Evaluation of Physical Layer Secret Key Generation for IoT Devices

Marko Jacovic*, Martin Kraus*, Geoffrey Mainland†, and Kapil R. Dandekar*
*College of Engineering, †College of Computing and Informatics,
Drexel University, Philadelphia, PA 19104, USA
Email: {mj355, mpk58, mainland, dandekar}@drexel.edu

***Abstract*—As aspects of our daily lives become more interconnected with the emergence of the Internet of Things (IoT), it is imperative that our devices are reliable and secure from threats. Vulnerabilities of Wi-Fi Protected Access (WPA/WPA2) have been exposed in the past, motivating the use of multiple security techniques, even with the release of WPA3. Physical layer security leverages existing components of communication systems to enable methods of protecting devices that are well-suited for IoT applications. In this work, we provide a low-complexity technique for generating secret keys at the Physical layer to enable improved IoT security. We leverage the existing carrier frequency offset (CFO) and channel estimation components of Orthogonal Frequency Division Multiplexing (OFDM) receivers for an efficient approach. The key generation algorithm we propose focuses on the unique CFO and channel experienced between a pair of desired nodes, and to the best of our understanding, the combination of the features has not been examined previously for the purpose of secret key generation. Our techniques are appropriate for IoT devices, as they do not require extensive processing capabilities and are based on second order statistics. We obtain experimental results using USRP N210 software defined radios and analyze the performance of our methods in post-processing. Our techniques improve the capability of desired nodes to establish matching secret keys, while hindering the threat of an eavesdropper, and are useful for protecting future IoT devices.**

***Index Terms*—Internet of Things, Data Security, OFDM, Physical Layer, Wireless communication**

I. Introduction

Emerging technologies such as 5G cellular, future local area networking, and the Internet of Things (IoT) must address the challenges related to supporting high density heterogeneous systems. The most constrained of these high density systems are IoT networks, which trade off certain system capabilities to reduce energy. For example, to conserve energy, these devices often have limited signal processing, limited storage/memory, and compact form factors. The number of devices that need to be supported continues to grow as different applications are developed to improve our quality of life. IoT has numerous consumer, enterprise, and commercial uses: health tracking watches, modern pacemakers, smart home and office systems, traffic or weather sensors, and smart utility meters are only a few examples of every-day usage. In each of the aforementioned scenarios, it is vital that current and future wireless communication systems are reliable and well protected [1]. Wi-Fi Protected Access (WPA/WPA2) has been shown to be vulnerable to attacks in the past [2]. As a result, the development of WPA3 has focused on reducing brute force attacks. Future home IoT devices will be designed to use Wi-Fi Easy Connect, a method of connecting systems without GUI capabilities to hand-held devices such as phones or tablets [3]. A malicious individual may be able to target a low-cost device and leverage the connective architecture to compromise the entire network, motivating the necessity for additional security options.

Physical layer security is a promising research avenue to secure future wireless technologies while adhering to processing and computational constraints of low cost IoT devices. The techniques may be used to strengthen the overall security capabilities of the wireless system, by complementing methods used at other levels of the protocol stack. The use of channel reciprocity to develop secret keys for the purpose of authentication has been studied extensively in the literature [4]–[7]. The physical (PHY) layer provides a significant opportunity for additional security due to the inherent random variations in the wireless medium. The unique channels experienced between nodes and channel reciprocity of links allow for the generation of unique keys. The high level of randomness and variability is achieved due to the wireless propagation channel between two nodes.

Multiple methods of utilizing randomness in the wireless channel have been proposed to generate secret keys from the PHY layer. A framework for secret key generation over an unauthenticated wireless channel was presented in [4]. The authors use a probing period in which packets are sent between intended nodes to measure channel characteristics, considering both the dominant component of the channel impulse response (CIR) and the received signal strength indicator (RSSI) values. The measurements are filtered, then quantized by comparing the results to positive and negative thresholds (representing bits 1 and 0) relative to the mean value. A window is used such that the threshold must be exceeded for a minimum duration in time. Assuming that the probing is performed at a high enough rate, the measurements should exhibit high similarity at both end points even in half-duplex mode, as confirmed by the real-time implementation described in [8]. The time indices at which the criteria are met are reported between both nodes on a shared communication channel, with the values that do not overlap being dropped. Despite the information being publicly sent, an adversary would not be able to determine the key since it would only have knowledge of the time

978-1-5386-9597-5/19/$31.00 ©2019 IEEE

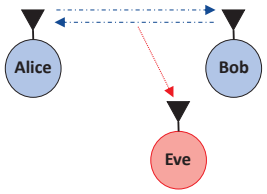

Fig. 1. **System Model.** Alice and Bob exchange packets to independently develop matching secret keys for secure communication, while Eve passively listens to the signals being sent and attempts to develop her own key.

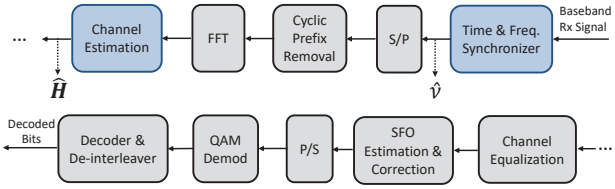

Fig. 2. **OFDM Receiver.** Generic pipeline used by OFDM systems; *Alice*, *Bob*, and *Eve* all share this structure.

index used, but not the actual bit values. Spoofing attacks are mitigated by adding an authentication code to the index sequence if a significant mismatch was observed; the code is used for privacy amplification (truncating part of the key). In our work, we incorporate carrier frequency offset (CFO) with the channel estimates to develop a key that is more difficult for an adversary to determine. Our resulting key is a function of both unique and reciprocal radio link features.

Adaptive thresholds were used to improve RSS quantization in [5] as a means to extend the work of [4]. The authors also considered single and multi-bit quantization levels to increase secret bit rate. Extensive experimental analysis determined that environments with low variation may be unsuitable for key generation due to low entropy. Mobile scenarios or significant movement within the wireless channel led to high entropy bits. The technique required desired nodes to send information about which time indices to drop from the key generation list, instead of which to include. A method was developed in [9] to avoid sending information over a shared channel by considering larger trend effects; however, due to the decrease in randomness of large scale fading, the secret bit generation rate was low.

Key generation using phase information of channel responses, in contrast to magnitude or RSS, has been shown through simulation to increase bit generation rate due to the randomness of phase on a per sub-carrier level [7], [10]. A higher level of quantization may be performed since decision boundaries may be designed throughout the 0 to 2π valid range. Experimental results of phase-based techniques have been challenging to obtain due to the requirement of high-precision, expensive, hardware to facilitate the quality of measurements needed. Overall, phase based techniques have been shown to provide better performance than signal strength methods; however, the Analog to Digital Converter requirements of these systems are infeasible for most devices, especially low-cost IoT applications [6].

In addition to reciprocal channels, different effects inherent to a radio pair may be used to distinguish a link. Carrier frequency offset (CFO) occurs naturally from a mismatch of local oscillators (LO) due to hardware accuracy or from Doppler shift resulting from mobility. Link authentication using measured CFO has been demonstrated to be successful in the past for practical systems [11]. Tracking of CFO was shown to be feasible even under high mobility through the use of a Kalman filter in [12]. Second order statistics of CFO were used to differentiate between Wi-Fi compliant devices in [13]. Despite its capability to provide authentication, CFO has not been used previously in combination with CSI to improve key generation techniques. We fuse these two sources of unique link characteristics, that inherently exist in wireless systems, to create more robust keys in comparison to what currently exists. We developed a new method of generating keys that provides greater security for a communication link by leveraging existing aspects of the radio while not requiring a drastic increase in computational resources. The remainder of this paper is organized as follows: in Section II the communication link scenario and threat model are introduced, the key generation algorithm is described in Section III, Section IV presents our experimental setup with results, and concluding remarks are provided in Section V.

II. System and Threat Model

In this study we consider a pair of users, $Alice$ and Bob, that attempt to secure their communication link by mutually establishing a secret key. A malicious actor, Eve, is separated from the intended users by at least a distance corresponding to half of the wavelength of the signal. Eve is a passive eavesdropper, that observes the communication link but does not interfere with the channel. $Alice$ and Bob exchange packets in an attempt to generate a key using the methods described in Section III, while Eve intercepts the messages sent by $Alice$, and attempts to replicate the key as shown in Figure 1.

We consider Orthogonal Frequency Division Multiplexing (OFDM) signals, given the widespread use of the modulation scheme, and in particular its adoption for smart-home IoT Wi-Fi enabled devices. The received signal of user j from user i is expressed as

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

where $h_{i,j}(t)$ is the channel impulse response between users i and j, $x_i(t)$ is the transmitted OFDM signal from user i, $\nu_{i,j}$ is the CFO relative to the sub-carrier spacing experienced by user j, N is the number of sub-channels, and $z(t)$ is a noise component. A generic single-input single-output (SISO) OFDM receiver design after down-conversion and sampling is shown in Figure 2. Time and frequency synchronization is performed to align the signal and correct for CFO in the time-domain, a serial to parallel conversion is used prior to the removal of the cyclic prefix used for inter-symbol interference mitigation, and multi-carrier demodulation is performed using a Fast Fourier Transform (FFT). Frequency-domain channel estimates per sub-carrier are calculated using reference symbols, then used to equalize the payload OFDM symbols.

Sampling frequency offset correction is completed on a per symbol basis using an average of estimates based on the phase of the pilot tones. Parallel to serial conversion, single-carrier demodulation, decoding, and de-interleaving are then performed to recover the bitstream. The CFO estimate $\hat{\nu}$ and channel estimates $\hat{\mathbf{H}}$ are inherently existing components of the OFDM receiver that we leverage in our key generation method.

CFO leads to performance degradation in general for communication systems, and is a very important challenge that must be compensated for when using OFDM. A fundamental advantage of using OFDM is that the orthogonality between adjacent sub-carriers removes the requirement of additional guard-bands. CFO directly results in the loss of orthogonality between sub-carriers, causing inter-carrier interference and severe symbol rotation. As described previously, CFO occurs in every wireless link due to either mobility, or with greater effect due to the mismatch of local oscillator clocks. Every physical radio uses a local oscillator that is specified in terms of parts per million tolerance relative to the carrier frequency. As both radios in a link will experience their own offset from the desired center frequency used, the CFO between two radios is unique. The CFO is calculated and compensated for in joint processing with time synchronization, prior to the FFT operation as shown in Figure 2; hence, it is a time-domain operation. The maximum likelihood CFO estimate of the signal received by user j from user i is calculated as

$$\hat{\nu}_{i,j} = \frac{1}{L}\angle\sum_{r=0}^{L-1} U_r U_{r+L}^* \tag{2}$$

where L is an observation length, chosen to be half the number of sub-carriers used in our work, $(.)^*$ is the complex conjugate operation, and U is a received OFDM symbol. The true frequency offset may be obtained by scaling the result by the sampling frequency of the signal. Building from our previous example, Bob's estimate $\hat{\nu}_{A,B}$ is the complement of $Alice$'s estimate $\hat{\nu}_{B,A}$; however, due to RF impairments and the quality of the algorithm used, there will be a difference between the estimates. We use quantization of the CFO estimate to compensate for the difference as will be described in Section III. The CFO measured by Eve, $\hat{\nu}_{A,E}$ and $\hat{\nu}_{B,E}$, do not correlate with the measurements obtained independently by $Alice$ and Bob as a result of the difference in hardware used. Eve is unable to infer $\hat{\nu}_{A,B}$ due to the time-varying nature of the CFO as the phase locked loop re-locks and the individual randomness of the LOs used by all parties.

Due to multi-path resulting from the environment, the wireless channel is a rich source of information and may be leveraged for random bit generation. In addition, the channel between two end-points is reciprocal and enables nodes to independently generate the same sequence of bits. The channel experienced by Bob, $h_{A,B}(t)$ is identical to the channel of $Alice$, $h_{B,A}(t)$. Channel estimation and correction components of the receiver chain as shown in Figure 2 are performed in frequency domain due to the ability to calculate simple single-tap complex equalizers on a per-sub-carrier level. The least square estimator for the channel from user i to user j in frequency domain is given as

$$\hat{H}_{i,j}[k] = \frac{Y_j[k]}{S[k]} \tag{3}$$

where k denotes the OFDM sub-carrier index, and $S[k]$ is a reference OFDM symbol known to the receiver. In our study we assume that $Alice$, Bob, and Eve all share the same knowledge of the value of $S[k]$. Due to half-duplex operation and the RF impairments described in (1) the measured frequency domain channel estimates, $\hat{H}_{A,B}[k]$ and $\hat{H}_{B,A}[k]$, will not match exactly, but will exhibit strong correlation as long as the sampling time is less than the coherence time of the channel. In contrast, Eve measures the channels as $\hat{H}_{A,E}[k]$ and $\hat{H}_{B,E}[k]$ for $Alice$ and Bob respectively. Her estimates will not be correlated with $\hat{H}_{A,B}[k]$ and $\hat{H}_{B,A}[k]$ unless she is physically located within a half-wavelength distance of either node. This is a reasonable assumption considering that at a carrier frequency of 2.4 GHz, Eve would need to be within 6.25 cm of either $Alice$ or Bob. Overall, we consider the case in which Eve has full knowledge of the receiver structure, reference signals, and algorithms used. Eve is physically separated from $Alice$ and Bob as described and observes the communication between the two without interfering.

III. Key Generation Methodology

Our key generation technique consists of three principle components: CFO quantization, channel estimate bit extraction, and key formulation. The overall method considers $Alice$ and Bob sending probe packets back and forth between each other until a maximum allocated time has elapsed. After computing quantized values of CFO and channel estimates, $Alice$ and Bob each send to each other locations of where the channel estimate bits are selected and the distance of the CFO value from the closest decision boundary of each quantizer. The received indices and boundary distances are compared to their own, and additional operations to be discussed are performed to combine the CFO and channel estimate information to form a key.

CFO quantization is necessary prior to key generation due to estimation error differences between nodes. The normalized CFO estimates $\hat{\nu}$ are observed over a window of length, L_d. The absolute value of the estimates is taken prior to computing the mean, and then scaling by the sampling frequency, F_s, to obtain μ_cfo. The result is quantized to different frequency precision levels, $\boldsymbol{q}$, in which for our system we consider 1 kHz, 500 Hz, and 250 Hz. As an example, for the 500 Hz quantizer, decision boundaries exist at 333 and 666 Hz. The distance of the closest decision boundary to μ_cfo is calculated and normalized by $\boldsymbol{q}$ to obtain $\boldsymbol{\delta_q}$ at each level. The resulting quantized CFO values and $\boldsymbol{\delta_q}$ are used in key generation after communicating between legitimate nodes. The procedure is summarized in Algorithm 1.

The bit-extraction method from channel estimates is based primarily on the work shown in [5] with a few key differences. We also compare the channel estimate magnitude to high and low thresholds to output bits 1 and 0 respectively. Instead of

Algorithm 1 CFO Quantization Reporting.

Input: $\hat{\boldsymbol{\nu}}$, $\boldsymbol{q}$, L_d, F_s
Output: ***cfo_quant***, $\boldsymbol{\delta_q}$

Compute absolute value of $\hat{\boldsymbol{\nu}}$ estimates from sample 0 to L_d-1 to obtain $\mid \hat{\boldsymbol{\nu}} \mid$
Calculate mean of $\mid \hat{\boldsymbol{\nu}} \mid$ and scale by F_s to obtain μ_cfo
Initialize: ***cfo_quant***, $\boldsymbol{\delta_q}$
for $k = 0$ to length of $\boldsymbol{q}$ -1 **do**
    Quantize μ_cfo to level $q(k)$ as *cfo_quant(k)*
    Determine distance of μ_cfo from closest decision boundary of quantization level $q(k)$
    Scale result by $q(k)^{-1}$ to obtain $\delta_q(k)$
end for
return ***cfo_quant***, $\boldsymbol{\delta_q}$

observing clusters above or below a threshold to determine a crossing, we adjust our thresholds by using local mean and standard deviation calculations. In addition, to reduce complexity we do not perform time-domain channel estimation, and focus on the estimates of a single band centered subcarrier. An additional condition is placed on our extraction as well, with a maximum time provided to terminate the process, as we leverage the measured CFO to account for incomplete keys. In Algorithm 2 we summarize the channel estimation-based bit extraction method described.

After user i performs the CFO quantization and channel estimation-based bit extraction described in the previous algorithms, an exchange of information is performed between nodes i and j. Both users receive each other's time and quantizer distance vectors to complete the key generation process. The quantized CFO value is selected which corresponds to the largest sum of distances from decision boundaries $i\_\boldsymbol{\delta_q}$ and $j\_\boldsymbol{\delta_q}$. A pseudo random number generator (PRNG) is leveraged to generate bits using the CFO value as its seed. The time indices between both users are compared and only the matching indices are kept, updating the channel bits. In our work if the length of the updated channel bits is less than a minimum required key length, the bits are padded by 0; otherwise, the updated channel bits are truncated to the largest base-2 number. The resulting channel bits are then used in an exclusive or operation with the CFO bits to generate the key. A description of our method is provided in Algorithm 3.

Overall by combining bits extracted from CFO and channel estimation we are able to provide methods to compensate for low bit generation and increase security by creating a more difficult key to determine. In terms of added complexity, our method requires three multiplications more than the channel estimation only derived key, as shown in Algorithm 1. The methods we propose in Algorithm 3 require a PRNG based on the widely adopted Mersenne Twister which consists of simple operations and avoids multiplications and divisions [14]. A small-sized variant of the Mersenne Twister may be implemented to reduce the state space to 127 bits [15].

IV. Performance Evaluation

In this section we assess the quality of our proposed key generation method through experimental results. We describe the hardware and software required for the experiment, discuss our test procedure, and analyze our performance results.

Algorithm 2 Channel Estimation-based Bit Extraction.

Input: $\hat{\boldsymbol{H}}$, S, L_s, L_b, α, T_{MAX}
Output: ***bits_ch***, ***t_index***

Select sub-carrier S from $\hat{\boldsymbol{H}}$ as $\hat{\boldsymbol{H}}_{\boldsymbol{s}}$
Compute magnitude of $\hat{\boldsymbol{H}}_{\boldsymbol{s}}$
Initialize: ***bits_ch***, ***t_index***
cnt = 0
while (length of ***bits_ch*** $< L_s$) & (cnt $< T_{MAX}$) **do**
    Calculate the mean μ_i from $\hat{H}_s$(i-$\frac{L_b}{2}$) to $\hat{H}_s$(i+$\frac{L_b}{2}$)
    Calculate standard deviation σ_i from $\hat{H}_s$(i-$\frac{L_b}{2}$) to $\hat{H}_s$(i+$\frac{L_b}{2}$)
    if $(H_s(i) > \mu_i + \alpha \cdot \sigma_i)$ **then**
        Append 1 to ***bits_ch***
        Append i to ***t_index***
    else if $(H_s(i) < \mu_i - \alpha \cdot \sigma_i)$ **then**
        Append 0 to ***bits_ch***
        Append i to ***t_index***
    end if
    cnt++
end while
return ***bits_ch***, ***t_index***

Algorithm 3 Key Generation of node i after exchange with node j.

Input: ***i_cfo_quant***, $\boldsymbol{i\_\delta_q}$, $\boldsymbol{j\_\delta_q}$, ***i_bits_ch***, ***i_t_index***, ***j_t_index***, L_{min}
Output: ***i_key***

Find index z where maximum of $\boldsymbol{i\_\delta_q + j\_\delta_q}$ occurs
Set PRNG seed to *i_cfo_quant(z)*
Determine ***m_t_index***, matching time indices of ***i_t_index*** and ***j_t_index***
i_bits_ch_match = *i_bits_ch* (***m_t_index***)
if (length of ***i_bits_ch_match*** $< L_{min}$) **then**
    i_bits_ch_upd = ***i_bits_ch_match***
    while (length of ***i_bits_ch_upd*** $< L_{min}$) **do**
        Append 0 to ***i_bits_ch_upd***
    end while
else
    i_bits_ch_upd = ***i_bits_ch_match*** truncated to largest base-2 number
end if
Generate length of ***i_bits_ch_upd*** samples from PRNG as ***i_bits_cfo***
i_key = ***i_bits_ch_upd*** $\bigoplus$ ***i_bits_cfo***
return ***i_key***

A. Experimental Procedure

Our experiment considers multiple Ettus USRP N210 [16] software defined radios (SDR) each equipped with a WBXv4 daughter board and an off the shelf omni-directional antenna. Each node uses DragonRadio, our in-house SDR, which leverages Liquid-DSP for its OFDM Physical layer with Media Access Control (MAC) options of Time Division Multiple Access (TDMA) and Frequency-division duplexing and a highly controllable link layer [17]. DragonRadio is capable of exceeding transmission rates of 3/bits/s/Hz; as an example, it achieves an average aggregate throughput of 20 Mbps for 10 nodes with 10 MHz of spectrum under 15 dB SNR. UHD (USRP Hardware Driver) by Ettus, provides drivers that are required to interact with the SDR. The CFO and channel estimates are written to a log file for every packet received with a valid header. All logs are time-stamped with the system time

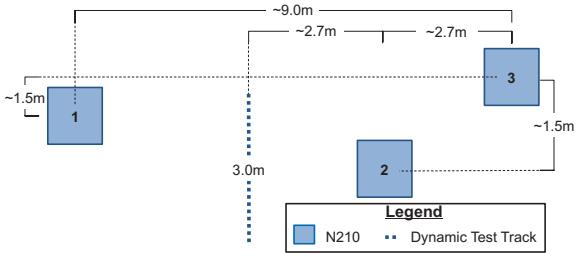

Fig. 3. **Physical Layout.** Position numbers of actors in relation to one another and dynamic test track.

which is approximately the same between SDRs to provide accurate comparisons between statistics.

As mentioned previously in Section II, our model includes two users, *Alice* and *Bob*, that are sending traffic to one another while a third malicious actor, *Eve*, passively obtains the traffic as an eavesdropper. With the intent of simulating an IoT device within a home network, each SDR is set with a carrier frequency of 2.1 GHz, bandwidth of 1 MHz, OFDM with 64 sub-carriers, and a cyclic prefix length of 8. The guard band consists of 11 sub-carriers, 4 sub-carriers are used as pilot tones, the DC-carrier is nulled, and QPSK is selected as the modulation type as summarized in Table I. Our study uses 2 TDMA timeslots: one for *Alice* and one for *Bob*. *Alice* is configured to transmit in timeslot 1 and listen in timeslot 2, while *Bob* is configured to transmit in timeslot 2 and listen in timeslot 1. In our attack model, *Eve* is configured to passively listen in timeslot 1 just as *Bob*, but never transmit. iPerf [18] is used to generate traffic between *Alice* and *Bob* in order to extract the CFO and channel estimates. *Alice* acts as a client, sending and requesting bidirectional UDP traffic to and from *Bob* at 1 Mbps for a duration of 10 seconds. *Bob* performs as the server in UDP mode listening for requests from *Alice* and sending back traffic, as *Eve* passively listens and obtains traffic originating from *Alice*. Using this setup, we are able to probe approximately 870 packets per trial. As mentioned previously, timestamps for each CFO and channel estimate reading are logged to align measurements.

Data is collected for two different environments: static and dynamic. In both scenarios, *Alice*, *Bob*, and *Eve* are placed in the locations marked 2, 1, and 3 respectively as shown in Figure 3. Each node is suspended approximately 3 meters in the air and separated by the distances marked. In the static environment, *Alice* and *Bob* communicate using bi-directional iPerf traffic in an empty room. Dynamic environments have been shown to improve physical layer key generation techniques due to the variations in the channel. The contributions of [5] demonstrated that an environment with movement present in it is rich in randomness, it is not necessary for the nodes themselves to be moving. To represent a dynamic environment, a sheet of metal with a dimension of 68.5cm $\times$ 48cm was moved in between *Alice* and *Bob* at a height of approximately 1.8 meters at a rate of approximately 0.5 meters per second. The sheet moves forward and backward along the 3 meter track described by Figure 3 while *Alice* and *Bob* send traffic to one another.

TABLE I
SIGNAL PARAMETERS IN EXPERIMENT. SHARED CONFIGURATION BETWEEN *Alice*, *Bob*, AND *Eve*.

Multi-Carrier Mod.	OFDM	**Single-Carrier Mod.**	QPSK
No. of Sub-carriers	64	**Multiple Access Scheme**	TDMA
Cyclic Prefix Length	8	**Carrier Frequency**	2.1 GHz
No. Pilot Tones	4	**Signal Bandwidth**	1 MHz
Total Guard Length	11	**DC Null**	Active

In our study, *Alice* and *Bob* send packets between one another to generate a key, while *Eve* attempts to replicate the key using the same method. *Alice* and *Bob* generate their keys using a combination of CFO and channel estimates, both of which are unique between a pair of SDRs. The channel estimates will be uncorrelated to a third-party, assuming that it is separated in distance by at least half a wavelength distance from one of the other SDRs.

B. Results

We generated keys based on our methods described in Section III using the collected SDR data logs in post-processing with MATLAB. CFO quantization was performed for each node using the process described in Algorithm 1 with an observation window of 100 estimates. Similarly, Algorithm 2 was used for channel estimate bit extraction for each node, using sub-carrier 26 (towards the center of the band), filter length of 10, and a maximum of 400 probing packets. Key generation was performed with Algorithm 3 using the outputs from the previous methods for three scenarios: *(i)* *Alice* and *Bob*; *(ii)* *Alice* and *Eve*; *(iii)* *Alice* and *Eve*, with *Eve* having perfect knowledge of Alice's channel estimates, time index vector, and boundary distance vector. Also, it was assumed that the time indices and boundary distances were received without error, and a minimum key length of 256 was selected.

In our analysis we consider the *Percentage of Matching Key Bits* as the percent of bits within computed keys that match over all of our trials. An important consideration is that *Eve* requires all bits to be identical in order to determine the key used by *Alice* and *Bob*. A comparison of matching bits between different nodes with varying threshold scaling factor is shown for both the static and dynamic environment experiments in Figure 4. The performance shown represents the number of matching bits across multiple trials; which includes instances of all bits matching to successfully generate a key. The ideal results in this study would show *Alice* and *Bob* exhibiting a high percentage, with *Alice* and *Eve* yielding a value of 50%. $A\text{-}B_P$ represents our performance of *Alice* and *Bob* using the methods described in Section III, while $A\text{-}B_{ch}$ shows a channel estimate only derived key (with same time indexing reconciliation techniques). Our method demonstrates a clear improvement in both the static and dynamic environments for the ability of *Alice* and *Bob* to generate reciprocal keys. In the static environment our technique provides a pronounced improvement with an increase of 19.75% at an α of 1 in comparison to the channel estimate only technique. As expected, the *Percentage of Matching Key Bits* was much higher in general for an environment with a mobile object in it. A minor gain of 0.8% at an α of

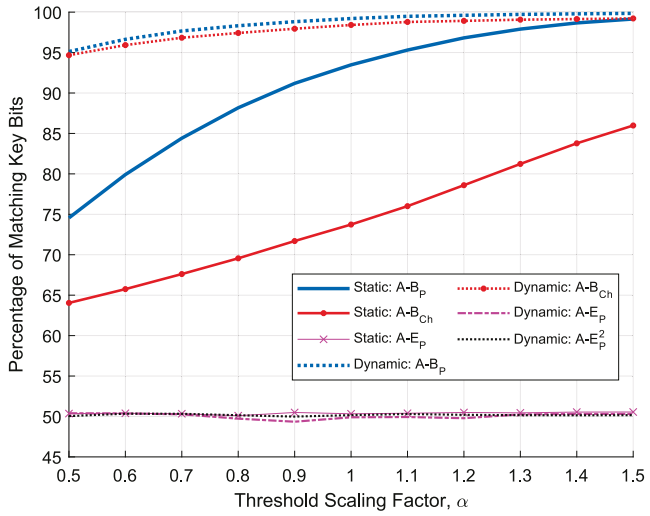

Fig. 4. **Percentage of Matching Key Bits for Both Environment Tests.** Similarity of generated keys between (i) *Alice* and *Bob* and (ii) *Alice* and *Eve* are shown for different scenarios with varying threshold scaling factor.

1 is demonstrated for the dynamic scenario, as the channel is inherently richer and more suitable for bit extraction. In addition, A-E_P shows $Eve's$ inability to match $Alice's$ key while our technique is implemented by *Alice* and *Bob*, even with $Eve's$ full knowledge of algorithms and parameters used. To further demonstrate the effect of our method, we provided *Eve* with perfect knowledge of $Alice's$ channel with *Bob*, the time indices of both nodes, and the boundary distance vectors communicated for quantization reconciliation. The performance for both environments of this scenario were identical, and only the dynamic case is shown in Figure 4. The result is indicated by A-E_P^2 and highlights that our proposed technique reduces the ability of *Eve* to determine the key bits even in an extreme case. Overall our techniques which we propose have demonstrated desirable results for both static and dynamic environments. In particular, we consider our methods as a way to enhance physical layer key generation for IoT devices under fixed environments or to reduce the packet probing duration and adhere to latency requirements. Future studies will explore scenarios with heterogenous equipment used between nodes, online processing of algorithms, and examining scalable solutions in more detail.

V. Conclusion

In this paper we provided a low-complexity method of performing key generation at the Physical layer that is suitable for the constraints of IoT devices. We discussed traditional OFDM receiver designs and modeled the effects of CFO and the wireless channel. The combination of channel estimates and CFO for key generation has not been explored in literature previously to the best of our knowledge. We presented our key generation algorithm which focuses on the unique CFO and channel characteristics of a pair of nodes, as we leverage existing components of the receiver without additional complexity. Our proposed techniques are based primarily on second order statistics and do not require extensive processing capabilities, allowing for our work to be desirable for IoT applications. We performed data collection using USRP N210 SDRs with identical daughter-cards and implemented our algorithms in post-processing. Overall, we demonstrated that our low-complexity techniques improve the ability of desired nodes to establish a key successfully while reducing the capability of an eavesdropper, and are beneficial to securing future IoT devices.

Acknowledgment

This material is based upon work supported by the National Science Foundation under grants CNS-1816387, DGE-1723606, and CCF-1717088.

References

[1] A. Mukherjee, "Physical-Layer Security in the Internet of Things: Sensing and Communication Confidentiality Under Resource Constraints," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1747–1761, Oct 2015.
[2] D. J. Fehr and B. Sandor, "Effects of the WPA2 KRACK Attack in Real Environment," in *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, Sep. 2018, pp. 239–242.
[3] Wi-Fi Alliance, "Discover Wi-Fi," https://www.wi-fi.org/discover-wi-fi.
[4] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *Proceedings of the ACM International Conference on Mobile Computing and Networking*, Sept. 2008, pp. 128–139.
[5] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "Secret Key Extraction from Wireless Signal Strength in Real Environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 917–930, May 2013.
[6] K. Ren, H. Su, and Q. Wang, "Secret key generation exploiting channel characteristics in wireless communications," *IEEE Wireless Communications*, vol. 18, no. 4, pp. 6–12, August 2011.
[7] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2008, pp. 3013–3016.
[8] B. Z. Katz, C. Sahin, and K. R. Dandekar, "Real-time wireless physical layer encryption," in *Proceedings of the IEEE Annual Wireless and Microwave Technology Conference*, Apr. 2016, pp. 1–4.
[9] C. Sahin, B. Katz, and K. R. Dandekar, "Secure and robust symmetric key generation using physical layer techniques under various wireless environments," in *Proceedings of the IEEE Radio and Wireless Symposium*, Jan. 2016, pp. 211–214.
[10] Q. Wang, H. Su, K. Ren, and K. Kim, "Fast and scalable secret key generation exploiting channel phase randomness in wireless networks," in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 1422–1430.
[11] C. G. Wheeler and D. R. Reising, "Assessment of the impact of CFO on RF-DNA fingerprint classification performance," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, Jan 2017, pp. 110–114.
[12] W. Hou, X. Wang, J. Chouinard, and A. Refaey, "Physical Layer Authentication for Mobile Systems with Time-Varying Carrier Frequency Offsets," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1658–1667, May 2014.
[13] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "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. New York, NY, USA: ACM, 2016, pp. 3–14.
[14] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, Jan. 1998.
[15] M. Saito and M. Matsumoto, "Tiny Mersenne Twister (TinyMT)," http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/TINYMT/.
[16] National Instruments, "Ettus Research." https://www.ettus.com/.
[17] J. D. Gaeddert, "Liquid DSP - Software-Defined Radio Digital Signal Processing Library," http://liquidsdr.org/.
[18] J. Dugan et al, "iPerf," https://iperf.fr/.