High-Rate Secret Key Generation Using Physical Layer Security and Physical Unclonable Functions

Tasneem Assaf ¹, Arafat Al-Dweik ², Youssef Iraqi ², Sobia Jangsher ², Anshul Pandey ², Jean-Pierre Giacalone ², Enas Abulibdeh ², Hani Saleh ², and Baker Mohammad ²

¹Khalifa University
²Affiliation not available

October 30, 2023

Abstract

Physical layer security (PLS) can be adopted for efficient key sharing in secured wireless systems. The random nature of the wireless channel and channel reciprocity (CR) are the main pillars for realizing PLS techniques. However, for applications that involve air-to-air (A2A) transmission, such as unmanned aerial vehicle (UAV) applications, the channel does not generally have sufficient randomness to enable reliable key generation. Therefore, this work proposes a novel system design to mitigate the channel randomness constraint and enable high-rate secret key generation (SKG) process. The proposed system integrates physically unclonable functions (PUFs) and CR principle to securely exchange secret keys between two nodes. Moreover, an adaptive and controllable artificial fading (AF) level with interleaving is used to mitigate the impact of low randomness variations in the wireless channel. The proposed system can operate efficiently even when the channel is nearly flat or time invariant. Consequently, the time required for generating and sharing a key is significantly shorter than conventional techniques.

We also propose a novel bit extraction scheme that reduces the number of overhead bits required to share the intermediate keys. The obtained Monte Carlo simulation results show that a key agreement can be reached at the first trial for moderate and high signal-to-noise ratios (SNRs), which is substantially faster than other PLS techniques. Moreover, the results show that inducing AF into static channels reduces the mismatch ratio between the generated secret sequences and degrades the eavesdropper’s capability to predict the secret keys.
High-Rate Secret Key Generation Using Physical Layer Security and Physical Unclonable Functions

T. Assaf, Member, IEEE, A. Al-Dweik, Senior Member, IEEE, Y. Iraqi, Senior Member, IEEE, S. Jangsher, Member, IEEE, A. Pandey, Member, IEEE, JP. Giacalone, Member, IEEE, E. Abulibdeh, Member, IEEE, H. Saleh, Senior Member, IEEE, and B. Mohammad, Senior Member, IEEE

Abstract—Physical layer security (PLS) can be adopted for efficient key sharing in secured wireless systems. The random nature of the wireless channel and channel reciprocity (CR) are the main pillars for realizing PLS techniques. However, for applications that involve air-to-air (A2A) transmission, such as unmanned aerial vehicle (UAV) applications, the channel does not generally have sufficient randomness to enable reliable key generation. Therefore, this work proposes a novel system design to mitigate the channel randomness constraint and enable high-rate secret key generation (SKG) process. The proposed system integrates physically unclonable functions (PUFs) and CR principle to securely exchange secret keys between two nodes. Moreover, an adaptive and controllable artificial fading (AF) level with interleaving is used to mitigate the impact of low randomness variations in the wireless channel. The proposed system can operate efficiently even when the channel is nearly flat or time invariant. Consequently, the time required for generating and sharing a key is significantly shorter than conventional techniques. We also propose a novel bit extraction scheme that reduces the number of overhead bits required to share the intermediate keys. The obtained Monte Carlo simulation results show that a key agreement can be reached at the first trial for moderate and high signal-to-noise ratios (SNRs), which is substantially faster than other PLS techniques. Moreover, the results show that inducing AF into static channels reduces the mismatch ratio between the generated secret sequences and degrades the eavesdropper’s capability to predict the secret keys.

Index Terms—Physical layer security (PLS), channel reciprocity (CR), physically unclonable function (PUF), secret key generation (SKG), static environments, artificial fading (AF), bit extraction (BE), received signal strength (RSS).

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are currently invading several sectors with applications in agriculture, logistics, transportation, energy, construction, media, entertainment, etc. Because of their operational flexibility, low cost, and small size, UAVs manifest themselves as the perfect tool for reconnaissance, surveillance, mapping, and surveying. The overall UAV market in 2021 is estimated to be $27.4 billion and is projected to reach $58.4 billion by 2026 [1]. The popularity of UAVs is highly correlated with the development of Internet of things (IoT) technology, and UAVs are currently considered as an integral element of IoT infrastructure where they are used for data collection, relaying, data distribution, etc. [2].

In certain applications, multiple UAVs can be jointly assigned a remote mission that requires secure data communications. For such applications, physical layer security (PLS) can be considered attractive due to the limited computational power and tight energy budget of the UAVs. In particular, PLS can facilitate the key generation and distribution processes and reduce the overhead signaling that is required for other key distribution techniques [3]. PLS techniques can be generally divided into two main categories: signal to interference plus noise ratio (SINR)-based and complexity-based PLS. The main focus of this work is the complexity-based PLS, which is associated with the mechanisms of extracting and sharing a secret sequence by utilizing the shared channel between legitimate users. PLS mechanisms leverage the random and reciprocal characteristics of wireless channels to achieve information-theoretical security [4]. Most existing complexity-based PLS schemes are designed for systems that adopt time division duplexing (TDD) to enable utilizing the channel reciprocity (CR) [3], [5], [6]. PLS generally requires rich and dynamic wireless channels. The richness of the channel is required to enable reliable key generation while the channel dynamics are required to maximize the difference between consecutive keys. Consequently, the channel information in time [7], [8], frequency [5], [9] and space domains [10], [11] can be utilized to enable reliable key generation.

In order to enhance the security and randomness levels of complexity-based PLS systems, physically unclonable functions (PUFs) can be added as a second layer of security. The concept of PUFs was first introduced in [12]. The idea is that the integrated circuits (ICs) have a uniqueness in their physical structure which is inherited from inevitable variations during the fabrication process. These unique characteristics are unpredictable before the end of the manufacturing process and can be considered as device fingerprints. Due to their physical unclonability and high resistance to reverse engineering, PUFs have shown great promise as hardware identification primi-
tives for cryptography applications such as authentication and secret key generation (SKG) [13]. The unclonability of PUFs means that for a given fabrication procedure, it is infeasible to reproduce the same physical structure [14]. Moreover, as compared to traditional cryptography techniques, PUFs require significantly less computational capacity as there is no need for permanent storage to secure the generated secret keys [15], [16]. PUFs are commonly characterized by a set of challenge-response pairs (CRPs) based on the unique circuits variations. The PUF response for a certain challenge that is measured under certain conditions, such as temperature and voltage, is called the "original response." The obtained responses from a PUF are sensitive to the environmental changes and physical conditions where the device is being tested. In other words, the readings from the PUFs are not perfectly reproducible. Therefore, error correction mechanisms such as fuzzy extractors are used to correct the mismatches with the original response [16].

A. Related Work

Complexity-based PLS techniques are used for key generation by exploiting the inherent randomness of the channel, and the principle of CR between the transmitter and receiver [3], [17], [18]. Unlike classical key distribution techniques, PLS does not involve the explicit exchange of keys. Therefore, it is difficult for eavesdroppers to tap the key. The PLS-based SKG is explored in the literature for various networks, and channel settings [17], [19]–[21]. In [19]–[21] and the references listed therein, SKG is studied for static and dynamic environments. The dynamic environments have high temporal variations that enable generating keys with high entropy, which leads to a high key generation rate (KGR).

SKG is a challenging task in poor scattering environments where the channel randomness or variations are limited due to the large coherence time or wide coherence bandwidth of the channel. Consequently, the key generation process, which requires strong time or frequency variations, will mostly fail causing low KGR. In [20], the authors conducted an experiment inside an underground concrete tunnel to exclude most external interference sources and the effects of channel variation due to any surroundings’ mobility. The obtained KGR was extremely low, about one 256 bit key every 7 minutes. In addition to the failure of the key generation process, the dominance of the independent hardware noise, i.e. additive white Gaussian noise (AWGN), at the legitimate nodes over the time or frequency selectivity of the channel, increases the key mismatch probability considerably. In the literature, several approaches were proposed to overcome such challenges, which include using relays [22], multiple-input multiple-output (MIMO) [23], intelligent reflective surface (IRS) [24], or by inducing artificial randomness [25]. In [26], [27], opportunistic randomized beamforming with a diversity mechanism is proposed. Generating artificial interference for the eavesdropper is proposed in [28], [29]. An induced randomness for SKG is studied in [6] for static channels. However, the work considers that the eavesdropper’s channel is independent of the legitimate users’ channel, which is not generally a valid assumption in several cases of interest. Moreover, the induced randomness is not common among the system users, and its level is not guaranteed or adaptive. This can lead to high estimation errors or extra complexity, which can be avoided in the case of high channel randomness. In [26], the authors propose using IRS with discrete phase shifts for SKG. The channel coefficients are used to generate the secret keys.

Furthermore, in the existing PLS work with poor scattering or static environments, it is assumed that the legitimate users’ channel is independent of the eavesdropper channel, given that the legitimate users are at least half a wavelength apart. However, this assumption is valid only in sufficiently rich scattering environments. In free space communications, which has poor scattering environments, such as air-to-air (A2A) and air-to-ground (A2G) channels, there might be a strong correlation between the channels of the legitimate and illegitimate users, even when there are large distances between the users [30]. Therefore, propagation environment reconstruction attacks can estimate the legitimate channel parameters with high accuracy [31].

The research on PUF in wireless communications applications is employed for node identification, authentication [32], and SKG [32]–[36]. In [33], quaternary PUF responses are used for key generation along with polar codes to ensure the secrecy leakage is low. The authors in [34] propose a method to produce reliable keys on field programmable gate arrays (FPGAs). The design uses a lookup table based on SLICEL components, which enables fine-tuning of the hamming weight of the PUF and increases the generated key uniqueness. A switched capacitor PUF is proposed in [35], which promises to provide a stable key for chip security with the use of metal blocks as a protective coating. An authentication and key establishment protocol based on PUF are proposed in [32].

Ideally, PUFs are unclonable, however, practical implementations have been prone to attacks such as physical cloning [37], side channel and reliability information-based attacks [38], machine learning (ML)-based modeling attacks [39], etc. The modeling attacks pose a greater threat compared to other attacks because in most cases, they do not require auxiliary information and can be based only on transmitted CRPs or leaked information during the exchange of data in the different stages of the SKG protocol. ML algorithms, such as logistic regression (LR), artificial neural network (ANN), support vector machine (SVM), etc, were applied successfully in various scenarios. Several solutions have been proposed in the literature to address the ML attack of PUFs. For example, in [40] Sbox transformation is introduced as an additional nonlinear operation to enhance the PUF resilience to modeling attacks. Other techniques are also proposed in [41], [42]. These mechanisms increase the implementation complexity and, consequently, the required energy and area cost, which is infeasible for resource-constrained UAV networks.

B. Motivation and Contributions

As can be noted from the cited literature, the references listed therein, and to the best of the authors’ knowledge,
line-of-sight (LoS) and poor scattering environments are still considered the main limitations for adopting PLS in practical systems, particularly UAV networks. In such networks, the spatial de-correlation assumption of the eavesdropper is invalid for several cases of interest, particularly if the eavesdropper is in close vicinity to a legitimate user, which jeopardizes the network. Therefore, this paper proposes a novel framework for high-rate SKG based on PLS by incorporating artificial fading (AF) and PUFs. The synergy of PLS and PUFs will increase resilience to ML modeling attacks because no CRPs are required to be transmitted over the air. Furthermore, in the proposed SKG the number of side-channel transmissions is reduced, which decreases the chance for eavesdroppers’ to collect more information to model the PUF. The main contributions of this work are:

1) Propose a novel SKG protocol based on the integration of PUF and CR. The proposed SKG protocol resolves the issue of static channels in the context of PLS with the aid of PUFs, which can enhance the reliability of the SKG process and increase the KGR. In the proposed SKG, CR between the legitimate users is used to generate a challenge at the communicating nodes, which is applied to the PUF or PUF emulator to generate the ultimate key.

2) Propose a novel mechanism to enhance the randomness level of PLS systems in static or low scattering environments. The proposed scheme, called AF, introduces common signal variations between legitimate nodes. The AF is an interleaved version of the channel frequency response (CFR) of the previously successful SKG session where a key agreement is achieved. The interleaving process of the CFR will significantly reduce the eavesdropper’s capability to estimate the legitimate users’ channel accurately, even if it was able to locate itself in close vicinity to a legitimate user.

3) Propose an efficient bit extraction (BE) scheme by modifying the adaptive secret bit extraction (ASBE) [20]. The new BE technique can reduce the number of transmissions between the nodes and also, reduce the required number of side-information bits.

4) The considered PUF is realized using a configurable ring oscillator (RO), which is implemented using FPGA, and its properties are validated.

The produced numerical results for the proposed and conventional SKG are compared in terms of randomness, key mismatch ratio (MMR), and the average number of sessions needed to reach a key agreement. The obtained results clearly show the superiority of the proposed scheme.

C. Paper Organization

The rest of the paper is organized as follows. Sec. II describes the signal and channel models. The intermediate key generation and sharing protocol is detailed in Sect. III. The proposed AF, BE, and PUF-based key generation are explained in Sec. IV. The numerical evaluation is presented and discussed in Sec. V. Section VI concludes the paper.

II. SIGNAL AND CHANNEL MODELS

This work considers two legitimate users, Alice and Bob, who aim to establish a secure and common key through an authenticated multipath wireless channel. An eavesdropper, Eve, is able to listen to all communications between Alice and Bob passively and intends to predict the generated key. The system model is shown in Fig. 1. Orthogonal frequency-division multiplexing (OFDM) with N subcarriers is adopted for the transmission where Alice, Bob, and Eve have the capability to transmit and receive OFDM signals. Every user is assumed to be equipped with a single antenna.

The key generation process should be initiated by one of the legitimate users, i.e., Alice or Bob, and then the negotiation to generate the secret key starts. Assuming that Alice starts the key generation process, then she should send an OFDM symbol to Bob. The transmitted OFDM symbol is generated by applying the data symbols’ vector \( \mathbf{x}_A = [x^0_A, x^1_A, ..., x^{N-1}_A]^T \), to an N-point inverse discrete Fourier transform (IDFT). Then, a cyclic prefix (CP) no less than the maximum delay spread is added as a preamble to prevent inter-symbol interference (ISI). In all OFDM transmission standards, certain subcarriers are modulated using pilot symbols for channel estimation and synchronization purposes. Therefore, the vector \( \mathbf{x}_A \) may consist of data and pilot symbols. The symbols \( x^i_A \in \{n \} \) are selected from an arbitrary constellation diagram and are considered to have unit average power, i.e., \( \mathbb{E} \left[ |x^i_A|^2 \right] = 1 \), where \( \mathbb{E} \cdot [\cdot] \) denotes the statistical expectation. For simplicity, we consider the quadratic phase shift keying (QPSK) modulation scheme.

In this work, we consider that the pilot symbols are generally distributed following the long-term evolution (LTE) resource block structure [43, Fig. 1]. The set of pilots is denoted by the vector \( \mathbf{u}_A = [x^0_A, x^1_A, ..., x^Q_A] \).

At Bob’s receiver, the CP is removed, and discrete Fourier transform (DFT) is used to separate and extract the symbols from the subcarriers. Assuming the channel is quasi-static, i.e., the channel remains fixed during one OFDM symbol period, and the CP is larger than the maximum delay spread of the channel [44], the DFT output at Bob’s receiver can be represented as

\[
\mathbf{r}_B = \mathbf{G}_{AB}\mathbf{x}_A + \mathbf{w}_B
\]  

where \( \{\mathbf{r}_B, \mathbf{w}_B\} \in \mathbb{C}^{N \times 1}, \mathbf{w}_B = [w^0_B, w^1_B, ..., w^{N-1}_B] \) is the AWGN vector whose elements are independent and identically distributed (i.i.d.) and \( w^i_B \sim \mathcal{CN}(0, 2\sigma^2_w) \). The channel matrix \( \mathbf{G}_{AB} \in \mathbb{C}^{N \times N} \) is the CFR matrix, which is given by

\[
\mathbf{G}_{AB} = \text{diag}\left\{ \left[ G^0_{AB}, G^1_{AB}, ..., G^{N-1}_{AB} \right] \right\}
\]  

where

\[
G^n_{AB} = \sum_{i=0}^{\mathcal{Q}} g_{AB}^i \exp \left( -j\frac{2\pi in}{N} \right), \forall n
\]  

and where \( g_{AB}^i \sim \mathcal{CN}(m_{g_{AB}}^i, 2\sigma^2_{g_{AB}}^i) \) denotes the \( i \)th multipath component gain and \( \mathcal{Q} + 1 \) represents the number of multipath components. The fading gains \( g_{AB}^i, i \in \{0, 1, ..., \mathcal{Q}\} \), are considered independent. Therefore, the envelope of the channel matrix elements is Rician and the frequency selectivity
of the channel depends on the gain and delays of the channel multipath components \(g_{AB}^i\). More specifically, \(|G_{AB}^i| \sim \mathcal{R}(K_{AB}^i, \Omega_{AB}^i)\), where \(K_{AB}^i = \frac{|m_{g_{AB}^i}|^2}{\sigma^2_{g_{AB}^i}}\), \(K_{AB}^i \in (0, \infty)\) and \(\Omega_{AB}^i = |m_{g_{AB}^i}|^2 + 2\sigma^2_{g_{AB}^i}\). A special case of interest is when the fading factor \(K = 0\), which corresponds to the Rayleigh fading scenario. It is worth noting that the diagonal elements in \(G_{AB} \triangleq d_{AB}\) are correlated with a correlation factor that depends on \(g_{AB}^i\) \(\forall i\) [45]. Because Alice’s signal is transmitted over a broadcast channel, Eve will also receive a copy, which can be written as

\[
 r_E = G_{AE}x_A + w_E
\]

where \(G_{AE}\) is the CFR from Alice to Eve.

Under the same assumptions and conditions, and in a similar fashion, Bob sends to Alice the vector \(x_B\), and the DFT output at Alice can be written as

\[
 r_A = G_{BA}x_B + w_A
\]

where \(G_{BA}\) is the CFR matrix from Bob to Alice. The IDFT output at Eve can be expressed as

\[
 r_E' = G_{BE}x_B + w_E'
\]

where \(G_{BE}\) is the CFR matrix from Bob to Eve.

The DFT outputs \(r_A\) and \(r_B\) can be used to obtain the channel state information (CSI) for both channels, i.e., \(G_{AB}\) and \(G_{BA}\). The process typically starts by estimating the CFR at the pilot subcarriers using techniques such as the least-square (LS) or minimum mean-square error (MMSE), and then interpolation can be used to compute the channel gains at the data subcarriers [46]. The communications between Alice and Bob are assumed to be conducted using TDD where the coherence time of the channel is larger than the TDD frame. In such scenarios, the channel reciprocity principle can be incorporated to consider that \(G_{AB} = G_{BA} \triangleq G\) [4], [20], [47]–[49]. Moreover, given that Eve is located at a relatively far distance from Bob, then \(G_{AB} \neq G_{AE}\). Consequently, Alice and Bob are the only nodes who have knowledge of \(G\). Therefore, Alice and Bob can use \(G\) to generate a secret key at both sides and use it for secure communications [5], [6], [50].

III. INTERMEDIATE KEY GENERATION AND SHARING

Conventional PLS-based SKG is described extensively in the literature, and hence, it is stated briefly in this section for the sake of completeness and to simplify the presentation of the proposed framework. The keys that are generated in this work can be classified as intermediate and final keys. The intermediate keys can be generated using various PLS key-sharing techniques as described in the following subsections. The intermediate keys go through the second stage of processing to generate the final keys using PUFs. The intermediate key generation and sharing processes using PLS can be briefly described as follows:

1) Channel Probing: The channel probing aims at estimating \(G_{AB}\) and \(G_{BA}\), or more specifically \(d_{AB}\) and \(d_{BA}\). The process starts when Alice transmits \(x_A\) to Bob, who computes \(r_B\) and uses it to estimate \(d_{AB}\) as described in Sec. II. In this work, LS method is used to estimate the channel coefficients at the pilot symbols, and then linear interpolation is used to obtain the coefficients at the data subcarriers. Similarly, Bob transmits \(x_B\) within the same TDD frame and Alice computes \(r_A\) and estimates \(d_{BA}\).

2) Intermediate Key Generation: Once the vectors \(d_{AB}\) and \(d_{BA}\) are estimated, they can be used to generate the intermediate keys, which are denoted by \(q_A\) and \(q_B\), respectively. In PLS, both the phase and amplitude of the channel coefficients can be used to extract the key bits from \(d\). Nevertheless, the phase is more sensitive to hardware imperfections, and hence, the amplitude is considered more attractive. Therefore, the amplitude, or equivalently the received signal strength (RSS) for QPSK or binary phase shift keying (BPSK) modulation schemes, \(\zeta = |r|\), is typically used to generate the bits of \(q_A\) and \(q_B\). Therefore, In the literature, the BE algorithm proposed in [20], named ASBE, has received significant attention due to its ability to generate high entropy bits at a high bit rate. Nevertheless, the algorithm performance may deteriorate significantly in static or flat-fading channels where it might take about 7 minutes to generate a 256 bits key [20]. Moreover, Alice and Bob must exchange the indices of the subcarriers that were dropped during the BE process, which can be considered a significant overhead. Furthermore, it causes some information leakage about the key. In order to address the disadvantages of ASBE, we propose a BE mechanism in Sec. IV, which has less transmission overhead, a low number of side-channel transmissions, and is computationally more efficient.

3) Error Reconciliation and Verification: For reliable communications, the keys \(q_A\) and \(q_B\) should be identical. However, the BE process is prone to errors due to AWGN, imperfect CSI estimation, and hardware mismatch. Therefore, additional processing is necessary to guarantee that \(q_A = q_B\), and both users should verify that they have identical keys. The verification process can be realized using cyclic redundancy check (CRC) where Alice generates the CRC bits and Bob will be able to verify and acknowledge if the CRC bits at both sides match [51]. Therefore, Alice computes the CRC bits of \(q_A\), denoted as \(\epsilon_A\). The error reconciliation aims at eliminating any discrepancy between \(q_A\) and \(q_B\). In this work, we adopt the code-offset secure sketch that is proposed in [6], and Bose–Chaudhuri–Hocquenghem (BCH) codes are used as the underlying coding scheme. The process starts when Alice randomly selects a codeword from the codebook of the corresponding BCH code, and then computes

\[
 s^q = [s_1^q, s_2^q] = [q_A \oplus v_A^q, c_A^q]
\]

where \(\oplus\) is the exclusive or (XOR) operation. Because \(q_A\) and \(q_B\) are not necessarily equal, we can write \(q_A = q_B \oplus \epsilon\), where \(\epsilon\) is the error pattern that represents the differences between \(q_A\) and \(q_B\). Therefore, \(\epsilon_i = 1\) if \(q_A^i = q_B^i\), otherwise \(\epsilon_i = 0\).
The AF is mathematically similar to the widely-known pre-fading channel is emulated and used at the transmitter side. To examine the impact of the channel interleaving on Eve’s capability to estimate the legitimate channel between Alice and Bob, for the worst case scenario, when Bob broadcasts $G_{B,A}x_B$, we can assume that Eve is in the middle between Alice and Bob, the received signal becomes

$$r = \hat{G}x + w$$

(12)

where $\hat{G} = GG$, and $G$ is interleaved version of $G$.

To implement the AF process, consider that a pre-designed fading channel whose channel matrix, denoted as $G$, can be represented as a diagonal matrix where the diagonal elements can be expressed by the vector $d = [d_0, d_1, \ldots, d_{N-1}]$. Therefore, the DFT output at any user’s receiver can be written as $r = GGx + w$. Because $G$ and $G$ are diagonal matrices, then $GG \triangleq G$, which is also a diagonal matrix whose diagonal elements vector can be written as $d = dd$. Given that the adjacent elements in $d$ are correlated, and similarly in $\bar{d}$, then the elements of $d$ and $\bar{d}$ can be correlated. Consequently, all users can estimate the CSI using conventional approaches as described in Sec. II. In the worst case that the channel is purely flat, i.e., $d_i = 1 \forall i$, then $d = d$, which corresponds to a frequency-selective channel, which can be used to generate a random bit sequence. However, most channel estimation algorithms for OFDM generally require the channel coefficients over adjacent subcarriers to be highly correlated [43], the selection of $G$ can be performed to decorrelate the overall fading matrix $G$ making it even more difficult for Eve to estimate $G$ or $\hat{G}$. For legitimate users, the estimation process starts by equalizing the effect of $d$, then $\bar{d}$ can be obtained. In this work, we adopt random interleaving to decorrelate the elements of $d$. The interleaved $d$ is denoted as $\bar{d}$. An example for $d$ with and without $\bar{d}$ interleaving is shown in Fig. 2. As can be noted from the figure, it will be hard to accurately estimate the channel coefficients at the non-pilot subcarriers before eliminating the impact of $\bar{d}$. When the interleaved vector $\bar{d}$ is used, the received signal becomes

$$r = \hat{G}x + w$$

(12)
Bob and is informed about the transmitted pilots. Hence, she can apply the channel estimation procedure explained in Sec. II to obtain $G_{BA}$. It should be noted that Eve is not aware of $G_{BA}$.

Fig. 3- (b) shows the MSE for the channel estimation of $G_{BA}$ at Alice and Eve over Rayleigh fading channel for two frequency selective channels $Ch_1$ and $Ch_2$ and implementing (12), $\hat{G}$ is obtained by interleaving $\hat{G}$ for the $Ch_1$ and $Ch_2$. Every channel has 5 taps with normalized delays of [0, 1, 4, 5, 11] samples. The average taps’ gains for the first channel, $Ch_1$, are [0.88, 0.07, 0.03, 0.01, 0.01] and for the second channel, $Ch_2$, are [0.3584, 0.247, 0.0928, 0.1851, 0.1167]. It can be noted that $Ch_1$ and $Ch_2$ represent, respectively, moderate and severe frequency selective environments. It can be seen that the MSE of Alice is much lower than that of Eve. This is due to the decorrelation impact of the interleaving on $\hat{G}_{BA}$, which makes it difficult for Eve to correctly estimate the channel at the data subcarriers. On the other hand, in Fig. 3-(a) the BER, $P_e$, is presented for the case where we do or do not perform interleaving to $d_{BA}$. Since Alice knows the pilots and $d_{BA}$, then she first divides $r_A$ over $d_{BA}$, and applies the channel estimation process in Sec. II. Clearly, for both channels $Ch_1$ and $Ch_2$, $P_e$ is identical for the two scenarios, which implies that the interleaving process does not affect the estimation capability of the legitimate users. Furthermore, Fig. 3-(a) shows the BER for Eve for $Ch_1$ and $Ch_2$. It can be noted that Eve’s BER performance is severely worse than Alice’s for all cases and signal-to-noise ratio (SNR) values which actually is an indicator of the high secrecy level provided by the AF and the associated interleaving process.

Although using the AF is generally beneficial even in frequency-selective channels, it introduces an additional computational complexity of $N$ complex multiplications at the transmitter and $N$ complex divisions at the receiver. To reduce the complexity, the AF can be applied only when the channel does not have sufficient frequency selectivity to produce a reliable bit sequence. In order to decide if a channel randomness level is not adequate to generate a shared challenge, we consider a counter for the number of sessions where Alice and Bob’s challenges are not matching. If the challenge sharing fails for a certain number of consecutive sessions, the channel is then considered not suitable, and AF is incorporated. It is also worth noting that the initially stored channel vector $d$ should not be used permanently and should be updated frequently. Toward this goal, we use the channel that was produced during the last successful sequence as the new channel for the AF process.

**Algorithm 1 : Artificial Fading**

1. **Input**: $l$, $x$, $G_l$
2. **Initialize**: $\hat{G}$
3. if $l > 0$ then
4. $\hat{G} = G_l$
5. end if
6. $G = \text{interleave}(\hat{G})$
7. $x = Gx$
8. **Output**: $x$

The AF process is described in Algorithm 1. In this scheme, it is assumed that each user has a buffer called channel emulator, which is used for storing the AF coefficients. The inputs to the AF algorithm are $l$, $x$ and $G_l$ where $l$ is the
counter for the number of successful final key agreement iterations and $G_l$ is the CFR of the last successful iteration. Prior to the implementation of the protocol, an initial CFR $G$ is generated and stored in the channel emulator. If $l > 0$ then $G$ is updated such that $G = G_l$. Then, we interleave $G$ using random interleaving, $\tilde{G} = \text{interleave}(G)$, consequently, the transmitted signal can be represented as
\[ \tilde{x} = \tilde{G}x. \]  

(13)

B. Proposed Adaptive Bit Extraction

Because the elements of $G_{AB}$ and $G_{BA}$ are analog values, they cannot be used directly for key generation. Therefore, additional processing is required for BE. In this work, we propose a BE scheme based on the ASBE presented in [20]. In the ASBE, the number of side-channel transmissions and required overhead are significant, particularly when the channel variations are limited and/or the SNR is low.

For notational simplicity, the indices $A$ and $B$ will be dropped unless it is necessary to include them. The proposed BE algorithm for Alice can be explained as follows:

1) Segment $\zeta$ into $M$ blocks $\{\zeta_1, \zeta_2, \ldots, \zeta_M\}$ each of which has $K$ elements. The set of indices for each block is denoted as $I_m$, $m \in \{1, 2, \ldots, M\}$. Because all blocks go through the same process, the block index $m$ will be dropped unless it is necessary to include it. Moreover, the same processes are applied to all blocks.

2) Evaluate two thresholds [20], $z^+ = \mu + \alpha \sigma$ and $z^- = \mu - \alpha \sigma$, where $\mu$, $\alpha$, and $\sigma$ are the mean, weight factor and standard deviation of the block, respectively.

3) Construct a $K \times 3$ matrix where the first column elements are $j = [1, 2, \ldots, N]$, the second column elements are $i = [1, 2, \ldots, N]$, and the third column contains the elements of $\zeta$. The first column elements represent the rows’ numbers while the second column elements represent the indices of $\zeta$.

4) Sort the elements of the second and third columns in a descending order based on the values of $\zeta$ elements. Note that the elements of $\zeta$ remain sorted.

5) Find the minimum element in the third column where $\zeta_{(1)} > z^+$. Store the value of $j$ for that element as $J_1$.

6) Find the maximum element in third column where $\zeta_{(1)} < z^-$. Store the value of $j$ for that element as $J_2$.

7) Assign a value of one for all elements in the third column, row $1$ to $J_1$ and zero to all elements in row $J_2$ to row $N$.

8) All rows with indices larger than $J_1$ and less than $J_2$ should be deleted.

9) Sort the values of columns two and three in a descending order based on the values of the second column, i.e., restore the order of the original elements.

10) To minimize the mismatch between Alice and Bob’s intermediate keys, Alice sends Bob the values of $J_1$ and $J_2$.

For Bob, steps 1 to 7 are identical to those of Alice. Bob aligns the regions by adjusting his range values such that $J_{1,B} = \max\{J_{1,A}, J_{1,B}\}$ and $J_{2,B} = \min\{J_{2,A}, J_{2,B}\}$. The remaining steps are also similar to those of Alice. However, Bob does not need to share his ranges with Alice. It is worth noting that unlike [20], the proposed algorithm does not leak information about the indices of the selected subcarriers, however, it tells the number of generated bits. Such information should not be critical since the key size is typically assumed to be known by Eve.

As for the ASBE, for the $m$th block, we must transmit $K$ bits from Alice to Bob and Bob to Alice. Thus, for the $M$ blocks, the total number of bits becomes $2 \times M \times K$. For the proposed BE scheme, we are required to transmit $2 \times M \times \lceil \log_2(K) \rceil$. For example, if $M = 2$, $K = 64$, the ASBE should transmit 512 bits while the proposed scheme requires only 24 bits.

In order to compare the performance of the ASBE and the proposed BE, Fig. 4 shows the MMR for three different Rayleigh fading channels, $Ch_2$, $Ch_3$, and $Ch_4$, where $Ch_3$ and $Ch_4$ have 5 taps with normalized delays of $[0, 1, 4, 5, 11]$ samples and the average taps’ gains are $[0.2, 0.2, 0.2, 0.2, 0.2]$ and $[0.97, 0.02, 0.005, 0.004, 0.001]$, respectively. For this figure, we consider applying the three steps: channel probing, BE with $M = 1$ and $\alpha = 0.4$ and error reconciliation. We use the same setup presented in the numerical results section for the OFDM structure and BCH code (63, 7, 15) for the error reconciliation step. It can be seen that the MMR difference between both techniques is negligible, which makes the proposed mechanism outweighs the ASBE in terms of the needed overhead. Fig. 5 shows the probability mass function (PMF), $f(\epsilon)$, and the cumulative distribution function (CDF), $F(\epsilon)$, comparison where $\epsilon = \sum_{j=1}^{63} |q_{A}^j - q_{B}^j|$. The channel probing and BE mechanisms are applied where the considered frequency selective channel is $Ch_2$ at $\gamma = 10$ dB. It can be seen from the PMF that the ASBE outperforms the proposed one. However, due to the correction capability of the BCH code which can correct up to 15 errors, both techniques will have comparable MMR performance as shown in Fig. 4.

C. Proposed PUF-based Final Key Generation

In principle, PUF utilizes the nano-scale manufacturing process variations of semiconductor devices to produce unique keys [16]. Mathematically, for a $v$-bit input (called $v$-bit challenge) and $\varsigma$-bit output (called $\varsigma$-bit response) PUF circuit can be represented by a Boolean function $f : \{0, 1\}^v \rightarrow \{0, 1\}^\varsigma$. The unclonability and uniqueness proprieties of PUFs are exploited to enhance the security level of the PLS- based SKG protocol as well as the KGR. In the proposed protocol, we input the intermediate keys, $q_A$ and $q_B$, to the PUF or equivalently its emulator and the produced hashed responses are considered as the final secret keys. It should be mentioned that the intermediate or final keys will not be distributed or shared at any step of the protocol. The previously mentioned characteristics of PUFs allow us to accept any number of bits (length) for the intermediate keys and this will not affect the secrecy level of the system. Consequently, unlike the
conventional PLS-based SKG, we do not need to wait until a specific number of bits is obtained from the RSS, thus utilizing the PUF leads to high KGR [6], [20]. In order to avoid transmitting the intermediate keys through the channel, both Alice and Bob should have the same set of CRPs obtained by the PUF. Due to the low computational and storage capabilities of UAVs, it is not feasible to store the CRPs at any node. Therefore, we propose to consider a PUF emulator at one side and the actual PUF at the other side. PUFs manufacturers can provide the legitimate parties by the PUF parameters such as gate delays and reliability distribution against voltage and temperature variations. In this paper, we assume that we can emulate the actual PUF using a set of gate delays and reliability models.

For the proposed protocol, we assume that Bob is equipped with a configurable RO PUF [53], which is a delay-based PUF that uses the RO frequencies as the random source for generating the responses, and Alice has its emulator. Due to the sensitivity of PUFs to temperature and voltage variations, we assume that the emulator response is the original response. Also, we assume that any attempt to tamper or separate the PUF will destroy it [54]. The process to generate the final key starts by inputting the intermediate keys $q_B$ and $q_A$ to the PUF and its emulator at Bob’s and Alice’s sides, respectively. The detailed steps are as follows:

1) **Response Generation**: Alice will input $q_A$ to the PUF whereas Bob will input $q_B$ to the PUF. The responses $y_A$ and $y_B$ are produced at Alice’s and Bob’s sides, respectively. Ideally speaking, the responses should be identical under any environmental setting.

2) **Error Reconciliation and Verification**: The aim of this step is to ensure that $y_A$ is identical to $y_B$ in the presence of temperate and voltage variations. A similar error reconciliation mechanism is followed as in Sec. III. Since we consider that the emulator can produce the original response, hence the encoder is located at Alice’s side and the decoder at Bob’s side. Moreover, the verification of the responses agreement is performed using CRC at Alice’s and Bob’s sides, $c^p_A$ and $c^p_B$, respectively. At Alice’s side, we compute

$$s^p = [s^p_1, s^p_2] = [y_A \oplus v^p_A, c^p_A]$$

where $v^p_A$ is a codeword with the same length as $y_A$. Then, Alice modulates and transmits $s^p$ to Bob who detects $\tilde{s}^p$ and computes $\tilde{c}$ as in Eq(11). Once $q_B$ is obtained, Bob calculates $c^p_B$ to compare it with $\tilde{c}^p_A$. If both CRCs are equal, then Bob will send an acknowledgment to Alice. Otherwise, a negative acknowledgment is to be sent. In the latter case, the final key generation steps (1) – (2) are repeated until a key generation agreement is reached.

3) **Hash Generation**: Some information about the shared challenges and responses is leaked to Eve during the error reconciliation steps. Thus, we utilize universal hash functions (UHFs) [55] to generate the final keys, $K_A = H(y_A)$ and $K_B = H(y_B)$, to enhance the randomness level.

### D. PUF Modeling Attack

We assume that Eve is aware of the proposed SKG protocol, including the decided parameters of the proposed BE and error reconciliation steps. As mentioned earlier in Sec. I, ML attacks are challenging for PUFs due to the possibility of modeling them using the transmitted CRPs and side-channel information without physical intervention. The key generation protocol can
be considered secure if Eve is not able to predict the correct keys given the knowledge of the used techniques and having full access to the transmitted data. We also assume that the benefit of an attack diminishes if Eve needs to continuously employ significant computing power beyond a reasonable time span [56]. In Fig. 1, we call the model resulting from the ML attack as "PUF prediction model." In our scheme, the following 4 secrets are shared over the channel: \( s^d, s^p, J_1 \) and \( J_2 \). We assume that the attacker has access to all data transmitted between Alice and Bob. The ML attacks require Eve to collect a sufficient subset of CRPs and side-channel information to build an accurate PUF. As presented earlier in Sections III and IV-C, the intermediate and final keys generation stages do not require any explicit transmission of the PUF CRPs. Moreover, the shared data \( s^d \) and \( s^p \) will not be useful for Eve unless she has the correct \( v^d \) and \( v^p \) to be able to accurately obtain \( q \) and \( y \) which is very unlikely because \( v \) is a codeword that corresponds to a random binary vector. As for \( J_1 \) and \( J_2 \), they only represent the range of the dropped indices during the BE step. Actually, if the RSS is not known, then these indices do not reveal useful information for Eve. Therefore, we can consider that the proposed SKG is secure since the leaked information is not significant to produce a subset of CRPs to model the PUF over a reasonable time span.

Fig. 6 shows the cross-correlation between Alice and Bob \( \rho_{AB} \) and Alice and Eve \( \rho_{AE} \) [57, Eq. (2.75)] in the best-case scenario for Eve, where she is located in the middle between Alice and Bob. This means that her fading channel is the same as the legitimate channel. However, the AF coefficients are known only to legitimate users. We assess the correlation between \( q_A \) and \( q_E \) and \( q_A \) and \( q_E \). Two Rayleigh fading channels are considered \( Ch_1 \) and \( Ch_2 \), and the OFDM, AF, BE and error reconciliation parameters are presented in Sec. V. It is clear that the level of correlation between Alice and Eve is considerably lower than between Alice and Bob, which is due to the impact of the induced AF at Alice's transmitters. Consequently, the intermediate key MMR between Eve and the legitimate users will be considerably high. Therefore, with regards to the PUF, it is challenging for Eve to estimate the challenge. Thus, generating a CRPs model is highly unlikely.

V. NUMERICAL RESULTS

This section presents a wide range of numerical results to evaluate the performance of proposed SKG protocol. The system model considers an OFDM system with \( N = 256 \) subcarriers that are modulated using QPSK. The number of CP samples, pilot and null subcarriers are \( N_c = 64, N_p = 25 \) and \( N_n = 53 \times 2 \), respectively. The null subcarriers split equally and are located at the edges of the subcarriers. The number of OFDM symbols considered to assess the performance of the proposed protocol is \( 1.2 \times 10^4 \) for each simulation point. The wireless channel is modeled as a quasi-static Rician frequency-selective fading channel with \( K \in \{-\infty, 15\} \) dB, where the channel remains fixed during a given OFDM symbol but changes randomly between adjacent symbols. The case where \( K = -\infty \) corresponds to the Rayleigh fading. Two multipath fading channel models are considered in this section, \( Ch_1 \) and \( Ch_2 \). Moreover, in order to examine the proposed protocol under practical conditions, we vary the correlation factor \( \rho_{AB} \) between the channels of Alice and Bob. For notational simplicity, we denote \( \rho_{AB} \) as \( \rho \). The chosen correlation values are \( \rho = \{1, 0.88\} \). As for the proposed BE, we chose \( M = 2 \) and \( \alpha = 0.4 \). With regards to the AF generation, we consider \( Ch_2 \) to generate the initially stored \( G_s \) in the channel emulator. For simplicity, we call our proposed SKG, P. SKG and the conventional SKG presented in [20] as C. SKG.

The PUF considered in this paper is presented in [53]. As tested in the paper, the uniqueness and uniformity are almost 50%. Also, in order to reflect the impact of temperature and voltage changes on the PUF responses, we use the presented reliability distribution in terms of the intra-hamming distance, which is obtained by conducting several experiments on FPGA. The reliability distribution of [53] is shown in Fig. 7. As mentioned earlier in Sec. I, the emulator response is considered as the original response. The reference temperature and voltage are, respectively, set as 26°C and 3V. The length of the response of the PUF and its emulator is 127 bits.

We have implemented the proposed configurable RO PUF of [53] on FPGA to verify its reliability. Due to the area limitation of the FPGA, the lengths of the challenge and response are chosen to be 32 bits. First, to ensure that the PUF can produce the same response to a certain challenge given fixed temperature and voltage, we ran it for 2500 times at 25°C and 3V. As expected, the same response is obtained in every run. Fig. 7 shows the intra-hamming distance distribution of the 32 bits responses under 4 different temperatures \([40°, 50°, 60°, 70°] \) C with reference temperature 25° C. For each temperature value, \( 5 \times 10^3 \) responses are produced. As can be noted, most of the measurements have 1 to 3 errors when compared to the
Moreover, it can be noticed that the C. SKG protocol performs significantly better than the conventional SKG protocol. This is due to the impact of the AF on the correlation between Alice and Bob. In other words, by applying the AF on both sides, the amount of common variations and correlation increase, which is similar to the impact of increasing the fading selectivity level, which increases the probability to agree on a shared key.

Fig. 10 shows the average number of sessions required for Alice and Bob to agree on a key, $K_A = K_B$ for the P. SKG and C. SKG protocols. It can be noted that for both protocols, as the SNR increases, the average number of sessions decreases, which is due to the reduced impact of the noise. Also, the P. SKG results in lower values due to the utilization of PUF and AF. The use of PUF does not restrict us on any intermediate key length, hence unlike the C. SKG protocol, we can generate the final keys from any number of bits and this does not affect the secrecy level of the system. In other words, instead of waiting until we get 63 bits for the $q$ in the C. SKG, in our protocol, we append the $q$ with zeros to get 63 bits. Also, due to the higher correlation between Alice and Bob resulting from applying AF, the average number of times required to achieve a key agreement is lower than the C. SKG.

In order to assess the randomness of the final keys generated
by the proposed SKG protocol, we use the NIST suite [58]. The suite consists of 15 tests and computes a probability value for each test, called p-value. For practical considerations related to the minimum input length required for every test, we decided to compute 8 tests [20]. The key can be considered random with 99% confidence if the corresponding p-values are greater than 0.01. We run our proposed protocol using the same coefficients and parameters listed previously for $\bar{\gamma} = 16$ dB. Table I shows the p-values of the NIST tests. Since the final keys pass all
the tests as shown in Table I, they are considered random with 99% confidence.

For comparison purposes, we ran the NIST test for the C. SKG protocol as shown in Table II. For a fair comparison with the settings of our protocol, we ensure that the length of the keys input to the hash generation step is 127 bits. We can note from Table II that the keys produced by this protocol pass the NIST test and are hence considered random.

VI. Conclusion

This work proposed a novel framework that integrates PLS with PUF to strengthen the secrecy and improve the efficiency of the key generation and sharing processes for dynamic and static wireless channels. More specifically, in the case of flat fading or poor scattering environments, we proposed a novel technique denoted as AF, which overlays a user-defined frequency-selective fading over the actual channel experienced between the legitimate users. Although AF results in higher computational complexity, it leads to a significant drop in the MMR compared to the conventional PLS-based SKG protocols. Further, it results in a lower average number of sessions needed to agree on a shared key. Furthermore, we proposed an efficient BE scheme that has reduced overhead, less number of side-channel transmissions, and increased secrecy, as compared to conventional schemes. The obtained numerical results showed a significant reduction in the MMR of the proposed protocol when compared to existing conventional SKG protocols. It is also shown that we can achieve a key agreement in a single session for moderate and high SNR ranges in rich scattering environments or equivalently when AF and PUF mechanisms are applied.

APPENDIX I: LIST OF ACRONYMS

A2A air-to-air.
A2G air-to-ground.
AF artificial fading.
ANN artificial neural network.
ASBE adaptive secret bit extraction.
AWGN additive white Gaussian noise.
BCH Bose–Chaudhuri–Hocquenghem.
BE bit extraction.
BER bit error rate.
BPSK binary phase shift keying.
CDF cumulative distribution function.
CFR channel frequency response.
CP cyclic prefix.
CR channel reciprocity.
CRC cyclic redundancy check.
CRP challenge-response pair.
CSI channel state information.
DFT discrete Fourier transform.
FPGA field programmable gate array.
i.i.d. independent and identically distributed.
IC integrated circuit.
IDFT inverse discrete Fourier transform.
IoT Internet of things.
IRS intelligent reflective surface.
ISI inter-symbol interference.
KGR key generation rate.
LoS line-of-sight.
LR logistic regression.
LS least-square.
LTE long-term evolution.
MIMO multiple-input multiple-output.
ML machine learning.
MMR mismatch ratio.
MMSSE minimum mean-square error.
MSE mean squared error.
NIST national institute of science and technology.
OFDM orthogonal frequency-division multiplexing.
PLS physical layer security.
PMF probability mass function.
PUF physically unclonable function.
QPSK quadrature phase shift keying.
RO ring oscillator.
RSS received signal strength.
SINR signal to interference plus noise ratio.
SKG secret key generation.
SNR signal-to-noise ratio.
SVM support vector machine.
TDD time division duplexing.
UAV unmanned aerial vehicle.
UFH universal hash function.
XOR exclusive or.

REFERENCES


