2, 5, 10\}$MHz in the lobby environment. We do not show the $L_{\text{security}}$ of ASBG and RSKE, since they have no security loss. When we increase the bandwidth, it is easy to observe that $L_{\text{security}}$ decreases under RT and TSCC (e.g., for RT, $L_{\text{security}}$ goes from 4 to 1). We can observe that the wireless signal with a larger bandwidth make the random bit sequence less correlated and therefore more random. From the efficiency perspective, increasing bandwidth can decrease $\mathbb{P}(T \text{ accepts } H_0)$ such that $L_{\text{efficiency}}$ becomes smaller (e.g., for ASBG, $L_{\text{efficiency}}$ decreases from 0.57 to 0.52).

**Evaluation of Key Generation Rates:** For testing the key generation rate of each method, we setup the 6 existing key generation methods for extracting 128-bit key sequence using the parameters in Table I under 5 meters laboratory environment. Under the same environment, we revise the parameters by our guideline and test the key generation rate. For each experiment, we collect 10,000 bits and calculate the rate (bits/sec). The results are shown in Figure 16. We note that RT and ZR has higher rates with their original setups but they both have security losses. Our setups ensure the maximum key generation rates (e.g., 7.49 bits/sec for ASBG) without security loss.

**Evaluation of Different Randomness Tests $T$ and $R_{\text{privacy}}$:** Table II indicates that no tests can achieve the optimization goal in (13) by using $\alpha = 0.01$. In other words, if we just follow the recommendation of NIST, we may not eliminate the security loss and guarantee efficiency of secret key generation. On the other hand, if we only focus on the security, the efficiency can be very low. Therefore, it is necessary to meet our guideline to consider both efficiency and security aspects. It is noted that Table II shows different values of $\alpha$ and $R_{\text{privacy}}$ for different randomness tests. This is due to the fact that each test has its own $\mathbb{P}(T \text{ accept } H_0)$, which is provided in Appendix C, in our guideline and we need to solve its own pair of $\alpha$ and $R_{\text{privacy}}$ based on the optimization (13a).

**Efficiency of Different Length of Secret Keys Generation:** Intuitively, if Eve's attack capability does not change, it should be more difficult to crack a longer key sequence. Thus, randomness test $T$ and privacy amplification should be set looser for a longer key sequence. However, the existing methods use the fixed value $\alpha$ and $R_{\text{privacy}}$ for generating different lengths of key sequence such that it is detrimental to the secret key generation efficiency (suppose no security loss). We evaluate $L_{\text{efficiency}}$ of 128, 256 and 512-bit secret key generation before and after revision based on our guideline under the laboratory scenario and frequency test. Eve's attack capability is $N = 2^{96}$.

In Table III, we can observe that it is more efficient to generate a longer secret key sequence based on our guideline (e.g., for TDS with revised setups, $L_{\text{efficiency}}$ is 0.57, 0.48 and 0.30 for the 128-bit, 256-bit and 512-bit cases, respectively). However, if we use the parameter setting in the existing studies, $L_{\text{efficiency}}$ does not change obviously, e.g., $L_{\text{efficiency}}$ of RSKE are $0.73$, $0.70$ and $0.68$. Hence, our design guideline also offers an adaptive method to generate keys with different lengths, which is not presented in existing studies.

## VII. Related Work

To provide the confidentiality of data transmission, secret key generation based on the information of wireless channels is promising because of the efficiency and security [43]–[46]. In [43], proximity attack requires the minimal distance from the eavesdropper to maintain perfect secrecy for secret key generation. The randomness test can provide a generic threshold on required distances from an eavesdropper and good key refreshing rates. [44] explores the use of wireless channel characteristics for establishing arbitrary length secret keys between Bluetooth devices. They verified the output secret bit streams generated by Bluetooth achieve high entropy by the randomness test. [45] tests the randomness of key bits, which quantifies a subcarrier's channel response with different coherent time. [46] proposes to defend against threats of eavesdropping and fake data injection in underwater acoustic networks, providing an overview of the advantages of RSSI based key generation and exploring the major challenges from the unique features of acoustic communications.

Key establishment using physical layer characteristics [47]–[49], which are much richer source of secret information but high computational cost overhead. [47] reviews different types of existing methods based on quantization, handling communication errors and the feasibility and security issues related to these methods. [48] proposes the reciprocity theorem, which

has become the most important theorem in this kind of method. The paper [49] provides an efficient secret key generation method using multipath relative delay from Ultra-wideband (UWB) channels. They study a statistical characterization of UWB channels in a residential scenario, and evaluate key mismatch probability. [50] presents the key establishment that uses the distance variation trends caused by the motion paths of two devices to each other.

Recently, some studies have started working in authenticating the transmitter and receiver based on prior coordination or secret sharing. [51] proposes physical-layer authentication schemes through adding low-power signal. [52] solves the authentication in IoT by exploiting the fading of the wireless links between devices to be authenticated and a set of trusted anchor nodes. [17] proposes a retroactive key setup to protect source authentication and path validation into the realm of practicality. [53], [54] adapt fingerprint embedding to keep message authentication and increased security by obscuring the authentication tag.

## VIII. Conclusion

This paper studies how to properly test the wireless channel randomness for security and efficiency. In particular, we proposes a new design guideline that can choose the P-value threshold, a critical parameter of the randomness test, to ensure the security of the wireless system as well as achieve a high secret bit generation rate with privacy amplification. Since the practical channel information (CSI, RSSI or phase shifts) is imperfectly independent, we come up with a new cracking key attack called MLTS, which searches the bit sequence by leveraging the Markov dependent property. By tuning a suitable P-value threshold and privacy amplification rate, we formulate an optimization problem to maximize the key generation rate under the constraint of no security loss. Our analysis indicates that the randomness test $T$ should be set in the wireless domain. We conduct different practical environments to validate the analysis of our guideline. By comparing to existing key generation methods, results show that our guideline can improve these methods to be more efficient and secure.

## Appendix A
## Proof of Theorem 1

Because we cannot know the correlation coefficient exactly, we need to consider the positive and negative correlation simultaneously. Let $\theta$ be the transition probability on the generation tree, and we assume $\theta < 0.5$. In [55], they proved that $\theta = \frac{1-\rho}{2}$. Because Eve cannot know the bit sequence is positive correlation or not, she needs to search from the most likely bit sequences happen with probabilities $\theta^L$ and $(1-\theta)^L$ simultaneously. MLTS searches for $X$ compute $K_T = f_c(X)$ from the most likely bit sequence towards the least likely one in $\{0,1\}^L$. Given the fact $K_D$ has been established, the MLTS success probability searching for $K_D$ is

$$\begin{aligned}
&\mathbb{P}(\text{MLTS succeeds} \mid K_D \text{ established}) \\
&= \sum_{i=0}^{n/2} \binom{L}{i} \theta^i (1-\theta)^{L-i} + \sum_{i=0}^{n/2} \binom{L}{i} (1-\theta)^i \theta^{L-i} \\
&= I_{\frac{1-\rho}{2}} \left( L - \frac{n}{2}, \frac{n}{2} + 1 \right) + I_{\frac{1+\rho}{2}} \left( L - \frac{n}{2}, \frac{n}{2} + 1 \right), \quad (16)
\end{aligned}$$

where $0 \le n \le L$, and $I_x(a,b)$ is the regularized incomplete beta function that has been defined in (10). □

## Appendix B
## MLTS for Multi-level Quantization

If we use $m$ levels to quantify the wireless information, each level can be represented by a $b$-bit string, where $m = \log_2 b$. The multi-level quantization is also called $m$-ary quantization. Here, we redefine the correlation coefficient $\rho_m$ of consecutive bit arrays $x_i^m$ and $x_{i+1}^m$, where $x_i^m = \{x_{i,1}^m, \ldots, x_{i,m}^m\}$ in bit sequence $X$ as follows

$$\rho_m = \frac{\sum_{j=1}^m (x_{i,j}^m - \bar{x}_i^m)(x_{i+1,j}^m - \bar{x}_{i+1}^m)}{\sqrt{\sum_{j=1}^m (x_{i,j}^m - \bar{x}_i^m)^2} \sqrt{\sum_{j=1}^m (x_{i+1,j}^m - \bar{x}_{i+1}^m)^2}}. \quad (17)$$

For the $b$-ary quantization, transition probability $\theta_{r,s}$ (i.e., $\mathbb{P}(x_{i+1}^m = s | x_i^m = r)$), where $r$ and $s$ are the states, $r, s \in \{1, 2, \ldots, m\}$ in transition matrix $\Phi \in \mathbb{R}^{m \times m}$ as $\theta_{r,s} = \rho\delta_{r,s} + (1-\rho)/2^m$, where $\delta_{r,s}$ is Kronecker delta [56]. $m$-ary searching is equivalent to dividing the $L$-bit sequence to $2^{m-1}$ blocks and the number of possible initialization is $\frac{n}{2^m}$. Thus, $\mathbb{P}(\text{MLTS succeeds} \mid K_D \text{ established})$ is given as

$$\begin{aligned}
&\mathbb{P}(\text{MLTS succeeds} \mid K_D \text{ established}) \\
&= \binom{\frac{L}{2^{m-1}}}{0} \theta_{1,1}^{\frac{L}{2^{m-1}}} + \cdots + \binom{\frac{L}{2^{m-1}}}{1} \theta_{1,1}^{\frac{L}{2^{m-1}}-1} \theta_{1,2} + \cdots \\
&+ \binom{\frac{L}{2^{m-1}}}{1} \theta_{1,1}^{\frac{L}{2^{m-1}}-1} \theta_{1m} + \cdots + \binom{\frac{L}{2^{m-1}}}{\frac{n}{2^m}} \theta_{1,1}^{\frac{L}{2^{m-1}}-\frac{n}{2^m}} \theta_{1,2}^{\frac{n}{2^m}} \\
&+ \cdots + \binom{\frac{L}{2^{m-1}}}{\frac{n}{2^m}} \theta_{1,1}^{\frac{L}{2^{m-1}}-\frac{n}{2^m}} \theta_{1,2}^{\frac{n}{2^m}} + \cdots \\
&= \sum_{s=1}^{m} \sum_{i=0}^{\frac{n}{2^m}} \binom{\frac{L}{2^{m-1}}}{i} \theta_{s,s}^{\frac{L}{2^{m-1}}-i} \theta_{s,\neg s}^i \\
&= \sum_{s=1}^{m} I_{\rho + \frac{1-\rho}{2^m}} \left( \frac{L}{2^{m-1}} - \frac{n}{2^m}, \frac{n}{2^m} + 1 \right),
\end{aligned}$$

where $\theta_{s,\neg s}$ is the transition probability from state $s$ to the rest of states except $s$.

## Appendix C
## Theoretical Results of NIST Randomness Tests

Due to the page limitation, we give the results of other 7 different tests and ignore the proof. From NIST test suite [21], we can conclude that the P-value can be calculated by Gaussian distribution: frequency test (Frequency), run test (Run) and DFT test (DFT) and Chi-square distribution: frequency test within a block test (BlockFreq), longest run of ones in a block test (LongRun), non-overlapping template matching test

(Nonoverlap), approximate entropy test (AppEntropy), first order serial (1storder) and second order serial (2ndorder).

**Run:** $\mathbb{P}(T_{\text{run}} \text{ accepts } H_0) = h_{\text{run}}(\rho,\alpha) = \frac{1}{2}\text{erf}\left(\frac{\alpha_f+\mu_f}{\sigma_f\sqrt{2}}\right) + \frac{1}{2}\text{erf}\left(\frac{\alpha_f-\mu_f}{\sigma_f\sqrt{2}}\right)$, where $\alpha_f = 2\sqrt{2L}\pi(1-\pi)\text{erfc}^{-1}(\alpha)$, $\mu_f = L\lambda - 2L(1-\pi)$ and $\sigma_f = \sqrt{L\lambda(1-\lambda)}$. The definition of $\pi$ is the probability of $I_l$, and $\lambda$ is the meaning of $I_l = 1$ if the $l$th element $\neq$ the ($l$-1)th element; $I_l = 0$ otherwise.

**DFT:** $\mathbb{P}(T_{\text{DFT}} \text{ accepts } H_0) = h_{\text{DFT}}(\rho,\alpha) = \frac{1}{2}\text{erf}\left(\frac{\alpha_f+\mu_f}{\sigma_f\sqrt{2}}\right) + \frac{1}{2}\text{erf}\left(\frac{\alpha_f-\mu_f}{\sigma_f\sqrt{2}}\right)$, where $\alpha_f = \text{erfc}^{-1}(\alpha)\sqrt{\mathbb{P}_{\text{DFT}}(1-\mathbb{P}_{\text{DFT}})\frac{n}{4}}$, $\mu_f = 0.95\frac{n}{2} - \mathbb{P}_{\text{DFT}}\frac{n}{2}$, and $\sigma_f = \sqrt{(0.95)(0.05)\frac{n}{4}}$. Accordingly, $\mathbb{P}_{\text{DFT}} = \mathbb{P}(n|S_j(R)| < -n\ln(0.05))$, where $S_j(R)$ is defined in NIST.

**BlockFreq:** The statistic is $\chi^2(\text{obs}) = 4M\sum_{i=1}^{N}(\pi_i - \frac{1}{2})^2$. Similar to the result of Frequency, $\pi_i \sim \mathcal{N}(\frac{1}{2}, \frac{4(1+\rho)}{1-\rho})$. The probability is $\mathbb{P}(T_{\text{longestrun}} \text{ accepts } H_0) = h_{\text{longestrun}}(\rho,\alpha) = \text{igam}\left(\frac{N}{2}, \frac{2\text{igamc}^{-1}(N/2,\alpha)\cdot\chi^2(\text{newobs})}{\chi^2(\text{obs})}\right) - \text{igam}\left(\frac{N}{2}, \chi^2(\text{newobs})\right)$, where igam is incomplete gamma integral function and $\text{igamc}^{-1}$ is inverse complemented incomplete gamma integral function. obs is the statistic by calculating the number and newobs is calculated by the exact distribution $\mathbb{P}(\pi_i)$.

**LongRun:** The statistic is $\chi^2(\text{obs}) = \sum_{i=0}^{K}\frac{(v_i - N\pi_i)^2}{N\pi_i}$, where $\pi_i$ is the statistical probability of obs for $i$, $K$ and $N$ are defined in NIST test suite. The exact distribution of $v_i$ is $\mathbb{P}(v_i) = \xi_0\mathbf{M}^i\mathbf{1}^T$, where $\xi_0$ is the initial $[1/2, 1/2, 0, \ldots, 0]_{1\times(i+1)}$, $\mathbf{1}^T$ is the transpose of the row vector $\mathbf{1} = [1, 1, \ldots, 1]_{1\times(i+1)}$, and the $(i+1)\times(1+1)$ matrix $\mathbf{M}$ is

$$\mathbf{M} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 \\ 0 & \mathbb{P}_{00} & \mathbb{P}_{01} & 0 & \cdots & 0 \\ 0 & \mathbb{P}_{10} & 0 & \mathbb{P}_{11} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \mathbb{P}_{10} & 0 & 0 & \cdots & \mathbb{P}_{11} \\ 0 & \mathbb{P}_{10} & 0 & 0 & \cdots & 0 \end{bmatrix}$$

where $\mathbb{P}_{00} = \mathbb{P}_{11} = 1/2 + \rho/2$ and $\mathbb{P}_{01} = \mathbb{P}_{10} = 1/2 - \rho/2$. The probability of longestrun is $\mathbb{P}(T_{\text{longestrun}} \text{ accepts } H_0) = h_{\text{longestrun}}(\rho,\alpha) = \text{igam}\left(\frac{K}{2}, \frac{2\text{igamc}^{-1}(K/2,\alpha)\cdot\chi^2(\text{newobs})}{\chi^2(\text{obs})}\right) - \text{igam}\left(\frac{K}{2}, \chi^2(\text{newobs})\right)$.

**Nonoverlap:** The statistic is $\chi^2(\text{obs}) = \sum_{i=1}^{N}\frac{(W_i-\mu)^2}{\sigma^2}$, where $\mu = (M-m+1)/2^m$ and $\sigma^2 = M\left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right)$. Now, we will compute the exact distribution of $W_i$. Let $W_i$ be a template with binary numbers from $\mathcal{A} = \{0,1\}$, and define the indicator variable $I_a(W_i)$ of the appearance of $W_i$ at the position $a$ is $I_a(W_i) = I[X_{a-m+1} = a_1, \ldots, X_a = a_m]$ with the expectation $\eta = \pi(a_1)\prod_{t=1}^{m-1}\mathbb{P}(a_t, a_{t+1})$. $\beta = (M-m+1)y$, where $y$ is the solution of equation $ye = \eta$. Because the Markovian hypothesis and the combinatorial structure of the problem requires some more notations in addition, $e = 1 + \sum_{t=1}^{m-1}\epsilon(t)C(t)$, where $\epsilon(t) = 1$, if there is an overlap of length $t$ two $W$s; $\epsilon(t) = 0$, otherwise. $C(t) = \mathbb{P}(a_m, a_{t+1})$, if $t = m-1$; $\mathbb{P}(a_m, a_{t+1})\prod_{l=t+1}^{m-1}\mathbb{P}(a_l, a_{l+1})$, if $t < m-1$.

The quantity $C(t)$ can be thought of as being the probability of observing the $m-t$ last letters of $W$ successively. Now, $W_i$ follows by Poisson distribution $W_i \sim Po(\eta)$. Thus, $\mathbb{P}(T_{\text{nonoverlapping}} \text{ accepts } H_0) = h_{\text{nonoverlapping}}(\rho,\alpha) = \text{igam}\left(\frac{N}{2}, \frac{2\text{igamc}^{-1}(K/2,\alpha)\cdot\chi^2(\text{newobs})}{\chi^2(\text{obs})}\right) - \text{igam}\left(\frac{N}{2}, \chi^2(\text{newobs})\right)$.

**AppEntropy:** The statistic is $\chi^2(\text{obs}) = 2N[\log 2 - ApEn(m)]$, where $ApEn = \phi^{(m)} - \phi^{(m+1)}$, $m$ is the overlapping block size and $\phi^{(m)} = \sum_{i=1}^{2^m}\frac{\text{freq of block } i}{N}\times\log(\frac{\text{freq of block } i}{N})$. Based on Markov dependent random variables, we define $\mathbb{P}_i^{(m)} = \mathbb{P}(U_{i_1})\mathbb{P}(U_{i_2}|U_{i_1})\cdot\mathbb{P}(U_{i_m}|U_{i_{m-1}})$, where the conditional probability can be calculated by transition matrix with $\rho$ in [55]. Now, we can define the new AppEntropy as $\chi^2(\text{newobs}) = 2N[\log 2 - \tilde{\phi^{(m)}} + \phi^{(m+1)}]$, where $\phi^{(m)} = \sum_{i=1}^{2^m}\mathbb{P}_i^{(m)}\log\mathbb{P}_i^{(m)}$. Thus, we obtain that $\mathbb{P}(T_{\text{AppEntropy}} \text{ accepts } H_0) = h_{\text{AppEntropy}}(\rho,\alpha) = \text{igam}\left(2^{m-1}, \frac{\text{igamc}^{-1}\left(2^{m-1},\alpha\right)\cdot\chi^2(\text{newobs})}{chi^2(\text{obs})}\right) - \text{igam}\left(2^{m-1}, \frac{\chi^2(\text{newobs})}{2}\right)$.

**1storder:** For serial test, $v_{i_m}$, $v_{i_{m-1}}$ and $v_{i_{m-2}}$ denotes the $m$-bit, $m-1$-bit and $m-2$-bit matching pattern. $U$ is each random variable from source emitting in one matching block. For Markov dependent random variables, we can obtain each matching probability of different matching pattern $\mathbb{P}_{i_m}(v_{i_m} = U_j) = \mathbb{P}(U_1)\times\mathbb{P}(U_2|U_1)\times\cdots\times\mathbb{P}(U_m|U_{m-1})$, and the expectation is $\mathbb{E}_{i_m}(v_{i_m} = U_j) = n\times\mathbb{P}(U_1)\times\cdots\times\mathbb{P}(U_m|U_{m-1})$. The statistic of $\psi_m^2 = \sum_{i_m=1}^{2^m}\frac{(v_{i_m}-\mathbb{E}_{i_m})^2}{\mathbb{E}_{i_m}}$. Then, we compute $\Delta\hat{\psi}_m^2 = \psi_m^2 - \psi_{m-1}^2$ and $\Delta^2\hat{\psi}_m^2 = \psi_m^2 - 2\psi_{m-1}^2 + \psi_{m-2}^2$. The probability of first order serial is $\mathbb{P}(T_{\text{firstorder}} \text{ accepts } H_0) = h_{\text{firstorder}}(\rho,\alpha) = \text{igam}\left(2^{m-2}, \frac{2\text{igamc}^{-1}(2^{m-2},\alpha)\cdot\Delta\hat{\psi}_m^2}{\Delta\psi_m^2}\right) - \text{igam}\left(2^{m-2}, \Delta\hat{\psi}_m^2\right)$.

**2ndorder:** Similarly, the probability of 2ndorder is $\mathbb{P}(T_{\text{secondorder}} \text{ accepts } H_0) = h_{\text{secondorder}}(\rho,\alpha) = \text{igam}\left(2^{m-3}, \frac{2\text{igamc}^{-1}(2^{m-3},\alpha)\cdot\Delta^2\hat{\psi}_m^2}{\Delta^2\psi_m^2}\right) - \text{igam}\left(2^{m-3}, \Delta^2\hat{\psi}_m^2\right)$.

## REFERENCES


[1] G. Wang, H. Cai, C. Qian, J. Han, X. Li, H. Ding, and J. Zhao, "Towards replay-resilient rfid authentication," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 385–399.

[2] X. Zhu, F. Xu, E. Novak, C. C. Tan, Q. Li, and G. Chen, "Extracting secret key from wireless link dynamics in vehicular environments," in *IEEE INFOCOM*, 2013, pp. 2283–2291.

[3] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *ACM MOBICOM*, 2009, pp. 321–332.

[4] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *IEEE INFOCOM*, 2012, pp. 927–935.

[5] S. N. Premnath, P. L. Gowda, S. K. Kasera, N. Patwari, and R. Ricci, "Secret key extraction using bluetooth wireless signal strength measurements," in *IEEE SECON*, 2014, pp. 293–301.

[6] W. Xi, C. Qian, J. Han, K. Zhao, S. Zhong, X.-Y. Li, and J. Zhao, "Instant and robust authentication and key agreement among mobile devices," in *ACM CCS*, 2016, pp. 616–627.

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

[8] Q. Wang, K. Xu, and K. Ren, "Cooperative secret key generation from phase estimation in narrowband fading channels," *IEEE J. Selected Areas in Communications*, vol. 30, no. 9, pp. 1666–1674, 2012.

[9] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *ACM MOBICOM*, 2008, pp. 128–139.

[10] J. W. Wallace and R. K. Sharma, "Automatic secret keys from reciprocal MIMO wireless channels: Measurement and analysis," *IEEE Trans. Information Forensics and Security*, vol. 5, no. 3, pp. 381–392, 2010.

[11] C. Chen and M. A. Jensen, "Secret key establishment using temporally and spatially correlated wireless channel coefficients," *IEEE Trans. Mobile Computing*, vol. 10, no. 2, pp. 205–215, 2011.

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

[13] G. R. Tsouri and D. M. Wagner, "Threshold constraints on symmetric key extraction from rician fading estimates," *IEEE Trans. Mobile Computing*, vol. 12, no. 12, pp. 2496–2506, 2013.

[14] S. T. Ali, V. Sivaraman, and D. Ostry, "Eliminating reconciliation cost in secret key generation for body-worn health monitoring devices," *IEEE Trans. Mobile Computing*, vol. 13, no. 12, pp. 2763–2776, 2014.

[15] Y. Liang, H. V. Poor, and S. Shamai, "Secure communication over fading channels," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2470–2492, 2008.

[16] W. Liu, X. Zhou, S. Durrani, and P. Popovski, "Secure communication with a wireless-powered friendly jammer," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 401–415, 2015.

[17] K. Argyraki, S. Diggavi, M. Duarte, C. Fragouli, M. Gatzianas, and P. Kostopoulos, "Creating secrets out of erasures," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 429–440.

[18] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Workshop on Theory and Application of Cryptographic Techniques*, 1993, pp. 410–423.

[19] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, "Generalized privacy amplification," *IEEE Trans. Information Theory*, vol. 41, no. 6, pp. 1915–1923, 1995.

[20] U. Maurer and S. Wolf, "Secret-key agreement over unauthenticated public channels - Part III: Privacy amplification," *IEEE Trans. Information Theory*, vol. 49, no. 4, pp. 839–851, 2003.

[21] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks *et al.*, *Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards & Technology, 2010.

[22] M. R. McKay and I. B. Collings, "General capacity bounds for spatially correlated rician mimo channels," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3121–3145, 2005.

[23] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[24] C. Kaufman, R. Perlman, and M. Speciner, *Network security: private communication in a public world*. Prentice Hall Press, 2002.

[25] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *ACM CCS*, 2007, pp. 401–410.

[26] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Trans. Mobile Computing*, vol. 9, no. 1, pp. 17–30, 2010.

[27] Y. Wei, K. Zeng, and P. Mohapatra, "Adaptive wireless channel probing for shared key generation based on PID controller," *IEEE Trans. Mobile Computing*, vol. 12, no. 9, pp. 1842–1852, 2013.

[28] S. Ponnaluri, B. Azimi-Sadjadi, Y.-K. Hue, T. Erpek, A. Komaee, and W. Trappe, "A practical wireless reciprocity-aware key establishment protocol," in *IEEE MILCOM*, 2016, pp. 1107–1113.

[29] C. Chen and M. A. Jensen, "Secret key establishment using temporally and spatially correlated wireless channel coefficients," *IEEE Trans. Mobile Computing*, vol. 10, no. 2, pp. 205–215, Feb 2011.

[30] A. Goldsmith, *Wireless communications*. Cambridge University Press, 2005.

[31] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full sha-1," in *Annual International Cryptology Conference*. Springer, 2017, pp. 570–596.

[32] H. Kim and J. Choi, "Channel estimation for one-bit massive mimo systems exploiting spatio-temporal correlations," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[33] S. Yasukawa, H. Iwai, and H. Sasaoka, "A secret key agreement scheme with multi-level quantization and parity check using fluctuation of radio channel property," in *2008 IEEE International Symposium on Information Theory*. IEEE, 2008, pp. 732–736.

[34] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.

[35] V. Henson, "An analysis of compare-by-hash." in *HotOS*, 2003, pp. 13–18.

[36] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle, "Recommendation for the entropy sources used for random bit generation," *NIST Special Publication*, vol. 800, p. 90B, 2018.

[37] C. S. Petrie and J. A. Connelly, "A noise-based ic random number generator for applications in cryptography," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 615–621, 2000.

[38] J. Machicao and O. M. Bruno, "Improving the pseudo-randomness properties of chaotic maps using deep-zoom," *Chaos: an interdisciplinary journal of nonlinear science*, vol. 27, no. 5, p. 053116, 2017.

[39] J. B. Lacy, D. P. Mitchell, and W. M. Schell, "Cryptolib: Cryptography in software." in *USENIX Security Symposium*, 1993.

[40] G. L. Jones *et al.*, "On the markov chain central limit theorem," *Probability surveys*, vol. 1, no. 299-320, pp. 5–1, 2004.

[41] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for ofdm," *IEEE transactions on communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[42] X. Huang and H.-C. Wu, "Robust and efficient intercarrier interference mitigation for ofdm systems in time-varying fading channels," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2517–2528, 2007.

[43] H. Taha and E. Alsusa, "Physical layer secret key exchange using phase randomization in mimo-ofdm," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[44] S. N. Premnath, J. Croft, N. Patwari, and S. K. Kasera, "Efficient high-rate secret key extraction in wireless sensor networks using collaboration," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 1, p. 2, 2014.

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

[46] Y. Luo, L. Pu, Z. Peng, and Z. Shi, "Rss-based secret key generation in underwater acoustic networks: advantages, challenges, and performance improvements," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 32–38, 2016.

[47] T. Wang, Y. Liu, and A. V. Vasilakos, "Survey on channel reciprocity based key establishment techniques for wireless systems," *Wireless Networks*, vol. 21, no. 6, pp. 1835–1846, 2015.

[48] 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.

[49] J. Huang and T. Jiang, "Dynamic secret key generation exploiting ultra-wideband wireless channel characteristics," in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2015, pp. 1701–1706.

[50] Z. Li, Q. Pei, I. Markwood, Y. Liu, and H. Zhu, "Secret key establishment via rss trajectory matching between wearable devices," *IEEE Transactions on Information Forensics and security*, vol. 13, no. 3, pp. 802–817, 2017.

[51] L. Y. Paul, J. S. Baras, and B. M. Sadler, "Physical-layer authentication," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 38–51, 2008.

[52] M. Zhao, A. Jha, Q. Liu, B. A. Millis, A. Mahadevan-Jansen, L. Lu, B. A. Landman, M. J. Tyskac, and Y. Huo, "Faster mean-shift: Gpu-accelerated embedding-clustering for cell segmentation and tracking," *arXiv preprint arXiv:2007.14283*, 2020.

[53] L. Y. Paul, G. Verma, and B. M. Sadler, "Wireless physical layer authentication via fingerprint embedding," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 48–53, 2015.

[54] J. Liu and X. Wang, "Physical layer authentication enhancement using two-dimensional channel quantization," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4171–4182, 2016.

[55] B. Lindqvist, "A note on bernoulli trials with dependence," *Scandinavian Journal of Statistics*, pp. 205–208, 1978.

[56] Y. Wang and Z. Yang, "On a markov multinomial distribution," *Mathematical Scientist*, vol. 20, no. 1, pp. 40–49, 1995.