Demonstrating and Mitigating the Risk of an FEC-Based Hardware Trojan in Wireless Networks

Kiruba Sankaran Subramani, *Student Member, IEEE*, Angelos Antonopoulos, *Member, IEEE*, Ahmed Attia Abotabl, *Member, IEEE*, Aria Nosratinia, *Fellow, IEEE*, and Yiorgos Makris, *Senior Member, IEEE*

***Abstract*— We discuss the threat that malicious circuitry (a.k.a. hardware Trojan) poses in wireless communications and propose a remedy for mitigating the risk. First, we present and theoretically analyze a stealthy hardware Trojan embedded in the forward error correction (FEC) block of an 802.11a/g transceiver. FEC seeks to shield the transmitted signal against noise and other imperfections. This capability, however, may be exploited by a hardware Trojan to establish a covert communication channel with a knowledgeable rogue receiver. At the same time, the unsuspecting legitimate receiver continues to correctly recover the original message, despite experiencing a slight reduction in signal-to-noise ratio (SNR) and, therefore, remains oblivious to the attack. Next, we implement this hardware Trojan on an experimental setup based on the Wireless Open Access Research Platform (WARP) and we demonstrate (i) attack robustness, i.e., the ability of the rogue receiver to correctly receive the leaked information and (ii) attack inconspicuousness, i.e., imperceptible impact on the legitimate transmission. Lastly, we theoretically analyze and experimentally evaluate a Trojan-agnostic detection mechanism, namely, channel noise profiling, which monitors the noise distribution to identify inconsistencies caused by hardware Trojans, regardless of their implementation details. The effectiveness of channel noise profiling is experimentally assessed using the proposed hardware Trojan under various channel conditions and a different covert Wi-Fi attack previously proposed in the literature.**

***Index Terms*— Channel noise profiling, forward error correction encoder, hardware Trojans, wireless networks.**

I. Introduction

Wireless networks have become an inseparable part of everyday life and are now prevalent in most electronic systems. With over 6.8 billion mobile phone subscribers worldwide [1] and over 30 billion Internet of Things (IoT) devices expected by 2020 [2], security and privacy concerns have, inevitably, become paramount. Such concerns are accentuated by the fact that wireless networks exchange sensitive information over public channels and are, therefore, an appealing target. Indeed, staging such attacks is far more plausible since an attacker does not need to obtain physical access to their nodes.

To mitigate security and privacy concerns, most wireless communication networks employ some form of encryption to protect the confidentiality of the information communicated over a public channel [1]. Interestingly, while this provides the user with an –often misleading– sense of security, it also entices attackers, who know that valuable secret information is stored and exchanged between communicating nodes. Hence, wireless networks have been the target of intense attacks [3]–[9], the majority of which are staged via software or firmware modifications that leverage communication protocol vulnerabilities, all the way down to the physical (PHY) layer.

Beyond the attacks exploiting legitimate capabilities of software and firmware, however, hardware-induced vulnerabilities introduce a dangerous new dimension for compromising security and privacy of wireless networks. Such vulnerabilities, wherein the hardware itself serves as the attack surface by exploiting malicious integrated circuit (IC) modifications known as hardware Trojans, have recently emerged due to globalization of the electronics supply chain [10]–[15]. Accordingly, hardware Trojans have become a topic of intense investigation by academic researchers, industry, and governmental entities alike [16], who are realizing the repercussions of deploying Trojan-infested ICs in sensitive applications (e.g. military, financial, infrastructure) and are developing appropriate remedies.

Motivated to address the serious threat that hardware Trojans pose on wireless networks, in [17] we presented a preliminary study of (i) a FEC-based Trojan, which stages an attack in an 802.11a/g network and leaks sensitive information to a rogue receiver, and (ii) a Trojan-agnostic detection method, which leverages the noise characteristics to detect malicious hardware. Herein, we extend this study by theoretically analyzing the operation and impact of the FEC-based Trojan on the legitimate communication link and experimentally validating the effectiveness of the proposed detection mechanism over various channel conditions, as well as for an additional hardware Trojan implementation.

Specifically, compared with [17], in this paper we make the following additional contributions:

- Analytically describe the Trojan operation and its impact on the legitimate transmission.

Manuscript received June 14, 2018; revised November 4, 2018; accepted February 1, 2019. Date of publication February 22, 2019; date of current version June 14, 2019. This work was supported by the National Science Foundation (NSF) under Grant 1514050. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Luc Danger. *(Corresponding author: Yiorgos Makris.)*

K. S. Subramani, A. Nosratinia and Y. Makris are with the Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: kiruba.subramani@utdallas.edu; aria@utdallas.edu; yiorgos.makris@utdallas.edu).

A. Antonopoulos is with u-blox Athens S.A., Maroussi, 15125 Athens, Greece (e-mail: aanton@utdallas.edu).

A. A. Abotabl is with the Samsung SOC Research and Development Laboratory, San Diego, CA 92121 USA (e-mail: axa120731@utdallas.edu).

Digital Object Identifier 10.1109/TIFS.2019.2900906

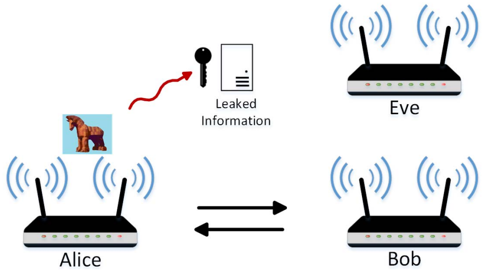
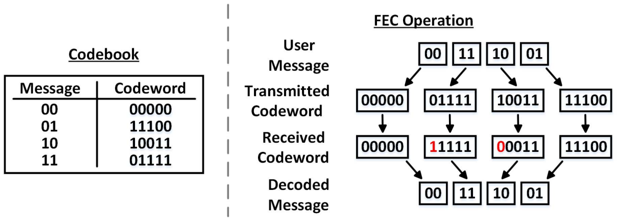


Fig. 1. Threat model.



Message	Codeword
00	00000
01	11100
10	10011
11	01111

Fig. 2. FEC operation.

- Investigate the effect of rogue data on Quadrature Amplitude Modulation (QAM) with respect to its position in the transmitted symbol.
- Experimentally demonstrate the trade-off between Trojan inconspicuousness and attack robustness.
- Analytically describe and experimentally verify the effectiveness of channel noise profiling under various channel conditions using over-the-air experiments.
- Validate the Trojan-agnostic characteristic of channel noise profiling by demonstrating its ability to detect a different, previously published hardware Trojan attack on wireless networks.

Overall, the proposed FEC-based attack exposes the ability of hardware Trojans to establish high-throughput, rogue communication channels in complex, standards-compliant wireless links, while remaining covert and undetectable by commonly employed test methods. Similarly, the proposed channel noise profiling method provides a general and effective defense mechanism which does not assume knowledge of the hardware Trojan specifics and which may not be tampered with by the attacker, as it is implemented on the receiver side.

The remainder of this paper is structured as follows. The threat model is discussed in Section II. The FEC-based Trojan attack along with a theoretical analysis and simulation-based characterization of its operation are presented in Section III. The proposed defense mechanism is introduced and its theoretical analysis is provided in Section IV. An experimental setup for evaluating effectiveness of both the proposed attack and defense is described in Section V. Attack and defense results are presented in Sections VI and VII, respectively. A comparison to related work is provided in Section VIII and conclusions are drawn in Section IX.

II. Threat Model

The threat model considered in this work is presented in Figure 1. “Alice” and “Bob” are two wireless nodes who have established a legitimate communication channel. Unbeknownst to her, “Alice” has been contaminated with a hardware Trojan which leaks sensitive data by embedding it within the legitimate transmission. “Eve” is a rogue receiver who is eavesdropping on the legitimate communication link between “Alice” and “Bob” and who may be using additional capabilities to retrieve the data leaked by the hardware Trojan embedded in “Alice”.

To remain clandestine, the hardware Trojan may only communicate alongside the legitimate transmission, without significantly affecting its quality. In other words, legitimate and rogue data must be transmitted at the same time, thereby increasing the net data transmitted by the Trojan-infested transmitter, compared with the Trojan-free version. In essence, this requires an increase in the outgoing data rate of the Trojan-infested transmitter, while it continues to operate within its circuit and wireless standard specifications. Interestingly, practical wireless devices facilitate such malicious activity because they rarely operate at their specification boundaries. Rather, due to a number of reasons outlined below, there typically exists a margin between the device operating point and the above mentioned boundaries, wherein the hardware Trojan finds room to hide. Such reasons include:

- Optimal transmission may not be pursued due to Intellectual Property (IP) considerations, standards compliance, or other such reasons.
- Optimal reception requires maximum likelihood decoders which have very high computational complexity. Many systems choose simpler decoding instead.
- The optimal transmission parameters are tied to channel conditions which are imperfectly known to transmitter and receiver.
- Circuits are designed conservatively to reduce cost and increase yield in the presence of process variations.

III. FEC-Based Hardware Trojan Attack

We now proceed to describe the FEC-based hardware Trojan attack in the context of an IEEE 802.11a/g network [18].[1] First, the general concept of the FEC-based hardware Trojan is introduced, followed by a theoretical analysis of its impact on the legitimate communication and a simulation-based exploration of its underlying design principles.

A. General Concept

The baseband circuitry of an IEEE 802.11a/g transmitter includes several error control blocks, such as scrambling, encoding, interleaving, cyclic redundancy checks, etc., whose purpose is to protect the transmitted message against noise in the communication link. Among these blocks, FEC provides error correcting capabilities to a receiver by encoding the transmitted message based on a predetermined “codebook”. An example of the FEC operation is shown in Figure 2, where corruption of the transmitted message due to noise is

[1] While the proposed hardware Trojan is demonstrated in an IEEE 802.11a/g network, implementing it in other protocols which use OFDM modulation and similar FEC mechanisms to encode the transmitted messages, such as the recently popular IEEE 802.11n protocol, is also possible.

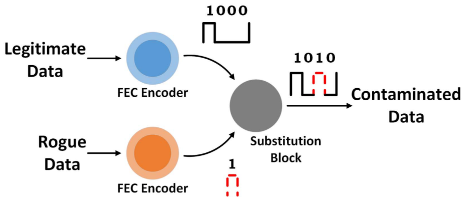


Fig. 3. FEC encoder attack.

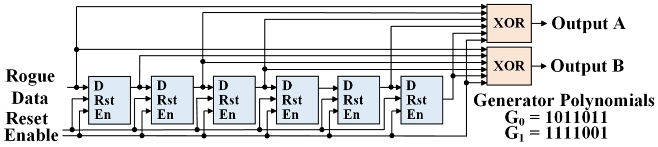


Fig. 4. Circuit diagram of optional FEC encoder.

represented by bit flips in dashed red color. To recover the transmitted message, the receiver selects the codeword which is the closest[2] to the received values.

Due to imperfectly estimated channel conditions and FEC granularity, as well as additional protocol-level error control methods, such as Automatic Repeat reQuest (ARQ), FEC often offers more protection than the channel needs. This, in turn, creates an opening which can be exploited by a knowledgeable adversary to stage a hardware Trojan attack. One such example is shown in Figure 3, where the Trojan replaces some of the user-encoded bits with rogue information. An unsuspecting legitimate receiver senses a slight deterioration in the signal to noise ratio (SNR) caused by the substituted rogue data. Nevertheless, the Viterbi decoder in the receiver chain enables the receiver to effectively retrieve the transmitted message despite the reduced SNR. On the other hand, a knowledgeable adversary (rogue receiver) who is aware of the rogue bit locations and rogue codebook, can use its own Viterbi decoder to recover the leaked information from the received message. In order to stage a successful attack, however, a hardware Trojan should establish a covert channel of reasonably high throughput, while minimizing its impact on the legitimate communication.

The hardware Trojan employs its own (additional) FEC encoder to protect the rogue message, since the rogue data are also susceptible to the same channel effects as the legitimate data (see Figure 3). Unlike decoders, which are often complex and resource consuming, many FEC encoders can be realized with a few shift registers and logic gates, thus enabling simple Trojan FEC circuits. This work concentrates on a communication protected by a convolutional code, where the inserted Trojan also uses a convolutional code to communicate the rogue data. The circuit diagram of the additional FEC encoder needed by the Trojan is shown in Figure 4. It comprises only 6 shift registers and 2 XOR gates to produce output bitstreams corresponding to the generator polynomials $g_0 = 1011011$ and $g_1 = 1111001$, respectively. Thus, without significantly increasing the required resources and, by extension, the risk of being detected, this additional FEC encoder can significantly increase attack robustness, as will be demonstrated experimentally in Section VI.

B. Theoretical Analysis

We now present a systematic study of the hardware Trojan's impact on the legitimate communication that is protected by a convolutional code. The error probability of a convolutional code depends on the input-output weight enumerating function, i.e., the number of input information sequences of weight M that result in an output sequence of weight N, for every M, N. Given a convolutional code of specific rate and constraint length, its error rate performance depends on its distance properties [19]. The distance spectrum of a code is defined as the number of valid codewords that are within a distance d of the all-zero codeword, and is denoted with A_d. The minimum free distance, denoted d_{free}, is the smallest value of d for which A_d is non-zero. As a rule of thumb, the higher the minimum free distance, the lower the probability that a Viterbi decoder will make an error when decoding a received encoded sequence.

The bit-error rate (BER) of the convolutional codes in the additive white Gaussian noise (AWGN) channel is bounded by

$$BER \leq \sum_{d=d_{free}}^{\infty} A_d Q\left(\sqrt{\frac{E_b}{N_o} d}\right) \tag{1}$$

where $\frac{E_b}{N_o}$ is the energy per information bit normalized by the noise power spectral density, and $Q(\cdot)$ is the standard normal Q function. This union bound is dominated by the first few terms in the sum.

Since the packet error event is the union of all possible bit error events, the packet error rate, P_b can be expressed as

$$P_b = 1 - (1 - BER)^k \tag{2}$$

where k is the number of data bits corresponding to the sequence being decoded, which is 414 in our case. The block error probability in terms of the distance spectrum is, then, given by

$$P_b = 1 - \left(1 - \sum_{d=d_{free}}^{\infty} A_d Q\left(\sqrt{\frac{E_b}{N_o} d}\right)\right)^k \tag{3}$$

For demonstration purposes, we use a rate 1/2 convolutional code, with a constraint length of 6 and generator polynomials $g_0 = 1011011$ and $g_1 = 1111001$, which is based on the IEEE 802.11a/g standard [18] and results in the distance spectrum shown in Table I.

In the present model, the Trojan inserts rogue data into the transmission stream via an XOR operation on certain (predetermined) transmission symbols at the output of the FEC encoder. Since, in general, rogue data symbols are equally probable to be zero or one, the FEC codewords are modified in that certain bit positions are flipped with probability 0.5. Recall that the resilience of FEC depends on how far the encoded sequences are from each other. Via flipping some of the encoded bits, the Trojan moves the encoded sequence

[2] In a sense that can depend on the encoding/decoding strategy.

TABLE I

DISTANCE SPECTRUM OF THE CONVOLUTIONAL CODE

d	A_d	d	A_d
10	11	18	7257
12	38	20	40406
14	193	22	234969
16	1331	24	1337714

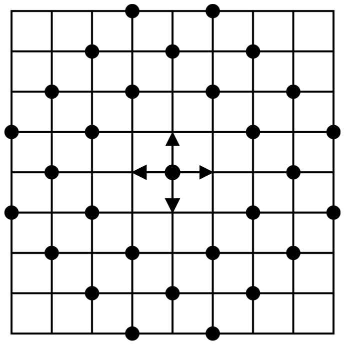


Fig. 5. Trojan impact in the Hamming space.

closer to some neighboring codewords and, also, away from some (other) neighboring codewords. The combination of the two constitutes the net effect of the Trojan on the probability of error of legitimate communication. Figure 5 demonstrates this effect. The grid represents all binary vectors, the center point represents a transmitted codeword that is being modified by the Trojan (arrows), while the solid dots represent nearby codewords whose distance to the transmitted codeword is manipulated by the Trojan.

These changes in the distances to nearby codewords can be modeled by a modification of the distance spectrum of the code. The exact value of the new distance spectrum depends on the rogue bits, which are not known in advance. However, since the probability of these bits is assumed to be uniform, we can attempt a probabilistic version of the distance spectrum. Thankfully this is sufficient for our purposes.

Lemma 1: Let X_i and Y_j be the input (information bits) and output (coded bits) of a binary convolutional code. Then

$$P_{Y_j}(y = \frac{1}{2} \quad \text{if} \quad P_{X_i}(x) = \frac{1}{2} \quad \forall i \tag{4}$$

Proof: We assume a convolutional encoder with a linear time-invariant response: $Y_j = \sum_{i=0}^{\infty} a_i X_{j-i}$, where a_i are the binary coefficients characterizing the impulse response of the code. We can write

$$Y_j = \sum_{i \in S} X_{j-i} \tag{5}$$

where S is the set of indices $i \geq 0$ where $a_i \neq 0$. Since the summation in (5) is a finite field addition, by the crypto lemma [20], if X_i are uniformly distributed, it follows that Y_j are also uniform. ■

If the set of FEC codewords with value 0 and 1 in position i are denoted with $\mathcal{C}_0^{(i)}$ and $\mathcal{C}_1^{(i)}$ respectively, the linearity of the code implies that

TABLE II

DISTANCE SPECTRUM OF THE CONVOLUTIONAL CODE WITH ONE OUTPUT BIT FLIPPED

d	A_d	d	A_d
9	6	17	4294
11	24	19	23822
13	116	21	137688
15	762	23	786342

$$|\mathcal{C}_0^{(i)}| = |\mathcal{C}_1^{(i)}| \tag{6}$$

where $|\mathcal{C}_0^{(i)}|$ denotes the cardinality of the set $\mathcal{C}_0^{(i)}$. This means that any Trojan bit flip will get the encoded FEC codeword closer to half of the neighboring code sequences (that were originally of distance d) and further away from the other half. In other words,

$$A_d^{(1)} = \frac{1}{2}\big(A_{(d+1)} + A_{(d-1)}\big) \tag{7}$$

where $A_d^{(1)}$ is the distance spectrum under one Trojan bit and A_d is the distance spectrum under no Trojan bits.

This idea is shown in Figure 5 and the effect of one Trojan bit on the distance spectrum is shown in Table II. When A_d is an odd number, in order to maintain an integer-valued spectrum we use a pessimistic estimate, i.e., we assume the encoded codeword gets closer to $\left\lceil \frac{A_d}{2} \right\rceil$ code sequences and further away from $\left\lfloor \frac{A_d}{2} \right\rfloor$.

In general, the Trojan will affect more than one bit of the FEC encoded codeword. The generalization of the above-mentioned ideas to multiple bits is straightforward; for example, for two rogue bits per codeword we define the sets $\mathcal{C}_{b_1,b_2}^{(i,j)}$ as the sets of code sequences with a bit of value $b_1 \in \{0, 1\}$ in location i and a bit of value $b_2 \in \{0, 1\}$ in location j. The uniform marginal distribution of the code suggests that

$$|\mathcal{C}_{0,0}^{(i,j)}| = |\mathcal{C}_{0,1}^{(i,j)}| = |\mathcal{C}_{1,0}^{(i,j)}| = |\mathcal{C}_{1,1}^{(i,j)}| \tag{8}$$

Since the number of rogue bits is smaller than the overall number of FEC encoded bits (because the Trojan needs to stay imperceptible), one may assume that the values of the FEC encoded bits at the Trojan-infested bit locations are statistically independent. From this it is straightforward to show that the Trojan bits reduce the distance spectrum to half of neighboring codewords and increase the distance to the other half. Furthermore, in each case half of the distance modifications are by one bit, and half of the modifications are by two bits.

The analysis of the Trojan bits for a convolutional code is also sensitive to the separation between the infected bits within a codeword. If the infected bits are far apart (generally this means far compared with the constraint length of the convolutional code) then each of the infected bits acts independently in the error analysis (although their effect is cumulative). The number of Trojan bits that should be jointly

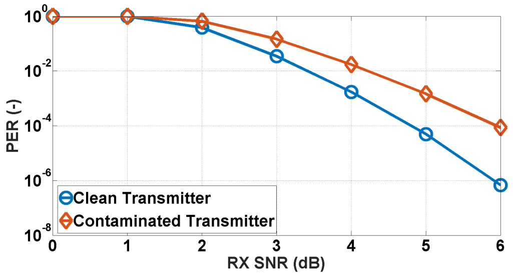


Fig. 6. Theoretical results for clean and contaminated transmitter.

analyzed are, then, the ones that reside within each other's sphere of influence. Altogether, the block error rate for the Trojan-infested codeword is given by

$$P_b = \sum_{i=0}^{T} 2^{-T}\binom{T}{i}\left(1-\left(1-\sum_{d=d_{free}^{(i)}}^{\infty} A_d^{(i)} Q\left(\sqrt{\frac{E_b}{N_o}d}\right)\right)^k\right) \tag{9}$$

where $\binom{T}{i}$ is the number of possible ways we can have i active Trojans out of T Trojan bits.

The packet error rate of binary convolutional codes in an AWGN channel is shown in Figure 6. The two curves correspond to the error probability of the clean and Trojan-infested transmitters operating under Binary Phase Shift Keying (BPSK) modulation. As mentioned earlier, the Trojan operation increases the error probability of the contaminated transmitter. Specifically, the contaminated transmitter requires approximately 0.75 dB more power to achieve an error probability of 10^{-3} compared with the clean transmitter. This additional power can be absorbed in the link margin, or is made up by the adaptive power control loop, since the legitimate receiver's channel estimation will conclude that the channel SNR has been reduced by a small amount, and will request more power from the transmitter.

1) Higher Order Modulations: Analyzing the Trojan's effect on the Quadrature Phase Shift Keying (QPSK) transmission is similar to BPSK. In QPSK, every transmitted symbol consists of two independent bits, each modulated via BPSK and combined in quadrature, therefore the error analysis for QPSK is identical to BPSK. The analysis for 16-QAM is more elaborate. In [21] it was shown that the corresponding packet error rate is upper-bounded by

$$P_b \leq \frac{1}{k}\sum_{d=d_{free}}^{\infty} A(d)P_2(d) \tag{10}$$

where

$$P_2(d) = Q\left(\sqrt{\frac{4}{5}dR_c\frac{E_b}{N_o}}\right) \tag{11}$$

The Trojan's effect on the distance spectrum of the code under 16-QAM can be analyzed in a manner similar to BPSK, but in this case the rogue bit's placement *within the 16-QAM symbol* also plays a role in determining its impact. This is because in 16-QAM, the four bits driving the modulation symbol are subject to several distinctions. The in-phase and quadrature components each consist of a most significant bit (MSB) and a least significant bit (LSB). Inserting the Trojan bit as the MSB vs the LSB has a different impact, because the separation between the clean and contaminated symbols in the signal constellation is different for the two cases. It is, in general, conceivable that a Trojan may decide to choose the least damaging of the two options in terms of SNR. It is also conceivable that a Trojan will attempt to spread the modifications among the LSB and MSB bits in order to better conceal the Trojan noise profile. To avoid any loss of generality, we assume a fraction of the Trojan bits are embedded into LSB and the remaining into the MSB, thus the overall error probability can be characterized by

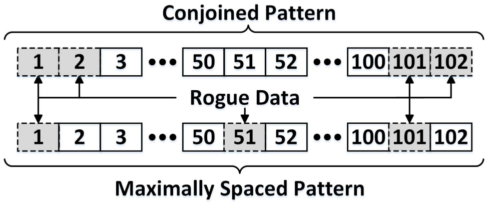


Fig. 7. Shift-error patterns.

$$P_b = P_{LSB}(1)Q\left(\sqrt{\frac{4}{5}dR_c\frac{E_{LSB}}{N_o}}\right) + P_{LSB}(0)Q\left(\sqrt{\frac{4}{5}dR_c\frac{E_{MSB}}{N_o}}\right) \tag{12}$$

where $P_{LSB}(1)$ is the probability that the Trojan bit is carried in the LSB and $P_{LSB}(0) = 1 - P_{LSB}(1)$ is the probability that the Trojan bit is carried in the MSB. E_{LSB} and E_{MSB} are the equivalent energy per bit for the bits transmitted over the LSB and MSB locations, respectively.

C. Trojan Design Principles

We now discuss two principles of designing the proposed hardware Trojan, namely (i) rogue data positioning and (ii) contamination rate, and use simulations to explore their impact on the legitimate communication and the attack throughput.

1) Rogue Data Positioning: The FEC considered in this work is a rate 1/2 convolutional code, which is commonly used in many wireless standards. In rate 1/2 encoders, for each input bit, two output bits are generated and are, then, serialized and passed on to successive blocks in the transmitter chain. The hardware Trojan substitutes some of the bits on these output bitstreams with the bits of the leaked information. We observed via experiments that the relative location of the contaminations in the two components of the FEC encoded bitstream affects the imperceptibility of the hardware Trojan. To analyze this phenomenon, we considered two different implementation scenarios, which are shown in Figure 7:

1) *Conjoined Pattern:* The leaked data bits are inserted in adjacent locations in the two components of the

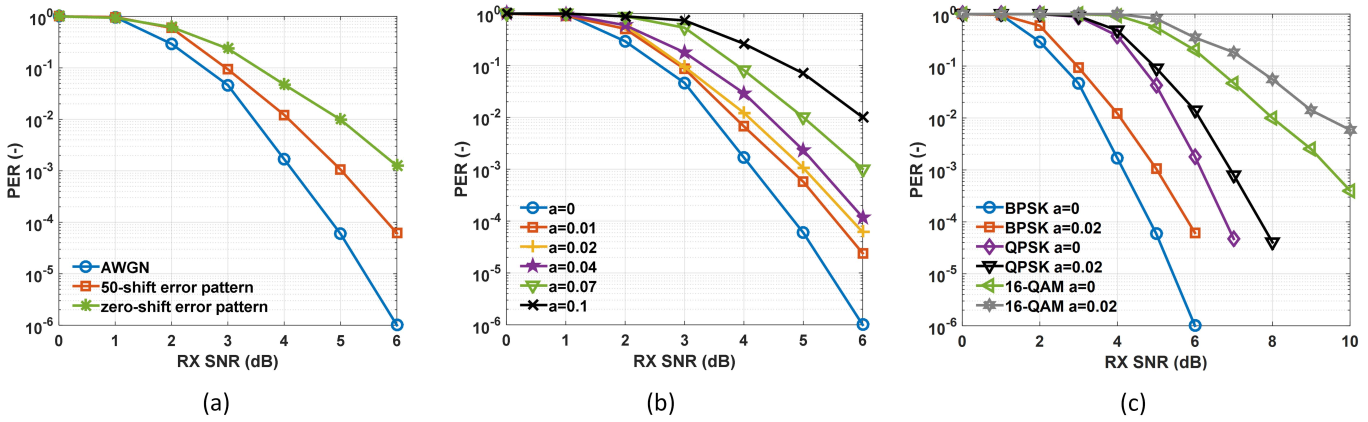


Fig. 8. Simulation of packet error rate (PER) vs. SNR for: (a) different shift-error patterns under BPSK modulation, (b) different contamination rates under BPSK modulation, and (c) clean and contaminated transmitters under BPSK, QPSK and 16QAM modulations.

FEC-output bitstream. For example, for a 2% contamination rate (i.e., 2 out of 100 bits in the legitimate communication are substituted with leaked data bits), these bit positions are at locations 1, 2, 101, 102 and so on.

2) *Maximally Spaced Pattern:* The leaked data bits are inserted in locations that are maximally apart in the two components of the FEC-output bitstream. For example, for a 2% contamination rate, these bit positions are at locations 1, 51, 101, 151 and so on.

Figure 8(a) depicts the simulation-based packet error rate (PER) for these two scenarios and the Trojan-free transmission (i.e., AWGN channel), as a function of SNR. The maximally spaced pattern experiences lower PER compared with the conjoined pattern. This is because Viterbi decoders are sensitive to multiple bit errors within one memory length, which is 6 for the implemented convolutional code. Therefore, maximally spaced pattern is used in the rest of the work to ensure inconspicuousness of the hardware Trojan.

2) Contamination Rate vs. Inconspicuousness: In an over-the-air communication, the main source of error is due to the noise added by the channel. However, in a Trojan-infested communication, the rogue activity is perceived by the legitimate receiver as additional noise. Therefore, it is important to characterize the impact of the hardware Trojan on the legitimate communication as a function of the number of substituted bits. This quantity, called *contamination rate* (α) and depicted in Figure 9, reflects the utility of the hardware Trojan but also impacts its imperceptibility. Higher α increases the rogue data rate at the cost of lower inconspicuousness (higher visibility), since the legitimate receiver demands additional power to achieve the same PER as a Trojan-free communication.

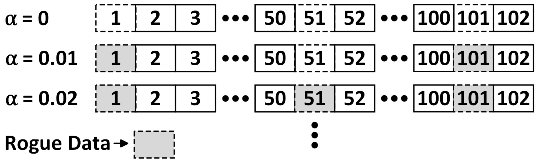


Fig. 9. Contamination rate.

In this work, we analyze contamination rates ranging from $\alpha = 0$ to $\alpha = 0.1$, where $\alpha = 0$ corresponds to a Trojan-free communication and $\alpha = 0.1$ represents the instance where 10% of the legitimate bits are replaced by leaked data bits. Figure 8(b) shows the PER vs. SNR plot for different contamination rates under BPSK modulation. The PER curve corresponding to the Trojan-free transmission (i.e., $\alpha = 0$) can be considered as the baseline. Higher contamination rates lead to a higher effective noise at the legitimate receiver, resulting in higher PER. A contamination rate of $\alpha = 0.1$ is shown to be the outer bound of what the Trojan can leak under the present system, since the Trojan-infested communication with $\alpha = 0.1$ requires significant additional power to achieve the same PER as the Trojan-free communication.

Operating the hardware Trojan at contamination rates below $\alpha = 0.02$ requires negligible additional power (i.e., below 0.5dB) to achieve the same PER as Trojan-free communication, which is well within the typical fluctuations of channel SNR. Therefore, in the remainder of this work a contamination rate of $\alpha = 0.02$ is used for studying and evaluating the proposed hardware Trojan implementation.

Using maximally spaced pattern and contamination rate $\alpha = 0.02$, we characterize the hardware Trojan impact on a communication link under BPSK, QPSK and 16-QAM modulation schemes. Simulation results are provided in Figure 8(c). A comparison with the theoretical analysis results depicted in Figure 6 for BPSK modulation shows a marked agreement between theory and simulation. For example, in both plots, a 0.7dB increase in power is demanded by the contaminated transmitter for a PER of 10^{-3}. The additional SNR required by the Trojan-infested transmission to achieve a PER of 10^{-3} is consistent across the three modulation schemes.

3) Rogue Data Positioning in 16-QAM and 64-QAM : In 16-QAM and 64-QAM, the hardware Trojan impact on the communication varies based on the position of the leaked data bits within the modulated symbol. For example, in a 16-QAM modulation, when the leaked data bit is substituted in the LSB, the corresponding clean and contaminated symbols are located in adjacent locations in the Gray-code mapping.

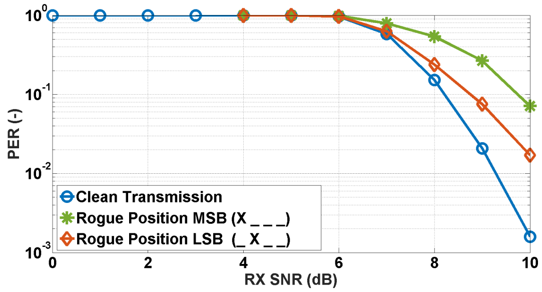


Fig. 10. 16-QAM simulation results.

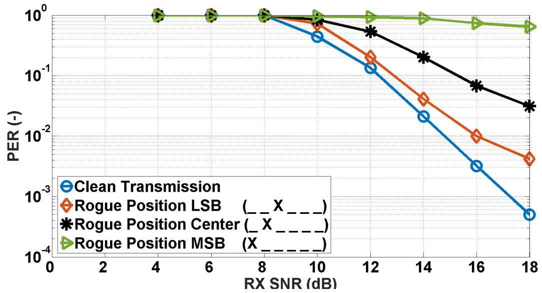


Fig. 11. 64-QAM simulation results.

In contrast, substituting the MSB with a leaked data bit, results in these two symbols being shifted maximally apart in the constellation. Therefore, substituting the LSB will have less impact on the PER than substituting the MSB. This conjecture is corroborated by the simulation results shown in Figure 10, where the PER for a Trojan-free 16-QAM transmission is contrasted to the above two scenarios. The MSB substitution results in significantly higher PER compared with the LSB substitution. A similar effect can be observed in Figure 11, where the plot shows the simulation results for a 64-QAM transmission. Since the constellation points in a 64-QAM modulation are even closer than in 16-QAM, the communication link becomes even more sensitive to the positioning of the leaked data bits within the modulated symbol. Indeed, a progressively higher PER is incurred as the position of the leaked data bit is changed from the LSB, to the center and to the MSB of the modulated symbol.

IV. Proposed Defense

Our objective is not just to expose hardware Trojan-induced vulnerabilities in wireless communications but also to develop appropriate remedies. Therefore, in this section, we introduce a Trojan-agnostic detection method which is based on the general principle of channel noise profiling. We first describe the general concept, followed by the details of the proposed method and a theoretical analysis of its effectiveness. We emphasize that the legitimate receiver is not privy to Trojan-specific information, such as the rogue codebook. Therefore, without additional capabilities, such as the proposed statistical method, the legitimate receiver can only perceive a slight SNR reduction, but cannot definitively attribute it to a Trojan.

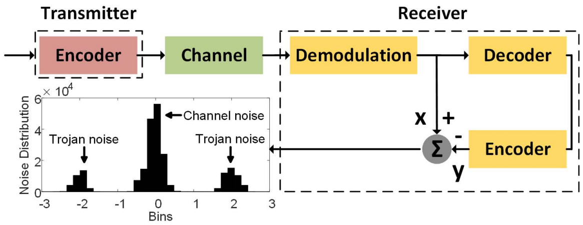


Fig. 12. Channel noise profiling.

A. General Concept

The rogue information which is introduced into a transmission stream inevitably changes, albeit slightly, the signals at the legitimate receiver: instead of the legitimate transmission and AWGN, the received value will consist of the legitimate transmission, AWGN, and the small fluctuations introduced by the Trojan to communicate rogue data. Assuming that the legitimate receiver is able to successfully decode the legitimate signal, it can subtract it from the received values to isolate the noise. This isolated noise can, then, be analyzed for traces of a systematic Trojan transmission. The proposed defense is, therefore, based on the general principle of profiling the channel noise distribution to identify unexpected systematic components which could be caused by Trojan activity. Since the proposed defense mechanism is implemented on the receiver side,[3] it is impervious to tampering by the attacker who has access to the transmitter design to insert the Trojan.

B. Channel Noise Profiling

Over-the-air signal transmissions are prone to the fading channel conditions and are received with additive noise. FEC techniques allow the receiver to recover the transmitted message even in the presence of noise. Channel noise profiling benefits from the ability of the decoders to reconstruct the original (uncontaminated) codewords, thus providing a baseline from which deviations of the received signal can be measured and examined for Trojan activity.

Figure 12 shows the general architecture of the proposed channel noise profiling defense method. Let ‘x’ denote the noisy signal at the input of the decoder. Since decoding recovers the legitimate data, re-encoding it to produce ‘y’ and subtracting it from the corresponding noisy version ‘x’ allows to estimate the effective noise distribution. In a Trojan-free communication through an AWGN channel, the extracted noise distribution is a zero-mean Gaussian. However, for a Trojan-infested transmission, the estimated noise distribution will, in addition to the Gaussian component, also show two impulsive components at positive and negative values corresponding to twice the Trojan BPSK modulation amplitude (see Figure 12). This is because the BPSK modulation represents its two symbols with $+1/-1$, and whenever a Trojan flips a modulated BPSK symbol, by going from $+1$ to -1 or vice versa, a net difference of ± 2 is created with respect to the legitimate signal. This results in impulsive components in the

[3] This defense method can also be implemented via a third party detector that receives the Trojan-infested transmit signal.

effective noise whose presence is utilized by channel noise profiling to detect hardware Trojan activity. The proposed defense mostly leverages existing blocks in a standard receiver, only requiring an additional encoder which operates in parallel with the receiver chain. Therefore, only limited area and power overhead, and no performance overhead, is incurred.

C. Theoretical Analysis

For the purpose of this analysis, we assume that the channel noise is a normalized Gaussian $N(0, 1)$. Furthermore, we assume that the transmitter uses a BPSK modulation with amplitude $\pm a$, and that a fraction 2β of the transmitted BPSK symbols are contaminated by the Trojan. Subject to these conditions, the channel noise profiler will observe a Gaussian mixture random variable with the following distribution

$$f_1 = (1-2\beta)f(0,1) + \beta f(a,1) + \beta f(-a,1)$$

where $f(\mu,\sigma) = \frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

We assume a channel noise profiler that uses a thresholding action on the observed samples. In order to analyze the probability distribution, we use the standard Q-function notation for the tail of the Gaussian probability distribution, i.e.,

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}}e^{-\frac{\alpha^2}{2}}d\alpha$$

Thus, assuming the channel noise profiler isolates the values whose absolute value is greater than t, the probability distribution of each individual observation triggering the channel noise profiler for the clean (non-Trojan) transmitter is simply $2Q(t)$, and for the Trojan-infested transmitter is

$$2(1-2\beta)Q(t) + 2\beta Q(t-a)$$

For the most desirable value of threshold t^*, a logical choice is the value of threshold that optimizes the likelihood function

$$t^* = \arg\max_t \frac{2(1-2\beta)Q(t) + 2\beta Q(t-a)}{2Q(t)}$$

Unfortunately, t^* is nominally dependent on β, the utilization factor of the Trojan, which is not known in advance. However, plotting t^* versus β, as shown in Figure 13, reveals that t^* is essentially insensitive to β and, therefore, one can choose the value of $t^* \approx 2$ which works well for all β.

In Figure 14, we also show the results of several simulations evaluating the detection probability of channel noise profiling, for various channel SNR values as well as for various Trojan contamination rates.

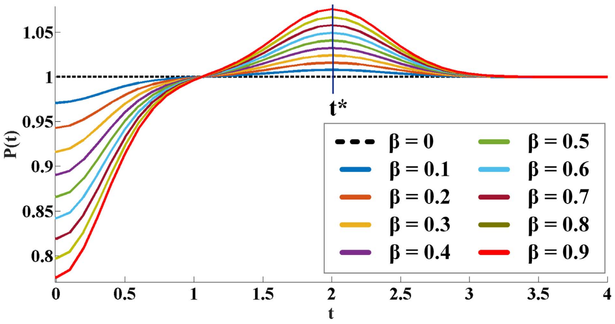


Fig. 13. Optimum threshold (t^*) vs contamination rate (β).

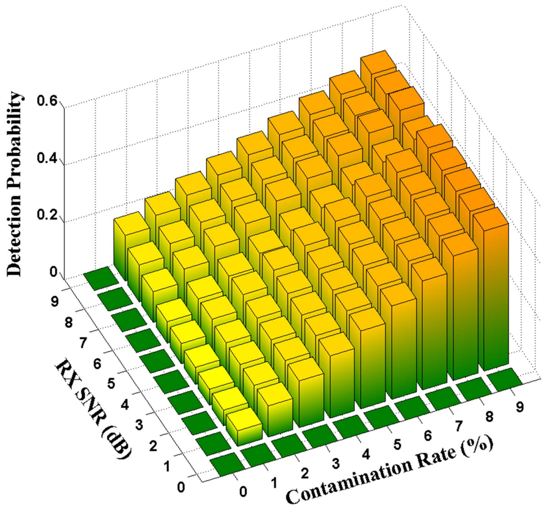


Fig. 14. Detection probability.

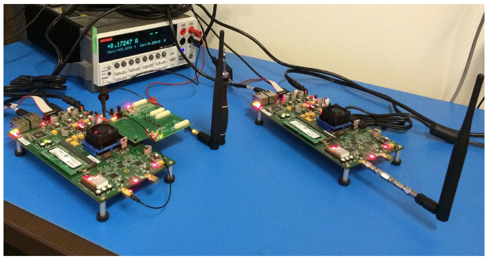


Fig. 15. WARP experimental platform.

V. Experimental Setup

The proposed hardware Trojan was implemented on the WARP experimental platform to investigate attack and defense effectiveness with over-the-air transmissions. Figure 15 shows the experimental setup, which consists of two WARP v3 boards communicating over-the-air using the IEEE 802.11a/g protocol. The two nodes are interfaced to a host PC via a gigabit Ethernet switch. This allows the PC to continuously monitor and control the MAC and PHY layers, which are implemented on an FPGA on each node, using a Python framework. The MAC layer has been programmed to run on the microblaze processors available on the FPGA and is responsible for handling network association requests, collision avoidance and MAC-PHY layer data interactions. The PHY layer, on the other hand, implements the baseband modules of the transceiver nodes using the FPGA fabric.

On the transmitter side, each data packet passes through various PHY layer processing blocks such as scrambling, encoding, puncturing, and interleaving. The FEC-based hardware Trojan, which substitutes some of the legitimate data with leaked data, is also implemented on the FPGA. The contaminated data at the output of the encoder is, then, passed on through several blocks in the baseband, handed over to the data converters and translated into the analog/RF domain.

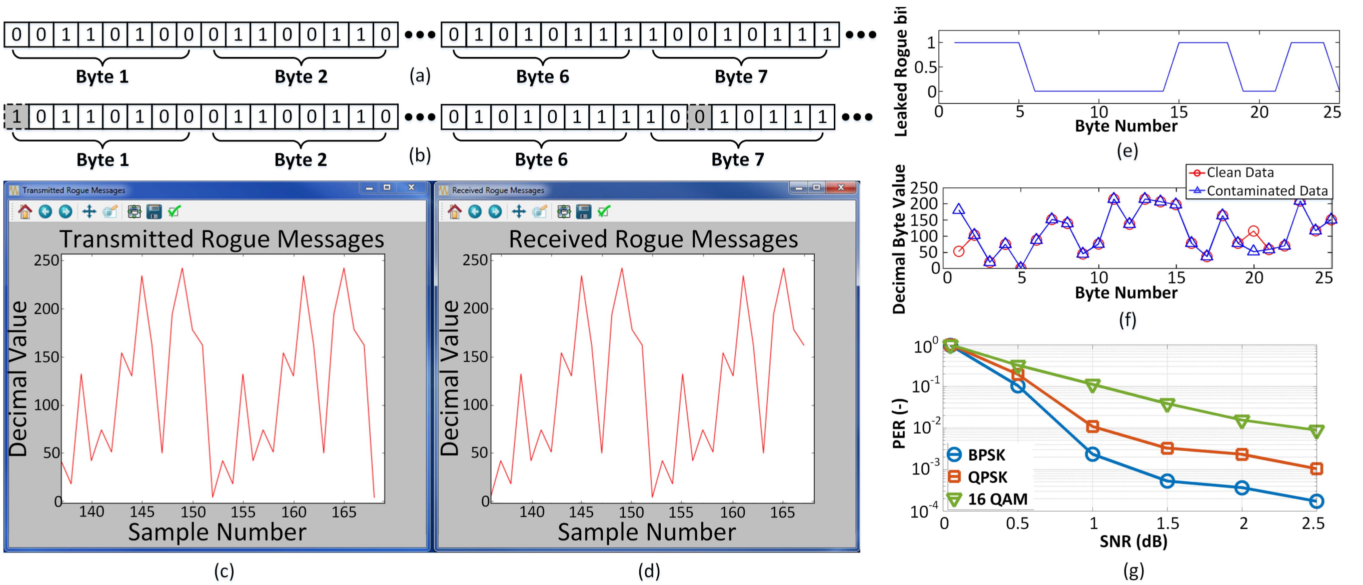


Fig. 16. End-to-end Trojan operation: (a) legitimate data, (b) contaminated data, (c) transmitted rogue message, (d) received rogue message, (e) leaked information bits, (f) comparison of clean and contaminated data and (g) rogue receiver characterization.

On the receiver side, once the data is received, downconverted and digitized, it enters the baseband PHY layer which is implemented on the FPGA. Here, the legitimate receiver uses a Viterbi decoder. The rogue receiver, being aware of the leaked data positions and the rogue codebook, uses an additional Viterbi decoder.[4] This additional decoder was also implemented on the receiver's FPGA to recover the leaked data.

Each WARP v3 FPGA board includes two programmable RF interfaces which allow communication in the 2.4GHz and 5GHz bands. During the experiments, the boards were configured to operate in the 2.4GHz band, with a signal bandwidth of 20MHz, with transmission power levels ranging from 1dBm to 7dBm, with a step size of 1dBm. PER and SNR values were calculated using the received power and frame check sequence for all received packets, which were logged through the Python framework. Experiments were repeated 10 times to ensure consistency and to reduce the impact of measurement noise. All experiments were conducted over-the-air to account for actual channel conditions.

VI. Attack Effectiveness

In this section, the end-to-end operation of the hardware Trojan is described, its impact on the legitimate communication is characterized, and the trade-off between robustness and inconspicuousness of the staged attack is studied.

A. End-to-End Trojan Operation

Figure 16 shows the end-to-end operation of the hardware Trojan and the ability of the rogue receiver to retrieve the leaked information from the received packets. The encoded legitimate data, depicted in Figure 16(a), is modified by the rogue data shown in Figure 16(e) to produce the contaminated data in Figure 16(b). The contamination process is shown in Figure 16(f), where the decimal values of the first 25 bytes of clean and contaminated data are plotted. The first rogue substitution occurs at the MSB of byte 1, changing its decimal value from 52 to 180. The second flip occurs in byte 7; however, since the rogue bit and the legitimate bit at this location are the same, the transmitted data are not affected. The ability of the rogue receiver to successfully retrieve the leaked information is demonstrated by the transmitted and received rogue message plots in Figure 16(c) and Figure 16(d), respectively. The 1 byte lag in the receiver is due to processing time. Finally, the rogue receiver performance is characterized by measuring its PER vs. SNR values. Results are presented in Figure 16(g) for BPSK, QPSK and 16-QAM.

In a practical 802.11a/g communication operating under BPSK modulation (6Mbps data rate), the hardware Trojan achieves a data throughput of 120Kbps for a contamination rate of 2%. Higher rogue throughput can be achieved either by operating the legitimate transmitter at higher data rates or by increasing the Trojan contamination rate while remaining within a reasonable range so that the legitimate communication is not significantly degraded.

B. Impact on Legitimate Communication

The following experiment evaluates the effect of the rogue data position in the transmitted packet. Figure 17(a) shows the PER vs. SNR results for conjoined patterns, maximally-spaced patterns, and for the Trojan-free transmission. The results corroborate the expectation that the maximally-spaced pattern is less prone to packet errors than the conjoined pattern. For example, to achieve equivalent performance to a Trojan-free transmission with PER of 10^{-3}, the maximally-spaced

[4] In this threat model, the rogue receiver is not resource-constrained and can possess additional capabilities (i.e., rogue codebook and additional Viterbi decoder) to support a successful attack.

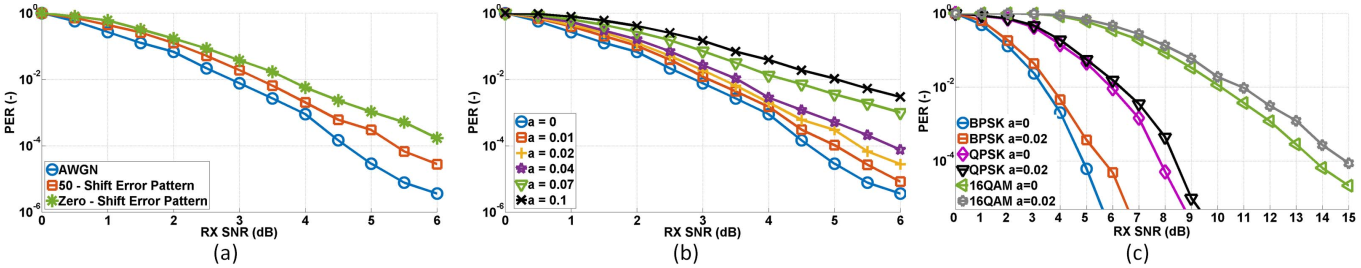


Fig. 17. Experimental results demonstrating PER vs. SNR for: (a) different shift-error patterns, (b) different contamination rates and (c) clean and contaminated transmitters under different modulation schemes.

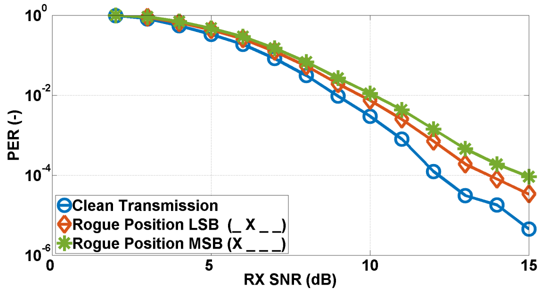


Fig. 18. 16-QAM experimental results.

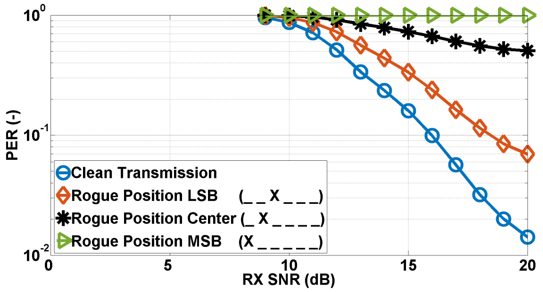


Fig. 19. 64-QAM experimental results.

pattern requires a 0.5dB boost in power, while the conjoined pattern requires 1.25dB.

Next, the effect of Trojan contamination rate on the communication link is studied by varying α from 0% to 10%. The corresponding results are shown in Figure 17(b). The plots are consistent with the simulation results presented in Figure 8(b). For example, both plots agree that a Trojan-infested transmitter with a contamination rate of 2% requires approximately 0.5dB of additional power over a Trojan-free transmission in order to sustain a PER of 10^{-3}.

Figure 17(c) shows experimental results for BPSK, QPSK, and 16-QAM under 2% contamination rate and a maximally-spaced pattern. Trojan-infested transmission requires up to 0.6dB extra power, compared to clean transmission, to achieve a PER value of 10^{-3}. This corroborates that the legitimate receiver perceives the Trojan activity as a slight increase in the background noise and is able to accurately decode its data. Despite hardware imperfections and measurement noise, the measured results are consistent with the simulation results shown in Figure 8(c).

The impact of rogue data positioning in 16-QAM and 64-QAM transmissions was also evaluated using experiments. The corresponding results are presented in Figure 18 and Figure 19. The slower PER waterfall, in comparison with the simulation results in Figure 10 and Figure 11, is attributed to channel and device noise which are not accounted for in the simulations. Despite the slower slope, the impact of leaked data bit position (i.e., LSB vs. center vs. MSB in the transmitted symbol) is consistent across experimental and simulation results.

C. Robustness vs. Inconspicuousness Trade-off

Recall that the Trojan utilizes a secondary FEC to transmit rogue data over the noisy channel (see Figure 4); this introduces a trade-off between attack robustness and inconspicuousness. To study this trade-off, the ability of the rogue receiver to correctly retrieve the leaked data was evaluated with and without the secondary FEC encoder. Figure 20(a) depicts the error probability curves obtained through experiments for both scenarios. As revealed by the lower PER values, encoding enables the rogue receiver to successfully retrieve the leaked data even at low SNR. For the Trojan highlighted in Figure 3, this error performance is once again captured by Equations (1)–(11). Uncoded rogue data, on the other hand, experience significant error to the level that makes it untenable.

An alternate approach to study this trade-off is via observing the leaked data bit errors experienced by the rogue receiver. To demonstrate this, 8 8-bit rogue messages were leaked to produce an 8 × 8 matrix. Each rogue message was transmitted 100 times and the number of bit errors experienced in each received message (byte) per bit location for the encoded and uncoded scenarios is shown using the colormaps of Figure 20(b) and Figure 20(c), respectively. When the leaked information is encoded, the main source of error is due to dropped packets. Therefore, Figure 20(b) has a more uniform coloring with a maximum of 4 bit errors experienced in the first message (byte). In the second scenario, where the rogue data is uncoded, the leaked information experiences many more bit errors in multiple bit positions, resulting in a non-uniform distribution of errors (colormap), as is evident in Figure 20(c).

VII. Defense Effectiveness

The experimental platform and the introduced FEC-based hardware Trojan are used to assess the effectiveness of the

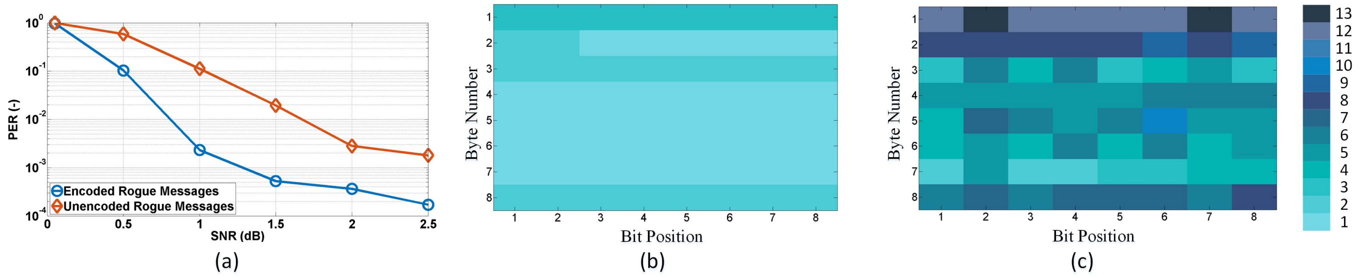


Fig. 20. Trade-off between robustness-inconspicuousness: (a) PER for encoded and uncoded rogue messages, (b) bit errors for encoded rogue messages and (c) bit errors for uncoded rogue messages.

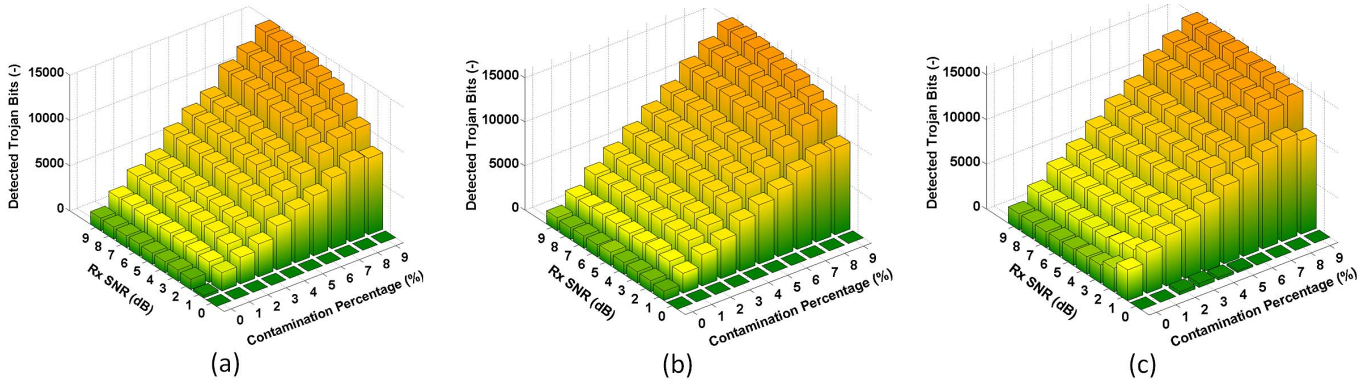


Fig. 21. Effectiveness of channel noise profiling for (a) BPSK, (b) QPSK and (c) 16-QAM.

proposed channel noise profiling method for various modulation schemes and channel conditions. Furthermore, effectiveness of the defense in detecting a different hardware Trojan implementation is demonstrated, thereby verifying its Trojan agnostic nature.

A. Different Modulation Schemes

As explained in Section IV, the number of samples deposited at bin locations 2 and -2 in the noise distribution depends on the SNR observed at the receiver and the contamination rate α used by the hardware Trojan. Therefore, we analyze the effectiveness of channel noise profiling as a function of these two parameters. In our experiments, SNR was varied from 0dB to 9dB and α from 0% to 9%, respectively. For each combination of SNR and α, the number of samples deposited in bins 2 and -2 is recorded and plotted in the form of a 3D plot. Figure 21 presents the results for BPSK, QPSK and 16-QAM modulations.

As SNR increases, the noise distribution exhibits less variation around bins 2 and -2, thereby depositing a larger number of samples exactly in these two bins. Also, as the contamination rate increases, the Trojan replaces more legitimate bits with leaked data bits, therefore a larger number of samples are, again, deposited in bins 2 and -2. These two observations are reflected in the results shown in Figure 21, where the number of detected leaked bits increases along both the SNR and the contamination rate axes. Even at very low SNR, there is a clear distinction in the number of detected Trojan bits between the Trojan-free ($\alpha = 0\%$) and minimal Trojan contamination rate ($\alpha = 1\%$), corroborating the effectiveness of the channel noise profiling as a hardware Trojan detection method. We note that the experimental results presented in Figure 21 are consistent with the theoretical results of Figure 14.

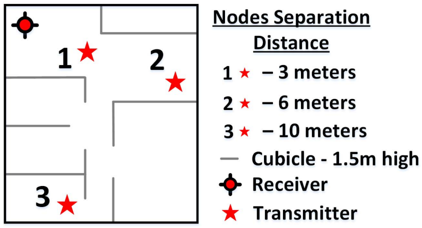


Fig. 22. Node positioning.

B. Effect of Channel Conditions

We now proceed to demonstrate the impact of channel conditions such as path loss, fading etc., on the effectiveness of channel noise profiling. To do so, we carry out various experiments where we position the nodes in various distances under both line-of-sight (LoS) and non line-of-sight (nLoS) communication.

Figure 22 shows the node placement in an office environment for three different separation distances. Positions 1 and 2 correspond to LoS communication with separation distance of 3m and 6m, respectively. Position 3 accounts for the effects of nLoS communication, where the transmitter node is placed inside a cubicle with a separation distance

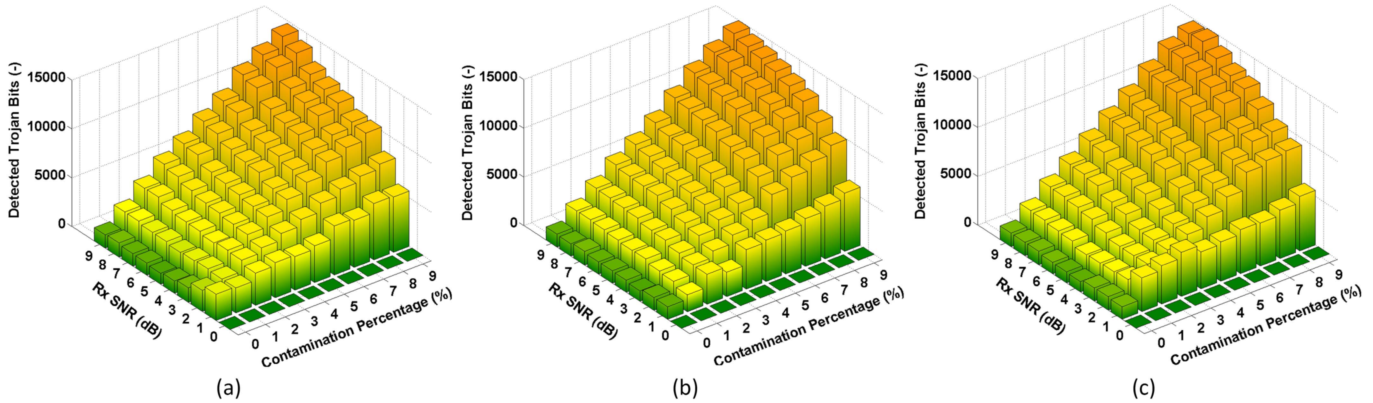


Fig. 23. Effectiveness of channel noise profiling for node separation distance of: (a) 3m (LoS communication), (b) 6m (LoS communication) and (c) 10m (nLoS communication).

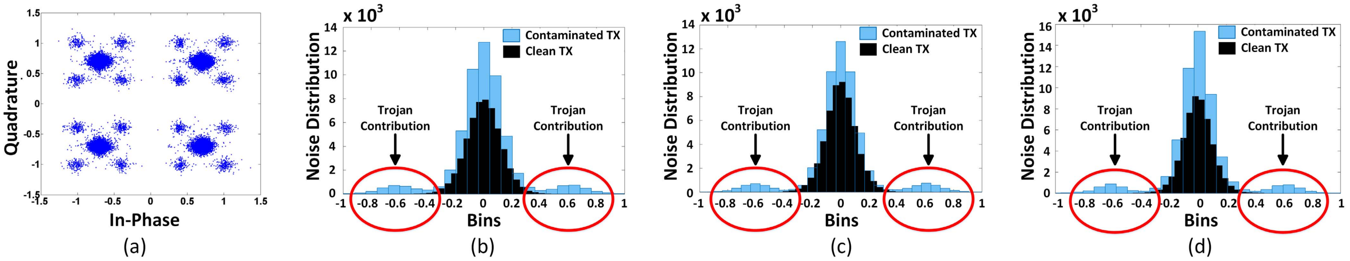


Fig. 24. Effectiveness of channel noise profiling against constellation-changing Trojan [22]: (a) dirty constellation (b) noise distribution at SNR = 10 dB, (c) noise distribution at SNR = 15 dB and (d) noise distribution at SNR = 20 dB.

of 10m. This causes the transmitted signal to undergo reflections and path loss from surrounding obstacles before reaching the receiver. For each of these three channel conditions, the experiment involves over-the-air communication, where the SNR and α parameters are varied from 0dB to 9dB and 0% to 9%, respectively. The detected Trojan bits deposited by channel noise profiling in bins 2 and -2 were measured and the corresponding results for the three scenarios are presented in Figure 23. The three plots are consistent with the trends of Figure 21, thereby confirming effectiveness of channel noise profiling in detecting hardware Trojans under different practical channel conditions.

C. Trojan-Agnostic Effectiveness of Channel Noise Profiling

The last experiment seeks to demonstrate the Trojan-agnostic attribute of channel noise profiling. To do so, a different type of hardware Trojan is used, which creates covert communication channels by modifying the constellation points of the transmitted signal, as explained by Dutta *et al.* [22]. Consistent with the description in [22], the implemented hardware Trojan creates sub-constellations around the QPSK modulation of the legitimate transmitter with a separation distance of $2/\sqrt{42}$. The mutual distance of the sub-constellation points mimics the separation distance of 64-QAM modulation, thereby deceiving the legitimate receiver to assume that normal operation is taking place. In contrast, the rogue receiver extracts the leaked data based on the relative position of the dirty symbols received, compared with the center of the expected clean constellation.

This hardware Trojan was implemented on the experimental platform and the resulting dirty constellation is shown in Figure 24(a). Consistent with the throughput and SNR values described in [22], in our experiments the hardware Trojan substitutes 10% of the transmitted symbols for SNR values of 10, 15 and 20 dB. The results of channel noise profiling are shown in Figure 24 for the three SNR values. In each plot, we have superimposed the channel noise profiling output for the clean and contaminated transmitters to facilitate comparison of the two distributions. The constellation changes introduced by the hardware Trojan result in distinct side lobes in the noise distribution, even for very low SNR values, which are exposed by channel noise profiling.[5]

VIII. Related Work

In this section, we contrast the introduced hardware Trojan threat and mitigation methods to the state of the art in attacks and defenses for wireless ICs and wireless networks.

A. Hardware Trojan Attacks and Defenses in Wireless ICs

1) Hardware Trojans: The first attack which leveraged wireless communication principles to leak information from

[5] In this experiment, channel noise profiling is applied at the input of the demodulator. This allows the defense to analyze the received signal as a modulated symbol, rather than as a binary representation.

a digital cryptographic IC was MOLES [23]. This attack added a code division multiple access (CDMA)-like channel to a crypto-processor to hide information below the noise floor. Hardware Trojans in wireless cryptographic ICs, capable of leaking sensitive information (i.e., the cryptographic key) by modulating transmission power or frequency, were proposed and demonstrated in silicon in [24] and [25]. Similarly, the ability of malicious circuitry to hide unauthorized signals within the ambient noise floor using spread spectrum techniques was shown in [26]. However, all of these attacks were demonstrated on simple wireless links.

2) Defenses: One of the most powerful hardware Trojan detection methods reported in the literature is statistical side-channel fingerprinting [11], [27]–[29]. This method distinguishes between Trojan-free and Trojan-infested chips during post-silicon testing based on the statistics of side channel parameters such as supply current, path delay, power consumption, temperature, or combinations thereof. In the context of wireless cryptographic ICs, effectiveness of statistical side-channel fingerprinting in detecting hardware Trojans was demonstrated using silicon measurements in [24] and [25]. Extensions to remove reliance on golden (i.e., Trojan-free) ICs and to continue hardware Trojan monitoring after deployment were proposed in [30] and [31], respectively. Similarly, in [32], a hardware Trojan detection method capable of self-referencing its performance to detect the Trojan activity without relying on golden ICs was also proposed. However, effectiveness of such methods in the context of wireless networks may largely depend on SNR and Trojan-to-circuit ratio. When applying these methods to detect the FEC-based Trojan, they all fell short, since the Trojan overhead and required SNR increase are negligible.

B. Covert Channel Attacks and Defenses in Wireless Networks

1) Covert Channel Attacks: The majority of attacks in wireless networks has been demonstrated either in theory or in simulation, as in [33] and [34], where information is hidden in unused frame bits or by exploiting software and firmware vulnerabilities [4], [7]. Practical covert channels operating in real hardware have only recently been demonstrated. Specifically, four practical attacks in the PHY layer of 802.11a/g were presented in [35]. However, traditional tests, such as EVM and spectral mask are capable of detecting these attacks, thereby exposing them. A covert channel encoding the leaked information using slight shifts in the constellation points was shown in [22]. However, this method is prone to environmental noise and fading and, thus, its effectiveness strongly depends on the channel model. A covert timing attack in the CSMA/CA protocol, where the spyware leaks information via inter-packet timing, was shown in [36]. However, the random back-off introduced by the CSMA/CA acts as noise, thereby limiting the throughput of the spyware. Finally, information leakage by modifying the channel state information (CSI), which deceives the communication between legitimate users to leak information to an adversary, was presented in [3]. However, this method is only applicable to multiple input multiple output (MIMO) networks.

2) Defenses: Regarding protection mechanisms against covert channel attacks in wireless networks, the state-of-the-art comprises only ad-hoc, attack-specific rather than generalizable approaches. For example, in [3] the transmitter sends known manipulated sequences to the receiver to mislead CSI estimation at clients before it is modified. This idea is based on [37], where communication secrecy and confidentiality are ensured through artificial-noise-aided security. Specifically, the source node generates interference signals that degrade the eavesdropper channel while the intended legitimate channel remains unaffected. This, however, comes at the cost of multiple transmit antennas which are needed to reduce the capacity of the wiretap channel. Moreover, the threat model in [37] is different from the one considered in this paper. Hence this defense is not applicable for detecting the FEC-based hardware Trojan attack.

C. Comparison

In contrast to prior attacks, the FEC-based hardware Trojan proposed herein *(i) goes beyond the capabilities of software and firmware, since it is staged in hardware, (ii) has higher rogue throughput, thereby constituting a more serious threat, and (iii) is compliant with and has been demonstrated using complex wireless standards and protocols, rather than simple links.* Similarly, in contrast to prior defenses, the proposed channel noise profiling method: *(i) is performed at the receiver side, hence its effectiveness cannot be undermined by the attacker, (ii) does not rely on availability of Trojan-free chips, (iii) does not require a MIMO communication scheme, and (iv) is based on general principles and does not assume knowledge of the Trojan attack specifics*.

D. Competing FEC Utilization

Most of the deployed IEEE 802.11a/g devices use FEC to protect the transmitted messages against errors introduced by noisy communication channels. This error correction capability is achieved by adding redundancy in the transmitted message, where the information is encoded based on a predetermined codebook. While the FEC operation improves the communication throughput, the added redundancy reduces the bandwidth of the wireless link. Therefore, various researchers [38]–[40] have proposed the use of adaptive FECs that can dynamically change their parameters based on the observed channel conditions. In other words, the number of redundant bits added to the transmitted message is varied, without affecting the quality of the communication. In such a scenario, the FEC-based Trojan would automatically reduce the rate of leaked information bits, as fewer FEC bits would be available. Therefore, the covert channel bandwidth would decrease. This is expected, as the legitimate communication is now occupying a larger portion of the channel capacity, hence the hardware Trojan has less breathing space to operate in.

IX. Conclusion

We presented a FEC-based hardware Trojan with negligible overhead, which is capable of covertly leaking sensitive information to a rogue receiver while a legitimate receiver remains oblivious to the attack. Trojan operation was theoretically analyzed and design trade-offs were investigated. Its robustness and inconspicuousness were verified using over-the-air experiments on a WARP-based platform implementing the 802.11a/g standard. We also devised a Trojan-agnostic detection mechanism, i.e., channel noise profiling, which is implemented at the receiver side and is capable of identifying any unexpected behavior in noise characteristics produced by malicious activity. Effectiveness of the method was verified for various modulation schemes and channel conditions, while its Trojan-agnostic nature was shown using a different Trojan implementation existing in the literature.

References

[1] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, Sep. 2016.

[2] (2016). *Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated*. [Online]. Available: http://spectrum.ieee.org/tech-talk/telecom/internet/popularinternet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

[3] Y.-C. Tung, S. Han, D. Chen, and K. G. Shin, "Vulnerability and protection of channel state information in multiuser MIMO networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Nov. 2014, pp. 775–786.

[4] A. Cassola, W. K. Robertson, E. Kirda, and G. Noubir, "A practical, targeted, and stealthy attack against WPA enterprise authentication," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2013, pp. 775–786.

[5] A. Díaz, P. Sanchez, J. Sancho, and J. Rico, "Wireless sensor network simulation for security and performance analysis," in *Proc. Conf. Design, Automat. Test Eur. (DATE)*, Mar. 2013, pp. 432–435.

[6] K. Gao, C. Corbett, and R. Beyah, "A passive approach to wireless device fingerprinting," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun./Jul. 2010, pp. 383–392.

[7] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. W. Chang, "Preventing wormhole attacks on wireless ad hoc networks: A graph theoretic approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2005, pp. 1193–1199.

[8] A. Sanatinia, S. Narain, and G. Noubir, "Wireless spreading of WiFi APs infections using WPS flaws: An epidemiological and experimental study," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2013, pp. 430–437.

[9] Y. Zhao, S. Vemuri, J. Chen, Y. Chen, H. Zhou, and Z. Fu, "Exception triggered DoS attacks on wireless networks," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun./Jul. 2009, pp. 13–22.

[10] S. Adee, "The hunt for the kill switch," *IEEE Spectr.*, vol. 45, no. 5, pp. 34–39, May 2008.

[11] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware Trojans: Lessons learned after one decade of research," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 22, no. 1, pp. 6:1–6:23, 2016.

[12] M. M. Tehranipoor, U. Guin, and S. Bhunia, "Invasion of the hardware snatchers: Cloned electronics pollute the market," *IEEE Spectr.*, vol. 54, no. 5, pp. 36–41, May 2017.

[13] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware Trojan attacks: Threat analysis and countermeasures," *Proc. IEEE*, vol. 102, no. 8, pp. 1229–1247, Aug. 2014.

[14] A. Antonopoulos, C. Kapatsori, and Y. Makris, "Hardware Trojans in analog, mixed-signal, and RF ICs," in *The Hardware Trojan War*. Cham, Switzerland: Springer, 2018, pp. 101–123.

[15] K. S. Subramani, A. Antonopoulos, A. A. Abotabl, A. Nosratinia, and Y. Makris, "ACE: Adaptive Channel Estimation for Detecting Analog/RF Trojans in WLAN Transceivers," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017, pp. 722–727.

[16] (2005). *Defense Science Board (DSB) Study on High Performance Microchip Supply*. [Online]. Available: http://www.acq.osd.mil/dsb/reports/2000s/ADA435563.pdf

[17] K. S. Subramani, A. Antonopoulos, A. A. Abotabl, A. Nosratinia, and Y. Makris, "INFECT: INconspicuous FEC-based Trojan: A hardware attack on an 802.11a/g wireless network," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2017, pp. 90–94.

[18] *IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2012, Mar. 2012. [Online]. Available: https://standards.ieee.org/findstds/standard/802.11-2012.html

[19] J. G. Proakis and M. Salehi, *Digital Communications*. New York, NY, USA: McGraw-Hill, 2008.

[20] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[21] R. C. Manso, "Performance analysis of M-QAM with Viterbi soft-decision decoding," M.S. thesis, Nav. Postgraduate School, Monterey, CA, USA, 2003.

[22] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "Secret agent radio: Covert communication through dirty constellations," in *Proc. Int. Workshop Inf. Hiding*, 2013, pp. 160–175.

[23] L. Lin, W. Burleson, and C. Paar, "Moles: Malicious off-chip leakage enabled by side-channels," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2009, pp. 117–122.

[24] Y. Liu, Y. Jin, and Y. Makris, "Hardware Trojans in wireless cryptographic ICs: Silicon demonstration & detection method evaluation," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2013, pp. 399–404.

[25] Y. Liu, Y. Jin, A. Nosratinia, and Y. Makris, "Silicon demonstration of hardware Trojan design and detection in wireless cryptographic ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 4, pp. 1506–1519, Apr. 2017.

[26] D. Chang, B. Bakkaloglu, and S. Ozev, "Enabling unauthorized RF transmission below noise floor with no detectable impact on primary communication performance," in *Proc. IEEE 33rd VLSI Test Symp. (VTS)*, Apr. 2015, pp. 1–4.

[27] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *IEEE Design Test Comput.*, vol. 27, no. 1, pp. 10–25, Jan./Feb. 2010.

[28] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *Proc. IEEE Int. Workshop Hardw.-Oriented Secur. Trust (HOST)*, Jun. 2008, pp. 51–57.

[29] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 296–310.

[30] Y. Liu, K. Huang, and Y. Makris, "Hardware Trojan detection through golden chip-free statistical side-channel fingerprinting," in *Proc. 51st Annu. Design Automat. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[31] Y. Liu, G. Volanis, K. Huang, and Y. Makris, "Concurrent hardware Trojan detection in wireless cryptographic ICs," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2015, pp. 1–8.

[32] F. Karabacak, U. Y. Ogras, and S. Ozev, "Detection of Malicious Hardware Components in Mobile Platforms," in *Proc. 17th Int. Symp. Qual. Electron. Design (ISQED)*, Mar. 2016, pp. 179–184.

[33] L. Frikha, Z. Trabelsi, and W. El-Hajj, "Implementation of a covert channel in the 802.11 header," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2008, pp. 594–599.

[34] K. Szczypiorski and W. Mazurczyk, "Steganography in IEEE 802.11 OFDM symbols," *Secur. Commun. Netw.*, vol. 9, no. 2, pp. 118–129, 2016.

[35] J. Classen, M. Schulz, and M. Hollick, "Practical covert channels for WiFi systems," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Sep. 2015, pp. 209–217.

[36] N. Kiyavash, F. Koushanfar, T. P. Coleman, and M. Rodrigues, "A timing channel spyware for the CSMA/CA protocol," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 477–487, Mar. 2013.

[37] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2180–2189, Jun. 2008.

[38] L. J. Chen, T. Sun, M. Y. Sanadidi, and M. Gerla, "Improving wireless link throughput via interleaved FEC," in *Proc. 9th Int. Symp. Comput. Commun. (ISCC)*, vol. 1, Jul. 2004, pp. 539–544.

[39] L. Baldantoni, H. Lundqvist, and G. Karlsson, "Adaptive end-to-end FEC for improving TCP performance over wireless links," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2004, pp. 4023–4027.

[40] T. Tsugawa, N. Fujita, T. Hama, H. Shimonishi, and T. Murase, "TCP-AFEC: An adaptive FEC code control for end-to-end bandwidth guarantee," in *Proc. IEEE Int. Packet Video Workshop (PV)*, Nov. 2007, pp. 294–301.

Kiruba Sankaran Subramani (S'16) received the B.Tech. degree (Hons.) in electronics and communication engineering from the Amrita School of Engineering, India, in 2009, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Dallas, in 2013 and 2018, respectively. He is currently a Post-Doctoral Research Associate in electrical and computer engineering with The University of Texas at Dallas. His research interests include hardware security in wireless networks and design of secure and robust analog/RF integrated circuits and systems. He was a recipient of the Best Hardware Demonstration Award from the 2018 IEEE Symposium on Hardware Oriented Security and Trust (HOST'18).

Angelos Antonopoulos (S'12–M'16) received the M.Eng., M.Sc., and Ph.D. degrees from the School of Electronic and Computer Engineering, Technical University of Crete, Chania, Greece, in 2005, 2008, and 2014, respectively. In 2015, he joined the Trusted and RELiable Architectures (TRELA) Research Laboratory, The University of Texas at Dallas (UTD), Richardson, TX, USA, as a Post-Doctoral Research Associate. He is currently a Patent Engineer with u-blox Athens S.A., Greece. His research interests include the design of trusted and reliable analog/RF integrated circuits and systems, hardware security in wireless networks, and design-oriented compact modeling of advanced semiconductor devices. He was a recipient of the Best Hardware Demonstration Award from the 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST'18).

Ahmed Attia Abotabl (S'15–M'17) received the B.S. degree (Hons.) from Alexandria University, Egypt, the M.Sc. degree from Nile University, Egypt, and the Ph.D. degree from The University of Texas at Dallas, Richardson, TX, USA, all in electrical engineering. Since 2017, he has been with the Samsung SOC Research and Development Laboratory, San Diego, CA, USA, where he is involved in algorithm development for 5G wireless modems. His research interests include information theory, coding theory and their applications in physical layer security, and machine learning. He was a recipient of the Erik Jonsson Graduate Fellowship in 2012, the Louis-Beecherl Jr. Award in 2015, and the UTD Electrical Engineering Industrial Advisory Board Award in 2016, from The University of Texas at Dallas.

Aria Nosratinia (S'87–M'97–SM'04–F'10) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1996. He has held visiting appointments at Princeton University, Rice University, and UCLA. He is currently an Erik Jonsson Distinguished Professor and the Associate Head of the Electrical and Computer Engineering Department, The University of Texas at Dallas. His research interests include information theory and signal processing, with application in wireless communications. He is a Fellow of the IEEE for contributions to multimedia and wireless communications. He was a recipient of the National Science Foundation Career Award, and the Outstanding Service Award from the IEEE Signal Processing Society, Dallas Chapter. He served on the organizing and technical program committees of a number of conferences, most recently as a General Co-Chair of the IEEE Information Theory Workshop 2018. He has served as an Editor and an Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and as an Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, the *IEEE Wireless Communications (Magazine)*, and the *Journal of Circuits, Systems, and Computers*. He was named as a highly cited researcher by Clarivate Analytics (formerly Thomson Reuters).

Yiorgos Makris (SM'08) received the Diploma degree from the University of Patras, Greece, in 1995, and the M.S. and Ph.D. degrees from the University of California San Diego, San Diego, CA, USA, in 1998 and 2001, respectively, all in computer engineering. After spending a decade on the faculty of Yale University, he joined The University of Texas at Dallas, where he is currently a Professor of electrical and computer engineering, leading the Trusted and RELiable Architectures (TRELA) Research Laboratory, and the Safety, Security and Healthcare Thrust Leader for the Texas Analog Center of Excellence (TxACE). His research interests include applications of machine learning and statistical analysis in the development of trusted and reliable integrated circuits and systems, with particular emphasis in the analog/RF domain. He was a recipient of the 2006 Sheffield Distinguished Teaching Award, Best Paper Awards from the 2013 IEEE/ACM Design Automation and Test in Europe (DATE'13) conference and the 2015 IEEE VLSI Test Symposium (VTS'15), and Best Hardware Demonstration Awards from the 2016 and the 2018 IEEE Hardware-Oriented Security and Trust Symposia (HOST'16 and HOST'18). He serves as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and has served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and the *IEEE Design & Test of Computers* Periodical, and as a Guest Editor for the IEEE TRANSACTIONS ON COMPUTERS and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS.