Testbed Implementation and Evaluation of Interleaved and Scrambled Coding for Physical-Layer Security

César Martins*†, Telmo Fernandes*†, Marco Gomes*†, and João Vilela‡
*Instituto de Telecomunicações, Portugal
†Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
‡CISUC and Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

***Abstract*—This paper presents a testbed implementation and evaluation of coding for secrecy schemes in a real environment through software defined radio platforms. These coding schemes rely on interleaving and scrambling with randomly generated keys to shuffle information before transmission. These keys are then encoded jointly with data and then hidden (erased) before transmission, thus only being retrievable through parity information resulting from encoded data. An advantage of the legitimate receiver (e.g. a better signal-to-noise ratio) on the reception of those keys provides the means to achieve secrecy against an adversary eavesdropper. Through this testbed implementation, we show the practical feasibility of coding for secrecy schemes in real-world environments, unveiling the usefulness of interleaving and scrambling with a hidden key to reduce the required advantage over an eavesdropper. We further describe and present solutions to a set of issues that appear when doing practical implementations of security schemes in software defined radio platforms.**

***Index Terms*—Wireless Security, Software Defined Radio, Physical Layer Security, Testbed Evaluation**

I. INTRODUCTION

The random nature of the wireless channel allows the physical layer to have a relevant role in communication security by exploiting that randomness and taking advantage of it when communicating over a wireless channel in the presence of an eavesdropper. Several works have been developed around this premise, where schemes are built based on interleaving, scrambling and puncturing, in order to guarantee secrecy in physical-layer security (PLS). Although much effort has been put into the development of schemes with practical applicability, physical-layer security [1] is yet to be mature enough to provide practical solutions capable of being included in current communication systems and standards. Scrambling for secrecy [2] and puncturing for secrecy [3] [4] are promising techniques that have been explored to take advantage of the inherent randomness of wireless networks, however they lack a testbed implementation in a real environment.

In previous work, a coding scheme for secrecy based on interleaving and puncturing techniques was proposed [5]. The Interleaved Coding for Secrecy with a Hidden Key (ICS-HK) scheme relies on a hidden interleaving key to conceal information from an eavesdropper. However, as well as the previously mentioned works, this scheme has not yet been tested in a real scenario. In this paper we present proof of concept of the ICS-HK scheme, through a real environment implementation in software defined radio (SDR) platforms. Besides we propose a novel Scrambled Coding for Secrecy with a Hidden Key (SCS-HK) scheme, where the interleaver is replaced by a scrambler to improve previous results. Resorting to SDR platforms, these mechanisms are implemented and evaluated in a real-world environment, for which relevant implementation challenges are also identified and addressed.

The remainder of this paper is organized as follows. Section II presents the background information necessary to understand the presented schemes. Section III describes the ICS-HK scheme and presents the new proposed SCS-HK scheme. Section IV describes the practical implementation of the presented schemes, with the corresponding evaluation in Section V. Finally, Section VI summarizes the major findings of the paper.

II. BACKGROUND

In this section we present a literature review about the wiretap channel model and secrecy metrics necessary to understand this work.

A. Wiretap Channel

The emergence and development of wireless networks brought within a greater challenge when protecting data, given the two main characteristics of these networks: diffusion and signal overlapping. As diffusion hardens the task of keeping the transmitted signals out of reach from illegal receivers, the overlapping of many different signals significantly reduces reliability. Although there has been research of channel coding for secrecy since the 70s, only with the emergence of these networks, in recent years, we have seen a renewed interest in this area, in part because of the need for an advantage of the legitimate user over an eavesdropper [6] [7]. Every development in this area is based on Wyner's work which, in 1975, has proved the existence of codes that guarantee, at the same time, reliability and confidentiality [8].

The wiretap channel, presented by Wyner and illustrated in Fig. 1, assumes the existence of three users, representing the simplest model where there is an attempt of intercepting information. Those are: a transmitter, a legitimate receiver and an eavesdropper, known as Alice, Bob and Eve, respectively. In this model, Alice wishes to send a private message M to Bob. The message is therefore coded resulting in the signal X^n which is sent over the main channel to Bob. The received signal, Y^n is decoded so that Bob can obtain an estimation of the original message, $\hat{M}$. Meanwhile, Eve is listening to the signal transmitted by Alice and gets a copy of X, Z^n, through the eavesdropper's channel. In this scenario, it is assumed that the eavesdropper's channel is noisier than the main channel and that the eavesdropper is passive, which means that Eve is listening without making itself noted. Besides, it is considered that Eve has complete knowledge of the decoding algorithm and has no boundaries with respect to its computational power.

B. Metrics

When evaluating the security that a scheme can guarantee, it is necessary to define the conditions that define what is secure and what it is not. In that sense, Shannon, in 1949, proposed the concept

This work was carried out in the scope of projects SWING2 (PTDC/EEI-TEL/3684/2014) and MobiWise (P2020 SAICTPAC/001/2015), funded by Fundos Europeus Estruturais e de Investimento (FEEI) through Programa Operacional Competitividade e Internacionalização - COMPETE 2020, by National Funds from FCT - Fundação para a Ciência e a Tecnologia, through projects POCI-01-0145-FEDER-016753 and UID/EEA/50008/2013, and European Union's ERDF (European Regional Development Fund).

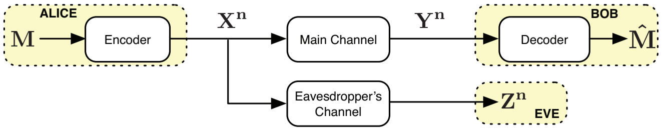


Fig. 1: The Wiretap Channel.

of perfect secrecy [9], which considers a code secure as long as it guarantees that the coded message does not leak any information about the original, that is the mutual information between M and X^n is null, $I(M; X^n) = 0$. However, Shannon also concluded that, in order to achieve perfect secrecy, it is necessary to use keys with at least the same size as the message, which is not practical [10].

A few years later, Wyner suggested a new secrecy metric, known as weak secrecy [8]. This metric does not demand that X^n does not have information about the message but instead that the received signal by Eve, Z^n, leaked a small amount of information about the message. This amount would be as small as the length of the message, n, increases, that is, $\lim_{n\to\infty} \frac{1}{n} I(M; Z^n) = 0$.

Nevertheless, it was proven that there are possible code constructions that satisfy this metric and yet leakage occurs [11]. And so, the concept of strong secrecy was introduced by Maurer [12], in which scheme a message is considered secure if the mutual information between M and Z^n is asymptotically zero, that is, $\lim_{n\to\infty} I(M; Z^n) = 0$.

The difficulty in applying these metrics for real channels and short blocklength codes [10] has led to the development of new, more operational security metrics, based on the bit-error rate (BER). In [3], a practical metric was introduced, which analyses the BER after the decoder and establishes desirable values for Bob and Eve. By matching the correspondent signal-to-noise ratio (SNR) values to those BER boundaries, a security gap (SG) is defined as the difference of SNR levels required for reliable communication for Bob and confidential communication against Eve. Thus, the advantage that Bob needs over Eve to achieve a secure transmission is known. Intuitively, we realize that the smaller the gap the better so that the quality difference between the Bob and Eve's channels is minimal.

Using BER as a security measure, although simple to implement, does not guarantee that all information is protected, because of its average nature. To cover this flaw, a bit-error rate-cumulative distribution function (BER-CDF) metric was proposed [10] that measures the probability of the measured BER in each transmitted/received codeword being superior to a value close to 0.5:

$$Pr(\hat{P}_b > 0.5 - \delta) \tag{1}$$

where $\hat{P}_b$ is the proportion of errors measured at the output of the decoder and δ is a value between 0 and 0.5, chosen accordingly of the intended security demands.

III. Finite Blocklength Coding Schemes

Code construction for the wiretap channel that does not leak information has shown to be a challenging task, with the first practical codes built in the last years [13] and only for ideal scenarios, such as a perfect channel to Bob and achieving the secrecy criteria at the asymptotic blocklength regime. As those codes fail to apply to real systems, some authors have dedicated themselves to the study of secrecy codes target to more realistic scenarios, such as solutions based on low-density parity-check (LDPC) codes [3] and polar codes [14] and/or techniques such as interleaving [5], [15], scrambling [2] and/or puncturing [3], [4]. All these proposed techniques are, however, lacking a practical testbed implementation and evaluation in a real scenario. This paper addresses such challenge for the case of the Interleaved Coding for Secrecy with a Hidden Key technique [5], and further proposes a new Scrambled Coding for Secrecy with a Hidden Key technique. Both ICS-HK and SCS-HK are described in this section.

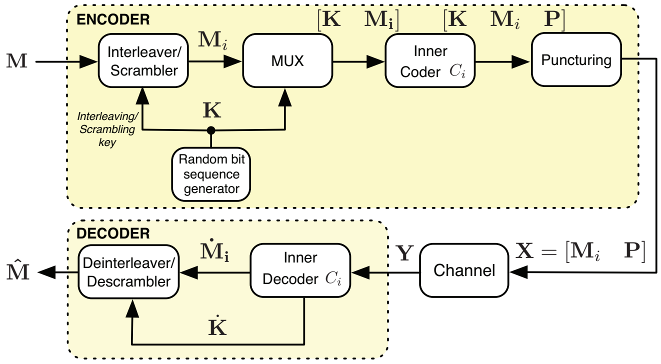

ENCODER
$[\mathbf{K} \;\; \mathbf{M_i}]$
$[\mathbf{K} \;\; \mathbf{M}_i \;\; \mathbf{P}]$
M
Interleaver/ Scrambler
$\mathbf{M}_i$
MUX
Inner Coder C_i
Puncturing
Interleaving/ Scrambling key
K
Random bit sequence generator
DECODER
$\hat{\mathbf{M}}$
Deinterleaver/ Descrambler
$\dot{\mathbf{M}}_\mathbf{i}$
Inner Decoder C_i
Y
Channel
$\mathbf{X} = [\mathbf{M}_i \;\; \mathbf{P}]$
$\dot{\mathbf{K}}$

Fig. 2: Interleaved/Scrambled Coding for Secrecy with a Hidden Key.

A. Interleaved Coding for Secrecy with a Hidden Key

In this model, depicted in Fig. 2, a random key K of length k is generated, which is used by the interleaver to shuffle the message to be transmitted, M, resulting in an interleaved message M_i. Next, the interleaving key is concatenated with the shuffled message M_i and so $[K \; M_i]$ is coded with the systematic code C_i, of size $(n, k+m)$. Then, before sending the coded sequence through the channel, the bits corresponding to the key are punctured, i.e., only the last $n-k$ bits are sent, which corresponds to the shuffled message and the parity bits, P, supplied by the inner code C_i. As such, the key is hidden from both receivers (Bob and Eve), but its information is embedded in the parity bits.

At the receiver, the received sequence is decoded, from which results an estimation of the shuffled message, $\dot{M}_i$, and an estimation of the interleaving key, $\dot{K}$, which is used to deinterleave $\dot{M}_i$, resulting in an estimation of the transmitted message, $\hat{M}$.

In this scheme, security comes from erasing/puncturing the interleaving key, which will then be harder to decode by an adversary eavesdropper in a disadvantageous situation, i.e., having a lower SNR (e.g. an eavesdropper behind a wall).

B. Scrambled Coding for Secrecy with a Hidden Key

Interleaving is fundamental in the previously described scheme as it is responsible for adding security through the random key used to shuffle the message. With it, an interleaving key that is not decoded properly can lead to an incorrectly decoded message. Nevertheless, even for an unsuccessful decoding of C_i, if the receiver (Bob or Eve) manages to recover correctly K^k, the de-interleaving of the message, $\dot{M}_i$, consists only in repositioning the wrong bits, as shown in Fig. 3, thus still possibly leaking information regarding the correct ones. Due to that, scrambling was considered as an alternative choice to that technique. Scrambling, by definition, manipulates the data information by eliminating undesirable long sequences of equal bits which can cause synchronization issues at the receiver, thereby increasing the density of bit transition [16]. As shown in Fig. 4, scramblers are defined based on k linear feedback shift registers, defined by a polynomial $1 + p_1 z^{-1} + p_2 z^{-2} + \cdots + p_k z^{-k}$ (where z^{-1} denotes a unit bit delay) that indicates the position of the switches which affect the sum at the output, with initial states $[\beta_1 \; \beta_2 ... \; \beta_k]$. At the receiver, the descrambler cancels the effects introduced by

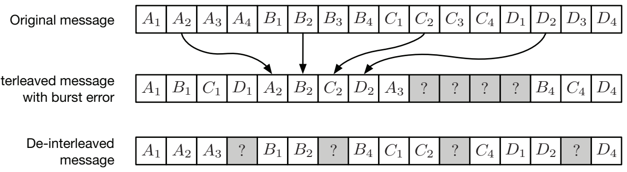

Original message
A_1 A_2 A_3 A_4 B_1 B_2 B_3 B_4 C_1 C_2 C_3 C_4 D_1 D_2 D_3 D_4
Interleaved message with burst error
A_1 B_1 C_1 D_1 A_2 B_2 C_2 D_2 A_3 ? ? ? ? B_4 C_4 D_4
De-interleaved message
A_1 A_2 A_3 ? B_1 B_2 ? B_4 C_1 C_2 ? C_4 D_1 D_2 ? D_4

Fig. 3: Example of interleaving.

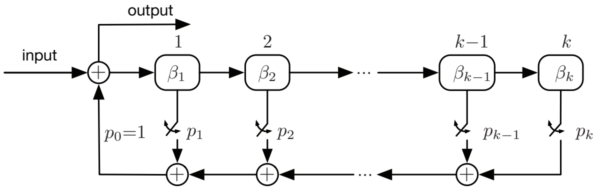


Fig. 4: Scrambler with k shift registers.

the scrambler. This way, scrambling manages to propagate an error along several bits due to its memory effect created by its use of shift registers in its operation (see Fig. 4).

In this paper we then consider scrambling as an alternative to interleaving in order to enhance error propagation. The new scheme is denominated Scrambled Coding for Secrecy with a Hidden Key, also depicted in Fig. 2. The random key K generated per message M to be transmitted, is, in the SCS-HK scheme, used as the initial conditions of the shift registers, giving rise to the scrambled message M_i. The remaining encoding procedure is similar to previously described ICS-HK scheme. At reception, after recovering the punctured key, $\dot{K}$, this is used as the initial conditions of the descrambler's shift registers. Note that, for the SCS-HK scheme, the existence of an error at the received message $\dot{M}_i$, would result in several errors at the descrambled message $\hat{M}$, even if the key $\dot{K}$ is received correctly.

IV. Testbed Practical Implementation

In this section we describe a real testbed implementation of the ICS-HK and SCS-HK coding for secrecy schemes, as well as the practical challenges that arose and how we dealt with them. We consider the use of an LDPC(n,$m+k$) systematic code as inner code C_i and a random interleaving/scrambling key of length k. For the practical deployment we used a set of Ettus USRP B210 SDR boards [17] with omnidirectional VERT2450 antennas, with the development being carried out on Matlab/Simulink, using the USRP Support from Communications System Toolbox [18].

A. Generic SDR Transceiver

The practical deployment of a system for a real testbed evaluation, like for the ICS-HK and SCS-HK, presents a considerable higher degree of complexity when compared to simple simulation, since we must deal with challenges such as carrier synchronization and clock recovery (among others), that are usually ignored when evaluating the performance of those schemes through simulation.

Fig. 5 presents a simplified block diagram with the basic blocks that constitute any SDR transceiver and had to be implemented for the practical deployment of the ICS-HK and SCS-HK schemes (more detail on the specific implementation of the coding for secrecy schemes is presented in the next section). The transmitter (Tx) has four components. The first block receives the message and codes it. Next, this goes through a modulator and a Nyquist pulse filter block (for bandwidth limitation) with the generated sequence being fed to the SDR frontend transmitter (e.g. Ettus USRP B210). This converts the digital signal to an analog signal, translates the signal to the chosen carrier frequency and also amplifies it before being transmitted by the antenna of the SDR.

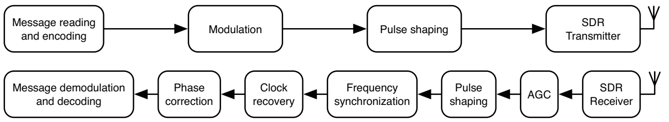


Fig. 5: Simplified block diagram of an SDR transceiver.

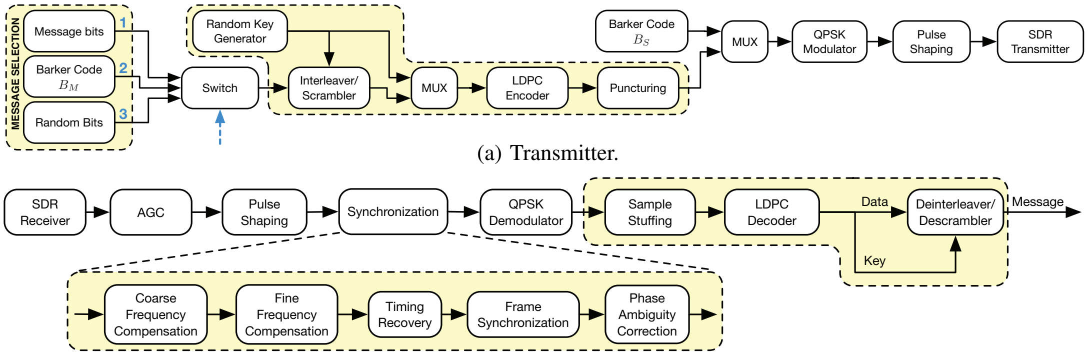
The receiver (Rx) system is considerably more complex. Right after the conversion of the signal from analog to digital at the SDR receiver frontend, the system has to identify and decipher the message in between the received samples. This implies the correction of the changes on the transmitted signal due to the transmission channel, such as frequency deviation and signal attenuation, the estimation of the best sampling time and the subsequent message decoding. Specifically, the signal captured by the antenna goes through an automatic gain controller (AGC) block, that adjusts the gain applied to the received signal, such that its amplitude is constant and high enough so that the signal can be processed. Next, the signal is filtered by a matched filter which maximizes the SNR at the reception and minimizes the inter-symbol interference (ISI) in the optimal sampling times. It follows the recovery of the frequency synchronism and estimation of the optimal sampling time (through clock synchronism). After this, the signal still needs a phase correction and, at last, it is demodulated and decoded to obtain an estimation of the transmitted message.

B. SDR Simulink Program

The vast majority of SDR boards/equipment works just as front-end transceivers, being only responsible for the digital-to-analog conversion (at Tx) and analog-to-digital (at Rx), the translation of the signal to the chosen carrier frequency (at Tx) or to baseband (at Rx) and amplification. All the remaining digital processing is carried out at a computer host to which the SDR board connects. Popular integrated development environment for the SDR deployment at the host are GNU radio [19] and Matlab/Simulink [18]. In this section, we present with more detail the SDR implementation of the ICS-HK and SCS-HK schemes for the case of Simulink.

The simplified block diagram of the ICS-HK and SCS-HK transmitter and receiver implemented in Simulink are represented in Fig. 6. The message file/stream to transmit is sliced into blocks of $n-k$ bits, that are input to the ICS-HK/SCS-HK coding scheme. A random key is generated and fed to the interleaver/scrambler block, which interleaves/scrambles the message. Then, the key is concatenated with the shuffled message and this new sequence is coded with a systematic LDPC code, resulting in a coded message with redundant parity bits. Finally, the bits corresponding to the key are punctured and a Barker code, B_S, of short length, j (set to 13 in our testbed), is appended for frame synchronism purposes (as explained in section IV-C) [20]–[22]. The resulting bit sequence is modulated in QPSK and filtered with a root raised cosine (RRC) filter to limit the transmitted signal bandwidth, and allow, upon match filtering at reception, the reduction of ISI and SNR maximization.

At the Rx, the signal fed by the SDR board is subject to an adaptive gain control, upon the algorithm presented in [21], before filtered by the RRC filter matching the Tx. Next, we perform synchronism to recover the frequency [20], [21], clock adjustment and beginning of the frame detection (see section IV-C) [22] and the signal is demodulated. Before decoding, it is yet necessary to add zeros to



(a) Transmitter.

(b) Receiver.

Fig. 6: Block diagrams of ICS-HK/SCS-HK model.

fulfill the bits punctured at the Tx, thus providing the expected sequence size to the decoder. So, the decoded signal is divided in two: the message and the key. After the deinterleaver/descrambler, we obtain an estimate of the original message.

C. Implementation Challenges

This section describes in more detail some of the most important faced challenges of the testbed implementation.

Synchronization: In what concerns to Rx's side, we need to perform several synchronization mechanisms [23]. First, a coarse frequency synchronization is performed using the M-Power method [20]. Then, a fine frequency compensation and clock recovery are done through phase-lock loop (PLL), with the errors detected with a maximum likelihood phase error detector and a zero-crossing timing error detector, following [21] (chapters 7 and 8, respectively). Clock correction is performed through interpolation. To resolve a possible phase ambiguity and, mainly, to enable frame detection, a Barker code B_S [22] is appended to each transmitted frame, so we are able to find, on the receiver's side, the beginning of each frame, through the calculation of the correlation between the received frame and the Barker code. This sequence is used due to its high autocorrelation peak at zero and low autocorrelation on non-zero values.

Frequency offset of SDR oscillators: Since there are imperfections on the construction of SDR oscillators, they do not oscillate strictly at the same frequency. To overcome this problem, a frequency calibration is needed for each transmitter-receiver pair. To do so, we set both frequencies to the desired value (5 GHz). Using a spectrum analyzer at the receiver's side, we were able to measure the offset between the received frequency and the expected one, enabling us to fix the frequency impairments at each receiver [20].

Turn on/off sequence of Tx and Rx(s): Due to the synchronism methods employed, it is necessary to turn the transmitter on before activating the receivers (Bob and Eve) [23], which causes data loss if the message is immediately transmitted, therefore distorting the results obtained with the practical testbed. Because of that, for synchronization of the data to be sent, the message selection block was designed with three states: transmitter sends random bits (switch 3 of Fig. 6), and as soon as the receivers are turned on, the switch is set in position 1, thus initializing the transmission of the data message. In order to truncate the files at the receiver such that only the message data is analyzed, a package of length n with a repeated Barker code, B_M, [22] (corresponding to switch at position 2) is sent at each transition $1 \leftrightarrow 3$, i.e., the turn on sequence follows the pattern $3 \rightarrow 2 \rightarrow 1$, while the turn off sequence follows the pattern $1 \rightarrow 2 \rightarrow 3$ of the message selection block. This additional package adds a necessary inefficiency because it decreases the message rate but solves the truncation problem. Also Barker codes are more appropriated because they show robustness even in poor channel conditions.

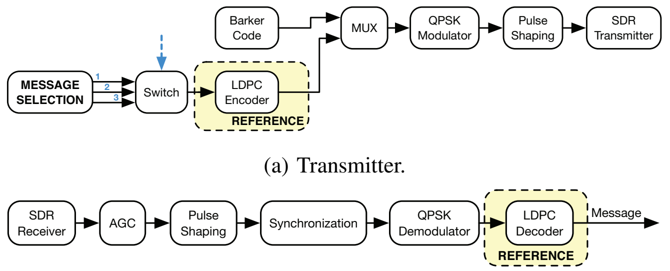
Testbed Rx data validation: Performing real environment tests introduces added unpredictability to the results, that would not occur otherwise in simulations. To address this issue, a Matlab script was written to verify every performed test and discard those instances where noise and/or interference present in the test environment could misrepresent the final results. In order to accomplish that, the correlation between the Barker code B_M and the received data was calculated. Then, the maximum correlation values were found, thus indicating the beginning and ending of the message transmission, and it was verified if the number of packages between these values was as expected. In the case where only the beginning of transmission is detected (as may happen for low SNR scenarios) the received bits are considered as valid data, if the identified maximum correlation value is above a threshold computed based on the cumulative distribution function (cdf) of all correlation values. Then, valid tests are considered when either both signaling packages (beginning and end of transmission) are detected, or when only the signaling package of beginning of transmission is detected, thus simulating an eavesdropper that starts to record as soon as it detects the beginning of transmission.

V. Testbed Scenario, Results and Evaluation

In this section, we present the results and evaluation for the testbed implementation described in section IV. We resort to a configuration with an LDPC(1536,1280) as inner code C_i, random interleaving/scrambling with key length set to $k = 60$, and Barker codes sequences B_S and B_M both having length $j = 13$. For comparison, we also provide results for a reference setup with a simpler scheme consisting solely of the LDPC(1536,1280) to ensure reliable transmission only, as pictured in Fig. 7.

A. Testbed Set-up and Attacker Model

The ICS-HK and SCS-HK coding schemes were designed to provide secrecy against an eavesdropper at the physical-layer for real channels, considering that Bob has an SNR advantage over Eve. The eavesdropper is assumed to be passive, but has complete knowledge of the encoding and decoding algorithms employed. The necessary SNR advantage of Bob over Eve may come from the environment (e.g. Bob appearing in a better location that Eve) or be forced through the generation of interference over Eve [24]. In this setup, we consider the former case, where Eve appears in a degraded location, i.e. it is



(a) Transmitter.

(b) Receiver.

Fig. 7: Block diagrams of reference model.

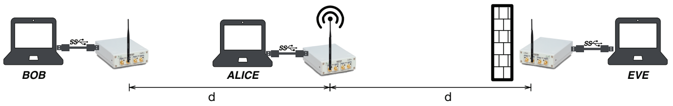


Fig. 8: Testbed scenario.

located in a disadvantageous position in relation to Bob. In order to replicate a real life scenario, Alice and Bob were located in the same compartment, separated by 1.5 meters, and Eve was located outside of it, against the wall, also separated from Alice by 1.5 meters, having the wall between them, as it is shown in Fig. 8. This way, we made sure that Bob was in a better condition than Eve and so it had some advantage introduced by the wall which degraded Eve's channel.

Table I presents the specifications of the host computers connected to the USRP B210 SDR boards for the scenario stakeholders. To avoid interference from surrounding communications, transmissions were performed at 5 GHz. Conditions were kept unchanged throughout the tests, and always carried in the same space, for fixed positions of Alice, Bob and Eve. Only the transmission power at Tx was increased within the range limited by the threshold power below which Bob detects nothing, until the threshold power above which Eve becomes able to decode the whole message correctly. For statistical purposes, for each Tx power level the test was repeated 30 times, with each test consisting of the transmission of 50 replicas of a file with 10^5 randomly generated bits[1]. This allowed us to obtain confidence interval values of 95% for the presented results.

B. Security Considerations

In terms of security, the system will be evaluated based on the notion of SG, which is the minimum difference between channel quality at Bob and Eve that warrants reliability for Bob and security against Eve. The security threshold is obtained by calculating the BER-CDF at Eve and the reliability threshold by calculating the BER from Bob. These calculations are performed as function of the Modulation Error Ratio (MER) [25] instead of the typical ratio of information bit energy to the noise spectral density (E_b/N_0) or the SNR as is used for simulations. This change was necessary because, in practical scenarios, noise power estimation at Rx is difficult. Moreover, it has been shown that the MER has a linear relation to SNR for AWGN channels [25]. The MER provides a measure on how degraded the received signal is, through calculation of the vectors of modulation error, which measure the distance between each received modulated symbol s_{Rx} to its respective ideal point S in the I-Q constellation plane, with MER being computed as:

$$MER = 10\log_{10}\left(\frac{\sum_{j=1}^{N}|S^{(j)}|^2}{\sum_{j=1}^{N}|s_{Rx}^{(j)} - S^{(j)}|^2}\right)[dB] \qquad (2)$$

where j is the index of and N the total number of received symbols.

TABLE I: Testbed host computers specifications.

	Alice	Bob	Eve
Model	Toshiba Satellite P50-C-15L	Asus X555LD	Asus FZ50VX
Processor	Intel Core i7-5500U	Intel Core i7-4510U	Intel Core i7-6700HQ
RAM	16 GB	8 GB	8 GB
Software	MATLAB R2016b @ Windows 10		
USB Version	3.0		

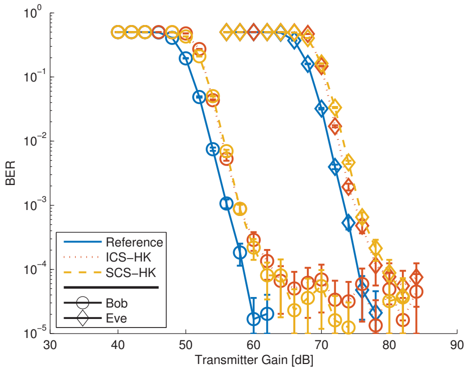


Fig. 9: Overlapped BER with confidence intervals of 95%.

C. Results

BER performance is presented in Fig. 9 as function of the Tx power amplifier gain. It is obvious the SNR advantage of Bob over Eve which requires a higher transmitted power to be able to successfully decode the received data, as expected. It is also observed a small penalty on the required transmit power to achieve the same BER with respect to the reference coding scheme that is used only for reliability. At low BER values, the results show greater variability, corresponding to non recoverable errors that occur over the key, that prevent the deinterleaving/descrambling of a correct received message. More importantly, there is a clear SG between successful (BER$<10^{-4}$) and catastrophic (BER $\approx$0.5) decoding.

To better evaluate the provided secrecy, the BER-CDF metric was used (see also section II-B). Due to the inherent greater variability of real environment tests, it was not viable to use the same BER-CDF conditions for security and reliability as for simulation scenarios [5]. We therefore considered a transmission secure with a BER-CDF greater or equal to 98.5% for $\delta = 0.05$, and reliable if the BER is lower than 10^{-4}. The results presented next show, with markers, the values measured upon reception at Bob and Eve (◊, and □, respectively) and, with a full line, the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) curve, used to evaluate the behavior of the techniques for the non-tested Tx gain points.

Figs. 10 present the results of BER and BER-CDF for the reference, ICS-HK and SCS-HK schemes, respectively. Given the criteria set above for secrecy and reliability, we define MER$_{E,max}$ and MER$_{B,min}$ as, respectively, the maximum MER at which Eve can operate for a secure transmission, and the minimum MER that Bob must possess for a reliable reception. The SG is then the difference between this two values, and we can observe for the reference case presented in Fig. 10(a) that MER$_{E,max}$$\approx$0.5 dB and MER$_{B,min}$$\approx$15 dB, thus leading to a SG of $15-0.5 = 14.5$ dB.

[1] For a matter of data successful validation in worst case conditions, i.e. low transmitted power, as discussed in section IV-C, a transmission of 50 replicas of a file with 10^5 randomly generated bits was preferred to the transmission of a unique file with length 5×10^6 bits.

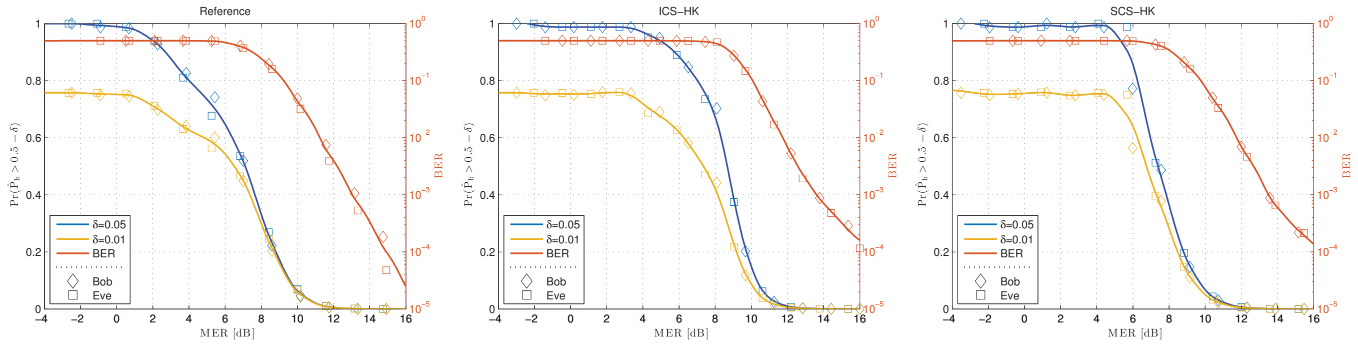


Fig. 10: Measured BER and BER-CDF using an LDPC(1536,1280) for the (a) reference, (b) ICS-HK and (c) SCS-HK schemes.

For the ICS-HK case, presented in Fig. 10(b), we get MER$_{E,max}$≈3.2 dB and MER$_{B,min}$≈17.3 dB, which represents a slight decrease in the SG, for $17.3-3.2 = 14.1$ dB. Although results show that ICS-HK provides a slight increase in secrecy, the practical testbed results were disappointing when compared to the simulations results presented in [5], where the same code C_i and key length were used. The explanation lies on the fact that under simulation, even for very low SNR, the error correcting code C_i presents on decoding a maximum BER of 0.2; the ICS-HK enables, in those same conditions, to closely approach a BER of 0.5 required to guarantee secrecy and thus providing an effective improvement of security against Eve. However, for the practical SDR implementation, the BER come close to 0.5 at low SNR, even for the reference value, since the received packets are simply dropped upon synchronization failure.

Finally, from Fig. 10(c) for the SCS-HK scheme we get MER$_{E,max}$≈4.7 dB and MER$_{B,min}$≈16.6 dB. Here it is observed a clear decrease of the SG to $16.6-4.7 = 11.9$ dB as desired. This represents a relevant increase of security and validates the assumption that led to the proposal of the alternative scheme employing a scrambler instead of interleaving, which was enhancing the propagation of errors on the message upon descrambling, for situations of low SNR where the eavesdropper is able to successful obtain the random key.

VI. Conclusions

We implemented and evaluated coding for secrecy schemes that rely on interleaving and scrambling with a random key to shuffle information before being sent through the channel. This was done in a practical setup with software defined radios, providing evidence of the usefulness of such coding for secrecy schemes in real environments. From the two approaches, the newly proposed scrambling scheme showed better security performance than interleaving, requiring a smaller gap between legitimate receiver and adversary eavesdropper. Along the process, we overcame a set of challenges of implementing coding schemes in software defined radio platforms, whose solutions are also herein described.

References

[1] M. Baldi and S. Tomasin, *Physical and Data-Link Security Techniques for Future Communication Systems*. Springer, 2016.

[2] M. Baldi, M. Bianchi, and F. Chiaraluce, "Coding with scrambling, concatenation, and HARQ for the AWGN wire-tap channel: A security gap analysis," *IEEE Trans. on Inform. Forensics and Security*, vol. 7, no. 3, pp. 883–894, 2012.

[3] D. Klinc, J. Ha, S. McLaughlin, J. Barros, and B. Kwak, "LDPC codes for the Gaussian wiretap channel," *IEEE Trans. on Inf. Forensics and Security*, vol. 6, no. 3, pp. 532–540, 2011.

[4] J. Almeida and J. Barros, "Random puncturing for secrecy," in *IEEE 2013 Asilomar Conference*, 2013, pp. 303–307.

[5] D. Sarmento, J. Vilela, W. K. Harrison, and M. Gomes, "Interleaved coding for secrecy with a hidden key," in *IEEE Globecom Workshops*, 2015, pp. 1–6.

[6] W. K. Harrison, J. Almeida, M. R. Bloch, S. W. McLaughlin, and J. Barros, "Coding for secrecy: An overview of error-control coding techniques for physical-layer security," *IEEE Signal Proc. Magazine*, vol. 30, no. 5, pp. 41–50, 2013.

[7] A. Mukherjee, S. Fakoorian, J. Huang, and A. L. Swindlehurst, "Principles of physical layer security in multiuser wireless networks: A survey," *IEEE Comm. Surveys & Tutorials*, vol. 16, no. 3, pp. 1550–1573, 2014.

[8] A. D. Wyner, "The wire-tap channel," *Bell Labs Tech. Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[9] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[10] W. Harrison, D. Sarmento, J. Vilela, and M. Gomes, "Analysis of short blocklength codes for secrecy," *arXiv preprint:1509.07092*, 2015.

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

[12] U. Maurer, "The strong secret key rate of discrete random triples," *Comm. and Crypt. - Two Sides of One Tapestry*, pp. 271–285, 1994.

[13] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. McLaughlin, and J. Merolla, "Applications of LDPC codes to the wiretap channel," *IEEE Trans. on Inform. Theory*, vol. 53, no. 8, pp. 2933–2945, 2007.

[14] H. Mahdavifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," *IEEE Trans. on Inform. Theory*, vol. 57, no. 10, pp. 6428–6443, 2011.

[15] J. Vilela, M. Gomes, W. Harrison, D. Sarmento, and F. Dias, "Interleaved concatenated coding for secrecy in the finite blocklength regime," *IEEE Signal Proc. Letters*, vol. 23, no. 3, pp. 356–360, 2016.

[16] A. B. Carlson, P. B. Crilly, and J. C. Rutledge, *Communication systems*, 4th ed. McGraw-Hill Higher Education, 2002.

[17] Ettus Research, "USRP B210 USB Software Defined Radio (SDR) - Ettus Research." [Online]. Available: https://www.ettus.com/product/details/UB210-KIT

[18] The Mathworks, Inc., "USRP Support Package from Communications System Toolbox." [Online]. Available: https://www.mathworks.com/hardware-support/usrp.html

[19] The GNU Radio Foundation, Inc, "GNU Radio." [Online]. Available: https://www.gnuradio.org

[20] D. Pu and A. M. Wyglinski, *Digital Communication Systems Engineering with Software-Defined Radio*. Artech House, 2013.

[21] M. Rice, *Digital Communications: A Discrete-Time Approach*. Prentice Hall, 2009.

[22] P. Borwein and M. J. Mossinghoff, "Barker sequences and flat polynomials," in *Number Theory and polynomials*. Cambridge University Press, 2008, pp. 71–88.

[23] The Mathworks, Inc., "QPSK Receiver with USRP Hardware - MATLAB & Simulink Example." [Online]. Available: https://www.mathworks.com/help/supportpkg/usrpradio/examples/qpsk-receiver-with-usrp-r-hardware-1.html

[24] J. Vilela, P. Pinto, and J. Barros, "Position-based Jamming for Enhanced Wireless Secrecy," *IEEE Trans. on Inform. Forensics and Security*, vol. 6, no. 3, pp. 616–627, September 2011.

[25] ETSI DVB Standard TR 101 290 v1.3.1, "Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems," European Telecomm. Stand. Institute, Tech. Rep., 7 2014.