Bit Security Estimation for Leakage-Prone Key Establishment Schemes

Marcus de Ree, *Member, IEEE,* Georgios Mantas, *Member, IEEE,* and
Jonathan Rodriguez, *Senior Member, IEEE*

***Abstract*—The security guarantees of cryptographic primitives are subject to the assumption that established keys are known only by the legitimate users and no information about the key bits is known by illegitimate users. Unfortunately, this assumption may not be applicable in leakage-prone key establishment schemes. Namely, information leakage about an established key (defined as a bit inference rate of an adversary that is strictly greater than 50%) reduces its computation effort required in an exhaustive key search. In this paper, we present a methodology and a polynomial-time algorithm that determines the exact impact of information leakage on a generated bit sequence and expressed these findings in terms of the achieved level of bit security. Additional simulation results enable us to determine the achieved level of bit security of a leakage-prone bit sequence or, conversely, enable us to determine the length of a bit sequence necessary to achieve a selected level of bit security.**

***Index Terms*—Cryptography, Information Leakage, Key Generation, Physical Layer Security, Security Analysis**

I. Introduction

KEY establishment covers a variety of techniques (e.g., key agreement, key distribution) which enable two honest nodes, commonly named Alice and Bob, to gain possession of a shared (i.e., symmetric) secret key [1]. For wireless communication, the physical layer can be leveraged for on-demand and low-complexity key establishment while independent from a complex system architecture [2]. These benefits make them attractive for resource constrained (e.g., Internet-of-Things) networks, delay-sensitive (e.g., autonomous vehicles) networks, and future 6G network [3]. From the literature, we can distinguish between two classes for physical layer-based key establishment:

The first class covers (information) theoretical schemes, initiated by Wyner [4], Maurer [5], and Ahlswede and Csiszar [6]. These schemes make abstractions about the channel between the legitimate nodes, Alice and Bob (i.e., *main channel*), and the channel towards an eavesdropper, Eve (i.e., *eavesdropper channel*). Unfortunately, the feasibility for key establishment relies on an advantage in the quality of the main channel with respect to the eavesdropper channel (i.e., a strictly positive secrecy capacity). This advantage can be either natural or created artificially (e.g., by taking advantage of supporting nodes which transmit signals simultaneously with Alice and Bob to provide constructive interference in the main channel and/or destructive interference in the eavesdropper channel [7]). Despite formal security proofs, there is a lack of experimental results which prove that the channel abstractions capture the real-world setting [8].

The second class covers practice-oriented schemes, inspired by works from Tope and McEachen [9], Aono et al. [10], and Mathur et al. [11]. These schemes exploit the principles of *channel reciprocity*, implying that characteristics of a shared wireless channel are reciprocal, and *temporal* and *spatial decorrelation*, implying that the channel characteristics decorrelate over time and in space, respectively. In these schemes, Alice and Bob exchange probing signals to estimate a channel characteristic (e.g., received signal strength) of which their measurements are highly correlated. The aim is to subsequently quantize these measurements into a corresponding sequence of bits such that a bit disagreement rate (BDR) of 0 is achieved (a commonly used performance metric which determines the fraction of bits that disagree in Alice and Bob's bit sequence). Many works (e.g., [11]–[14]) assume an environment where channel characteristics rapidly decorrelate in space and Eve to be sufficiently separated from Alice and Bob (i.e., at least one-half or a few wavelengths) such that Eve's measurements are independent of those observed by Alice and Bob [15], [16]. The BDR between Eve's and Alice and Bob's bit sequence would therefore be approximately 0.5. Alice and Bob can therefore generate an n-bit sequence that achieves n bits of uncertainty at Eve. We omit details related to *information reconciliation* and *privacy amplification* since these are not relevant for the remainder of this paper.

Experimental studies have shown that Eve's measurements are at least weakly correlated such that a fraction strictly greater than 0.5 would agree between Eve's bit sequence and Alice and Bob's bit sequence. We can also phrase this as the fraction of bits that Eve correctly infers from her measurements and refer to this metric as the bit inference rate (BIR), the converse of the BDR (i.e., BIR = 1 - BDR). The experimental platforms of [17] and [18] simulated various indoor and outdoor environments with Eve closely located to Alice (one-half to seven wavelengths). Both platforms implemented a quantization scheme and determined the fraction of Eve's bit sequence that disagrees (or conversely, agrees) with Alice and Bob's bit sequence. In [17], it was shown that Eve

This research was sponsored by the NATO Science for Peace and Security Programme under grant SPS G5797.

Marcus de Ree is with the Instituto de Telecomunicações, 3810-193 Aveiro, Portugal (e-mail: mderee@av.it.pt).

Georgios Mantas is with the Instituto de Telecomunicações, 3810-193 Aveiro, Portugal, and also with the Faculty of Engineering and Science, University of Greenwich, Chatham Maritime ME4 4TB, U.K. (email: gimantas@av.it.pt).

Jonathan Rodriguez is with the Instituto de Telecomunicações, 3810-193 Aveiro, Portugal, and also with the Faculty of Computing, Engineering and Science, University of South Wales, Pontypridd CF37 1DL, U.K. (e-mail: jonathan@av.it.pt).

can achieve a BDR of approximately 0.38 when located at one wavelength from Alice, corresponding to a BIR of 0.62. In [18], it was shown that Eve can achieve a BIR of 0.55 when located at seven wavelengths of Alice, corresponding to a BDR of 0.45. Although Eve has no way of knowing which bits were incorrectly inferred, the leaked information could improve her chances of uncovering Alice and Bob's bit sequence. Namely, the leaked information reduces the effort required from Eve since its search space follows a *non-uniform probability distribution* (i.e., certain bit sequences are more likely than others). Furthermore, it is important to mention that all nodes had the same hardware specifications in the experimental platforms [17], [18]. We may therefore assume that Eve's ability to infer bits of Alice and Bob's bit sequence improves with better hardware.

A. *Assumptions*

Eve has complete knowledge of Alice and Bob's key establishment and encryption scheme. Furthermore, the encryption scheme is assumed *not* to be the one-time pad such that the key is used to encrypt large amounts of plaintext data. The encrypted (i.e., ciphertext) data produced by one key, which can be eavesdropped on and stored by Eve, is assumed to be sufficient such that Eve can conclusively determine that key. Namely, Eve can trial all possible keys in the key space such that the decryption algorithm will return a logical plaintext if and only if the inputs of the decryption algorithm are the ciphertext data and Alice and Bob's key. Based on these assumptions, it is necessary to evaluate the level of computational security that Alice and Bob's key achieve since unconditional security (i.e., information-theoretic security) does not apply.

Computational security concerns with the computational effort required to break a security system. The security system is *computationally secure* if the best algorithm for breaking it requires some very large number ($N = 2^n$) of operations [19], where n represents the measure of bit security. In the context of this paper, we assume that the optimal exhaustive key search attack (see section II-A) is the best algorithm to break a leakage-prone key establishment system. Determining the computational effort in launching the optimal exhaustive key search attack was first attempted by Massey [20] and followed-up by McEliece and Yu [21] although they merely provided upper and lower bounds of N (i.e., the computational effort required by the adversary). To the authors' best knowledge, a polynomial-time solution to determine the adversary's computational effort has never been published.

B. *Contributions*

This paper's contributions can be summarized as follows:

- This paper presents a polynomial-time solution to determine the exact level of computational security (i.e., bit security) achieved by an n-bit key against an optimal exhaustive key search where Eve infers the value of secret key bits at a rate *strictly greater than* 0.5 (i.e., BIR) or, conversely, that Eve incorrectly infers the value of secret key bits at a rate *strictly smaller than* 0.5 (i.e., BDR).
- The results provided in this paper, with respect to the problem of information leakage in the generated bit sequence of a physical layer-based key establishment scheme following the channel reciprocity principle, enable us to determine the length of the n-bit sequence (i.e., key) that Alice and Bob must generate in order for this sequence to achieve m bits of security at Eve.
- The presented work is also applicable to other security systems (e.g., resource constrained devices) that lack a trusted platform module (TPM) due to their potential to leak information about the bits of generated or derived keys from side channel attacks (e.g., time and power analysis).

II. Mathematical Model

A. *Exhaustive Key Search*

We consider a discrete random variable K, which consists of the key space $\mathcal{K} = \mathbb{Z}_2^n$ containing all possible n-bit keys. The discrete random variable K takes on the value $\boldsymbol{k}$, where $\boldsymbol{k}$ represents the n-bit key selected at random by an honest node and unknown to an adversary. We denote the probability that the discrete random variable K takes on the value of key $\boldsymbol{k}_i \in \mathcal{K}$ as follows:

$$\Pr[K = \boldsymbol{k}_i \mid 1 \leq i \leq 2^n] = p_i \tag{1}$$

Without loss of generality, we suppose that the probability distribution $P_K = (p_1, p_2, \cdots, p_{2^n})$ satisfies $p_1 \geq p_2 \geq \cdots \geq p_{2^n}$. Following this notation, the optimal strategy of the adversary is to trial keys $\boldsymbol{k}_i$ for increasing values of i. This can be interpreted as the adversary executing the decryption algorithm (with the eavesdropped ciphertext data and the trialed key $\boldsymbol{k}_i$ as inputs) and determine whether the trialed key $\boldsymbol{k}_i$ was correct based on whether the decryption algorithm returned a logical plaintext.

B. *Definition of Guesswork*

The guesswork (also referred to as "guessing entropy") of a discrete random variable estimates the expected number of guesses (i.e., the N operations) required by an adversary to determine its value while following an optimal strategy [22], [23]:

$$N = \sum_{i=1}^{2^n} i \, p_i \tag{2}$$

Massey [20] and McEliece and Yu [21] were the first to study this problem and showed that the guesswork is under bounded and upper bounded, respectively, in terms of the entropy function $H(K)$ as follows:

$$2^{H(K)-2} + 1 \leq N \leq \frac{2^n - 1}{2n} H(K) + 1 \tag{3}$$

$$H(K) = -\sum_{\boldsymbol{k}_i \in \mathcal{K}} p_i \log_2(p_i) \tag{4}$$

We will show that these bounds become, unfortunately, inaccurate for determining the level of computational security (i.e., bit security) achieved for increasing levels of information leakage (see Section IV-A).

III. Bit Security Estimation for Leakage-Prone Key Establishment Schemes

In this section, we provide a polynomial-time solution for estimating the guesswork of an adversary under the assumption that information leakage allows the adversary to infer bits of the n-bit secret key $\boldsymbol{k}$ with a BIR denoted by α. The optimal exhaustive key search consists of trialing keys $\boldsymbol{k}_i$ in order of non-increasing probability. We denote the n-bit secret key as $\boldsymbol{k} = [b_1, \cdots, b_n]$ and the adversary's estimate of each key bit b_i as $\hat{b}_i$ such that $\Pr[b_i = \hat{b_i}] = \alpha$.

The adversary starts its key search with the most probable key $\boldsymbol{k}_1 = [\hat{b}_1, \cdots, \hat{b}_n]$ such that $\Pr[K = \boldsymbol{k}_1] = \alpha^n$. If the adversary finds that $K \neq \boldsymbol{k}_1$, it will assume that one of the estimated bits $\hat{b}_i$ was incorrectly inferred and proceeds to trial the second-most probable keys $\{\boldsymbol{k}_2, \cdots, \boldsymbol{k}_{n+1}\}$ where one of the n inferred bits $\hat{b}_i$ is flipped such that $\Pr[K = \boldsymbol{k}_i \mid 2 \leq i \leq n+1] = \alpha^{n-1}(1-\alpha)$. This process continues until the adversary finds secret key $\boldsymbol{k}$.

We can generalize this process as an exhaustive key search that consists of $n+1$ iterations. Initially, no mismatches between the inferred bits $\hat{b}_i$ and the key bits b_i are assumed, whereas subsequent iterations consider one additional mismatch. We denote the set of keys to be trialed during iteration k by Ω_k (for $0 \leq k \leq n$) such that the probability of each key $\boldsymbol{k}_i \in \Omega_k$ trialed during iteration k equals:

$$\Pr[K = \boldsymbol{k}_i \mid \boldsymbol{k}_i \in \Omega_k] = \alpha^{n-k}(1-\alpha)^k \tag{5}$$

We must also determine the ordinalities at which the keys are being trialed. First, we determine the number of keys that are trialed during iteration k (i.e., the cardinality of Ω_k), denoted as $|\Omega_k|$, using the binomial formula:

$$|\Omega_k| = \binom{n}{k} \tag{6}$$

Based on these cardinalities, we can determine the ordinalities of the keys trialed during iteration k. Namely, the ordinality of the first and last key trialed during iteration k can be estimated by the cardinalities of the previous iterations and the current iteration. The sum of ordinalities for iteration k, given key length n, can be described as a function $s(n,k)$:

$$s(n,k) = \sum_{i=a}^{b} i \text{ for } \begin{cases} a = \sum_{j=0}^{k} |\Omega_{j-1 \bmod n+1}| \\ b = \sum_{j=0}^{k} |\Omega_j| \end{cases} \tag{7}$$

For algorithmic purposes, we can utilize the sum of consecutive integers (i.e., $\sum_{i=a}^{b} i = \frac{(a+b)(b-a+1)}{2}$) to compute the value of the function $s(n,k)$. Combining the results from (5) and (7) enables us to rewrite (2) such that the guesswork of an adversary can be expressed in terms of the key length (n) and the BIR of the adversary (α):

$$N = \sum_{k=0}^{n} s(n,k)\, \alpha^{n-k}(1-\alpha)^k \tag{8}$$

Finally, we convert the guesswork N, required by an adversary to uncover an n-bit secret key, into the achieved level of bit security m as follows:

$$m = \log_2(2N - 1) \tag{9}$$

The legitimacy of the conversion formula can be verified by showing that it satisfies the following two properties:

1) The bit security level m should equal the key length n when no information is leaked to the adversary (i.e., $m = n$ for $\alpha = 0.5$).

Proof. We prove that the above property is satisfied through algebraic derivation.

$$\begin{aligned} N &= \sum_{k=0}^{n} s(n,k)\, 2^{-(n-k)}\, 2^{-k} && \text{(10a)} \\ &= 2^{-n} \sum_{k=0}^{n} s(n,k) && \text{(10b)} \\ &= 2^{-n}\, \frac{2^n\,(2^n+1)}{2} && \text{(10c)} \\ &= \frac{2^n+1}{2} && \text{(10d)} \end{aligned}$$

Combining (9) and (10d) yields:

$$m = \log_2(2\,(\frac{2^n+1}{2}) - 1) = n \tag{11}$$

■

2) The bit security level m should equal zero when all information about the bits are leaked to the adversary (i.e., $m = 0$ for $\alpha = 1$).

Proof. We prove that the above property is satisfied through algebraic derivation (where $0^0 = 1$).

$$\begin{aligned} N &= \sum_{k=0}^{n} s(n,k)\, 1^{n-k}\, (1-1)^k && \text{(12a)} \\ &= s(n,0) + \sum_{k=1}^{n} s(n,k)\, 1^{n-k}\, 0^k && \text{(12b)} \\ &= 1 + 0 && \text{(12c)} \end{aligned}$$

Combining (9) and (12c) yields:

$$m = \log_2(2 - 1) = 0 \tag{13}$$

■

Based on the above described methodology which allowed us to formulate (8), we designed a polynomial-time algorithm[1] (see Algorithm 1). It inputs a key length (n) and adversarial BIR (α) to estimate the guesswork for each of the $n+1$ iterations (represented as a for-loop), sums these estimations to obtain the total guesswork, and converts this into the output bit security (m). The computational complexity of the algorithm is determined by this for-loop and therefore equals $O(n)$.

IV. Simulation Results

This section present our three main results: (i) the confirmation that earlier bounds on an adversary's guesswork can be inaccurate, (ii) the impact of information leakage in terms of (relative) bit security achieved, and (iii) the key length required to achieve 128- and 256-bit security against computational adversaries under varying information leakage assumptions.

[1] Available online as a Python script at https://github.com/mderee/Public-Scripts/blob/main/Security-vs-Leakage for reproducibility purposes.

Algorithm 1: Bit Security Estimation from Guesswork

```
Input  : Key length $n \in \mathbb{Z}^+$, adversarial BIR
         $\alpha \in [0.5, 1]$.
Output: Bit security $m$.
begin
    $b \leftarrow 0$
    $N \leftarrow 0$
    for $k = 0$ to $n$ do
        $a \leftarrow b + 1$
        $|\Omega| \leftarrow \binom{n}{k}$
        $b \leftarrow b + |\Omega|$
        $s(n, k) \leftarrow \frac{(a+b)(b-a+1)}{2}$
        $N \leftarrow N + s(n, k)\, \alpha^{n-k}(1 - \alpha)^k$
    end
    $m \leftarrow \log_2(2N - 1)$
    return $m$
end
```

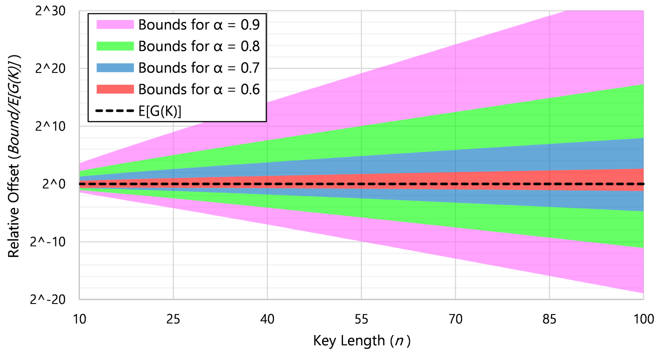

Fig. 1. The accuracy of the lower and upper bounds on guesswork are shown as a relative offset compared to the baseline ($N = 1$). The lower and upper bounds are shown below and above the baseline, respectively.

A. *Evaluation of the Guesswork Bounds*

Massey [20] and McEliece and Yu [21] proposed a lower bound and upper bound, respectively, on the guesswork required by an adversary. To determine the accuracy of these bounds, we considered varying levels of information leakage (such as $\alpha \in \{0.6, 0.7, 0.8, 0.9\}$) and key lengths ($10 \leq n \leq 100$). For each combination, we computed the lower and upper bounds on the adversary's guesswork using (3) and we determined the actual guesswork using Alg. 1 (excluding the step that converts the guesswork to bit security). These results were subsequently normalized such that the lower and upper bounds become a value relative to the guesswork baseline. These results are graphed and shown in Fig. 1.

The results show that the lower and upper bounds are relative close approximations of the guesswork when there is limited information leakage. For example, for $\alpha = 0.6$ and $n = 72$, the lower bound is approximately half (2^{-1}) and the upper bound is approximately quadruple (2^2) the actual guesswork. These bounds could therefore recommend a key length (aiming to achieve a specified level of bit security) that will only be one bit longer than necessary. However, these bounds become very inaccurate when we make more conservative assumptions on the level of information leakage. Namely, these bounds would recommend key lengths approximately 20% larger than necessary when we assume $\alpha = 0.9$.

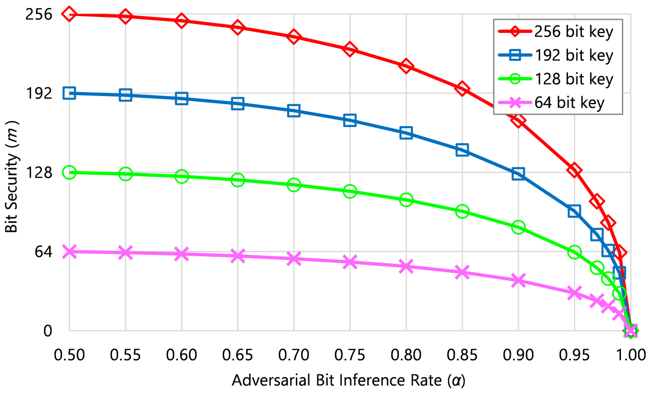

Fig. 2. The level of bit security (m) achieved as a function of the adversarial bit inference rate (α) for varying key lengths.

B. *Bit Security Loss from Information Leakage*

In this subsection, we determine the level of bit security m that a key of length n achieves despite an adversarial BIR of α. For a given key length n and adversarial BIR α, we utilized Alg. 1 to determine its bit security m. For a given key length (such as $n \in \{64, 128, 192, 256\}$), we plotted the achieved level of bit security m as a function of the adversarial BIR α as shown in Fig. 2.

From Fig. 2, we can see that even a moderate amount of information leakage ($\alpha \approx 0.7$) has a relatively low impact on the achieved level of bit security. This indicates that relatively few additional bits are necessary in a leakage-prone key establishment scheme to establish keys that achieve a specified level of bit security. However, when considering a very high amount of information leakage ($\alpha \approx 0.95$) the key size has to be doubled to achieve a specified level of bit security.

From these results, we can determine a relative level of achieved bit security (m/n) as shown in Fig. 3. It can be observed that the relative level of achieved bit security is nearly identical for key lengths varying between 64 and 256. For these key lengths, the relative level of achieved bit security is approximately 97.5%, 90%, 75%, and 50% for low ($\alpha \approx 0.6$), moderate ($\alpha \approx 0.7$), high ($\alpha \approx 0.85$), and very high ($\alpha \approx 0.95$) amounts of information leakage, respectively.

C. *Key Length to Achieve Security*

In this subsection, we determine the key lengths necessary to achieve an arbitrary level of bit security under the assumption of information leakage. To obtain these results, we considered varying levels of information leakage (such as $\alpha \in \{0.60, 0.70, 0.80, 0.90, 0.95\}$) and key lengths ($0 \leq n \leq 500$) after which we computed the achieved level of bit security for each combination utilizing Alg. 1. The resulting data is shown in Fig. 4.

The required key length can be determined through intersection. For example, given a low amount of information leakage

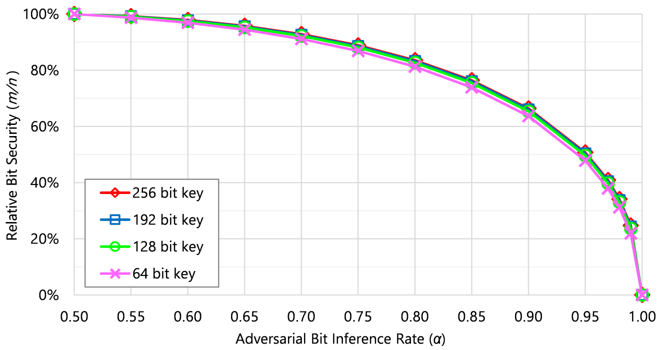

Fig. 3. The relative level of bit security (m/n) achieved as a function of the adversarial bit inference rate (α) for varying key lengths.

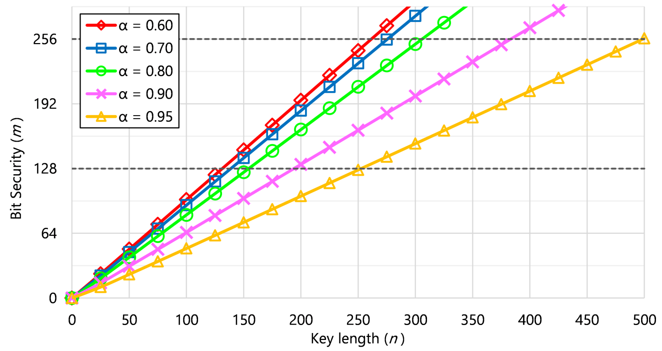

Fig. 4. The level of bit security achieved (m) as a function of the key length (n) under varying information leakage assumptions. The key length required to achieve 128- or 256-bit security based on the adversarial bit inference rate α can be determined from the intersection with lines $m = 128$ and $m = 256$, respectively.

($\alpha = 0.6$) then a 132-bit key is sufficient to achieve 128 bits of security (262-bit key to achieve 256 bits of security). However, if we assume a very high amount of information leakage ($\alpha = 0.95$) then a 253-bit key is required to achieve 128 bits of security (499-bit key to achieve 256 bits of security). In practice, this can be applied by having the leakage-prone key establishment scheme generate a 253 (499)-bit sequence that is subsequently fed into a secure hashing algorithm which returns a 128 (256)-bit key providing the estimated 128 (256) bits of security.

V. Conclusion

This paper tackled the problem of measuring the impact of information leakage in leakage-prone key establishment schemes. The paper presents a polynomial-time solution that enables the user to estimate the key length necessary to achieve a specified level of bit security under particular information leakage assumptions. Furthermore, it was shown that conservative assumptions on information leakage can be made without having to cause a severe bit security loss (0-10%). A reader can utilize the presented solution to determine how long the bit sequence should be, generated by a leakage-prone key establishment scheme (achieving a pre-selected level of bit security), such that it can subsequently hash this bit sequence to determine a key where the length of the key and the achieved level of bit security correspond.

References

[1] M. de Ree, G. Mantas, A. Radwan, S. Mumtaz, J. Rodriguez, and I. E. Otung, "Key Management for Beyond 5G Mobile Small Cells: A Survey," *IEEE Access*, vol. 7, pp. 59200-59236, 2019.

[2] M. Bloch, and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering*, Cambridge University Press, 2011.

[3] M. Ylianttila, R. Kantola, A. Gurtov, L. Mucchi, and I. Oppermann, *6G White Paper: Research Challenges for Trust, Security and Privacy*, University of Oulo, 2020. [Online]. Available: http://jultika.oulu.fi/files/isbn9789526226804.pdf

[4] A. D. Wyner, "The Wiretap Channel," *Bell System Technical Journal*, vol. 54, pp. 1355-1387, 1975.

[5] U. M. Maurer, "Secret Key Agreement by Public Discussion from Common Information," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 733-742, 1993.

[6] R. Ahlswede and I. Csiszar, "Common Randomness in Information Theory and Cryptography: I. Secret Sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121-1132, 1993.

[7] P. Mukherjee, J. Xie, and S. Ulukus, "Secure Degrees of Freedom of One-Hop Wireless Networks with no Eavesdropper CSIT," *IEEE Trans. Inf. Theory*, vol. 63, no. 3, pp. 1898-1922, 2017.

[8] M. Edman, A. Kiayias, Q. Tang and B. Yener, "On the Security of Key Extraction from Measuring Physical Quantities," *IEEE. Trans. Inf. Forensics Secur.*, vol. 11, no. 8, pp. 1796-1806, 2016.

[9] M. A. Tope and J. C. McEachen, "Unconditionally Secure Communications over Fading Channels," *IEEE Mil. Commun. Conf. (MILCOM)*, McLean, VA, USA, 2001, pp. 54-58.

[10] T. Aono, K. Higuachi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless Secret Key Generation Exploiting Reactance-Domain Scalar Response of Multipath Fading Channels," *IEEE Trans. Antennas Propag.*, vol. 53, no. 11, pp. 3776-3784, 2005.

[11] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-Telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel," *ACM Int. Conf. Mob. Comput. Sys. (MobiCom)*, San Francisco, CA, USA, 2008, pp. 128-139.

[12] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krisnamurthy, "Secret Key Extraction from Wireless Signal Strength in Real Environments," *IEEE Trans. Mob. Comput.*, vol. 12, no. 5, pp. 917-930, 2013.

[13] J. Zhang, S. Rajendran, Z. Sun, R. Woods, and L. Hanzo, "Physical Layer Security for the Internet of Things: Authentication and Key Generation," *IEEE Wirel. Commun.*, vol. 26, no. 5, pp. 92-98, 2019.

[14] N. Aldaghri and H. Mahdavifar, "Physical Layer Secret Key Generation in Static Environments," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2692-2705, 2020.

[15] W. Trappe, "The Challenges Facing Physical Layer Security," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 16-20, 2015.

[16] M. de Ree, G. Mantas, and J. Rodriguez, "A Cryptographic Perspective to Achieve Practical Physical Layer Security," in *IEEE Global Commun. Conf. (GLOBECOM)*, Rio de Janeiro, Brazil, 2022, pp. 4038-4043.

[17] R. Guillaume, F. Winzer, and A. Czylwik, "Bringing PHY-based Key Generation into the Field: An Evaluation for Practical Scenarios," *IEEE Vehic. Technol. Conf. (VTC)*, Boston, MA, USA, 2015, pp. 1-5.

[18] M. Edman, A. Kiayias, and B. Yener, "On Passive Inference Attacks against Physical-Layer Key Extraction," *ACM Europ. Wksp. Sys. Secur. (EuroSec)*, Salzburg, Austria, 2011, pp. 1-6.

[19] D. R. Stinson, *Cryptography: Theory and Practice*, 3rd ed., CRC Press, 2006.

[20] J. L. Massey, "Guessing and Entropy," in *Int. Symp. Inf. Theory (ISIT)*, Trondheim, Norway, 1994, pp. 204.

[21] R. J. McEliece, and Z. Yu, "An Inequality on Entropy," in *Int. Symp. Inf. Theory (ISIT)*, Whistler, BC, Canada, 1995, pp. 329.

[22] C. Cachin, *Entropy Measures and Unconditional Security in Cryptography*, Ph.D. dissertation, ETH Zürich, Zürich, Switzerland, 1997.

[23] J. O. Pliam, "Guesswork and Variation Distance as Measures of Cipher Security," in *Int. Wksp. Selected Areas in Cryptography (SAC)*, Kingston, ON, Canada, 1999, pp. 62-77.