# INFECT: INconspicuous FEC-based Trojan: a Hardware Attack on an 802.11a/g Wireless Network

Kiruba Sankaran Subramani, Angelos Antonopoulos, Ahmed Attia Abotabl, Aria Nosratinia and Yiorgos Makris
Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080
Email: {kiruba.subramani, aanton, ahmed.abotabl, aria, yiorgos.makris}@utdallas.edu

***Abstract*—We discuss the threat that hardware Trojans (HTs) impose on wireless networks, along with possible remedies for mitigating the risk. We first present an HT attack on an 802.11a/g transmitter (TX), which exploits Forward Error Correction (FEC) encoding. While FEC seeks to protect the transmitted signal against channel noise, it often offers more protection than needed by the actual channel. This margin is precisely where our HT finds room to stage an attack. We, then, introduce a Trojan-agnostic method which can be applied at the receiver (RX) to detect such attacks. This method monitors the noise distribution, to identify systematic inconsistencies which may be caused by an HT. Lastly, we describe a Wireless open-Access Research Platform (WARP) based experimental setup to investigate the feasibility and effectiveness of the proposed attack and defense. More specifically, we evaluate (i) the ability of a rogue RX to extract the leaked information, while an unsuspecting, legitimate RX accurately recovers the original message and remains oblivious to the attack, and (ii) the ability of channel noise profiling to detect the presence of the HT.**

## I. Introduction

Wireless networks have become an inseparable part of everyday life and are now prevalent in most electronic systems, due to the rapid growth of telecommunications and the Internet of Things. At the same time, they are particularly vulnerable and constitute an appealing target for malicious attacks; indeed, since they exchange information over public channels, an attacker does not need to obtain physical access to the nodes, making such attacks far more plausible. Most wireless communication networks employ some form of encryption in order to protect the privacy of the information communicated over a public channel. 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 the communicating nodes. As a result, wireless networks have been the target of covert channel attacks [1]–[11], most of which are staged via software or firmware modifications that leverage communication protocol vulnerabilities, all the way down to the physical (PHY) layer. Beyond such attacks, however, which exploit legitimate hardware and protocol capabilities, malicious hardware modifications known as HTs can be introduced by a knowledgeable adversary to cause erroneous results, steal sensitive data, or incapacitate a chip [12].

Motivated to address this serious threat, in this work we investigate the risk that HTs pose in this context. Specifically, this paper makes the following *contributions*:

- An HT staging a covert channel attack in an 802.11a/g TX, exploiting the inherent ability of FEC encoders to suppress occasional bit-flips. Leaked information is hidden within legitimate transmission, from where an informed rogue RX can easily extract it, while an unsuspecting RX only senses a slight increase in channel noise.
- A *Trojan-agnostic* detection method, which operates at the RX and monitors channel noise characteristics in order to identify systematic inconsistencies and expose the presence of an HT in the TX.
- An experimental demonstration of the HT attack and defense using WARP [13], which investigates the trade off between attack robustness and detectability and evaluates effectiveness of channel noise profiling.

## II. FEC Attack

**General Idea:** The baseband of an 802.11a/g TX is a complex system with blocks performing scrambling, encoding, interleaving, etc. Among these blocks, FEC is designed to provide immunity against channel noise. However, due to reasons including engineering robustness, conservative calculation of link budget and granularity of FEC capabilities, the amount of protection offered by FEC is, typically, more than the channel needs. Moreover, use of Automatic Repeat reQuest (ARQ) offers additional mechanisms so that the demands on FEC are not exacting (i.e., occasional failure of FEC can be tolerated). The operational margins offered by such capabilities introduce a region wherein an HT may covertly operate.

More specifically, since FEC provides robustness against signal deviations, an HT can be carefully designed such that the deviation it introduces remains within the error handling capability of the FEC. Besides noticing a slight increase in channel noise, the legitimate RX remains unaffected and oblivious to the presence of the HT. The rogue RX, however, is aware of the deviations introduced by the HT and is able to extract the leaked information from the compromised signal.

Evidently, for an HT to successfully stage an attack, the following two conditions should hold: **(i) Inconspicuousness:** the legitimate RX should correctly receive the legitimate data, and **(ii) Robustness:** the rogue RX should correctly receive the rogue data. This, in turn, introduces a trade-off between the bandwidth of the HT and the risk of being detected, as we will highlight in our results.

**TX Modification:** The TX modification required to stage a FEC attack is shown in Figure 1. Specifically, this attack

978-1-5386-3929-0/17/$31.00 

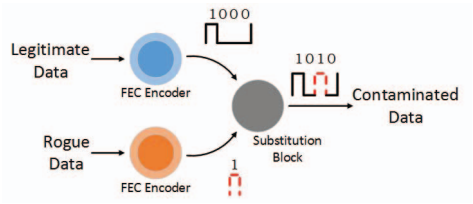


Fig. 1: FEC encoder attack.

hijacks some of the FEC-encoded legitimate output bits, substituting their values with rogue information bits. However, considering that the modified bits themselves go through a hostile (noisy) channel, an additional FEC encoder is used for encoding the rogue data prior to the substitution[1]. Through this process, the output of the HT-infected FEC encoder consists of a mixture of both legitimate and rogue information, which then propagates though the rest of the transmission blocks.

**Rogue RX Design:** Upon reception of the contaminated data, the legitimate RX uses a Viterbi decoder to correct any errors, independent of whether such errors were introduced by the channel or the HT. Indeed, the legitimate RX is unaware of the presence of rogue data. The rogue RX, however, knows not only that rogue data is present but also where to look for it. Therefore, the rogue RX extracts the encoded leaked information from the received message and decodes it using a second Viterbi decoder.

**Rogue Data Positioning:** We consider a commonly used rate-1/2 FEC, wherein for an input bitstream, two encoded output bitstreams are generated, serialized and transmitted. The HT inserts rogue bits on both output streams. The exact location of the rogue bits within the two streams determines the impact of rogue data on the legitimate system (i.e., its imperceptibility). To further elucidate this point, Figure 2 shows two scenarios:

1) **Zero-shift error pattern:** The encoded rogue bits are inserted in adjacent locations.
2) **50-shift error pattern:** The encoded rogue bits are inserted in positions that are maximally apart.

Figure 3(a) shows the packet error rate (PER) at the legitimate RX versus the RX signal-to-noise ratio (SNR) for these two scenarios for Binary Phase Shift Keying (BPSK) modulation. Both plots correspond to a rogue transmission where one out of every 50 legitimate bits (i.e., 2%) is replaced by a rogue bit. As may be observed, the legitimate RX achieves a lower PER for the second scenario, because the Viterbi decoder is more sensitive to bit errors that are within one memory length, and this convolutional encoder has constraint length (memory) of 6. Thus, in order to achieve the same PER as the clean transmission, the 50-shift error pattern requires lower SNR at the legitimate RX than the zero-shift option. Therefore, our FEC attack uses the 50-shift error pattern.

[1]We note that FEC encoders are often very simple; for example, a convolutional FEC encoder consists of only a few shift registers. Nevertheless, if the area/power footprint of the added FEC encoder is a concern, it can be simplified or omitted. In this case, the error probability of the rogue data will be increased, as it will be transmitted with less or no FEC capabilities. In essence, attack robustness is traded off with attack inconspicuousness.

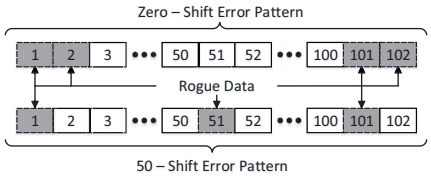


Fig. 2: Shift-error patterns.

**Rogue Data Rate, Inconspicuousness & Robustness:** The required SNR increase for decoding the contaminated packet is sensed by the legitimate RX as extra noise. By controlling the number of leaked bits, the HT can ensure that this "extra noise" remains low and hidden among the various uncertainties and variations in the communication process, thereby going unnoticed. The number of Trojan-leaked bits is expressed as the *contamination rate,* $a$. Higher rogue data rate comes at the cost of lower inconspicuousness, since the legitimate RX demands extra power to compensate for an increase in $a$ and continue to operate within a target error probability.

This trade-off between rogue data rate and attack inconspicuousness is demonstrated in Figure 3(b), wherein the simulated PER of the legitimate RX is plotted against the SNR, for an Additive white Gaussian noise (AWGN) channel, under various levels of contamination rate for the 50-shift error pattern. The transmitted bits are modulated using BPSK with a coding rate of $1/2$. The ratio $a$ spans from $0$, where no rogue bits are inserted, to $0.1$ where one rogue bit is inserted for every 10 legitimate bits. Evidently, a $10\%$ contamination rate may raise a red flag since an additional 2.5dB of power is required to obtain a PER of $10^{-2}$, as compared to the clean transmission. On the other hand, at the expense of lower rogue data throughput, contamination rates of $1\% - 2\%$ leave a very small trace in terms of SNR, as they require less than 0.5dB of additional power to obtain a PER of $10^{-2}$. Therefore, in the rest of this work, we use $a = 0.02$ as a compromise between rogue data rate and Trojan inconspicuousness when implementing the FEC attack.

Figure 3(c) provides simulation results for the clean and $2\%$ contamination rate transmissions for three modulations schemes, namely BPSK, Quadrature Phase-Shift Keying (QPSK) and Quadrature Amplitude Modulation (16-QAM), all with a coding rate of $1/2$. In all cases, an additional SNR of approximately 0.5dB-1.0dB is needed for the Trojan-infested transmission to achieve a PER of $10^{-2}$, verifying robustness of the proposed attack across modulation schemes.

## III. Detection Mechanism

Our objective is not just to expose hardware-induced vulnerabilities but also to develop appropriate remedies. Therefore, in this section we describe a *Trojan-agnostic* detection method based on the general principle of *channel noise profiling*. We emphasize that the legitimate RX is not privy to the Trojan-specific information (i.e. rogue codebook). Therefore, without additional capabilities, such as the proposed statistical method, the legitimate RX cannot perceive Trojan existence.

The foundation of the proposed method is the observation that HTs which seek to hide their impact within channel noise

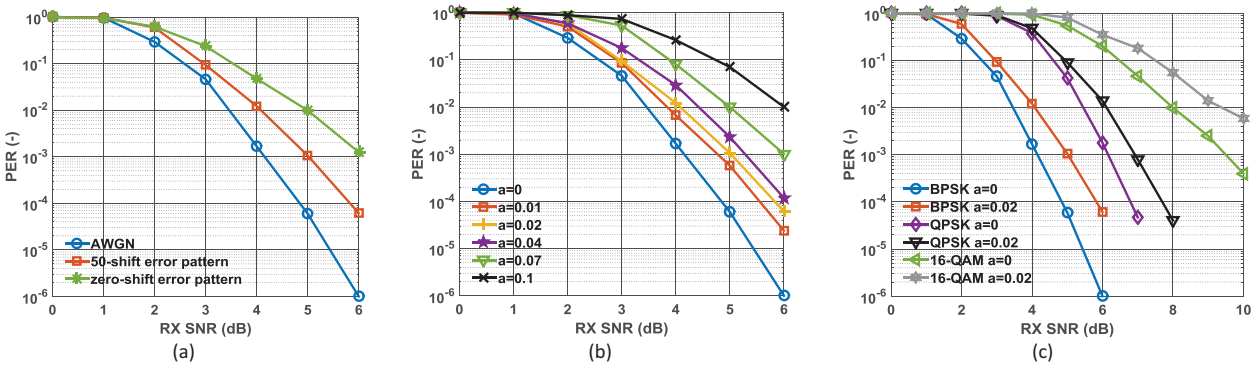


Fig. 3: Matlab simulations of PER vs. SNR for (a) different shift-error patterns for BPSK, (b) different contamination rates for BPSK, and (c) clean and 2% contamination rates for BPSK, QPSK and 16-QAM.

margins, must distort noise distribution *systematically*. Otherwise, the rogue RX will be unable to distinguish the rogue data from channel noise. Therefore, our detection method monitors noise distribution and identifies inconsistencies with respect to the expected noise characteristics of the channel.

As shown in Figure 4, the proposed detection method is implemented at the RX side and employs an additional encoder. The received noisy encoded data, $x$, is first decoded to extract the legitimate message. Such decoding removes noise from the signal. Therefore, if we re-encode the signal, to produce $y$, and subtract it from the noisy signal $x$, we obtain the noise distribution. Assuming an AWGN channel, in a Trojan-free communication the noise distribution will ideally be concentrated around point 0. However, in a Trojan-infected transmission, additional noise components are present around points -2 and 2, whose magnitude depends on the value of the legitimate bits that were substituted by rogue data. For example, when a legitimate bit is flipped from -1 to 1, channel noise profiling will return a value equal to -2.

We note that outliers at points -2 and 2 may also occur due to high channel noise. However, they only correspond to a very small percentage of the additional components. In Figure 4, a high SNR is assumed; thus, additional bins are entirely due to the HT impact. We also point out that the proposed detection method is effective across all modulation schemes, since demodulation occurs before decoding.

## IV. Experimental Results

We employ a WARP-based experimental platform to expose the security risks introduced by HTs in 802.11a/g networks. Two nodes, shown in Figure 5, are used for demonstrating the FEC based attack. One acts as the TX, incorporating both the legitimate and rogue components, while the other acts as the legitimate and rogue RX. All experiments were performed over the air and were repeated 10 times to ensure robustness.

### A. Experimental Platform

A local traffic generator produces a random sequence of legitimate bits, which along with the rogue data is encoded using two FEC encoders. The orthogonal frequency division multiplexing (OFDM) symbols are upconverted to the appropriate frequency band and transmitted through the antenna.

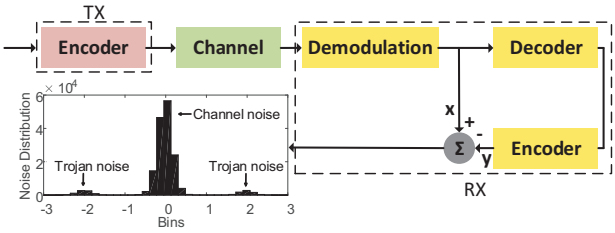


Fig. 4: Channel noise profiling.

### B. Attack Specifics & Impact on Legitimate RX

The experiment runs at the 2.4GHz band with a 20MHz signal bandwidth. The payload for each frame is set at 1400 bytes. Measurements were conducted for transmission power levels from 1dBm to 15dBm with a 1dBm step. In the RX, a Received Signal Strength Indicator (RSSI) is provided by the Python framework for each packet; It is worth mentioning that a noise floor of -89dBm was measured for the BPSK modulation, which is the value provided by the 802.11a/g standard [14]. PER is obtained using the frame check sequence, which indicates if a received packet has been correctly decoded.

Experiments were performed for a clean and a 2% contamination rate with 50-shift error pattern, for BPSK, QPSK, and 16-QAM. The measured results are presented in Figure 6. Despite hardware imperfections, experimental results are consistent with the respective simulation results of Figure 3(c). For example, for BPSK, a PER of $10^{-4}$ is obtained at the cost of an additional 0.8dB of SNR for both simulated and experimental results. Therefore, experiments verify that for a legitimate RX, the HT is inconspicuous, as it essentially appears as an increase in background noise, requiring extra power to correctly decode the original message.

### C. End-to-end Attack Demonstration & Overhead

An end-to-end walk-through of the rogue TX operation and the ability of rogue RX to retrieve the leaked information is given in Figure 7. In this example, 2% of the encoded serialized clean data, shown in Figure 7(a), are replaced by rogue data, shown in Figure 7(e), using the 50-shift error pattern, thereby producing the contaminated data shown in Figure 7(b). A representation of the contamination process is given in Figure 7(f), in which the decimal byte value for 25 consecutive bytes is plotted for both clean and contaminated

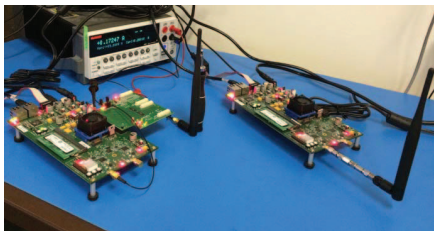

Fig. 5: Experimental setup with WARP boards.

data. The MSB of the first byte, which has a value of "0", is replaced by a rogue bit "1", thereby changing its decimal value from 52 to 180. The next flip occurs in byte 7; however, the contamination process does not modify the corresponding byte value, since the rogue bit is the same as the clean bit. Effectiveness of the rogue RX in correctly decoding the leaked message is presented in Figure 7(c) and Figure 7(d), where the transmitted and received rogue messages are shown, respectively. The 1 byte lag in the RX is due to processing time. Characterization of the rogue RX in terms of error probability versus RX SNR is presented in Figure 7(g) for BPSK, QPSK and 16-QAM. The low PER of the rogue message is due to: (i) small rogue message size, (ii) use of encoding, and (iii) distribution of rogue data across the legitimate packet. The rapid drop in PER confirms the effectiveness of the FEC-based HT even under noisy channel conditions.

In an 802.11 BPSK modulation scheme (i.e., 6Mbps data rate), a 256-bit encryption key requires 4.26msec to be leaked at a contamination rate of as low as 1%. Increasing the contamination rate, while staying within a reasonable range, would allow an attacker to exploit a higher rogue throughput to leak much more data in a short time, such as user profile, instead of just an encryption key.

The overhead introduced by the HT is 27 slice registers and 12 look-up tables, which amounts to a 0.03% increase as compared to the resources required for a clean transmission.

### D. Defense Demonstration

Effectiveness of channel noise profiling is shown in Figure 8, where the detected rogue bits, i.e., the bits corresponding to bins -2 and 2 of Figure 4, are measured for a wide range of SNR values and contamination rates for BPSK, QPSK, and 16-QAM. Experimental results reveal that detection of the HT becomes more probable as SNR and contamination rate increase, since the contribution of the HT to the overall noise becomes more prominent. Presence of interference, e.g. by additional TXs under different modulation schemes, will not reduce effectiveness of noise profiling, since the HT noise positions remain unaffected.

The few inconsistent points of Figure 8, corresponding to high contamination rates (8% - 9%) alongside low SNR (0dB - 1dB), are because of the aggregate impact of HT and channel noise, due to which a limited number of packets are successfully received. This, in turn, reduces the number of detected HT bits. Operating under such conditions, however, is both impractical and risky for the attacker: the rate of leaked information is very low, while the high packet drop rate may alert the legitimate RX.

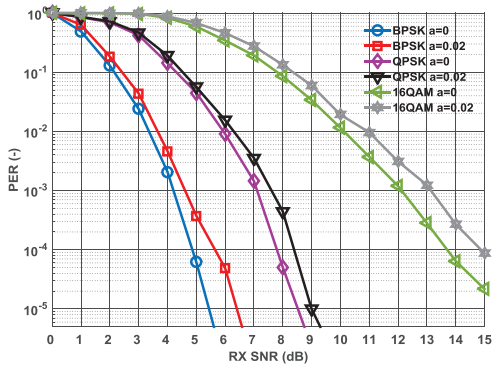


Fig. 6: Experimental PER vs. SNR for WARP.

## V. Related Work

**Covert Channel Attacks in Wireless Networks:** The majority of attacks in wireless networks have focused on exploiting software and firmware vulnerabilities [1]–[3]. Covert channel attacks in WiFi networks, have been mostly studied in theory and simulation, by hiding information in unused frame bits [5], [6], [8], [9]. In [10], four practical covert channels in the PHY layer of 802.11a/g were implemented using WARP. However, depending on the amount of leaked information, these attacks can be detected by traditional tests, such as Carrier Frequency Offset (CFO) and Error Vector Magnitude (EVM). In [11], a covert timing attack was shown in the CSMA/CA protocol, where the spyware leaks information via inter-packet timing. However, the random backoff introduced by the CSMA/CA acts as noise, thereby limiting the throughput of the spyware.

**Hardware Trojans in ICs:** An attack termed MOLES, adding a CDMA-like channel to a crypto-processor to leak information below the noise floor was shown in [15]. HTs in wireless cryptographic ICs, capable of leaking sensitive information (i.e. cryptographic key) by modulating transmission power or frequency have been proposed and demonstrated in silicon [16], [17]. Similarly, the ability of HTs to hide unauthorized signals within the ambient noise floor, using spread spectrum techniques, was shown in [18]. In all these cases, however, attacks were demonstrated on simple wireless links.

**Hardware Trojan Detection:** Statistical side-channel analysis, based on supply current, path delay or power consumption, has been proposed to distinguish Trojan-infected and Trojan-free devices during post-silicon testing [12]. However, effectiveness of such methods in the context of wireless networks may largely depend on (i) SNR, (ii) Trojan-to-circuit ratio, and (iii) availability of Trojan-free chips (golden ICs).

In contrast, the FEC-based attack proposed herein *(i) goes beyond the capabilities of software and firmware, as it is staged in hardware, (ii) has higher rogue throughput, thereby constituting a more serious threat, and (iii) is compliant and demonstrated using complex wireless standards and protocols, rather than simple links.* Respectively, our detection method: *(i) is performed at the RX side, hence its effectiveness cannot be undermined by the attacker, (ii) does not rely on availability of Trojan-free chips, and (iii) is based on general principles and does not assume knowledge of the Trojan attack specifics.*

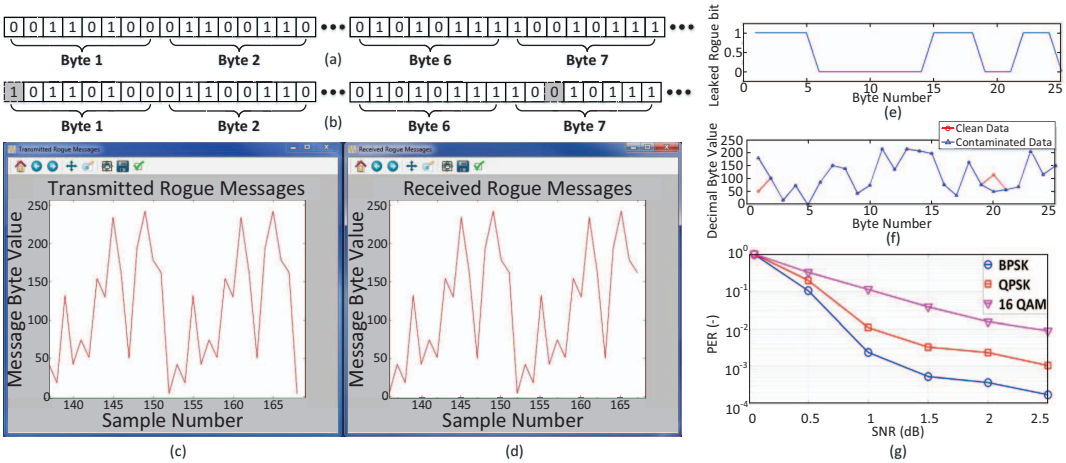


Fig. 7: End-to-end Trojan operation: (a) legitimate data, (b) contaminated data, (c) transmitted rogue message, (d) received rogue message, (e) leaked rogue bits, (f) comparison of clean and contaminated data, and (g) rogue RX characterization.

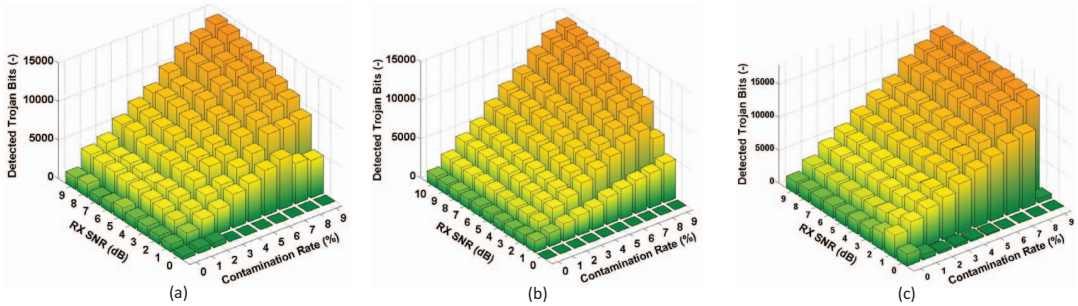


Fig. 8: Effectiveness of the defense mechanism for: (a) BPSK, (b) QPSK, and (c) 16-QAM.

## VI. Conclusion

We devised an HT attack which exploits the error correcting capabilities of the FEC encoder in the baseband circuitry of an 802.11a/g TX. We also introduced a Trojan-agnostic defense wherein channel noise is profiled by the RX to identify systematic deviations due to compromised transmission. As corroborated through attack demonstration on the WARP platform, the concern that HTs can inconspicuously leak sensitive data from a contaminated TX to a rogue RX without disrupting communication with the legitimate RX or being detected is well-justified. Defensive solutions akin to the one proposed herein are, hence, important to wireless network security.

## Acknowledgment

This work is funded by the National Science Foundation under grant 1514050.

## References

[1] 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 *IEEE Wireless Communications and Networking Conference*, vol. 2, 2005, pp. 1193–1199.

[2] Y.-C. Tung, S. Han, D. Chen, and K. G. Shin, "Vulnerability and Protection of Channel State Information in Multiuser MIMO Networks," in *Computer and Communications Security*, 2014, pp. 775–786.

[3] A. Cassola, W. Robertson, E. Kirda, and G. Noubir, "A Practical, Targeted, and Stealthy Attack Against WPA Enterprise Authentication," in *Network and Distributed System Security Symposium*, 2013.

[4] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," *IEEE Communications Surveys Tutorials*, vol. 9, no. 3, pp. 44–57, 2007.

[5] C. Krätzer, J. Dittmann, A. Lang, and T. Kühne, "WLAN Steganography: A First Practical Review," in *Workshop on Multimedia and Security*, 2006, pp. 17–22.

[6] K. Szczypiorski, "A performance analysis of HICCUPS-a steganographic system for WLAN," *Telecommunication Systems*, vol. 49, no. 2, pp. 255–259, 2012.

[7] H. Zhao, "Covert channels in 802.11e wireless networks," in *Wireless Telecommunications Symposium*, 2014, pp. 1–5.

[8] L. Frikha, Z. Trabelsi, and W. El-Hajj, "Implementation of a Covert Channel in the 802.11 Header," in *International Wireless Communications and Mobile Computing Conference*, 2008, pp. 594–599.

[9] K. Szczypiorski and W. Mazurczyk, "Steganography in IEEE 802.11 OFDM symbols," *Security and Communication Networks*, vol. 9, no. 2, pp. 118–129, 2016.

[10] J. Classen, M. Schulz, and M. Hollick, "Practical covert channels for WiFi systems," in *IEEE Conference on Communications and Network Security*, 2015, pp. 209–217.

[11] N. Kiyavash, F. Koushanfar, T. P. Coleman, and M. Rodrigues, "A Timing Channel Spyware for the CSMA/CA Protocol," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 477–487, 2013.

[12] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware Trojans: Lessons learned after one decade of research," *Transactions on Design Automation of Electronic Systems*, vol. 22, no. 1, pp. 6:1–6:23, 2016.

[13] *Wireless Open-Access Research Platform - WARP (http://www.warpproject.org)*.

[14] *IEEE 802.11-2012 Standard for Information Technology (https://standards.ieee.org/findstds/standard/802.11-2012.html)*.

[15] L. Lin, W. Burleson, and C. Paar, "MOLES: Malicious Off-chip Leakage Enabled by Side-channels," in *International Conference on Computer-Aided Design*, 2009, pp. 117–122.

[16] Y. Jin and Y. Makris, "Hardware Trojans in Wireless Cryptographic ICs," *IEEE Design Test of Computers*, vol. 27, no. 1, pp. 26–35, 2010.

[17] Y. Liu, Y. Jin, and Y. Makris, "Hardware Trojans in wireless cryptographic ICs: Silicon demonstration & detection method evaluation," in *International Conference on Computer-Aided Design*, 2013, pp. 399–404.

[18] D. Chang, B. Bakkaloglu, and S. Ozev, "Enabling unauthorized RF transmission below noise floor with no detectable impact on primary communication performance," in *IEEE VLSI Test Symposium*, 2015, pp. 1–4.