Journal of Hardware and Systems Security (2020) 4:230–245
https://doi.org/10.1007/s41635-020-00097-8

Physical Layer Encryption for Wireless OFDM Communication Systems

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

Received: 7 November 2019 / Accepted: 19 May 2020 / Published online: 23 July 2020

Abstract

Our everyday lives are impacted by the widespread adoption of wireless communication systems integral to residential, industrial, and commercial settings. Devices must be secure and reliable to support the emergence of large scale heterogeneous networks. Higher layer encryption techniques such as Wi-Fi Protected Access (WPA/WPA2) are vulnerable to threats, including even the latest WPA3 release. Physical layer security leverages existing components of the physical or PHY layer to provide a low-complexity solution appropriate for wireless devices. This work presents a PHY layer encryption technique based on frequency induction for Orthogonal Frequency Division Multiplexing (OFDM) signals to increase security against eavesdroppers. The secure transceiver consists of a key to frequency shift mapper, encryption module, and modified synchronizer for decryption. The system has been implemented on a Virtex-7 FPGA. The additional hardware overhead incurred on the Virtex-7 for both the transmitter and the receiver is low. Both simulation and hardware evaluation results demonstrate that the proposed system is capable of providing secure communication from an eavesdropper with no decrease in performance as compared with the baseline case of a standard OFDM transceiver. The techniques developed in this paper provide greater security to OFDM-based wireless communication systems.

Keywords Communication system security · OFDM · Physical layer · System implementation

1 Introduction

Eavesdropping and man-in-the-middle attacks are serious security threats in wireless communication. Encryption schemes at the software layer such as Wi-Fi Protected Access (WPA/WPA2) have been previously compromised [1]. Although WPA3 is designed to be resistant to brute force attacks, the Dragonblood Attack exploited the vulnerabilities of WPA3 to dictionary, group downgrade, and side-channel attacks [2]. Further security methods must, therefore, be developed to protect communication systems.

Physical layer security provides additional protection of wireless communication between devices. The security techniques implemented at the physical (PHY) layer are used to complement and strengthen security protocols at higher layers. PHY layer security consists of two primary components: key generation and encryption [3]. Secret keys are generated using properties of the wireless channel to help secure a communication link. Encryption in this context consists of encoding signals by introducing strategic modifications to stages within the transceiver pipeline to secure the communication channel. This work explores PHY layer encryption of Orthogonal Frequency Division Multiplexing (OFDM) signals, which is a widely adopted communication technique [4].

OFDM is a digital multi-carrier transmission scheme used in IEEE 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), Long Term Evolution (LTE) Advanced, New Radio (5G NR), and various other communication systems [5]. Advantages of OFDM include high spectral efficiency, resistance to severe channel conditions, and an efficient implementation using the Fast Fourier Transform (FFT). However, OFDM has reduced temporal efficiency, high peak to average power ratio, and is highly sensitive to frequency offsets.

A method is developed to secure wireless OFDM communication links by inducing frequency offsets to the payload portion of the packet. *Carrier frequency offset*

✉ Marko Jacovic
mj355@drexel.edu

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

(*CFO*) occurs due to a mismatch between local oscillators of radios and a Doppler shift resulting from movement of the radios. OFDM signals are negatively impacted by CFO as the shifts in frequency result in a loss of orthogonality between sub-carriers, which leads to performance degradation. The frequency synchronization requirements of OFDM are leveraged to secure communications using PHY layer encryption. In the scenario in which a user *Alice* attempts to communicate securely with *Bob*, she can prevent an eavesdropper *Eve* from recovering her original message by using encryption as shown in Fig. 1. *Alice* and *Bob* share a common key, whereas *Eve*'s key differs. Such a key can be generated at the PHY layer using properties inherent to the wireless medium, as described in Section 4. Specifically, keys that are generated using frequency domain channel estimates and CFO values are considered, as measured by the synchronizer. The wireless channel between a pair of nodes is uncorrelated from the channel observed by an unintended node, assuming that the third party is separated by a distance of at least half the signal wavelength. For a carrier frequency of 2.4 GHz, the minimum distance corresponding to a half wavelength is 6.25 cm, which indicates a fair practical assumption. The CFO is also a feature that is unique to a pair of nodes, based on the RF hardware used by each system. The channel estimates and CFO are exclusive to the specific communication link. Therefore, an eavesdropper is unable to reproduce the matching key, despite having knowledge of the algorithm applied to generate keys. However, the use of other keys matching the requirements of the application are also plausible. Next, an algorithm is developed that uses the generated key along with the measured mean CFO value to induce frequency shifts during wireless communication—which are applied at different granularities to the payload of an OFDM packet. *Bob* uses the previously agreed-upon key to recover the frequency shifts and compensate for the offsets, thereby decrypting the original signal from *Alice*. *Eve*, however is not able to decrypt the original signal despite knowing all of the implemented techniques, algorithms, and system parameters used. The key that *Eve* generates does not match that of *Alice* and *Bob*, and results in *Eve* applying incorrect frequency shifts to the intercepted signal. *Eve*'s ability to recover the signal is impacted by the loss of orthogonality between sub-carriers upon receiving the sent signal, which results from the mismatch in frequency shifts calculated by *Alice* and *Eve*.

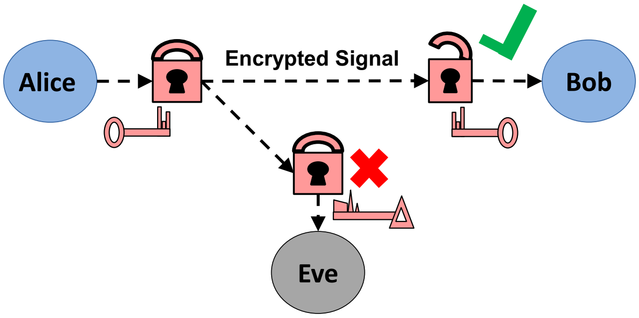


Fig. 1 Communication scenario of interest. *Alice* is the transmitter with *Bob* and *Eve* acting as receivers. *Alice* encrypts her signal, which *Bob* decrypts using the matching key, while *Eve* is unable to decrypt the signal with her key

The developed methods described in this paper are applicable to OFDM-based communication systems, including Wi-Fi, cellular, and Internet of Things (IoT) devices. Future home IoT devices are slated to use Wi-Fi Easy Connect, in which the edge devices are connected to tablets or cell phones [6]. There are possible security vulnerabilities with the Easy Connect architecture, in which a low-cost device is targeted and leveraged to compromise other components on the network. The proposed techniques are suitable for IoT devices constrained by lower processing capability.

To the best of our knowledge, this is the first technique to artificially induce frequency shifts at the PHY layer for encryption during wireless communication, which provides a method of low-complexity to secure OFDM-based wireless communication links. The rest of the paper is organized as follows. Background on OFDM transmitter and receiver design is provided in Section 2. Both the system and threat models are introduced in Section 3. A policy for key generation and management is described in Section 4. Encryption and decryption techniques as well as the hardware implementation of each are described in Sections 5 and 6, respectively. The performance of the proposed technique to secure the channel is evaluated in Section 7. Related work is discussed in Section 8, and concluding remarks are provided in Section 9.

2 OFDM System Design

The baseband components of a standard OFDM transmitter and receiver are described in this section.

2.1 Transmitter Operation

The existing modules of an OFDM transmitter and the location of the proposed encryption scheme in the pipeline is shown in Fig. 2. Standard components that currently exist in OFDM transmitters are shown in blue, while the proposed module and the corresponding inputs from higher layers of the protocol stack are in red stripes. An OFDM transmitter consists of an input bit stream, which undergoes coding when redundancy is introduced for forward error correction (FEC) and interleaving is applied to reduce the effects of burst errors. The coded or uncoded bits are passed into

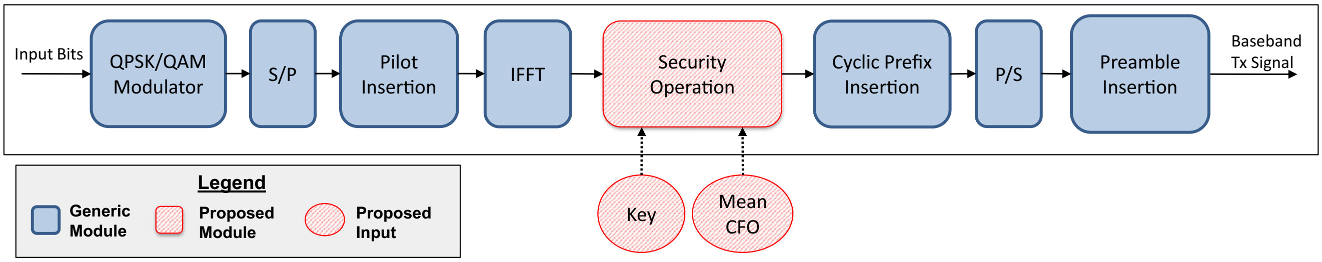


Fig. 2 System transmitter design. Proposed modifications to the OFDM pipeline are shown in red stripes

either a quadrature amplitude modulator (QAM) or a phase shift keying (PSK) modulator to produce frequency domain symbols. A serial to parallel converter is then applied to generate N streams, representing the total number of sub-carriers available. Pilot tones, which are reserved reference values used for training purposes, are then inserted into the appropriate locations. The parallel streams of frequency domain symbols undergo an N-point Inverse Fast Fourier Transform (IFFT) operation that results in time-domain values. The proposed security operation is described in Section 5 and is not a standard component of a transmitter. A cyclic prefix is appended to the time-domain parallel data by repeating a set number of values from the end of the IFFT to the beginning of the output, which reduces the effects of inter-symbol interference (ISI). In addition, the use of a cyclic prefix allows for a circular convolution of the resulting signal with the channel rather than a linear convolution. A parallel to serial converter is then used to form the payload of the OFDM signal. Training data is appended to the beginning of the signal for synchronization and equalization at the receiver. The baseband OFDM signal is converted to the analog domain and is up-converted to a carrier frequency for transmission.

2.2 Receiver Operation

The standard OFDM receiver modules and proposed inputs to the decryption scheme are shown in Fig. 3. Standard components are shown in blue, with the modified synchronizer module and higher layer inputs in red stripes. The baseband signal is extracted by down-converting and digital sampling the received signal at the carrier frequency. Timing and frequency synchronization is implemented by applying a joint algorithm that utilizes the received preamble and known reference signals. Details of the proposed decryption method and modified synchronizer are provided in Section 6. A serial-to-parallel conversion is performed after synchronization to form the same number of streams that were utilized by the transmitter, while the samples used for synchronization are removed. The samples used for the cyclic prefix are discarded prior to performing the FFT operation. The use of the FFT provides significant benefit to OFDM systems, as alternative multi-carrier demodulation techniques require the use of a large number of parallel band pass filters. Channel estimation is performed by dividing the received frequency domain training data with the original reference values known to the receiver. Under the assumption that the length of the cyclic prefix is at least one less than the length of the channel, simple channel estimation using single tap complex values per sub-carrier is possible due to the properties of circular convolution. The resulting channel estimates are utilized to equalize the remaining frequency domain payload data. Phase tracking is performed on the payload data using the embedded pilot tones as reference signals. After correction techniques are applied, the values of the pilot tones are separated from the payload symbols. A parallel-to-serial conversion is performed prior to the demodulation of the symbols to recover the bits. The implemented demodulator corresponds to the QAM or PSK modulator used by the

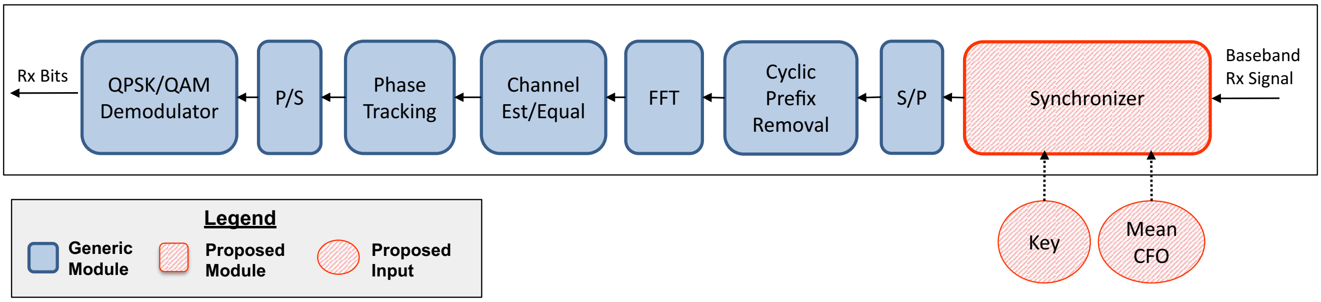


Fig. 3 System receiver design. Proposed modifications to the OFDM pipeline are shown in red stripes

transmitter. Finally, decoding and interleaving techniques are applied if forward error correction was implemented, which results in the generation of the received bit stream.

3 System and Threat Models

$Alice$ encrypts the payload portion of the OFDM signal after the IFFT operation at the transmitter but prior to cyclic prefix insertion, as shown in Fig. 2. The module implementing the security operation embeds the frequency shifts $\vec{\psi}$, shown in Fig. 4, to the time-domain payload samples. Extraction of $\vec{\psi}$ from the key and the mean CFO is discussed in detail in Section 5.1. The encryption technique is described in further detail in Section 5.2. The encrypted signal received by user j from user i is expressed as

$$y_j^{\vec{\psi}}(n) = h_{i,j}(n) * x_i^{\vec{\psi}}(n) e^{\frac{j2\pi\nu_{i,j}n}{N}} + z(n), \tag{1}$$

where $h_{i,j}(n)$ is the channel impulse response between the users, $x_i^{\vec{\psi}}(n)$ is the encrypted OFDM signal from user i, $\nu_{i,j}$ is the CFO relative to the sub-carrier spacing observed by user j, N is the number of sub-channels, and $z(n)$ is a noise component. Channel effects and CFO are experienced by the entire OFDM signal, while the encryption technique is only applied to the payload portion.

Bob and Eve utilize the receiver structure shown in Fig. 3, with the standard components in blue and the proposed blocks in red stripes. Decryption is performed by modifying the existing synchronization module to compensate for the induced frequency offsets according to the key-extracted shifts. $Alice$ and Bob possess matching keys that are unique to the key used by Eve.

Finally, the following assumptions are made regarding the eavesdropper Eve:

- Eve knows the signal structure, reference signals, as well as the encryption and decryption algorithms used by $Alice$ and Bob.
- Eve is separated from both $Alice$ and Bob by a distance of at least half of the signal wavelength. For example, at 2.4 GHz this corresponds to 6.25 cm.
- Eve does not interfere with the communication between $Alice$ and Bob or manipulate the physical environment.

4 Key Generation and Management Policy

The proposed encryption method begins with $Alice$ and Bob agreeing on a common key, with the proviso that Eve always obtains a different key, even knowing fully well the agreement algorithm implemented by $Alice$ and Bob. The focus of this work is on the development and implementation of an encryption technique once the key is established; however, a summary of a method to generate keys is provided here for completeness.

Key generation relies on physical layer properties inherent to wireless OFDM communication and utilizes an algorithm previously developed in Jacovic et al. [7]. $Alice$ and Bob exchange packets during a probing period, extract channel estimates and CFO measurements, reconcile the PHY layer features with each other, and create unique and reciprocal secret keys. The mean CFO value is calculated using packet statistics at a higher layer of the protocol stack; therefore, the calculated statistics are not considered a PHY layer operation. The randomness of the wireless channel is used to extract bits at threshold crossings of the channel estimate magnitudes for each user, while the mean CFO value is taken at different quantization levels. $Alice$ and Bob exchange time indices of the channel estimate threshold crossings and distances from the quantization boundaries of the CFO. Bits corresponding to matching time indices are retained, while the remaining are discarded. The quantized CFO is selected such that the sum of the boundary distances measured by $Alice$ and Bob is the largest. The exchange allows both $Alice$ and Bob to reconcile the extracted bit sequences and CFO estimates. The outputted key for each user is generated by performing an XOR operation of the extracted bit sequence with bits produced by a pseudorandom number generator (PRNG), where the CFO value is used as a seed. The length of the generated key is 256 bits. Eve observes packet exchanges between $Alice$ and Bob during the reconciliation stage and develops her own key using the same agreement algorithm. The channel

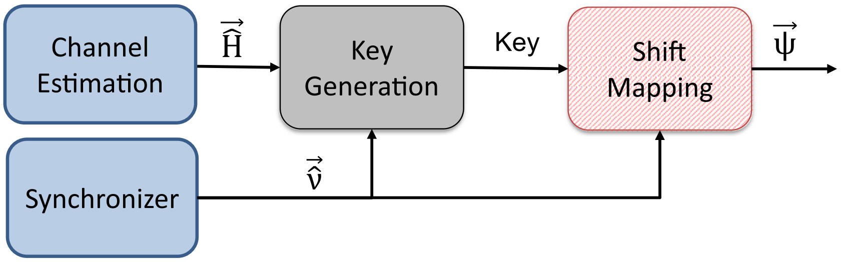


Fig. 4 Key generation and mapping to frequency shifts. The focus of this work is on the shift mapping technique. Keys generated applying different methods are compatible

properties and CFO between radio pairs are reciprocal; however, do not match exactly due to RF impairments and measurement error. The estimated channels between $Alice$ and Bob exhibit correlation with each other if the sampling time is less than the coherence time of the channel. However, Eve's channel differs from the one between $Alice$ and Bob as she is at a distance greater than half of the signal wavelength, as described in Section 3. The channel estimates that form the basis for her generated key differ, which results in a different extracted bit sequence for *Eve*.

The method to generate the key is secure based on the assumptions made in Section 3 and the properties of wireless channels. Fading is an inherent characteristic of wireless communication systems, which occurs due to multi-path propagation of signals and obstructions in the environment. The fading channels experienced by distinct receivers from a single source are statistically independent if the receiver antennas are separated by a distance corresponding to at least half of the signal wavelength [4, 8, 9]. In addition, Eve is assumed to not interfere with the probing process or change the physical surroundings between $Alice$ and Bob. The constraints prevent Eve from creating a deterministic channel between the legitimate nodes. The keys generated by $Alice$ and Bob are guaranteed to be secure from Eve due to the given assumptions. Channel estimate based key generation for security has been validated experimentally under various environments in the past [10, 11].

Once a key bit is generated, the bit is stored in a shift register that is updated periodically with new channel estimate and CFO information by applying the management protocol previously developed by Chacko et al. [12]. Periodically, contents of the shift register are transferred to a register holding the key bits used during the given transmit period, where the period is based on the number of packets transmitted or a fixed time interval. In addition, the frequency mapping algorithm requires a pointer to an 8 bit sub-key taken from the set of key bits stored in the key register. The pointer offset is based on a counter that is advanced contingent upon the shift frequency of the sub-key, as discussed in Section 5.1. Another method to extend the duration of the key is to utilize the stored value from the key register as an initial seed to a PRNG. The output sequence is then used as the extracted sub-key for the mapping algorithm.

5 Encryption at Transmitter

The encryption scheme at the transmitter is described in this section, with corresponding details of the implementation provided.

5.1 Frequency Shift Mapping

The frequency mapper shown in Fig. 5 takes as input the average CFO value, represented in IEEE 754 single-precision floating-point format [13], and alters selected bits of the CFO value using XOR translation with an 8-bit sub-key. The three least significant bits of the exponent field are each XORed with a key bit, as well as the five most significant bits of the fraction. The eight specific bits are chosen based on the influence each has on the translated CFO value, with further analysis provided in Section 7. As the fifth most significant fractional bit is used, the minimum possible offset is 1/32. An increase in the minimum offset is possible by applying more bits of the exponent rather than the fraction. The maximum offset occurs when the key is the inverse of the original 8-bit component of the average CFO value. For the 8-bits chosen for the mapping, the maximum offset is 7.96875 when including the second least significant bit to the first most significant bit of the fraction[1].

Recall that the key bits are generated via the PHY layer approach described in Section 4, and only an 8-bit portion of the key is used for the current data transmission. Updates of the sub-key are possible on a packet basis, or on a more frequent update of the fine-grained symbols. As the XOR cipher used for the security technique is vulnerable to attack due to multiple transmissions, the frequency of the update of the sub-key is important to allow for one-time pad security. Note that the adversary is assumed not able to insert a known message into the transmitter, which implies known-plaintext attacks are avoided. However, the frequent update of the sub-key and key register make a known-plaintext attack difficult to execute.

In addition, an attacker is unable to determine the frequency shifts and decode the encrypted signal through an exhaustive search. The 8-bit sub-key is used to modify the mean CFO value and produce a single frequency shift. The mean CFO observed from a window of samples changes over time, and the length of the secret key is much larger than the sub-key to facilitate multiple frequency shifts per packet.

The FPGA implementation of the proposed encryption technique consists of a frequency shift mapper and a sequence buffer to align the frequency shifts $\overrightarrow{\psi}$ with samples of the in-phase and quadrature phase components. The inputs consist of the key, the reconciled mean CFO value, and sequence duration parameters, while the output is $\overrightarrow{\psi}$. The shift mapper requires the Boolean key bitstream, the reconciled mean CFO value in 32-bit floating-point format, and a corresponding "key valid" signal. The enable

[1] The maximum offset results from the sum of the fixed point value of each bit . For the 8-bit example described: $2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} = 7.96875$.

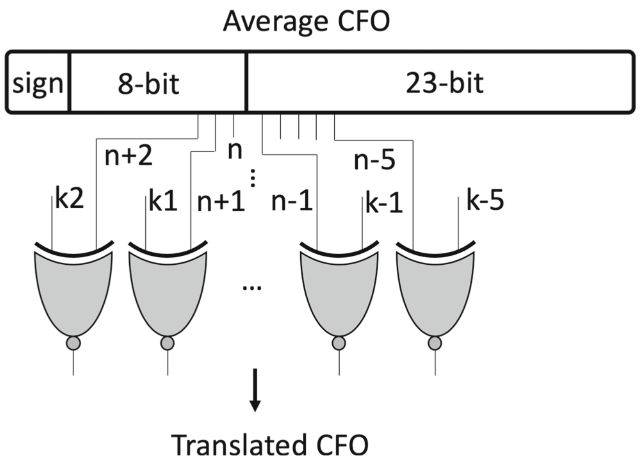


Fig. 5 Frequency mapping technique that uses the average CFO value stored in a register and translates a subset of the bits via XOR operations. Data is stored as a 32-bit floating-point value as per the IEEE 754 format [13]

signal is delayed to align with the output. Buffering of the resulting signals requires additional inputs of the Boolean enable signal corresponding to the payload samples and the sequence duration parameters. A Boolean constant is provided to select the duration of each element of $\vec{\psi}$ to either match the desired shift granularity or to use a single shift for the entire payload. Both lengths are provided as constant values and are configurable. A single port RAM is used to write and read the values of $\vec{\psi}$. The shift duration and enable signals are utilized with three 12-bit free running counters to control the operation of the RAM and provide the properly repeated and aligned shift sequence. The output of the shift mapping is a 32-bit signal $\vec{\psi}$. The payload and enable signals are delayed to align with the output.

5.2 Method of Encryption

An overview of the encryption technique implemented at the transmitter is shown in Fig. 6. The technique was implemented in the time-domain due to limitations of applying the shifts in the frequency domain. For example, integer frequency induction requires minimal complexity to implement as the operation is simply a circular shift in the frequency domain. However, the method suffers from (i) a lower possible number of shifts and (ii) the ability of an eavesdropper to determine the shifts through frequency domain cross-correlation. In addition, fractional frequency induction in the frequency domain requires upsampling, a larger FFT, scaling of the output, and complex control signals. The use of a smaller FFT is possible, but requires multiple stages that increase latency and the multiplication by complex exponentials, which defeats the purpose of operating in the frequency domain. Shift values in the frequency domain are a quantized equivalent of that achieved in the time-domain. Operating in the time-domain also permits the application of streamlined shift values. The time-domain samples $u[n]$, after the IFFT operation, are shifted in frequency according to

$$w[n] = u[n] \exp\left(\frac{j2\pi\psi_n n}{N}\right), \tag{2}$$

where n is the time index, ψ_n is the normalized frequency shift as described in Section 5.1 at n, and N is the number of sub-channels. The real and imaginary signal components from calculation of (2) are separated as follows. The encrypted signal prior to the cyclic prefix insertion is represented as

$$w[n] = w^I[n] + jw^Q[n], \tag{3}$$

where the superscripts $\{\cdot\}^I$ and $\{\cdot\}^Q$ represent the in-phase and quadrature components of a signal, respectively. After applying Euler's formula, the real component of the encrypted signal is given as

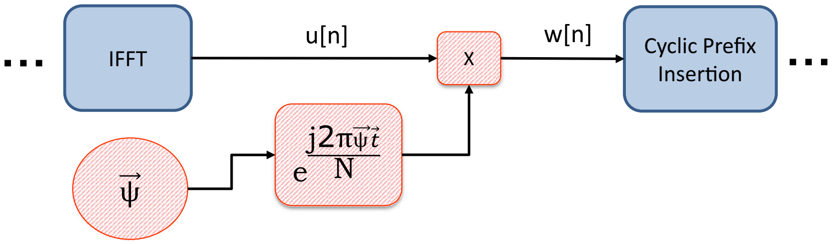
$$w^I[n] = u^I[n]\cos\left(\frac{2\pi\psi_n n}{N}\right) - u^Q[n]\sin\left(\frac{2\pi\psi_n n}{N}\right), \tag{4}$$

and the quadrature phase component as

$$w^Q[n] = u^I[n]\sin\left(\frac{2\pi\psi_n n}{N}\right) + u^Q[n]\cos\left(\frac{2\pi\psi_n n}{N}\right). \tag{5}$$

The implementation consists of an instance of the shift mapper from Section 5.1 and a time-domain frequency shifter with an operating response described by (2). The frequency shifter requires $\vec{\psi}$ from the mapper, the real and imaginary components of the payload samples with data type S16_14, which indicates a signed 16-bit signal with a binary point of 14, and the corresponding Boolean payload valid signal. The exponent is calculated using a constant multiplier and a 32-bit counter. A COordinate Rotation DIgital Computer (CORDIC) module is used to implement

Fig. 6 PHY layer encryption technique at transmitter. The proposed method is shown in red stripes



the sin and cos components of (4) and (5). The input argument to the CORDIC requires a modulo-2π operation and a mapping to $\pm\pi$ due to module constraints, which are performed using relational operators and bit slices. Four multipliers and two adders are required to implement the complex operations of $w[n]$ given by (3). The resulting real and imaginary components of the payload are of the same data type as the input and are sequentially aligned with the corresponding valid out signal.

6 Decryption at Receiver

The decryption method at the receiver is described in this section with corresponding details of the implementation provided. The frequency shift mapping implemented for the encryption method discussed in Section 5 is needed at the receiver as well. The design of the baseline synchronizer is first described to provide context for the modifications required to implement the proposed decryption methodology.

6.1 OFDM Synchronizer

A joint time and frequency OFDM synchronizer for the IEEE 802.11g physical layer was developed based on techniques described in [14] and [15]. The frequency acquisition range, normalized to the sub-carrier spacing, was limited to the fractional CFO, which is a practical choice for an indoor Wi-Fi setting. The signal bandwidth is 20 MHz with 64 sub-channels, resulting in a sub-carrier spacing of 312.5 kHz. The maximum error in frequency per local oscillator is 25 parts per million (ppm), which results in a maximum possible offset of ±120 kHz at a carrier frequency of 2.4 GHz [16]. Based on the Doppler shift of a signal given by

$$f_d = \frac{v}{\lambda}, \tag{6}$$

where v is the velocity in meters per second and λ is the wavelength of the signal, a velocity of 24.1 km/s is required for integer CFO to occur.

The stages of the synchronizer are shown in Fig. 7, in which the received signal is subject to coarse timing acquisition, fractional CFO estimation, CFO correction, and finally fine timing acquisition. The execution of coarse timing and CFO estimation are primarily based on the contributions described in [14], in which a modified correlation is applied to leverage a repetitive preamble structure. An iterative representation of the modified correlation is given as

$$\begin{aligned} P(0) &= 0, \\ P[n+1] &= P[n] + r^*[n+L]r[n+2L] - r^*[n]r[n+L], \end{aligned} \tag{7}$$

where n denotes the time index, r the received signal at a given time n, and L the lag set to the number of sub-carriers N. A detailed description of the implementation of the coarse timing acquisition is provided in Appendix A.1. The fractional CFO is calculated by analyzing the phase of the modified correlation given by (7) at the estimated coarse timing point t_{coarse} as

$$\hat{\nu} = \frac{\angle P(t_{coarse})}{2\pi}. \tag{8}$$

The estimation algorithm described in [14] consists of both fractional and integer components, which results in the merging of the two estimates. For example, if the true CFO of a signal is 3.7 sub-carrier spacings, the fractional estimate is −0.3 with integer 4 since the search space is limited to even sub-carriers. In the case that no integer CFO estimate is calculated, the fractional estimate is either −0.3 or 0.7. The actual CFO is determined as an element of

$$\overrightarrow{V} = \{\hat{\nu}, -sign(\hat{\nu})(1-|\hat{\nu}|)\}, \tag{9}$$

due to the integer estimation component of the algorithm. Two copies of the received signal are fractionally corrected, and then a cross-correlator bank is applied to compensate for the difference in the algorithm, where only the fractional component is considered. The corrected unencrypted signal is given as

$$\hat{x}_i[n] = y[n]\exp\left(\frac{-j2\pi V_i n}{N}\right), \tag{10}$$

where the subscript i denotes the unique CFO estimate, $y[n]$ is the unencrypted received signal, and n the time index. Similar to Section 5.2, the real component of the corrected signal is given as

$$\hat{x}_i^I[n] = y^I[n]\cos\left(\frac{-2\pi V_i n}{N}\right) - y_i^Q[n]\sin\left(\frac{-2\pi V_i n}{N}\right), \tag{11}$$

and the quadrature phase component as

$$\hat{x}_i^Q[n] = y^I[n]\sin\left(\frac{-2\pi V_i n}{N}\right) + y^Q[n]\cos\left(\frac{-2\pi V_i n}{N}\right). \tag{12}$$

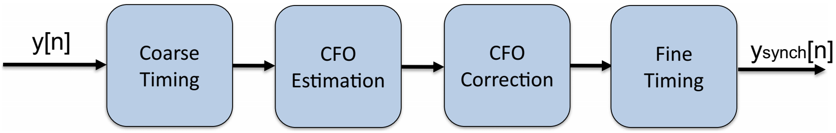


Fig. 7 Block diagram of the stages of the synchronizer

Substituting (11) and (12) into (10), the original unencrypted corrected signal is given as

$$\hat{x}_i[n] = \hat{x}_i^I[n] + j\hat{x}_i^Q[n]. \tag{13}$$

Fine timing and determination of the correct CFO estimate are performed using cross-correlation of the corrected signals with the known reference signal. Threshold crossings are used to align the signals, and the maximum values of the correlation are compared to determine the CFO from the array of possible CFO values $\overrightarrow{V}$. A detailed description of the fine timing component of the synchronizer is provided in Appendix A.2

The inputs to the OFDM synchronizer block are the real and imaginary samples of the signal each with data type S16_14 (signed 16-bit signal with binary point 14) and the timing comparison thresholds with data type U16_0 (unsigned 16-bit signal with binary point 0). The thresholds are used as addresses to the Block RAMs, which allows for the setting of fractional values. The modified correlation of $P(d)$ with data type S32_23 is utilized at the coarse timing point to estimate the CFO given by (9) using a CORDIC module that is executed 5 times. Estimates are set to S16_14 since only the fractional CFOs are considered. The input to the CORDIC used for CFO correction also requires the modulo-2π operator and a mapping to $\pm\pi$ as described in Section 5.2. Calculations of (11) and (12) are performed in parallel for both CFO estimates. A total of three CORDIC modules are required in the design of the unmodified synchronizer: one for estimation and two for correction. The corrected signals are of data type S16_14 and are inputs to the implemented cross-correlation function described in Appendix A.1. The outputs of the overall synchronization system are the in-phase and quadrature phase frequency corrected samples with data type S16_14 and the corresponding timing pulses of data type U1_0 from the correlator for packet alignment.

6.2 Method of Decryption

When decrypting the payload portion of the packet, the frequency shifts are removed. The existing synchronizer is modified to enable the decryption operation. The details on the design and implementation of the decryption technique are described in this Section.

An overview of the decryption technique at the receiver is shown in Fig. 8. The existing frequency correction module is leveraged to minimize system complexity, with the primary modification being the shift applied to the payload samples. The real component of the decrypted signal is given as

$$\hat{u}_i^I[n] = y^I[n]\cos\left(\frac{-2\pi(\zeta_n + V_i)n}{N}\right) - y^Q[n]\sin\left(\frac{-2\pi(\zeta_n + V_i)n}{N}\right), \tag{14}$$

where ζ_n is defined as

$$\begin{cases} 0 & n \le L_p \\ \psi_n & n > L_p \end{cases}$$

The normalized frequency shift ψ_n is described in Section 5.1 at time index n, where L_p is the length of the preamble. Similarly, the quadrature phase component is given by

$$\hat{u}_i^Q[n] = y^I[n]\sin\left(\frac{-2\pi(\zeta_n + V_i)n}{N}\right) + y^Q[n]\cos\left(\frac{-2\pi(\zeta_n + V_i)n}{N}\right), \tag{15}$$

which results in the decrypted signal of

$$\hat{u}_i[n] = \hat{u}_i^I[n] + j\hat{u}_i^Q[n]. \tag{16}$$

An instance of the shift mapper from Section 5.1 is used to calculate $\overrightarrow{\psi}$. A payload valid signal from the coarse timing module is utilized to apply the additional security-based frequency shift to only the payload portion of the packet. The modified synchronizer of the receiver outputs the updated normalized frequency shifts with data type S32_23 to the CFO correction module, which compensates for the inherent CFO of the packet and decrypts the payload.

7 Performance Evaluation

The PHY layer encryption technique is evaluated via both simulation and hardware assessment for the following scenarios: (i) the effect of frequency shifting on an eavesdropper without knowledge of the proposed decryption methods; (ii) the end-to-end performance of an intended and eavesdropping receiver applying the proposed security method with keys generated and managed as described in Section 4, and (iii) the performance of the System Generator models with the same scenario as described by (ii). The resource

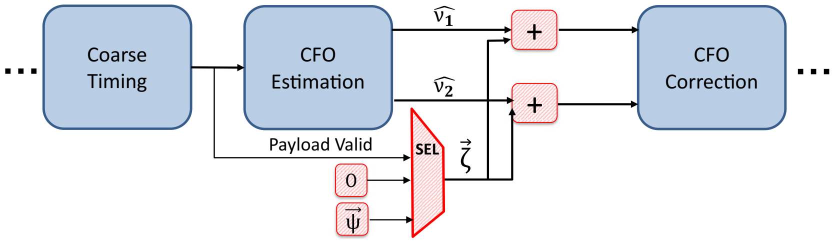


Fig. 8 Decryption modules at the receiver. The possible CFO values from (9) are estimated as $\hat{v}_1$ and $\hat{v}_2$. The proposed modifications are shown in red stripes

utilization, the timing, and the power requirements of the various implementations are also provided.

7.1 Effect of Frequency Shifting

A simulation was performed to analyze the effect of frequency shifting on system performance, independent of the key and mapping techniques applied. The system model described in Section 3 was considered with *Alice* performing as a transmitter, and *Bob* and *Eve-Blind* as receivers. *Alice* applied frequency shifts on a per-symbol basis to the payload portion of the packet. *Bob* is aware of the applied shifts and compensates for them, while *Eve-Blind*, in this scenario, has knowledge of the signal characteristics as well as the reference signals.

The threat model represented by *Eve-Blind* differs from that described in Section 3 and represented by *Eve* in that the decryption algorithm is unknown to *Eve-Blind*. A characterization of the effect of frequency shifting on the performance of an OFDM receiver is provided in this section. The analysis performed in Sections 7.2 and 7.3 considers *Eve* as the threat model.

The OFDM signals utilized 64 sub-carriers N, a cyclic prefix length C of 16, guard band G of 11 sub-carriers, 4 pilot tones per symbol, 31 OFDM symbols per packet, Quadrature Phase Shift Keying (QPSK) single-carrier modulation, a carrier frequency f_c of 2.4 GHz, and a signal bandwidth of 10 MHz. Channel equalization based on Minimum Mean Square Error (MMSE) is applied at each receiver with an ideal noise estimate. A summary of the OFDM parameters used by *Alice*, *Bob*, and *Eve-Blind* is provided in Table 1. Frequency shifts were randomly selected according to a uniform distribution bounded by a maximum shift of ψ_{max}. The received signals were assumed to experience additive white Gaussian noise (AWGN) with a varying energy per bit to noise power spectral density ratio (E_b/N_0). CFO was applied to the signals according to a uniform distribution bounded by the signal parameters and a maximum local oscillator offset of 20 ppm. The bit error rate (BER) for *Bob* and *Eve-Blind* was calculated under repeated trials. The ψ_{max} parameter was swept from 0.5 to 5 sub-carrier spacings as shown in Fig. 9. *Bob*'s BER is superior to *Eve-Blind* in all cases as expected, while *Eve-Blind*'s BER increases with ψ_{max}. The effect of ψ_{max} on the BER for E_b/N_0 values of 5 and 10 dB is characterized, with results as shown in Fig. 10. Based on the results, selecting a ψ_{max} greater than or equal to 2 produces the theoretical worst BER of 0.5 for *Eve-Blind*. The effect of the number of shifts per OFDM symbol is characterized with results as shown in Fig. 11. A comparison between 1, 2, and 4 shifts per OFDM symbol are shown for ψ_{max} values of 0.25 and 0.5. A BER of 0.5 was observed for *Eve-Blind* with 4 shifts per symbol, which indicates that even for small frequency shifts in the symbols the desired effect on the BER is achieved by increasing the number of total shifts per symbol.

Table 1 Parameters of the OFDM signal utilized in the study

N	64	Modulation	QPSK
C	16	f_C	2.4 GHz
G	11	Bandwidth	10 MHz
No. Pilots	4	Equalizer	MMSE
No. Symbols	31	Noise Estimate	Ideal

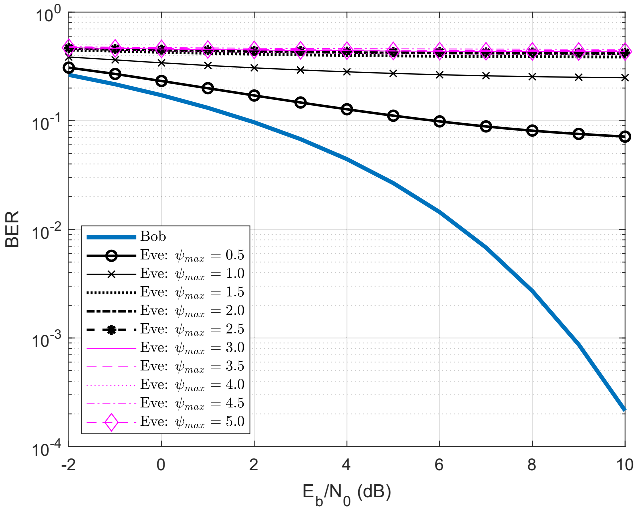


Fig. 9 BER performance of *Eve-Blind* as a function of the parameters ψ_{max} and E_b/N_0

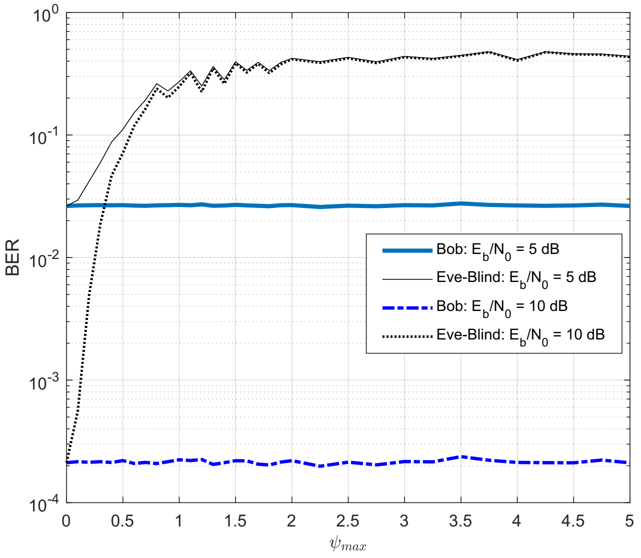


Fig. 10 BER performance of *Eve-Blind* when varying ψ_{max}. The BER is characterized for E_b/N_0 ratios of 5 dB and 10 dB

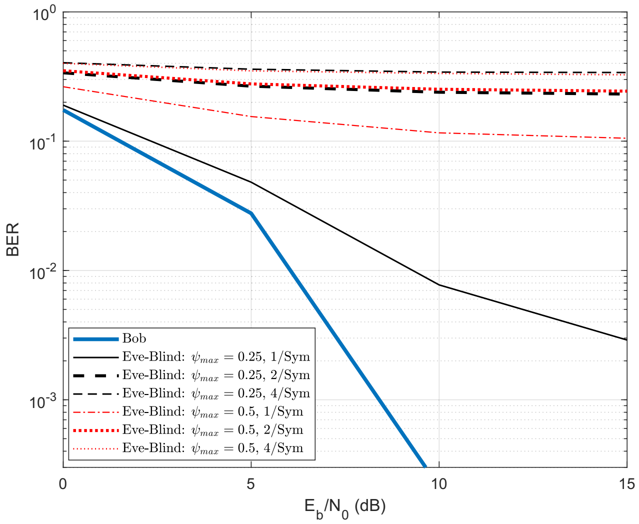


Fig. 11 BER performance as a function of the number of shifts per OFDM symbol

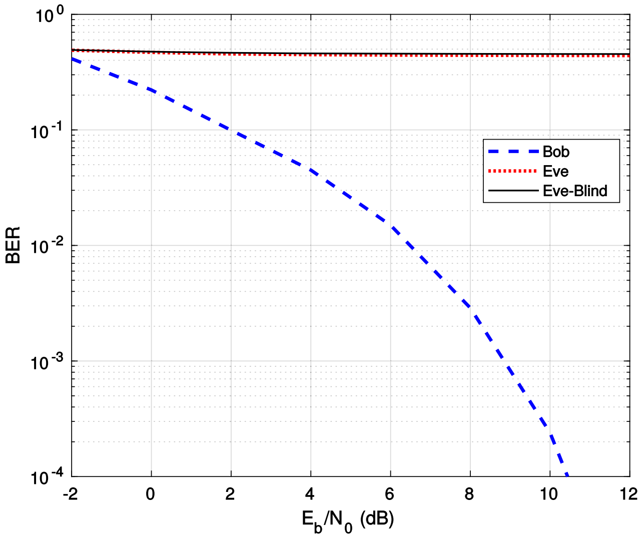


Fig. 12 BER performance when applying the key mapping algorithm that utilizes experimentally collected keys

7.2 Security Performance

The simulation framework from Section 7.1 is extended to include the developed key mapping based encryption and decryption techniques described in Section 5 and Section 6, respectively. The keys generated for *Alice*, *Bob*, and *Eve* were obtained by implementing the technique described in [7], as discussed in Section 4. The channel estimates and CFO were experimentally measured with Ettus USRP N210 radios using the software defined radio network testbed described in [17]. The threat model described in Section 3 is assumed, in which *Eve* has full knowledge of system parameters and applies the identical decryption technique as *Bob*. Shifts were updated at a per OFDM symbol basis, with ψ_{max} set to 7.96875 sub-carrier spacings based on the frequency shift mapping described in Section 5.1.

A comparison between the BER of *Bob*, *Eve*, and *Eve-Blind*, in which *Eve-Blind* does not utilize the decryption technique, is shown in Fig. 12. The performance of *Bob* is not impacted by the implementation of the proposed security technique, while both *Eve* and *Eve-Blind* suffer a BER of 0.5 for all values of E_b/N_0. Despite *Eve* knowing the applied reference signals, packet structure, and decryption algorithm, she is unable to recover the message due to the difference in the measured mean CFO, which results in an incorrect extracted key. *Eve* uses the identical algorithms to generate the keys as *Alice* and *Bob*. However, although *Eve* implements the decryption method applied by *Bob*, the differences in key value and mean CFO result in deviations in the calculated frequency shifts from the values used by *Alice* during encryption. The mean CFO is measured over an interval of received packets and is not constant over time. Therefore, determining the base CFO utilized in Section 5.1 is an additional challenge *Eve* faces. Instead of removing the frequency offsets, *Eve* further corrupts the received signal by applying an incorrect series of shifts. *Eve-Blind* in contrast does not have knowledge of or apply the decryption method and, therefore, processes the received signal under normal operation. The identically high BER shown for *Eve* and *Eve-Blind* demonstrates that the developed encryption technique provides a suitable method to secure communication links.

7.3 Analysis of the Hardware Implementation

The performance of the transceiver system implementing the proposed hardware encryption and decryption modules is verified using a combination of MATLAB scripts and the System Generator (sysgen) simulation environment. Signal parameters listed in Table 1 were selected with the exception of using Least Squares channel estimation instead of MMSE. The keys and measured CFO values for *Bob* and *Eve* were the same as those used in the performance study described in Section 7.2, and frequency shifts were also updated at each OFDM symbol with ψ_{max} set to 7.96875. OFDM payload samples were generated in script and provided as inputs to the encryption module in the sysgen environment. The model uses the key, CFO value, and signals described in Section 5.2 to apply the security technique based on frequency shifting to the payload samples.

Execution of the script results in the insertion of the cyclic prefix to the output signal of the sysgen model, which occurs prior to parallel to serial conversion and appending of the preamble. AWGN and CFO effects were added to the transmit signal prior to being provided as input signals to the receiver. Two modified synchronization sysgen modules were implemented, each consisting of the synchronizer from Section 6.1 and the decryption block described in Section 6.2. The module instances represent *Bob* and *Eve*, with each providing their corresponding key and CFO measurements as system inputs. Each sysgen module performs coarse timing, CFO estimation, CFO correction, and decryption according to the given generated key. The cross-correlation function of the synchronizers was bypassed, and perfect timing synchronization was used in the script to isolate the effects of the encryption technique for characterization. The remaining receiver components shown in Fig. 3 were implemented in the script to extract the OFDM payload. Error vector magnitude (EVM) was calculated for *Bob* and *Eve* by comparing the corrected received QPSK symbols to the original symbols transmitted by *Alice*. In addition, the case of no implemented security features is considered as the benchmark configuration with results as shown in Fig. 13. The results from the hardware demonstration indicate that *Bob* is able to decrypt the signal without any performance degradation as compared to the case that features no added security, while *Eve* is unable to recover the message despite having full knowledge of the signal parameters and execution of the same decryption algorithm.

The hardware is implemented such that the number of components required is minimized as FPGA resources are limited, while also assuming that resource efficient modules are more readily integrated within larger circuit topologies. The encryption module, baseline synchronizer, and decryption-modified synchronizer are synthesized on a Xilinx Virtex-7 FPGA VC707 Evaluation board to characterize the additional resources utilized for each component. The Xilinx Virtex-7 series is a high-end FPGA option suitable for both Wi-Fi and cellular prototyping, and acts as a gateway to a heterogeneous IoT sensor network as discussed in [18]. The proposed techniques can also be implemented on lower-end FPGA boards such as the Xilinx Artix-7. The primary limiting resource on the FPGA boards is the number of available multipliers; while the proposed encryption/decryption technique utilizes substantially less than the 740 multipliers available on the Artix-7 AC701 evaluation board.

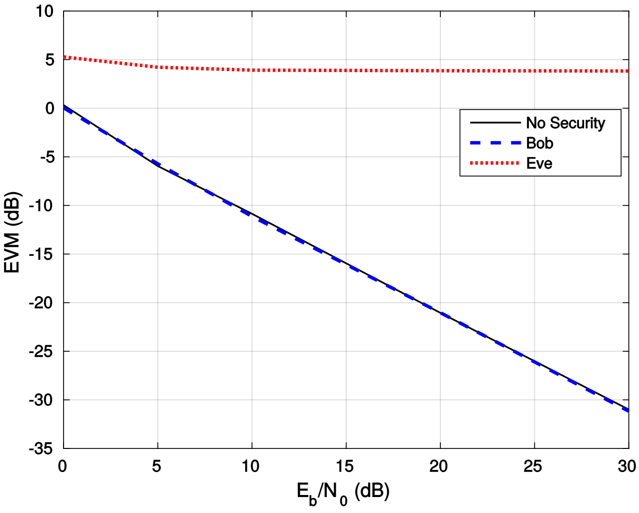


Fig. 13 Comparison of the error-vector magnitudes achieved using the sysgen environment for *Bob*, *Eve*, and no security

Much of the complexity of the security technique is in the implementation of the transmitter. The resources required for the encryption module discussed in Section 5.2 are summarized in Table 2. Look-up Tables (LUTs) and Flip-flops (FFs) consume 1.65% and 0.8% of the total FPGA resources, respectively. Four Block Random Access Memory (BRAM) units are implemented to buffer the shift sequences, and ten multipliers are used to apply the modulo-2π scaling and to implement the complex function given by (3). The BRAM and multipliers occupy about 0.4% of the resources of the evaluation board. The FPGA resources required to implement the baseline synchronizer, which consists of the coarse timing unit, CFO estimation unit, CFO correction unit, and cross-correlation modules as described in Section 6.1, are listed in Table 3. The synchronizer consists of a greater number of circuit blocks since it is a more complex component. The results provide a benchmark to compare against after implementing the decryption module. The unit count and the percentage increase in the utilization of each resource (LUTs, FFs, BRAMs, and Mult/DSP48) as compared to the baseline synchronizer module are listed in Table 4. The number of LUTs and FFs is significantly less than that required by the encryption module, whereas the number of BRAMs remains the same as when an identical mapping module is used. Most notably, there is no increase in the number of multipliers since the decryption module reuses components of the synchronizer. As the encryption module is the primary source of added complexity, a Xilinx Artix-7 AC701 FPGA board was utilized to implement the module. The resource utilization report of the encryption module is listed in Table 5, where the resource count and the increase in the percent utilization over the baseline transmitter are provided.

The timing and power requirements of the encryption and decryption modules were analyzed through implementation on a Xilinx Artix-7 AC701 FPGA board. A 100 MHz system clock was set to match the Ettus USRP N210 SDR[19]. The power and timing requirements are summarized in Table 6. The encryption module results in a 0.39 µs timing overhead attributed to the CORDIC and multiplication operations described in Section 5.2. A total on-chip power of 142 mW was consumed by the encryption module, which consists of 22 mW of dynamic and 120 mW of static power. A baseline model that consists of delaying input signals by one clock cycle was considered to determine the power consumption attributed to the selected FPGA board. The simple module also consumed a static power of 120 mW. As a result, only the dynamic power was determined to contribute to the total power consumption from the proposed security modules. Therefore, the power consumption attributed to the encryption technique is 22 mW. The overhead in power is primarily a result of the resources utilized to implement the BRAM and CORDIC. The decryption module does not result in a timing overhead, and requires 11 mW of dynamic power for the FPGA resources utilized to implement the BRAM.

IoT communication consists predominately of Low Power Wide Area Network (LPWAN) protocols, including OFDM-based cellular systems, and short range networks related to ZigBee [20]. OFDM-based short range IoT that leverage Wi-Fi systems have been developed to provide increased data rates, support non-line-of-sight communication, and enhance area coverage [21]. Wi-Fi systems require frame reception and response transmission to occur within the time duration defined by the Short Interframe Space (SIFS) criterion of the IEEE 802.11 standard [22]. The 0.39 μs timing overhead listed in Table 6 for encryption accounts for only 3.9% of the 10 μs SIFS described by IEEE 802.11n, and does not hinder the ability of the communication protocol to meet standard requirements. An empirical characterization of the power consumption of an IEEE 802.11n network adapter was performed for different configurations of the transmitter and the receiver in [23]. The single input multiple output (SIMO) transmitter was determined to use 1.28 W of power when active, while the single input single output (SISO) receiver consumed 0.94 W when active. Based on the results listed in Table 6, the encryption technique results in a 1.72% increase in power consumption relative to IEEE 802.11n, while decryption results in an increase of 1.17%. The power consumption of both the SIMO transmitter and SISO receiver is analyzed for active communication. Wi-Fi data is typically sent in bursts, which allows for long periods of inactivity to conserve power. As the encryption and decryption blocks do not contribute to the static power consumption, the increase in the overall power consumption for both modules is acceptable.

Table 2 Estimated resource utilization of the encryption module to be integrated into the transmitter. The module is implemented on the Xilinx Virtex-7 FPGA VC707 Board

Resource name	Count	Utilization
FF’s	4878	0.8
BRAM’s	4	0.39
Mult/DSP48	10	0.36

Table 3 Estimated resource utilization of the unmodified synchronizer implemented on the Xilinx Virtex-7 FPGA VC707 Board

Resource name	Count	Utilization
LUT’s	24176	7.96
FF’s	20397	3.36
BRAM’s	2	0.15
Mult/DSP48	44	1.57

The low overhead in FPGA resource utilization, timing, and power requirements as shown by the results listed in Tables 5 and 6 indicate that the proposed encryption techniques are applicable to low-power wireless systems. More importantly, the benefit of applying the security technique outweighs the additional requirements of both the transmitter and receiver.

8 Related Work

Securing signals for an authenticated receiver by making the transmitted signals indiscernable to an attacker is an essential area of focus for wireless communication systems, particularly for low-cost devices. A variety of related work on PHY layer security has been previously reported as existing radio components are leveraged to provide an additional layer of security. Time-domain scrambling was proposed in [24], which is performed after the Inverse Fast Fourier Transform (IFFT) and before cyclic prefix insertion in the OFDM pipeline. While additional security was provided, a complex transformation of the frequency domain constellation is required at the receiver. In addition, the work relied on a pre-shared key to ensure encrypted operation between two parties sharing a communication channel. In contrast, the proposed technique is capable of using either pre-shared keys or keys generated based on PHY layer properties as described in Section 4. Phase rotation and induced noise were applied at the transmitter to secure communication signals in [25]. However, the transmission pilot tones were unaltered throughout operation of

Table 4 Overhead incurred by the synchronizer as a result of implementing the frequency shifting based decryption technique as compared with the baseline

Resource name	Count increase	Utilization increase
LUT’s	644	0.22
FF’s	652	0.11
BRAM’s	4	0.38
Mult/DSP48	0	0

Table 5 Estimated resource utilization of the encryption module to be integrated into the transmitter. The module is implemented on the Xilinx Artix-7 AC701 FPGA Board

Resource name	Count	Utilization
LUT's	4994	3.73
FF's	4878	1.82
BRAM's	4	1.10
Mult/DSP48	10	1.35

the transceiver, which leaves the pilot tones vulnerable to an attacker. In addition, noise induction is an energy inefficient technique and impedes system performance. Pilot symbol manipulation was implemented in [26] by shifting the phase of certain sub-carriers based on a measured amplitude or phase threshold. Through simulation, the amplitude based approach reduced the successful execution of an eavesdropping attack while maintaining an acceptable level of performance. Both methods require perfect channel knowledge and consider reference signals of the preamble without discussion of pilot tone manipulation from within the payload. An attacker synchronized to the victim transceiver is able to recover the signal using pilot tone based channel estimation techniques with interpolation methods. Additional approaches to manipulate pilot symbols based on channel characteristics were considered in [27]. A measured threshold for channel gain was applied to generate pilot symbols that represented whether a sub-carrier contained legitimate information or dummy data throughout the transmission of the signal. The method also assumed perfect channel knowledge at the transmitter, did not consider pilot tones within the payload of the signal, and required a large overhead in transmission bandwidth as an equal number of data bearing and dummy sub-carriers were used.

The interleaver stage of the OFDM pipeline was modified to secure communication channels by using keys to alter the permutation sequence in [12]. The modified pipeline was implementated on a Virtex-6 FPGA, which resulted in less than a 1% increase in the resources utilized and a minor loss in performance as compared with the standard IEEE 802.11a interleaver. Exclusive OR (XOR) based encryption was applied to the preamble portion of an OFDM packet to limit the cross-correlation timing capabilities of an eavesdropper in [28]. The technique does not protect the payload of a packet in the event that an adversary is capable of detecting the signal energy and synchronize in time by applying auto-correlation methods.

Table 6 Estimated timing and power requirements of the encryption and decryption modules implemented on the Xilinx Artix-7 AC701 FPGA Board.

Module type	Timing (μs)	Power (mW)
Encryption	0.39	22
Decryption	0	11

In this paper, the frequency synchronization requirements of an OFDM signal are leveraged instead of the timing requirements. The CFO has previously been explored for authentication. The CFO measured between devices was demonstrated as a suitable property for authentication of communication links in [29], while second-order statistics were used to distinguish Wi-Fi compliant devices in [30]. Authentication using CFO was shown feasible for highly mobile communication channels in [31]. CFO was induced on a per-frame basis to allow for blind authentication by a regulatory device in [32]. The blind receiver required the implementation of highly complex techniques, which are not feasible in most communication systems based on OFDM. In addition, the CFO induction was applied only on a per-frame basis, in contrast to the proposed PHY layer encryption system that permits greater granularity in frequency shifting. CFO was previously leveraged by combining measurements with channel estimates for the purpose of PHY layer key generation in [7]. The keys are used as inputs to the system described in this paper, as the objective is on securing a communication link after a key has been established. A PHY layer encryption technique and the corresponding hardware components are described that induce frequency offsets to the payload portion of a packet to secure signals for wireless communication.

9 Conclusion

The wide adoption of wireless devices and the vulnerabilities of WPA2 and WPA3 have motivated additional research in developing low-cost security solutions. A PHY layer encryption method is described for OFDM signals based on frequency induction that complements existing security techniques. Keys are used to map frequency shifts to the payload of a packet, with extensive details on each implemented system component provided. Simulation and hardware results demonstrate that the proposed encryption technique is capable of preventing an eavesdropper from recovering the original message, while maintaining the desired performance of an intended receiver. The implementation of the techniques on a Xilinx Virtex-7 FPGA VC707 board resulted in an increase of 1.65% in the number of LUTs, 0.8% in the number of FFs, and less than 0.4% in the number of BRAM and multipliers required at the transmitter. Implementation of the decryption technique at the receiver required minimal additional FPGA resources, with the utilized LUTs, FFs, BRAM, and multipliers each increasing by less than 0.4%.

The techniques were implemented on a low-power Xilinx Artix-7 AC701 FPGA Board to validate suitability for OFDM-based applications. An increase of 3.73% in the number of LUTs, 1.82% in the number of FFs, 1.10% in the number of BRAMs, and 1.35% in the number of multipliers is required at the transmitter. Both the encryption and decryption modules were analyzed for timing and power requirements, and were compared with IEEE 802.11n standard systems. The encryption method produces a delay corresponding to 3.9% of the SIFS constraint, while the decryption technique does not result in a timing overhead. The power consumption of the transmitter and receiver increases by 1.72% and 1.17%, respectively.

The increase in system requirements is a minor trade-off for the added security provided. The technique is, therefore, suitable for a wide range of applications that require wireless communication.

Funding Information This research was supported by the National Science Foundation Grant No. CNS-1816387, the Department of Education Graduate Assistance in Areas of National Need (GAANN) program under award P200A180082, and the Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

Appendix

Specific details regarding the coarse timing (Section A.1) and fine timing (Section A.2) implementation of the OFDM synchronizer described in Section 6.1 are provided.

A.1 Coarse Timing

The coarse timing component of the synchronizer utilizes the received signal samples to determine the coarse start time of the packet and the iterative calculation of the modified correlation given by (7). The outputs of the module are provided to the CFO estimation block. The real and imaginary components of (7) are given as

$$Re\{P[n+1]\} = Re\{P[n]\} + r^I[n+L]r^I[n+2L] - r^I[n]r^I[n+L] \\ - r^Q[n]r^Q[n+L] + r^Q[n+L]r^Q[n+2L], and \tag{17}$$

$$Im\{P[n+1]\} = Im\{P[n]\} + r^I[n+L]r^Q[n+2L] \\ - r^I[n+2L]r^Q[n+L] - r^I[n]r^Q[n+L] + r^I[n+L]r^Q[n], \tag{18}$$

respectively, where the superscript $\{\cdot\}^I$ denotes the in-phase component of the received signal and $\{\cdot\}^Q$ represents the quadrature phase. The real-valued calculation of the signal energy is derived as

$$R[n+1] = R[n] + |(r^I[n])^2 + (r^Q[n])^2 - \\ (r^I[n+2L])^2 - (r^Q[n+2L])^2|^2. \tag{19}$$

A filtered timing metric is calculated as

$$M_{LPF}[n] = \frac{1}{C}\sum_{k=0}^{C-1} \frac{|P^I[n-k] + P^Q[n-k]|^2}{|R[n-k]|^2}, \tag{20}$$

where the window is set to the cyclic prefix length C. The strength of $P[n]$ varies and, therefore, requires normalization by $R[n]$, with a result as given by $M_{LPF}[n]$, which is used to set a hard threshold. The coarse timing point is selected at the maximum of the $M_{LPF}[n]$ metric. The inputs to the implemented unmodified synchronizer are the real and imaginary signal samples each with data type S16_14 and a coarse timing comparison threshold of data type U16_0. The resulting threshold addresses a Block RAM, which allows for the setting of fractional values. The calculations of (17), (18), and (19) are performed in parallel to reduce latency, but require 12 multipliers to implement the complex operations. The timing metric given by (20) is calculated using a divider block after bit-shifting the auto-correlation values to meet input constraints, with the resulting output being of data type U16_14. Low pass filtering is implemented with parallel addressable shift registers, cascaded addition, and a constant multiplier. Relational operators are used for threshold crossing throughout the implementation of the synchronizer. The number of time samples above the timing metric threshold is calculated using a counter, with crossings determined by rise and fall edge detection. The output of the counter is used to address shift registers for proper signal alignment in time.

A.2 Fine Timing

Fine timing is completed using the output samples of the CFO correction block and a threshold value, with the output being the corrected samples and a corresponding alignment signal. Fine timing and determination of the correct CFO estimate are performed by calculating the cross-correlation given by

$$F_i[n] = \sum_m \hat{x}_i[m+n]S^*[m], \tag{21}$$

where $S[m]$ is the reference signal provided to the receiver. The separate real and imaginary components of the cross-correlation are given as

$$Re\{F_i[n]\} = F_i^{II}[n] + F_i^{QQ}[n], \text{ and} \tag{22}$$

$$Im\{F_i[n]\} = F_i^{QI}[n] - F_i^{IQ}[n], \tag{23}$$

respectively, where the superscript pair corresponds to the component of the corrected signal and the component of the reference signal used in individual cross-correlation calculations. The absolute value of the F_i signals are compared with a hard threshold to detect the start of the packet. Due to the structure of the preamble, there are ideally two events at which a crossing occurs, spaced by N samples. The first crossing corresponds to the start of the packet. The distance between multiple threshold events is used to filter incorrect estimates. The maximum values of F_1 and F_2 are compared to determine which CFO estimate is correct. The incorrectly adjusted signal results in a non-coherent correlation and yields low values.

The real and imaginary components of the CFO corrected signals each with data type S16_14 and the fine timing comparison threshold with data type U16_0 are provided as inputs to the implementation of the unmodified synchronizer on the FPGA. Multiplications are avoided by quantizing the signals to ±1 and using conditional sign inversion with multiplexers. The reference signals are extracted using bit slices of data type U32_0 for decimal equivalent constants of the binary sequence. Cascaded addition is used to improve computational efficiency, and relational operators are used to compare the computed correlations with the hard threshold. The CFO corrected signals given by (10) are delayed to align with the calculated correlations to minimize loss in precision. The outputs of the unmodified synchronizer for packet alignment are the frequency corrected in-phase and quadrature phase samples with data type S16_14 and the corresponding timing pulses of data type U1_0 from the correlator.

References

1. Fehér DJ, Sandor B (2018) Effects of the WPA2 KRACK attack in real environment. In: Proceedings of the IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY), pp 239–242
2. Cimpanu C Dragonblood Vulnerabilities Disclosed in WiFi WPA3 Standard. https://www.zdnet.com/article/dragonblood-vulnerabilities-disclosed-in-wifi-wpa3-standard/
3. Mukherjee A (2015) Physical-layer security in the Internet of Things: sensing and communication confidentiality under resource constraints. Proc IEEE 103(10):1747–1761
4. Proakis JG, Salehi M (2007) Digital communications, 5th edn. McGraw-Hill, Boston Mass
5. Li YG (2009) Orthogonal frequency division multiplexing for wireless communications. Springer, Berlin
6. Wi-Fi Alliance, Discover Wi-Fi https://www.wi-fi.org/discover-wi-fi
7. Jacovic M, Kraus M, Mainland G, Dandekar K (2019) Evaluation of physical layer secret key generation for IoT devices. In: Proceedings of the IEEE 20th Wireless and Microwave Technology Conference (WAMICON), pp 1–6
8. Molisch A (2005) Wireless communications, 2nd edn. John Wiley & Sons Ltd, New York
9. Da-Shan Shiu G, Foschini J, Gans MJ, Kahn JM (2000) Fading correlation and its effect on the capacity of multielement antenna systems. IEEE Trans Commun 48(3):502–513
10. 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
11. 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
12. Chacko J, Juretus K, Jacovic M, Sahin C, Kandasamy N, Savidis I, Dandekar KR (2019) Securing wireless communication via hardware-based packet obfuscation. Journal of Hardware and Systems Security, [Online]. Available: https://doi.org/10.1007/s41635-019-00070-0
13. IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2008, 1–70, 2008
14. Schmidl TM, Cox DC (1997) Robust frequency and timing synchronization for OFDM. IEEE Trans Commun 45(12):1613–1621
15. Minn H, Bhargava VK, Letaief KB (2003) A robust timing and frequency synchronization for OFDM, Systems. IEEE trans Wireless Commun 2(4):822–839
16. IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part Ii: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001), pp. i–67, 2003.
17. Dandekar K, Begashaw S, Jacovic M, Lackpour A, Rasheed I, Rivas Rey X, Sahin C, Shaher S, Mainland G (2019) Grid software defined radio network testbed for hybrid measurement and emulation. In: Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON) (SECON 2019). USA, Boston
18. de la Piedra A, Braeken A, Touhafi A (2012) Sensor systems based on FPGAs and their applications: a survey. Sensors (Basel, Switzerland) 12:12235–64, 12
19. National Instruments, Ettus Research, https://www.ettus.com/
20. Al-Sarawi S, Anbar M, Alieyan K, Alzubaidi M (2017) Internet of Things (IoT) communication protocols: review. In: Proceedings of the IEEE International Conference on Information Technology (ICIT), pp 685–690
21. Li L, Xiaoguang H, Ke C, Ketai H (2011) The Applications of WiFi-based wireless sensor network in Internet of Things and Smart Grid. In: Proceedings of the IEEE Conference on Industrial Electronics and Applications, pp 789–793
22. IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput, IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009), pp. 1–565, Oct 2009.
23. Halperin D, Greenstein B, Sheth A, Wetherall D (2010) Demystifying 802.11n power consumption, In Proceedings of the 2010 International Conference on Power Aware Computing and Systems, ser. HotPower'10USA: USENIX Association, 1
24. Li H, Wang X, Hou W (2013) Secure transmission in OFDM systems by using time domain scrambling. In: Proceedings of the IEEE 77th Vehicular Technology Conference (VTC Spring), pp 1–5
25. Reilly D, Kanter GS (2009) Noise-enhanced encryption for physical layer security in an OFDM radio. In: Proceedings of the IEEE Radio and Wireless Symposium, pp 344–347

26. Soltani M, Baykaş T, Arslan H (2015) Achieving secure communication through pilot manipulation. In: Personal Indoor, and Mobile Radio Communications (PIMRC,), 2015 IEEE 26th Annual International Symposium on, pp 527–531
27. Umebayashi K, nakabayashi F, Suzuki Y (2014) A study on secure pilot signal design for OFDM systems. In: Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, pp 1–5
28. 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 InternationalConference on Computing, Networking and Communications, pp 293–297
29. Wheeler CG, Reising DR (2017) Assessment of the impact of CFO on RF - DNA fingerprint classification performance, pp 110–114
30. Vo-Huu TD, Vo-Huu TD, Noubir G (2016) Fingerprinting Wi-Fi devices using software defined radios. In: Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks, ser. WiSec ’16. USA:ACM, pp 3–14
31. Hou W, Wang X, Chouinard J, Refaey A (2014) Physical layer authentication for mobile systems with time-varying carrier frequency offsets. IEEE Trans Commun 62(5):1658–1667
32. Kumar V, Park JM, Bian K (2014) Blind transmitter authentication for spectrum security and enforcement. In:Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS ’14. USA:ACM [Online]. Available: https://doi.org/10.1145/2660267.2660318