Journal of Hardware and Systems Security
https://doi.org/10.1007/s41635-019-00070-0

Securing Wireless Communication via Hardware-Based Packet Obfuscation

James Chacko[1] · Kyle Juretus[1] · Marko Jacovic[1] · Cem Sahin[1] · Nagarajan Kandasamy[1] · Ioannis Savidis[1] · Kapil R. Dandekar[1]

Received: 28 June 2018 / Accepted: 9 April 2019

Abstract

Obfuscation of the orthogonal frequency-division multiplexing (OFDM) physical layer is described in this paper as a means to enhance the security of wireless communication. The standardization of the communication channel between two trusted parties results in a variety of security threats, including vulnerabilities in WPA/WPA2 protocols that allow for the extraction of the software layer encryption key. Obfuscating the physical layer of the OFDM pipeline provides an additional layer of security in the event that the software layer key is compromised and allows for rolling updates of the physical layer key without altering the software layer key. The interleaver stage of the OFDM pipeline is redesigned to utilize a physical layer key, which is termed Phy-Leave. The Phy-Leave interleaver is evaluated through both MATLAB simulation and hardware prototyping on the Software Defined Communication (SDC) testbed using a Virtex6 FPGA. The implemented rolling physical layer key policy and Phy-Leave system resulted in a less than 1% increase in the area of a Virtex6 FPGA, demonstrating physical layer obfuscation as a means to increase the security of wireless communication without a significant cost in hardware.

Keywords Wireless security · Physical layer security · Logic obfuscation · Secure interleaving · Software defined radios (SDR)

1 Introduction

Consider the case of Bob and Alice exchanging messages over a wireless channel in the presence of Eve, the eavesdropper. To detect a message in flight and to correctly decode it, Bob and Alice must mutually agree on the rules governing the structure, or syntax, of the message—specifically, the length and pattern of the preamble, placement of pilot patterns for signal-distortion compensation, and the error detection and correction scheme. If Eve has prior knowledge of the structure, she too can successfully decode the message. If, however, the structure is obfuscated at run time using a secret key known only to Bob and Alice, Eve cannot decode the message. A real-time packet obfuscation method at the physical (PHY) layer is described in this paper for communication protocols that use orthogonal frequency-division multiplexing (OFDM) to deter the eavesdropper Eve.

The interleaver and de-interleaver stages of an OFDM pipeline are selected for the obfuscation of a packet. The operation of a basic interleaver, where the encoded data enters in-order and is scrambled based on a mapping scheme to produce a different output-order, is shown in Fig. 1. Scrambling the data reduces the effect of burst errors, as the errors are distributed throughout the transmitted data when de-interleaved at the receiver. Transmitted data with more evenly distributed errors provide a higher probability of being corrected by a Viterbi decoder [1]. Therefore, interleaving is widely used in communication systems to improve the performance of forward error correcting (FEC) codes when observing burst errors. In a typical OFDM pipeline, interleaving is done immediately after the encoder stage and de-interleaving right before the decoder stage.

Although the primary function of an interleaver is to improve the reliability of transmitting a message, the interleaver also provides an opportunity to obfuscate communication transmissions from Eve. The design and

✉ Kyle Juretus
kjj39@drexel.edu

Extended author information available on the last page of the article.

Published online: 16 May 2019

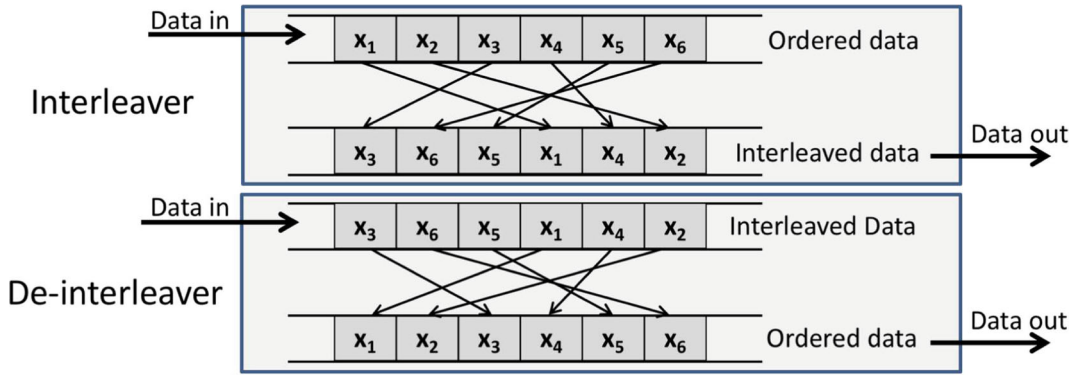

Fig. 1 Operation of a typical interleaver and de-interleaver. The individual bits comprising the packet are represented by x. "Data in" at the transmitter represents the encoded data into the interleaver block from the encoder block and "Data out" represents the movement of the binary reordered data to the modulation block. Similarly, "Data in" at the receiver represents the demodulated data as input to the de-interleaver block from the demodulation block, and "Data out" represents the ordered encoded data heading to the decoding block

implementation of a real-time reconfigurable interleaver and de-interleaver stage within the OFDM pipeline is described, which dynamically changes the input-output mapping as shown in Fig. 1, subject to a secret key known only to Bob and Alice. The reconfigurable structures are implemented such that the area and computational overhead to secure the OFDM pipeline is minimal. The obfuscation of the packet is performed at the PHY layer rather than at a higher-level software layer to support real-time operation of the pipeline. A policy by which Bob and Alice extract and agree on secure keys is described, which is also implemented at the PHY layer, that governs the mapping strategy at the interleaver and de-interleaver. The generation of the key is dynamic in that the key is regenerated periodically at run time based upon an agreed synchronization schedule. The key policy also utilizes shared channel-state estimation at the PHY layer between the transmitter and the receiver to generate symmetric encryption keys at run time [2–4]. The methods described in [2–4] to generate keys guard against protocol vulnerabilities in Wired Equivalent Privacy (WEP) and WiFi Protected Access (WPA) due to the use of pre-shared keys [5, 6] that enable man-in-the-middle and eavesdropping attacks.

The performance of the secure OFDM pipeline is evaluated through both simulation and experimental implementation on a Virtex6 FPGA board. Results under different channel conditions indicate that Eve is unable to recover the transmitted message without knowledge of the secret key. The area overhead as compared to the baseline OFDM pipeline is less than 1% of used FPGA resources. The effect on pipeline throughput is also quantified, which is a function of the interleaver depth and the length of the key. The impact on error correction during normal operation of the proposed interleaving scheme is characterized to confirm continued functionality through channel conditions that include additive white Gaussian noise (AWGN).

The paper is organized as follows: Related work in the area of secure interleaving is discussed in Section 2. An overview of key generation at the PHY layer and the assumed threat model is provided in Section 3. The design and implementation of the packet obfuscation technique are described in Section 4. Experimental results are discussed in Section 5, and concluding remarks are provided in Section 6.

2 Related Work

A methodology to secure Direct-Sequence Code Division Multiple Access by applying Advanced Encryption Standard (AES) based interleaving has previously been explored in [7, 8]. The secure block interleaving scheme presented by Ling et al. [8] loads the data into a $M \times N$ matrix and performs row and column permutations based on the AES algorithm to generate an interleaved sequence. The row and column permutations are completed by utilizing a plaintext and a key. The AES key is shared between the transmitter and receiver, generating a ciphertext that is used as the row or column index for the interleaved output. The permutations are repeated for each row and column in the $M \times N$ matrix until the interleaved sequence is fully generated. Transmitting both the plaintext and key for each row and column index to the receiver provides an adversary with additional opportunities to intercept the shared information. Using an expensive AES operation to generate each row and column index for the interleaved output also limits the ability to update the interleaver mapping. Finally, using AES generated row and column indices does not increase the effort required for a brute-force attack to recreate the $M \times N$ matrix. The technique was extended for use in orthogonal frequency division multiple access (OFDMA) systems in [9], which remapped subcarriers based on an AES permutation.

Khan et al. [10] propose a method to enhance security by implementing interleaving after the modulation phase in the communication pipeline. The technique involves the remapping of complex values representing in-phase and quadrature phase pairs. Performing interleaving on fixed point data increases the memory overhead as storage of 16 bit fixed point numbers is required, as compared to the single bit storage required for the technique described in this paper. The increased overhead limits the implementation of inter-symbol interleaving between frames.

The proposed Phy-Leave technique includes a flexible interleaver with low hardware overhead and provides the ability to rapidly update the key used to generate the interleaving sequence. The size of the interleaver is only bounded by the size of the data buffer, allowing for the interleaving of multiple symbols, which results in an additional barrier for an adversary to overcome when decoding a transmitted message. A time-varying key at the PHY layer is also applied to provide security when the software-based session key is compromised. The developed technique secures the transmitted message at the PHY layer without significantly increasing the overhead in area and performance, as described in Section 5.3.

3 Threat Model and Key Generation

The PHY-based method to generate keys—one that exploits shared channel state information—is described in this section. Other security methods such as the use of pre-shared keys are compatible with the proposed methodology; however, the security vulnerabilities discussed in the introduction apply to the software layer encryption key.

3.1 Threat Model

The primary threats considered are eavesdropping and a man-in-the-middle attack in which Eve tries to impersonate either Alice or Bob. The assumption is that Eve has clear access to the channel and is able to probe the channel between herself and Alice or Bob, including when the secure channel is formed. A further assumption is that Eve has complete knowledge of all the algorithms used to secure the communication between Alice and Bob, as well as knowledge of the hardware that implements the proposed Phy-Leave packet obfuscation. However, Eve is not co-located with Alice or Bob; there is a spatial separation of at least one wavelength. As Eve is also aware of the key extraction algorithm used by Alice and Bob, it is possible that Eve records the transmitted information for analysis at a later time. Finally, a scenario in which Eve is able to determine the software encryption key is also considered, and the generation of a time-varying key, as described in Section 4, is implemented as a means to mitigate the threat. As Eve is not co-located with Alice or Bob, it is assumed that Eve does not have access to the hardware of either party and is, therefore, not able to perform a known-plaintext attack. Additional measures to protect against known-plaintext attacks are possible, including obfuscation of the preamble as described in [11].

Active attack scenarios are also possible, where Eve is actively interfering with the transmission of data between Alice and Bob. Active attacks include, but are not limited to, masquerade attacks [12], replay attacks [13], selective forwarding [14], node replication [15], wormhole attacks [16], and sybil attacks [17]. These attacks all involve Eve recording or manipulating a valid transmission by Alice to either gain unauthorized access or disrupt communication. The generation and use of a time-varying key, as discussed in Section 4, provides a means to limit an adversary from using a previous transmission for nefarious purposes, as the receiver no longer recognizes the data as valid with an incorrect key.

3.2 Key Generation

The wireless channel between Alice and Bob is used as a source of common randomness to generate a correlated random bit sequence that is then utilized to produce a shared secret key. The secret key is applied to secure the communication link between Alice and Bob, while Eve is unable to generate the same key as the channel reciprocity between Alice and Bob is not available to Eve.

Generation of keys based on temporal-spatial properties has previously been explored [18], where keys are generated for low-complexity body sensors. Mathur et al. [2] develop a key generation technique based on channel reciprocity that uses an initial analysis period during which the channel between two radios is sampled using probing packets. The probe packets are assembled at both ends of the communication link and, once enough packets are exchanged, each node independently filters the estimated channel measurements to reduce the impact of fast fading. Once filtering is complete, the nodes compute the standard deviation of the measured channel transmissions, which is used as a threshold to determine if the sampled bit at each time index is a 1, 0, or undefined. A window is then applied to the extracted bits. A bit is considered present at both radios within the window if there are N consecutive bits of the same value. Finally, one radio sends the indices of the estimated bit locations to the other radio, and the other radio replies with a list of confirmed indices that it agrees contain useful bits. The bit sequence is extracted from the final list of indices and is agreed upon as the shared key. Note that Eve only has information pertaining to which samples are used as the bits, but not the values of the samples

themselves. In addition, as the wireless channel is reciprocal only between the two cooperating radios, Eve cannot extract the same bit sequence, leaving her with a useless key.

The method described in [2] was recently extended into a real-time technique that relies on 802.11 preamble information rather than using dedicated probing packets [4]. The bursty and asymmetric nature of the application-layer traffic is accounted for by applying a sampling technique.

On each radio, an internal timer interrupt occurs approximately once per channel-coherence time interval, which is dependent on the environment and ranges between tens to hundreds of milliseconds [19]. When an interrupt occurs, the next received packet transmitted from a participating radio is used in the channel estimation process. While mobility improves the uniqueness of the channel and provides increased confidence in the extracted bits [20], high mobility also results in shorter channel coherence times [3]. If a key update occurs solely based on interrupts, scenarios with high mobility lead to increased computational overhead due to frequent updates of the key. To limit the computational overhead, the key is valid for a variable amount of time independent of the frequency of the occuring interrupts.

4 System Design and Implementation

The implementation of the proposed obfuscation technique to secure a communication link is described in this section, including the policy for generating time-varying keys at the PHY layer. The data interleaving policy, which is a function of the obtained key, is discussed. An implementation of the obfuscation technique on an FPGA is also described.

4.1 PHY-Based Packet Obfuscation

The method used to generate the key at the PHY layer and the use of the created key to determine the input-output mapping for the interleaver are illustrated in Fig. 2. The steps involved in the generation and application of the key are described as Step 1 and Step 2, respectively.

Step 1: Generating the Shared Key

The shared key is generated as described in Section 3.2 using the technique developed by Katz et al. [4]. The channel state is estimated independently at the transmitter and the receiver based on the packets exchanged between the two. The state is used to generate bits based on the channel symmetry that form the key, which are continuously placed into a shift register as shown in Fig. 2. The bits are continuously updated until a system specified event occurs, upon which the current key value is transferred from the shift registers into memory. The system event is either time-triggered based on an interrupt or event-triggered based on the number of packets successfully transmitted or received. At this point, the key is (1) used on its own, (2) mixed with a software encryption key from the application layer, or (3) mixed with the previously valid physical encryption key to generate a more secure key. The key mixing function shown in Fig. 2 is applied with XORs, however, a hash or AES-based implementation of the mixing function is also possible to secure against the threat of a one-time pad. By generating the shared key from prior valid keys, adversaries are restricted to a fixed window to determine and utilize a given key.

Step 2: Controlling the Interleaver Mapping

The PHY-layer key generated by the process described in Step 1 is used to control the interleaving sequence within the OFDM pipeline. The proposed system supports two different implementations of interleaver sequencing. For the first approach (Method A), the lower-order k bits are extracted from the PHY-layer key and a hash is applied as an address to index into the memory (ROM) that stores multiple pre-computed input-output mapping schemes for the interleaver to use. The second approach (Method B) uses a Fibonacci Linear-Feedback Shift Register (LFSR) to generate pseudo-random interleaving mappings on the fly and stores the mappings into volatile memory (RAM). For Method B, the PHY-layer key is used as the initial seed to the LFSR, and the output from the LFSR is used to produce random memory mappings that are stored in the RAM. As the output bits of the LFSR are not transmitted, the adversary is not able to use the Berlekamp-Massey algorithm, as is discussed in [21].

The primary difference between Method A and Method B is the execution time required to produce an optimal mapping of the interleaving. To avoid non-deterministic key generation times possible when applying Method B and sharing predetermined keys as done with Method A, a mixed technique is implemented that applies Method B to generate the mapping sequences placed into the RAM and then uses Method A to hash the sequences. Placing mapping schemes into a RAM allows for caching sequences, which permits faster key updates and deterministic update policies. Further work is required to analyze the threat of an adversary determining future physical layer keys based on attacks to stored potential keys in RAM.

Mapping sequences are utilized to create an interleaved data sequence at the transmitter and a de-interleaved sequence at the receiver. A large Hamming distance is needed for the generated mapping, in contrast to the standard-based interleaver sequence used for wireless communication, as a small difference between the two interleaving sequences allows adversaries to apply FEC

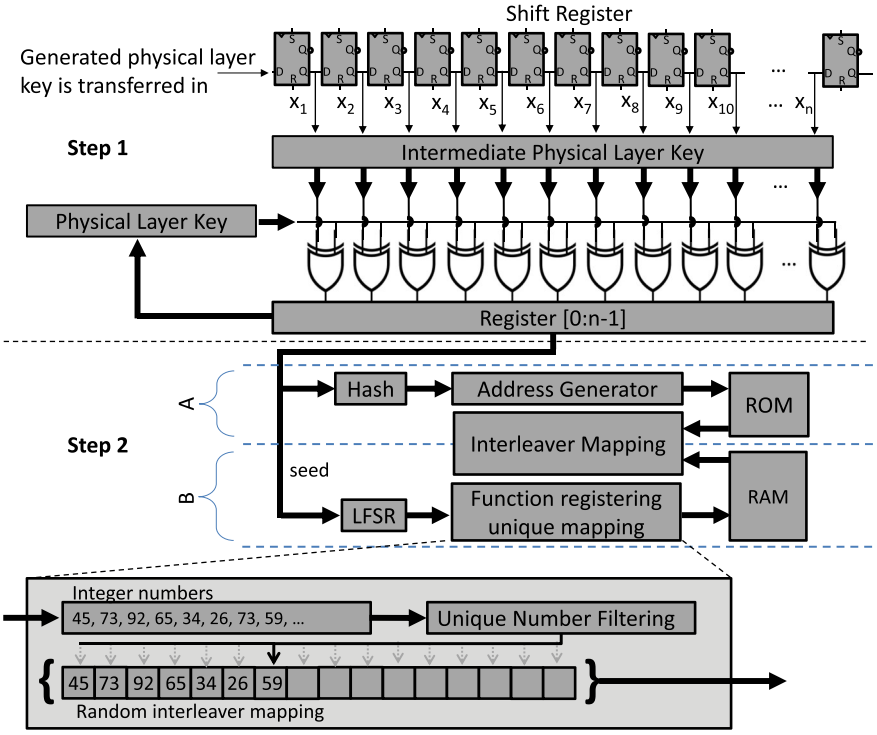

Fig. 2 Overview of the proposed packet obfuscation technique

algorithms to decode the transmission. Point A in Fig. 3 represents the case where the generated mapping scheme does not produce a significant Hamming distance from the standard interleaver sequence. In such a case, the interleaver sequence is regenerated to produce a larger Hamming distance from the standard-based interleaver.

4.2 Challenges in Synchronization

Synchronously updating the mapping of both the transmitter and receiver interleavers with a new unique key is the primary challenge of the proposed key policy. To address synchronization issues including clock drift, the proposed

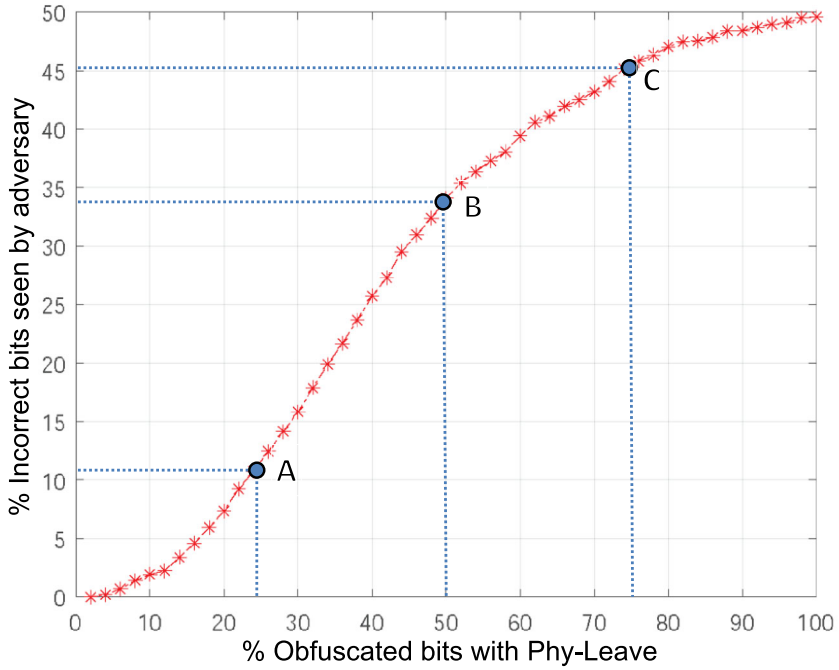

Fig. 3 Comparison of the percentage of correct bits seen by the adversary with respect to the percentage of the transmission frames secured using Phy-Leave. Based on the bit obfuscation policy, if the randomly generated key produces outcome A, where 25% of the bits are obfuscated and the resulting Hamming distance from the correct bit sequence is 10%, the code iterates to produce another key corresponding to point B or C that increases the errors observed by the adversary

key policy uses a combination of asynchronous and synchronous phases between the transmitter and receiver. During the asynchronous phase, a unique channel-derived key is generated every 5 to 10 packets, which is then loaded into the shift registers present at both the transmitter and receiver on a packet-based interval independent of the system clock.

The entire proposed physical layer-based security mechanism, after the availability of an intermediate physical layer key (see Fig. 2) at both the transmitter and receiver, requires strict synchronization. Once the key bits are selected and transferred to the intermediate physical layer key stage, the bits are available to the software layer as a session key. To avoid dependence on board clocks, the proposed security methodology is driven by interrupts generated based on successful packet transmissions observed at the transmitter and receiver. The packet-based interrupt mechanism generates one pulse per successful packet transfer, which is indicated by the reception of an ACK packet from the receiver to the transmitter. The recovery from lost packets in the channel medium involves shifting the LFSR forward by a known number of steps, which is derived from the initial physical layer key both at the transmitter and receiver, at the end of a timeout with a fixed duration.

A strict packet-based synchronization is assumed within the experimental framework through an external signal whose frequency is assigned from a set of predetermined values. The external signal is available to the circuit block at both the transmitter and receiver implementing the physical layer security technique. The use of the MAC layer for cross-layer synchronization to remove the considered assumptions is beyond the scope of this paper.

4.3 FPGA-Based Implementation

Hardware verification of the secure interleaving method is implemented on the Software Defined Communication (SDC) testbed [22]. The SDC testbed is a highly flexible physical layer implementation of an OFDM pipeline that enables rapid prototyping for wireless research, allowing for the pipeline modifications necessary to implement the Phy-Leave system. The modifications to the OFDM pipeline, as shown in Fig. 4, are applied in the SDC testbed to experimentally characterize the Phy-Leave system.

4.3.1 Implementation of Key Policy on FPGA

The key policy discussed in Section 4.1 is adapted to an FPGA fabric. To implement the key policy on an FPGA, the bits generated from the real-time algorithm described in Section 3.2 are placed into a shift register capable of storing N bits, where N is the size of the key. After each interrupt, a new key bit is generated, replacing one of the previous key bits. A separate counter that is either time or packet dependent (time in the current implementation) is added to determine when the content of the shift register is stored in memory. The content stored in memory serves as the key for the software layer for the current time session. The current session key applied within the software layer is XORed with the physical layer key, which was initialized to all zeros, to create a new temporally dependent key. The XORing of the software and physical layer keys increases the number of key sequences an adversary must attempt to successfully determine the physical layer key. The key generation procedure results in more bits than needed, 1024 bits in this work. A frame of 100 bits is extracted from the 1024 bits to use with the proposed interleaving scheme.

The 1024 bits are produced through the implementation of the key generation algorithm described in [3], which generates keys based on a few packet exchanges between the active non-adversarial transmitter and receiver. Although only a fraction of the 1024 bits are utilized for both implementations (Methods A and B), the additional bits allow for securing other circuit sub-blocks of the OFDM pipeline.

4.3.2 Implementation of the Key Update Procedure on FPGA

The interleaver block controller updates the key to allow for scalability and modularity. In the first phase, the block controller finishes interleaving the data already present within the buffer belonging to the frame currently being processed. The obfuscated interleaving of further data is stalled until the new physical layer key is generated, which prevents the loss of valid data that was in transit. The duration of the stall is dependent on the mixing function utilized in the proposed packet obfuscation technique shown in Fig. 2, with the XOR mixing function only requiring a stall of a single cycle. If the duration of the stall of the mixing function is too large, the process to update the key is further optimized by computing the key in parallel to the transmission of a data packet. However, the additional pipelining (parallelism) results in increased area and power consumption. Once the new physical layer key is produced, a control signal is generated to update the mapping scheme and to write the scheme into the addressable block memory on the FPGA, which is referred to as mapping_buff.

Both Methods A and B described in Section 4.1 were implemented using a combination of Xilinx SysGen hardware modules and Verilog-based control functions. Since the spectrum characteristics used to derive the keys are random, a function written in Verilog uniquely identifies integer numbers from the LFSRs to create unique mapping sequences for the specified interleaver depth. Once validated, the mapping sequences are written into the mapping_buff implemented using the RAM.

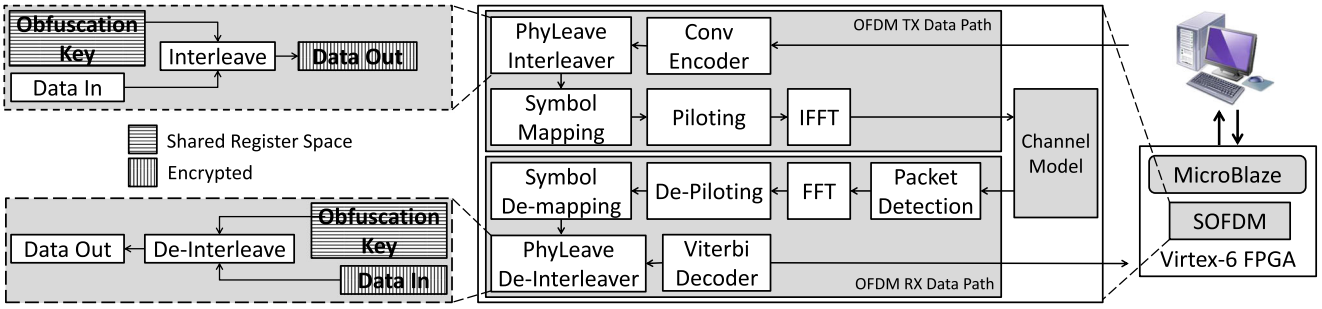

Fig. 4 System layout of the SDC testbed showing the (de)obfuscation based (de)interleaving modules in the transmit and receive chain

With the mapping_buff loaded with the interleaver mapping sequences, the system sends a start signal to the interleaver. Upon receiving the start signal, the data to interleave is read from the input buffers to an addressable block memory on the FPGA, which is referred to as data_buff. Once data_buff has sufficient data to comprise a frame, the functional module interleaves the data by reading data_buff out-of-order based on the addresses stored in mapping_buff that correspond to the new mapping scheme. The base processing delay due to the interleaving of any given encryption mapping is a function of the buffer size required to store the last of the data being read out. Optimization of the time required for buffering is not a viable option as the proposed key generation scheme uses a random mapping order that results in a base delay corresponding to the maximum frame size that is interleaved.

Switching into the Phy-Leave secure mode on the FPGA requires the rest of the pipeline to accommodate the additional delay due to base processing. The interleaver of the baseband pipeline is placed between the convolutional encoder and the QAM symbol mapping modulation units, as shown in Fig. 4. The encoder must delay sending more data if the interleaver is switching to the Phy-Leave secured mode or implementing other policy changes. In addition, the QAM modulator delays reading partial (or null) data while waiting for the interleaver to switch into secure mode or make policy changes. All the sub-blocks that constitute the scalable OFDM core within the SDC testbed are insensitive to functional latencies with respect to each baseband module. Therefore, the extra processing latency introduced by the implementation of the Phy-Leave security protocol in the interleaver does not interfere with the physical layer implementation of the baseband [23].

4.3.3 Implementation of Phy-Leave on FPGA

Interleaving is typically completed using a convolution or block based methodology, which differ in terms of resources required and the speed of operation. The Phy-Leave technique described in this paper applies block based interleaving, which allows for increased flexibility when interleaving various frame sizes. The physical layer key generated by the methodology discussed in Section 4.1 is used as a seed for a memory address generator, of which two implementations are provided in Section 5. The memory address generator utilizes a dynamic sequence to place the content of the memory into a data sequence set for transmission. The receiver requires a de-interleaver that performs the opposite function of the interleaver. The receiver is provided the same physical layer key as the transmitter, generated with the technique described in Section 3. The memory address generator stores the received data in the original pre-interleaved order. The memory is read to verify transmission of the original message.

5 Performance Evaluation

The Phy-Leave security technique uses keys generated by the process defined in [4], which was implemented using the Wireless Open-Access Research Platform (WARP) [24] with a 802.11-2012 based experimental setup. The implementation of the Phy-Leave system through both MATLAB simulation and on a Virtex 6 ML605 FPGA is described in this section.

5.1 Interleaver Validation

As interleavers are designed to mitigate the effects of burst errors by spreading the error bits across the packet, which enables forward error correction, the effect on the bit error rate (BER) is characterized when implementing the Phy-Leave technique. An analysis and comparison of the BER between Phy-Leave interleaving with half-rate convolutional coding, IEEE 802.11a interleaving with half-rate convolutional coding, and without coding and interleaving is performed through simulation [25]. Note that the single stream mode of IEEE 802.11n and IEEE 802.11a apply identical interleaving operations. However, an additional permutation is required for multiple

input multiple output (MIMO) 802.11n systems. As a single input single output (SISO) system is considered in this work, the use of the IEEE 802.11a interleaver is appropriate and interchangeable with IEEE 802.11n operating in SISO mode. A 20 MHz OFDM signal is considered with Quadrature Phase Shift Keying (QPSK) modulation, 64 sub-carriers (48 used for data), a cyclic prefix of 16, a minimum mean square error channel equalizer with ideal noise variance estimates, and perfect synchronization. The length of the Phy-Leave interleaver was constrained to match the block-type interleaver used in IEEE 802.11a for fair comparison. A 802.11n TGn multipath fading channel model was applied, which included the power delay profile F and varying levels of bit energy to noise spectral density ratios (normalized signal-to-noise ratio) [26]. The selected model represents a large indoor and/or outdoor hotspot environment. In addition, the model consists of six clusters and includes an RMS delay spread of 150 ns, a maximum delay of 1050 ns, and a Rician K-factor of 6 dB. As an additional point of comparison, the analysis of an uncoded QPSK without interleaving is provided to characterize and emphasize the benefits of coding and interleaving provided by the proposed system transmitting over a model of a WLAN TGn channel.

The interleaver performance for cases with and without an active noise signal jammer is characterized through analysis of the bit error rate (BER), with results shown in Fig. 5. A signal-to-interference ratio of 5 dB relative to the received signal is specified. For scenario 0 (s0), no active jamming is applied, while a jammer is present in all other cases. The s1, s2, and s3 scenarios correspond to, respectively, 1 MHz, 2.5 MHz, and 5 MHz bandwidths of the jammer.

Coding and interleaving is shown to provide a benefit over the uncoded QPSK technique for energy per bit to noise power spectral density ratios $\frac{E_b}{N_0}$ greater than 10 dB. An $\frac{E_b}{N_0}$ crossover point known as the coding threshold exists, at which point an interleaver implementing forward error correction produces a lower performance as measured by an increase in the BER when compared with an uncoded scheme [27]. The BER is shown to increase with the bandwidth of the jammer. The use of Phy-Leave results in a small reduction in the performance of the wireless system as compared to the IEEE 802.11a standard interleaver, which is expected as the standard method is designed to optimally spread burst errors. The observed loss of 0.3 dB for a BER of 10^{-2} without a jammer and 0.7 dB for a BER of 10^{-2} with a 1 MHz bandwidth jammer is considered an acceptable tradeoff for the security provided to the communication channel.

5.2 Experimental Setup

In order to implement and characterize the Phy-Leave obfuscation technique, a simulation framework was developed in MATLAB sysGen to transmit and receive packets

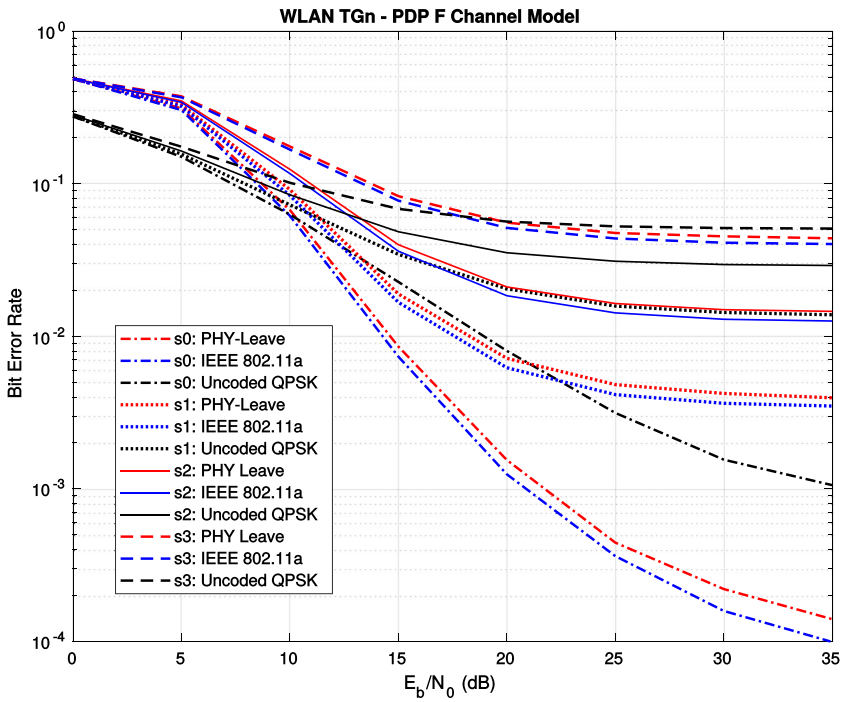

Fig. 5 Comparison of the bit error rate (BER) of the Phy-Leave and the IEEE 802.11a interleaver using a WLAN TGn channel model with power delay profile F and jamming scenarios s0 (no jamming), s1 (1 MHz jamming), s2 (2.5 MHz jamming), and s3 (5 MHz jamming)

through a simulated channel. The simulator accepts a string of data, a key sequence, a target signal-to-noise ratio (SNR), and burst noise control to accurately analyze data through the stages of the transmitter and receiver depicted in Fig. 4. The simulation framework allows for the characterization of the system under various noise constraints as well as when partial key information is known by the adversary. The wireless transmitter and receiver were implemented through an AWGN channel with adjustable SNR to allow for a controlled testing environment.

QPSK was selected as the per sub-carrier modulation scheme for the payload. The effects of noise on QPSK are shown in Fig. 6a. Based on the quadrant a signal is received in, the demodulator at the receiver determines if the received symbol is bit-mapped to "00," "01," "10," or "11," which represents the original bit sequence. A channel with a lower SNR shifts the received symbol further from the quadrant at which it is expected and, therefore, results in a demodulation to an incorrect pair of bits.

5.3 Interleaver Performance

The MATLAB simulation described in Section 5.2 is used to analyze the Phy-Leave technique for SNRs of 2, 4, 6, 8, and 12 dB and with the following experimental configurations:

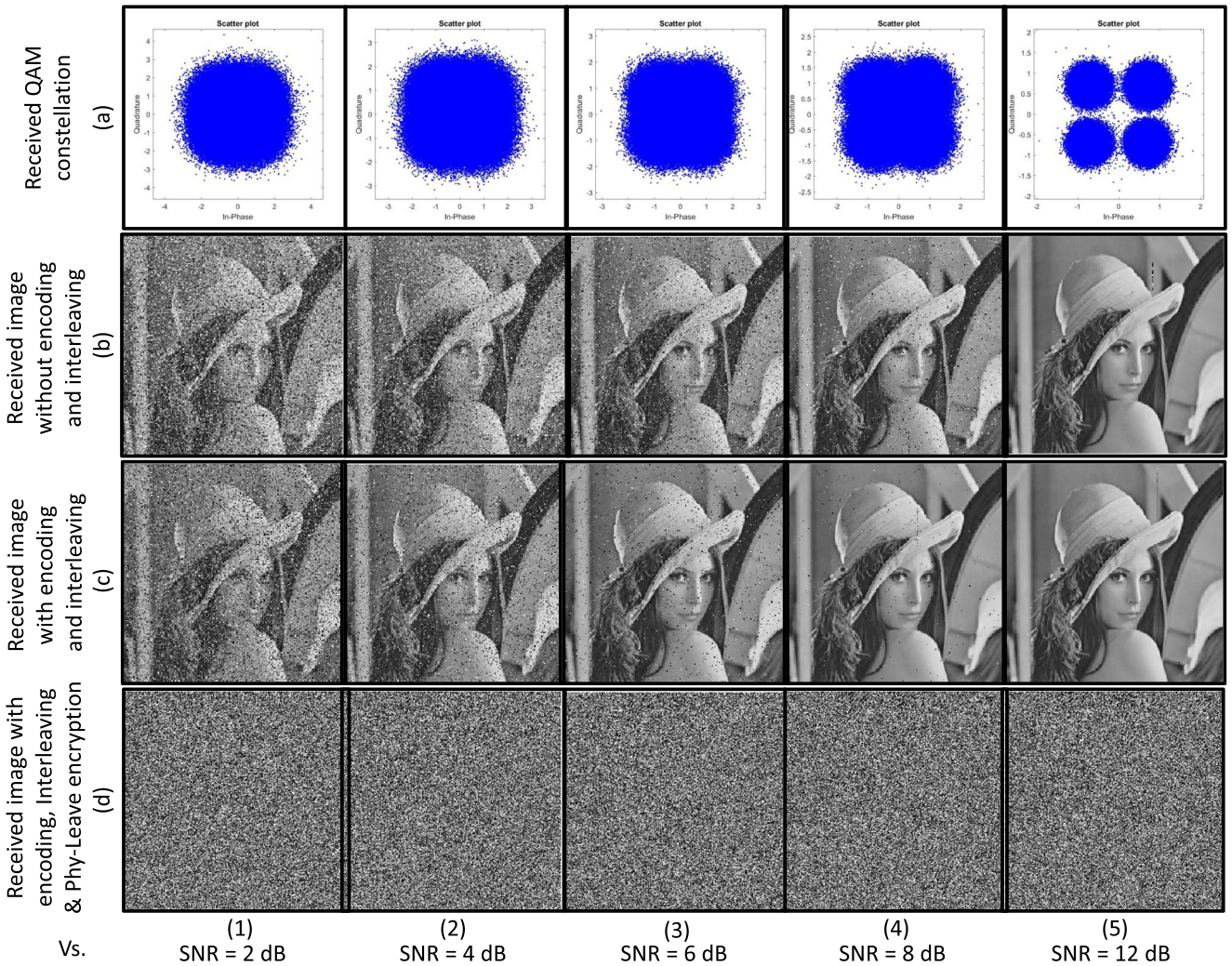

Fig. 6 Experimental results showing **a** the received constellation at the receiver, **b** the post-processed received image of a transceiver baseband with no interleaver and no FEC encoding, **c** the post processed received image of a transceiver baseband with both the interleaver and FEC encoding, and **d** the post processed received image of a Phy-Leave encrypted payload with transceiver baseband implementing both interleaver and FEC encoding. All the above results are shown for an AWGN channel 2, 4, 6, 8, and 12 dB of SNR

1. without FEC encoding/decoding and interleaving,
2. with FEC encoding/decoding and interleaving, and
3. with FEC encoding/decoding and the Phy-Leave interleaving system.

The results characterizing the received QPSK constellation and the three experimental configurations are shown in Fig. 6, where the sub-figures are described as:

- a(1) to a(5): The quality of the received symbols improves as the SNR increases.
- b(1) to b(5): The processed and decoded result corresponding to SNRs of 2, 4, 6, 8, and 12 dB for the case where there was no encoding or interleaving. The received image improves as the SNR increases.
- c(1) to c(5): The case where coding and interleaving is applied. The image is received with improved quality as compared to b(1) to b(5).
- d(1) to d(5): The case where the receiver, without knowledge of the key, is attempting to decode the received symbols that were obfuscated using Phy-Leave interleaving at the transmitter.

The results indicate that the receiver is not able to recover the transmitted image without knowledge of the key. The data is decoded correctly, similar to the case shown in c(1) to c(5), when the independently generated key is used to properly set the mapping within the Phy-Leave decryption block of the receiver. For FPGA and MATLAB simulations of the implemented Phy-Leave technique, the key is assumed known at the receiver to correctly decode the transmission. Otherwise, the BER is close to the 50% theoretical worst.

5.4 Resource Utilization and Pipeline Performance

The FPGA resources needed to implement the Phy-Leave system and the standard interleaving method are listed in Table 1. The tabulated results indicate an average utilization difference of 17 occupied slices, 46 flip-flops, and 27 look-up tables (LUTs) when Phy-Leave is implemented instead of the standard interleaver; a nominal increase in resource utilization. The resources required to implement the Phy-Leave core are less than 1% of the FPGA fabric of a Virtex6 ML605 board used for experimental verification of the obfuscation technique.

The scaling of the Phy-Leave core to secure larger symbol sizes (2^7, 2^8, 2^9) required an additional 14 occupied slices, 2 flip-flops, and 7 LUTs. Although the Phy-Leave core can be modified to accommodate encryption across OFDM symbols, it is recommended that obfuscation based on Phy-Leave is executed within an OFDM symbol to reduce the complexity of implementing other frame-based baseband modules.

Table 1 Resource utilization as a function of total slices, flip-flops (FFs), and lookup tables (LUTs) on the Virtex6 FPGA for symbol sizes of 2^7, 2^8, and 2^9 as compared to a standard interleaver implementation. Results indicate a resource overhead of up to 7% over the standard interleaver. However, the overhead of the total system that includes the Phy-Leave interleaver is less than 1% of the resources available in the Virtex6 FPGA ML605 Evaluation board

Resource utilization						
Size	Slices		FFs		LUTs	
	Base	Secured	Base	Secured	Base	Secured
2^7	332	354	631	677	698	725
2^8	338	347	634	680	708	725
2^9	342	362	635	681	702	739

The only additional latency of the Phy-Leave system over the standard implementation is when updating the key used by the interleaver. The interleaver stalls the pipeline to perform an update only if the securing policy provides no idle time between consecutive packets. One possible technique that masks the latency of updating the key utilizes the idle time between packet frames, as implemented in this paper for a transceiver that includes an interleaver symbol size of 100 and applies QAM modulation over a 128 wide OFDM symbol size.

The time required to complete a key update in the event of a stall is directly proportional to the size of the interleaver. Therefore, the loading time of a mapping scheme for a 100 point interleaver is less than the time for a 200 point interleaver. The FPGA resources required to implement a large interleaver depth is negligible, with a maximum increase of 7% in total slices, flip-flops, and lookup tables, as indicated by the data listed in Table 1. The latency in updating a key is further reduced by parallelizing the process of loading the interleaver mapping registers.

6 Conclusion

The obfuscation of the physical layer with the Phy-Leave interleaver is presented in this paper to increase the security of wireless communication. A key policy capable of rolling updates and an interleaver system that obfuscates packet transmissions is presented and implemented on a Virtex6 FPGA with less than 1% area overhead and no impact to the clock frequency. Experimental results demonstrate that when an incorrect key is applied, a BER of approximately 50% (essentially random) is seen at the receiver. The increased security, including the determination of the physical layer key and the rolling updates to the physical key, and the low overhead to implement the Phy-Leave technique demonstrate that packet obfuscation at the

physical layer provides a means to increase the security of wireless communication.

Funding Information This research was supported by the National Science Foundation Grant No. CNS-1228847, CNS-1730140, CNS-1816387, and DUE-1241631. Additional support was provided by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowships and 32 CFR 168a.

References

1. Haykin S (1988) Digital communications. Wiley, New York
2. Mathur S, Trappe W, Mandayam N, Ye C, Reznik A (2008) Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In: Proceedings of the ACM international conference on mobile computing and networking, pp 128–139
3. Sahin C, Katz B, Dandekar K (2016) Secure and robust symmetric key generation using physical layer techniques under various wireless environments. In: Proceedings of the IEEE radio and wireless symposium, pp 211–214
4. Katz BZ, Sahin C, Dandekar K (2016) Real-time wireless physical layer encryption. In: Proceedings of the IEEE annual wireless and microwave technology conference, pp 1–4
5. Fluhrer S, Mantin I, Shamir A (2001) Weaknesses in the key scheduling algorithm of RC4. In: Proceedings of the annual international workshop on selected areas in cryptography, pp 1–24
6. Wifi Pineapple, https://www.wifipineapple.com
7. Ahmad A, Biri A, Afifi H (2008) Study of a new physical layer encryption concept. In: Proceedings of the Ieee international conference on mobile Ad Hoc and sensor systems, pp 860–865
8. Ling Q, Li T, Ren J (2005) Physical layer built-in security enhancement of DS-CDMA systems using secure block interleaving. In: Proceedings of the IEEE global telecommunications conference, vol 3, pp 1–5
9. Lightfoot L, Zhang L, Ren J, Li T (2009) Secure collision-free frequency hopping for OFDMA-based wireless networks. EURASIP J Adv Signal Process 2009(1):361063
10. Khan MA, Jeoti V, Manzoor RS (2011) Secure interleaving-physical layer security enhancement of OFDM based system. e-Technologies and Networks for Development, pp 349–361
11. Chacko J, Juretus K, Jacovic M, Sahin C, Kandasamy N, Savidis I, Dandekar K (2017) Physical gate based preamble obfuscation for securing wireless communication. In: Proceedings of the IEEE international conference on computing, networking and communications, pp 293–297
12. Shiu Y, Chang SY, Wu H, Huang SC, Chen H (2011) Physical layer security in wireless networks: a tutorial. IEEE Wirel Commun 18(2):66–74
13. Syverson P (1994) A taxonomy of replay attacks [cryptographic protocols]. In: Proceedings of the computer security foundations workshop, pp 187–191
14. Bysani LK, Turuk AK (2011) A survey on selective forwarding attack in wireless sensor networks. In: Proeedings of the IEEE international conference on devices and communications, pp 1–5
15. Zhu WT, Zhou J, Deng RH, Bao F (2012) Detecting node replication attacks in wireless sensor networks: a survey. J Netw Comput Appl 35(3):1022–1034
16. Khalil I, Bagchi S, Shroff NB (2005) LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks. In: Proceedings of the IEEE international conference on dependable systems and networks, pp 612–621
17. Karlof C, Wagner D (2003) Secure routing in wireless sensor networks: attacks and countermeasures. In: Proceedings of the IEEE international workshop on sensor network protocols and applications, pp 113–127
18. Ali ST, Sivaraman V, Ostry D (2010) Secret key generation rate vs. reconciliation cost using wireless channel characteristics in body area networks. In: Proceedings of IEEE/IFIP international conference on embedded and ubiquitous computing, pp 644–650
19. MacLeod H, Loadman C, Chen Z (2005) Experimental studies of the 2.4-GHz ISM wireless indoor channel. In: Proceedings of the annual communication networks and services research conference, pp 63–68
20. Premnath SN, Jana S, Croft J, Gowda PL, Clark M, Kasera SK, Patwari N, Krishnamurthy SV (2013) Secret key extraction from wireless signal strength in real environments. IEEE Trans Mob Comput 12(5):917–930
21. Massey J (1969) Shift-register synthesis and BCH decoding. IEEE Trans Inf Theory 15(1):122–127
22. Chacko J, Sahin C, Pfiel D, Kandasamy N, Dandekar K (2015) Rapid prototyping of wireless physical layer modules using flexible software/hardware design flow. In: Proceedings of the ACM/SIGDA international symposium on field-programmable gate arrays, pp 32–35
23. Chacko J, Sahin C, Nguyen D, Pfeil D, Kandasamy N, Dandekar K (2014) FPGA-based latency-insensitive OFDM pipeline for wireless research. In: Proceedings of the IEEE high performance extreme computing conference, pp 1–6
24. WARP Project, http://warpproject.org
25. ISO/IEC Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (Includes IEEE Std 802.11, 1999 Edition; IEEE Std 802.11A.-1999; IEEE Std 802.11B.-1999; IEEE Std 802.11B.-1999/Cor 1-2001; and IEEE Std 802.11D.-2001), ISO/IEC 8802-11 IEEE Std 802.11 Second edition 2005-08-01 ISO/IEC 8802 11:2005(E) IEEE Std 802.11i-2003 Edition, pp 1–721 (2005)
26. Erceg V, Schumacher L, Kyritsi P (2004) IEEE P802.11 wireless LANs: TGn channel model (IEEE 802.11-03/940r4)
27. Lin S, Costello DJ (2004) Error control coding, 2nd edn. Prentice-Hall, Inc., Upper Saddle River

Affiliations

James Chacko[1] · Kyle Juretus[1] · Marko Jacovic[1] · Cem Sahin[1] · Nagarajan Kandasamy[1] · Ioannis Savidis[1] · Kapil R. Dandekar[1]

James Chacko
jjc652@drexel.edu

Marko Jacovic
mj355@drexel.edu

Cem Sahin
cs486@drexel.edu

Nagarajan Kandasamy
nk78@drexel.edu

Ioannis Savidis
is338@drexel.edu

Kapil R. Dandekar
krd26@drexel.edu

[1] Drexel University, 3141 Chestnut Street, Philadelphia, PA, 19104, USA