Leaking Wireless ICs via Hardware Trojan-Infected Synchronization

Alán Rodrigo Díaz-Rizo, Hassan Aboushady, Haralampos-G. Stratigopoulos

Leaking Wireless ICs via Hardware Trojan-Infected Synchronization

Alán Rodrigo Díaz-Rizo, Hassan Aboushady, *Senior Member, IEEE*, and Haralampos-G. Stratigopoulos, *Member, IEEE*

Abstract—We propose a Hardware Trojan (HT) attack in wireless Integrated Circuits (ICs) that aims at leaking sensitive information within a legitimate transmission. The HT is hidden inside the transmitter modulating the sensitive information into the preamble of each transmitted frame which is used for the synchronization of the transmitter with the receiver. The data leakage does not affect synchronization and is imperceptible by the inconspicuous nominal receiver as it does not incur any performance penalty in the communication. A knowledgeable rogue receiver, however, can recover the data using signal processing that is too expensive and impractical to be used during run-time in nominal receivers. The HT mechanism is designed at circuit-level and is embedded entirely into the digital section of the RF transceiver having a tiny footprint. The proposed HT attack is demonstrated with measurements on a hardware platform. We demonstrate the stealthiness of the attack, i.e., its ability to evade defenses based on testing and run-time monitoring, and the robustness of the attack, i.e., the ability of the rogue receiver to recover the leaked information even under unfavorable channel conditions.

Index Terms—Hardware security and trust, hardware Trojans, wireless integrated circuits, covert communication channel, information leakage.

1 Introduction

A Hardware Trojan (HT) is a malicious modification of the hardware performed by an adversary [1], [2], [3], [4], [5], [6]. HTs are classified according to the insertion phase (i.e., design, fabrication, assembly, post-silicon, etc.), insertion level (i.e., RTL, gate-level, transistor-level, layout, etc.), location on die (i.e., processor, memory, analog, etc.), triggering mechanism (i.e., always-on, activation after some operation time elapses, activation when some rare input condition is met), and payload or effect (i.e., performance degradation, denial-of-service, or leaking of sensitive data such as a cipher key).

From the attacker's perspective, the goal is to design a small footprint and stealthy HT that evades detection. From the defender's perspective, the goal is to prevent HT insertion or detect the presence of a HT, for example with reverse engineering, post-manufacturing testing, or during run-time.

Numerous HT designs have been proposed in the literature, the vast majority of which target digital Integrated Circuits (ICs). The simplest HT is a combinational circuit that monitors a set of nodes to generate a trigger on the simultaneous occurrence of rare node conditions and, subsequently, once the trigger is activated, the payload is simply flipping the value of another node. Another common HT design are the sequential HTs which are triggered with a sequence of conditions and not with a specific state or condition like the combinational HTs. More complex HTs include silicon wearout mechanisms [7], hidden side-channels [8], changing dopant polarity in active areas of transistors [9], siphoning charge from victim wires known as A2 attack [10], [11], activating a row in DRAM to corrupt data in nearby rows known as rowhammer attack [12], exploiting capacitive crosstalk effects [13], leveraging characteristics of emerging Non-Volatile Memories (NVMs) [14], etc.

For analog ICs, HT design is more challenging because analog performance is sensitive to circuit alterations, thus a HT-infected analog IC is likely not to pass testing, and also because analog layouts have few components, thus a HT can be easily detected via reverse engineering and layout inspection. Proposed HT designs for analog ICs include bringing the circuit into an undesired state or operation mode [15], [16], [17], [18], [19] and digital-to-analog HTs that exploit the on-chip test infrastructure [20], [21]. In the latter scenario, the HT resides inside a digital IP where it is triggered. The generated payload is transferred to the victim analog IP via the common test access mechanism and is applied to it via its built-in self-test or programming interface to the test access mechanism. There exist also analog HT designs to infect digital ICs, such as the A2 attack [10], [11]. The A2 attack can be used to infect the digital section of a mixed-signal IC, but it has not been demonstrated inside the analog section.

This paper concerns a HT attack in wireless ICs that implements a covert communication channel aiming at leaking sensitive information from the transmitter within a legitimate signal transmission. A rogue receiver can listen to the transmission to recover the sensitive information, while the legitimate receiver is inconspicuous and does not realize the information leaking. Several studies have demonstrated this type of HT attack. The HT can be embedded within

• *Alán Rodrigo Díaz-Rizo, Hassan Aboushady, and Haralampos-G. Stratigopoulos are with the Sorbonne Université, CNRS, LIP6, 75005 Paris, France.*
E-mail: Alan-Rodrigo.Diaz-Rizo@lip6.fr, hassan.aboushady@lip6.fr, haralampos.stratigopoulos@lip6.fr.
Manuscript received 13 Dec. 2021; revised 11 July 2022; accepted 9 Oct. 2022.
(Corresponding author: Haralampos-G. Stratigopoulos.)
Digital Object Identifier 10.1109/TDSC.2022.XXXXXXX

the Medium Access Control (MAC) protocol [22], within the digital baseband physical layer (PHY) [23], [24], [25], [26], [27], or its payload mechanism can partially act upon the Analog Front-End (AFE) [28], [29], [30], [31], [32]. In parallel, these studies propose defenses for detecting the HT attack at test time or during run-time. The prior art on attack models and defenses will be discussed in more detail in Section 2.

In this paper, we propose a novel and practical HT attack in this context where the HT operates exclusively in the digital baseband PHY. The underlying idea is to modulate leaked data with the preamble part of the transmitted frames which serves for the synchronization of the receiver with the transmitter. By doing so the synchronization still succeeds and the HT is non-intrusive and transparent to the communication link as the transmitted data is not affected. The proposed HT attack is generally applicable to any RF transceiver. We show that it evades defenses at test time or during run-time, including all the known defenses described in Section 2 and a new defense that we propose which acts specifically on the infected preamble. We embed the HT mechanism in the PHY of an RF transceiver implemented in the Software Defined Radio (SDR) bladeRF board from Nuand™ [33]. We show the HT design at circuit-level and we demonstrate with hardware measurements the ability of the HT to evade detection, thus being indistinguishable from normal operation, yet decipherable by the intended rogue receiver. The HT overhead is estimated to be 0.109% of the PHY hardware and the throughput of the covert channel can reach 12 bits per frame. We also present a use case demonstrator where the attacker steals the encryption key and subsequently decrypts a transmitted image message. The demonstrator allows us to study in addition the reliability of the covert channel for different channel conditions. The attacker can receive several repetitions of the noisy key and perform a voting scheme to extract the correct key. Finally, for the purpose of completeness, we review generic HT countermeasures based on pre-silicon prevention and post-silicon detection. Their applicability is related to the threat model, i.e., the HT insertion phase. Evaluating these defenses is an area of future work.

The rest of the article is structured as follows. In Section 2, we discuss the prior art on covert channel attacks for wireless ICs and defenses. In Section 3, we describe the theory of the proposed HT attack, including the threat model, working principle, and applicability. In Section 4, we discuss the HT attack implementation, including the circuit-level design, overhead, rogue receiver design, and throughput of the covert channel. In Section 5, we present the hardware platform and we physically demonstrate the attack with measurements, including its transparency to the legitimate communication, its resilience to test-based and run-time defenses, a demonstrator from the attacker's perspective, and the reliability analysis of the covert channel. In Section 6, we discuss generic HT countermeasures potentially applicable in this context. Section 7 concludes this article.

2 Prior Art on Covert Communication Channels

Table 1 provides a concise summary of existing attack models and corresponding defenses. These are explained below in more detail.

2.1 Attack models

In [22], a spyware is demonstrated exploiting the timing channel resulting from interarrival times of the legitimate transmitted packets. It can be embedded within any MAC protocol that avoids collision in packet retransmissions by using an exponential back-off rule, i.e., the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. In [23], covert messages are hidden within a "dirty" payload data constellation by taking advantage of the I/Q impairments and noisy channel conditions. In [24], four covert channel schemes are shown for the PHY of the IEEE 802.11. These include leaking data by: (a) introducing an additional phase shift in the Short Training Sequence (STS) symbols of the preamble; (b) introducing an artificial Carrier Frequency Offset (CFO) into each Orthogonal Frequency-Division Multiplexing (OFDM) symbol; (c) introducing extra camouflage subcarriers to the OFDM signal; and (d) by replacement of the OFDM Cyclic Prefix (CP). In [27], the attack is staged in the Forward Error Correction (FEC) block, exploiting the fact that the FEC block offers more error correcting capabilities than the channel needs. In [28], [29], [30], the idea is to exploit the margins that exist between the operating point of the circuit and the boundaries defined by the circuit and communication standard specifications. In particular, the HT performs minute modifications in the parameters of the transmitted signal, such as amplitude and frequency, to leak sensitive information from the tampered device. Two HT payload mechanisms are shown in [30], one that uses a single pole double throw switch and a pair of resistors to alter the input termination impedance of the power amplifier, and another one that reprograms the gain stages. In [31], it is proposed to use spread spectrum techniques to hide an unauthorized transmission signal within the legitimate signal below the noise level. In [32], first feasible transmitter impairments that do not affect appreciably Bit Error Rate (BER) are determined, then leaked data are mapped on such artificially introduced impairments. The adversary learns these impairments and extracts the leaked data using deep learning.

2.2 Defenses

Most of the aforementioned studies also examine the resilience to various defenses, oftentimes finding a working defense. Defenses range from standard measurements, i.e., measuring Signal-to-Noise Ratio (SNR), Error Vector Magnitude (EVM) or BER, examining compliance with the spectral mask specifications, analyzing I/Q constellation diagrams, etc., to more elaborate techniques, i.e., Statistical Side-Channel Fingerprinting (SSCF) [29], Adaptive Channel Estimation (ACE) [30], and channel noise profiling. SSCF consists in training a one-class classifier in a feature space composed of parametric measurements, e.g., transmitted power, from golden HT-free devices. The HT-infected devices have a feature vector that lies outside the classification boundary and, thereby, can be distinguished from HT-free devices. The ACE defense leverages the slow-fading characteristics of indoor communication channels to distinguish between channel impairments and HT activity. In [30], it

TABLE 1
HT attack models and defenses.

Ref.	Attack model	Defense mechanism
[22]	Modifies the MAC layer CSMA/CA protocol to leak data into the timings of the transmitted packet sequence.	Evades statistical tests that detect covert timing channels. No other defense is studied.
[23]	Encodes leaked data on the I/Q mapping and hides the encoding by introducing imperfections to the transmitted signal.	Certain tests, such as EVM, show a distinguishing behavior compared to HT-free operation.
[24]-1	Leaks data by introducing an additional phase shift into all STS symbols of the preamble.	Analysis of the preamble constellation.
[24]-2	Leaks data by introducing artificial CFO into each OFDM symbol.	Analysis of CFO changes over time.
[24]-3, [25]	Leaks data in extra camouflage subcarriers added to the OFDM signal.	Decode the signal field to determine if the number of subcarriers is correct.
[24]-4, [26]	Leaks data into parts of the OFDM CP.	Compare the last 16 samples of an OFDM symbol with its CP; spectrum analysis.
[27]	Leaks data by substituting some legitimate data in the FEC block.	Channel noise profiling.
[28], [29], [30]	Leaks data by modulating amplitude and/or frequency of transmitted signal.	SSCF; ACE; Use hardware dithering as a prevention mechanism [34] .
[31]	Leaks data using spread spectrum techniques.	Spectral analysis.
[32]	Leaks data into controlled artificial impairments.	No defenses are studied.
This work	Leaks data through amplitude modulation of some subcarriers in the STS of the preamble.	Evades any known defense for α <15%.

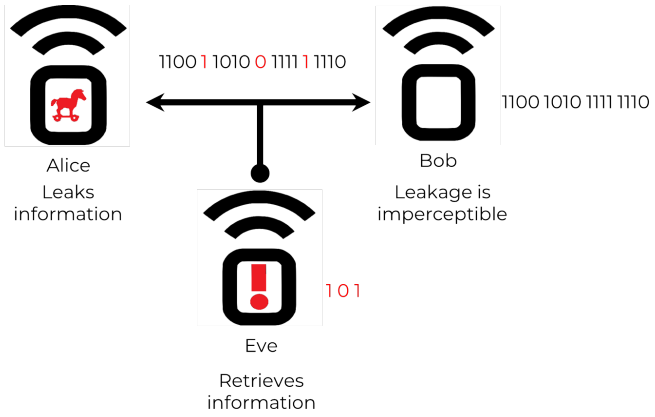


Fig. 1. Threat model.

is claimed that the ACE defense is successful in detecting any HT regardless of the attack specifics. There exist also defenses that are specific to the attack model focusing on the particular encoding of the leaked data. Finally, it is possible to design a proactive defense mechanism that challenges the operation principle of the HT with the aim to neutralize it. An example is the hardware dithering technique proposed in [34].

3 Proposed Attack: Theory

3.1 Threat Model

We consider three wireless ICs as depicted in Fig. 1, namely Alice that has been tampered with by an attacker, Bob that establishes a communication link with Alice and is a legitimate receiver, and Eve that is a rogue receiver having knowledge of the existence of the HT into Alice's transmitter hardware. The HT has been implanted into the baseband of the transmitter of Alice during the design or fabrication phases. Unbeknownst to Alice, while performing an authorized communication with Bob, she is disclosing valuable information to Eve, i.e., the secret cipher key that is thereafter used to decrypt the transmitted data, sensitive data from body sensors or other Internet of Things (IoT) devices or the weights of a proprietary Deep Neural Network (DNN) model. The information is encoded into bits that are well hidden within the transmission signal of Alice.

The attacker can be: (a) a third-party IP (3PIP) provider that delivers the HT-infected digital section of the RF transceiver to the inconspicuous design house which then integrates it together with the AFE; (b) the design house itself that has easy access to perform the malicious modifications; or (c) the foundry which receives the Graphic Database System II (GDSII) file and can insert the HT either by directly modifying the layout or by first reverse engineering the file to extract the circuit netlist and then inserting the HT at the transistor-level or gate-level [35].

3.2 Working principle

In any wireless communication protocol the payload is transmitted along with the PHY specifications. The baseband Digital Signal Processor (DSP) prepares the payload in a frame format for transmission. The Physical layer Protocol Data Unit (PPDU) frame format of an OFDM IEEE 802.11 transmission consists of several OFDM symbols. These symbols are divided into 3 parts, namely preamble (a.k.a. SYNC), header (a.k.a. SIGNAL), and payload (a.k.a. DATA). The preamble section is composed of two different training symbol sequences, namely a STS and a Long Training Sequence (LTS). Fig. 2 shows the PPDU of an IEEE 802.11 transmission with the 3 mentioned parts as defined in the IEEE 802.11 standard [36]. The STS field consists of ten identical short symbol repetitions and it is used for timing acquisition based on the Schmidl and Cox algorithm [37], i.e., to detect the start of the frame, and for coarse CFO

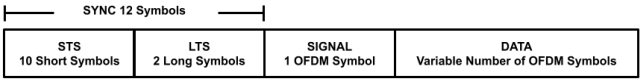


Fig. 2. PPDU frame format of an OFDM IEEE 802.11 transmission.

TABLE 2
STS_F as defined in the IEEE 802.11 standard [36].

Index (k)	Floating-point (I,Q)	Fixed-point (I,Q)
-24, -16, -4, 12, 16, 20, 24	1.4720 , 1.4720	16'h5E35 , 16'h5E35
-20, -12, -8, 4, 8	-1.4720 , -1.4720	16'hA1CA , 16'hA1CA
others	0.0 , 0.0	16'h0000 , 16'h0000

estimation [36]. The LTS field consists of two long symbol repetitions and is used for channel estimation and fine CFO estimation [36].

The STS is described by the IEEE 802.11 standard [36] using its frequency domain representation, denoted by STS_F. STS_F is composed of 64 complex values, i.e., having real (I) and imaginary (Q) components, also called subcarriers or frequency bins, each of 16-bit length, i.e., STS_F has $N = (64 + 64) * 16$ bits = 2048 bits. The 64 subcarriers are indexed from -32 to 31, and there are 12 non-zero subcarriers as shown in Table 2, with the signed decimal floating-point value and hexadecimal fixed-point value using a 16-bit word length shown in the second and third columns of Table 2, respectively.

The STS that is prepended to the frame is derived by performing an Inverse Fast Fourier Transform (IFFT) on the 64 subcarriers composing STS_F, as depicted in Fig. 3. The IFFT generates a time-domain representation of the STS, denoted by STS_t, of the same size as STS_F, composed of 64 I/Q samples, each of 16-bit length. STS_t has a periodicity of 16 samples, i.e., it contains 4 repetitions of 16 samples. The final STS is formed by concatenating two and a half STS_t to obtain the 10 repetitions required by the standard. The transmitter performs those operations at the OFDM modulation block, which contains an IFFT unit, and at the framer block, performing the concatenation operation, as depicted in Fig. 4. The same operation is performed to the frequency-domain representation of the LTS, denoted by LTS_F, to obtain the time-domain version, denoted by LTS_t. The preamble is formed by concatenating the final STS and two and a half LTS_t. The resulting preamble is prepended to any transmitted frame as it is depicted in Fig. 2.

In this work, we propose a HT attack which leaks secret information through the baseband STS_F. Fig. 4 shows a block diagram where sensitive information from outside the baseband processor is driven to the STS generation block, as it is depicted by a red dotted line, where the HT will be implemented. For simplicity, Fig. 4 shows only parts of the transmitter of the wireless IC. This leakage scheme where secret information in one part of the design is driven to another part of the design, i.e., the DSP or AFE, is used in all of the previous works in Table 1. Indeed, in an Application-Specific Integrated Circuit (ASIC) implementation of the RF transceiver where all parts are integrated on the same substrate, sniffing the valuable information from the memory where it is stored and establishing the connection to the

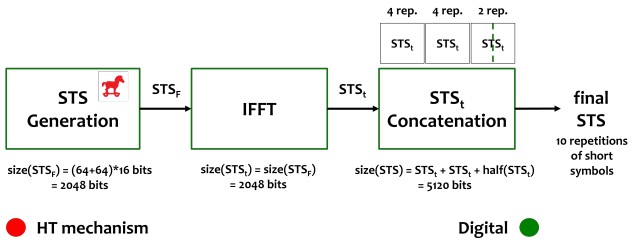


Fig. 3. The detailed STS generation.

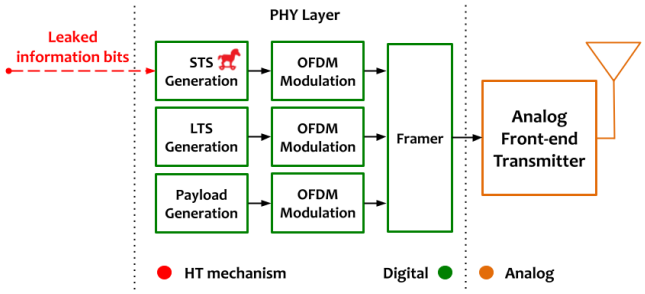


Fig. 4. AM STS HT location within the architecture of a wireless IC, showing only part of the transmitter's sub-blocks.

STS generation block is totally feasible if the attacker is the design house or foundry. More specifically, the proposed HT attack, called *Amplitude Modulation (AM) STS HT attack*, consists in modulating the amplitude of the STS_F subcarriers with the information bits being stolen.

In detail, from the 12 non-zero subcarriers of the STS_F, we choose to leak one byte of information using 8 subcarriers and we use the other 4 subcarriers to set a non-modulated amplitude threshold to reduce the error rate for the rogue receiver. Therefore, in each frame, a byte of the disclosed information is leaked into 8 subcarriers, called *corrupted subcarriers*, with the indexes of these subcarriers being the same for all frames. The amplitude of the corrupted subcarriers is multiplied by $\alpha < 1$, i.e., it is slightly lowered when the leaked bit is '1', otherwise the amplitude is preserved for leaked bits '0'.

Let us consider for example that the 8 corrupted subcarriers have indexes $k = \{-24, -20, -16, -8, 4, 8, 16, 24\}$. The STS_F infected by the AM STS HT is then given in floating-point values by

$STS^{HT}_{F_{-32,31}}$ ={0 0 0 0 0 0 0 0 α'_{-24}(1.472+1.472j) 0 0 0 α'_{-20}(-1.472-1.472j) 0 0 0 α'_{-16}(1.472+1.472j) 0 0 0 -1.472-1.472j 0 0 0 α'_{-8}(-1.472-1.472j) 0 0 0 1.472+1.472j 0 0 0 0 0 0 0 α'_{4}(-1.472-1.472j) 0 0 0 α'_{8}(-1.472-1.472j) 0 0 0 1.472+1.472j 0 0 0 α'_{16}(1.472+1.472j) 0 0 0 1.472+1.472j 0 0 0 α'_{24}(1.472+1.472j) 0 0 0 0 0 0 0}

where $\alpha'_k = \alpha$ if the leaked bit in subcarrier k is 1 and $\alpha'_k = 1$ if the leaked bit in subcarrier k is 0.

At the receiver side, the synchronization process to find the start of the frame consists of performing a cross-correlation operation between the received samples and the ideal OFDM modulated samples of the standard-defined

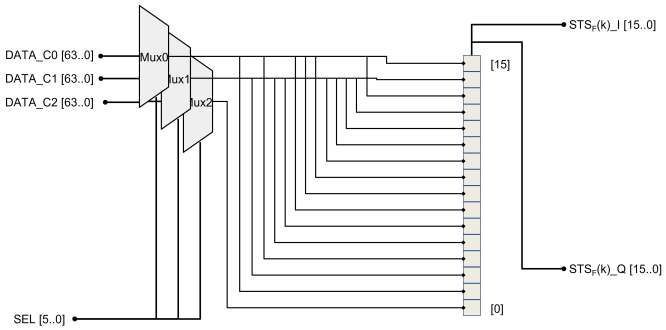


Fig. 5. Circuit schematic of the nominal STS block.

STS_F. Since the synchronization process is done with the received STS_t, the STS_F generation at the transmitter side can be freely modulated by the attacker to leak data under the condition that the correlation properties at the receiver side are kept. Therefore, the amplitude modulation operation in the proposed HT attack, dictated by the choice of α, is performed while ensuring that it will have no effect on the correlation properties. In this way, the synchronization process is not affected and the covert channel is unnoticed by the inconspicuous receiver.

As we will see in Section 5, the magnitude of α represents a trade-off between stealthiness of the AM STS HT attack and effective recovery of leaked data by the rogue receiver. Larger α increases the probability of detection while reducing the probability of error in the recovered data. The attacker can choose a small α to circumvent detection, and exploit several consecutive transmissions of the sensitive bits to perform an error correction scheme.

3.3 Applicability

A synchronization process is present and necessary in any wireless communication protocol. For instance, Wireless Local Area Network (WLAN) IEEE 802.11 (i.e., Wi-Fi), Wireless Personal Area Network (WPAN) IEEE 802.15.1 (i.e., Bluetooth), and Low-Rate Wireless Personal Area Network (LR-WPAN) IEEE 802.15.4 (i.e., Zigbee), use correlation-based synchronization algorithms. Moreover, all of the above standards use a preamble for synchronization, thus the proposed AM STS HT attack is virtually applicable to all of them.

4 Proposed Attack: Implementation

4.1 Circuit-level design

Fig. 5 shows the hardware implementation of the STS block in Fig. 4. The STS block creates the 16-bit fixed-point values for each element of the sequence $\text{STS}_{\text{F}_{-32,31}}$, starting with the element with index $k = 0$ up to $k = 31$, then from $k = -32$ up to $k = -1$. These fixed-point values are reported in Table 2. Note that the real (I) and imaginary (Q) components have the same value. Therefore, for simplicity and without loss of generality, in the following examples we refer only to the real value. Taking as an example the element with index $k = 0$ that has floating-point value 0.0, the STS block has to create the fixed-point value 16'h0000. The SEL input is a 6-bit word and selects the index k of one of the 64 subcarriers to be created. The three multiplexers (MUXes) in Fig. 5 receive a fixed 64-bit value as shown in the upper part of Table 3, where DATA_CX and MuxX denote the input and output of MUX X, respectively. The position of the selected bit transferred at the output of each MUX equals the decimal representation of the SEL input. The output values of the MUXes are then concatenated according to the scheme in the first row in the upper part of Table 4 to create the 16-bit fixed-point value of the selected element of the sequence $\text{STS}_{\text{F}_{-32,31}}$.

For example, for element with index $k = 4$ that has fixed-point value 16'hA1CA in hexadecimal representation, SEL = 6'b000100, bit position $k + 1 = 5$ is selected at the inputs of the MUXes as shown in blue in the upper part of Table 3, and the MUXes output concatenation is as shown in blue in the upper part of Table 4 resulting in the desired value of 16'hA1CA. The same hardware and concatenation operations as shown in the upper part of Table 4 are used to generate any element k by setting the input SEL equal to k in decimal.

Fig. 6 shows the hardware modifications in the STS block to implement the AM STS HT attack. The stolen bits are streamed into the STS block whose output is the 16-bit fixed-point values of the elements of the sequence $\text{STS}^{\text{HT}}_{\text{F}_{-32,31}}$. The attacker needs to define which subcarriers will be corrupted, as well as the amplitude modulation α. Without loss of generality, similarly to Section 3.2, let us assume that 8 subcarriers are corrupted with indexes $k = \{-24, -20, -16, -8, 4, 8, 16, 24\}$. Let us also assume $\alpha = 10\%$. As shown in Fig. 6, the design is modified to add two extra MUXes. The inputs of the five MUXes are shown in the lower part of Table 3. The first two MUXes have constant input values, while MUXes 2-4 have some constant bits and some bits coming from the leaked secret information. The 8 stolen bits per frame are denoted by $\sigma[j]$, $j = 0, \cdots, 7$ in Table 3, where $\sigma[0]$ is the least significant bit and $\neg\sigma[j]$ denotes the inverse of $\sigma[j]$. In this implementation, stolen bits $\{0, \cdots, 7\}$ are mapped to subcarriers with indexes $\{4, 8, 16, 24, -24, -20, -16, -8\}$ in this exact order. When the value of the leaked bit is 0 the corresponding subcarrier according to this mapping has the same amplitude as in the HT-free case. On the other hand, when the value of the leaked bit is 1 the amplitude of the corresponding subcarrier according to this mapping is multiplied by α. The concatenation of the outputs of the MUXes is shown in the first row of the lower part of Table 4.

As an example, let us assume that the leaked byte with the secret information is σ = 8'b00010001. In this scenario, subcarrier with index $k = 4$ will have a modulated amplitude of $0.9 \times (-1.4720) = -1.3248$, subcarrier with index $k = -24$ will have a modulated amplitude of $0.9 \times (1.4720) = 1.3248$, while the rest of the subcarriers will remain at the non-modulated amplitude, i.e., -1.4720 or +1.4720. Let us further consider subcarrier with index $k = 4$. The outputs of the MUXes for SEL input $k+1 = 5$ are shown in red in the lower part of Table 3. The second row of the lower part of Table 4 shows in red the generation of the element of $\text{STS}^{\text{HT}}_{\text{F}}$ with index $k = 4$ after the concatenation operation. Certain bits depend on the leaked bit $\sigma[0]$. The

TABLE 3
Input values of MUXes.

HT-free STS block implementation (Fig. 5)		
MUX	Name	Input (binary value)
Mux0	DATA_C0	64'b0000 0001 0001 0000 0001 0000 0000 0000 0000 0000 0000 0000 0000 0001 0001 0000
Mux1	DATA_C1	64'b0001 0000 0000 0001 0000 0001 0000 0000 0000 0001 0001 0001 0001 0000 0000 0000
Mux2	DATA_C2	64'b0001 0000 0000 0001 0000 0001 0000 0000 0000 0001 0001 0001 0001 0000 0000 0000
HT-Infected STS block implementation (Fig. 6)		
MUX	Name	Input (binary value)
Mux0	DATA_C0	64'b0000 0001 0001 0000 0001 0000 0000 0000 0000 0000 0000 0000 0000 0001 0001 0000
Mux1	DATA_C1	64'b0001 0000 0000 0001 0000 0001 0000 0000 0000 0001 0001 0001 0001 0000 0000 0000
Mux2	DATA_C2	64'b0001 000σ[7] 0000 000$\neg\sigma$[6] 000σ[5] 000$\neg\sigma$[4] 0000 0000 0000 000$\neg\sigma$[3] 0001 000$\neg\sigma$[2] 0001 000σ[1] 000σ[0] 0000
Mux3	DATA_C3	64'b0000 000$\neg\sigma$[7] 0001 000σ[6] 000$\neg\sigma$[5] 000σ[4] 0000 0000 0000 000σ[3] 0000 000σ[2] 0000 000$\neg\sigma$[1] 000$\neg\sigma$[0] 0000
Mux4	DATA_C4	64'b0000 0001 0001 000σ[6] 0001 000σ[4] 0000 0000 0000 000σ[3] 0000 000σ[2] 0000 0001 0001 0000

Note: $\neg\sigma$ is the inverse of σ.

TABLE 4
Concatenation operation of the outputs of the MUXes.

HT-free STS block implementation (Fig. 5)					
	Concatenation of MUXes (M#)	M0,M1,M0,M1	M1,M1,M1,M0	M0,M0,M1,M1	M0,M1,M0,M2
Subcarriers (k): -24, -16, -4, 12, 16, 20, 24	Fixed-point value Hexadecimal value	0101 5	1110 E	0011 3	0101 5
Subcarriers (k): -20, -12, -8, 4, 8	Fixed-point value Hexadecimal value	1010 A	0001 1	1100 C	1010 A
Subcarriers (k): others	Fixed-point value Hexadecimal value	0000 0	0000 0	0000 0	0000 0
HT-infected STS block implementation (Fig. 6)					
	Concatenation of MUXes (M#)	M0,M1,M0,M1	M2,M1,M2,M0	M3,M3,M2,M2	M3,M2,M4,M2
Concatenated output of subcarrier $k = 4$	Fixed-point value	**1010**	σ[0] **0** σ[0] **1**	$\neg\sigma$[0]$\neg\sigma$[0]σ[0]σ[0]	$\neg\sigma$[0]σ[0] **1** σ[0]
Leaked bit $\sigma[0] = 0$	Fixed-point value Floating point value	1010	0001	1100 -1.4720	1010
Leaked bit $\sigma[0] = 1$	Fixed-point value Floating point value	1010	1011	0011 -1.3248	0111

Note: $\neg\sigma$ is the inverse of σ.

third and fourth rows of the lower part of Table 4 show the 16-bit fixed-point and floating-point values in the case where $\sigma[0] = 0$ and $\sigma[0] = 1$, respectively. For $\sigma[0] = 0$ the concatenated output value remains at -1.4720, whereas for $\sigma[0] = 1$ the concatenated output value is reduced to -1.3248, i.e., 10% below the standard.

The HT-infected STS block implementation shown in Fig. 6 resulted from a synthesis using the software Quartus II 16.0 from Intel™. It should be noted that the implementation is not unique and different implementations can result using different optimization levels of synthesis.

Moreover, the implementation of the PHY of the IEEE 802.11 standard can vary depending on available resources or developer preferences, as long as it still complies with the standard. Herein, we showed the HT design in the case where STS_t is re-computed for every frame. Alternatively, the STS_t may be computed once, then stored and reused for each frame. In this scenario, the HT mechanism would modulate the stored STS_t samples to implement the information leakage.

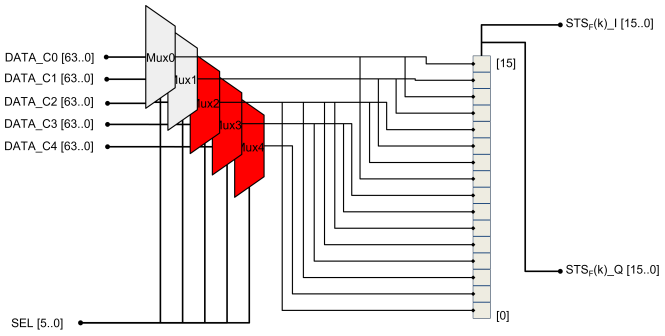


Fig. 6. Circuit schematic of the HT-infected STS block.

4.2 Overhead

To calculate the HT overhead and prove the low footprint of the proposed HT attack, we utilized as base HT-free implementation an open-source IEEE 802.11 compatible SDR VHDL modem [38]. The project is called *bladeRF-wiphy* as it implements the PHY of the IEEE 802.11 in the FPGA of the bladeRF board [33]. More details about the bladeRF board will be given in Section 5.1. Starting from the HT-free implementation, we made the modifications of Section 4.1 to incorporate the HT into the PHY of the modem and we re-synthesized the project using Quartus II 16.0 from Intel™ to find the resultant overhead. The HT design requires the addition of 58 Adaptive Look-Up Tables (ALUTs) which translates into a negligible increase of 0.109% in the total

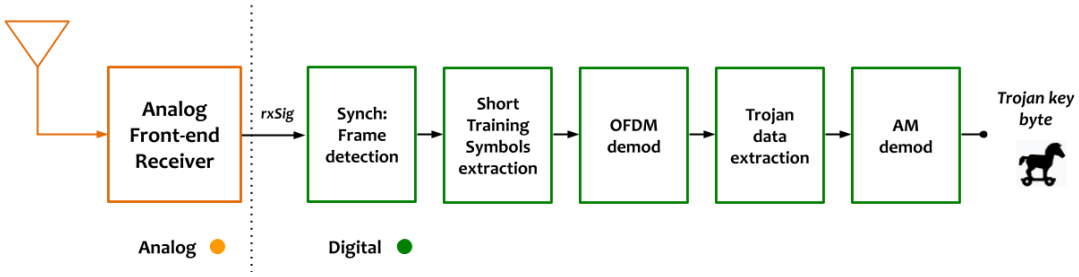


Fig. 7. Rogue receiver block architecture for the AM STS HT attack.

number of required ALUTs to implement the entire PHY of the modem. According to Intel, the physical resources represented by an ALUT differ depending on the device family. For example, in Stratix V, Arria V and Cyclone V devices, there is one combinational LUT and 2 registers per ALUT.

4.3 Rogue receiver

To extract the leaked data from the received signal, after synchronization the rogue receiver needs to demodulate the OFDM symbols containing the STS_t, extract the values stored in the corrupted subcarriers, and demodulate the AM signal. The hardware needed to retrieve the leaked data is depicted in Fig. 7.

The above process is not performed in the nominal receiver since after finding the start of the frame the STS_t is no longer processed. In Section 5, we investigate different defenses aiming to spot the HT activity. One of them, namely the STS constellation test, looks for anomalies in the constellation of the demodulated STS_t, i.e., it emulates some of the stages performed by the rogue receiver. Although this defense may be able to detect HT attacks that act on the STS, it is not cost-effective to be implemented into every nominal receiver because the purpose-specific hardware needed to achieve this would incur a large area, power, and delay overhead. In addition, the nominal receiver does not know which subcarriers are corrupted and which are affected only by noise, therefore a very high SNR would be required to detect any anomalies. We will return to this point in our experimental results in Section 5.

4.4 Throughput of the covert channel

In the proposed HT attack, the throughput of the covert channel can reach 12 bits per frame (bpf), whereas the number of transmitted frames per second (fps), denoted by n_{fps}, depends on the length of the transmitted frames. There are three types of IEEE 802.11 frames, namely management, control, and data. The second layer of the Open Systems Interconnection (OSI) model, namely the data link layer, defines the number of bytes contained in each of the different IEEE 802.11 frame types. All types include a PHY preamble, meaning that every transmitted frame can carry up to 12 bits of the covert channel payload. Therefore, the throughput of the covert channel, denoted by Th_{HT}, expressed in bits per second (bps) is given by $Th_{HT} = 12 \cdot n_{fps}$. For instance, short control frames, such as the Acknowledge (ACK) and Clear to Send (CTS), have a frame length of 16 µs, which implies a frame rate of 62,500 fps, and thus a throughput of 750,000 bps. Longer control frames, such as the Request to Send (RTS) or variable-length data frames, reduce the number of fps according to their frame duration, thus the throughput of the covert channel is also reduced.

5 Measurement Results

5.1 Hardware Platform

To demonstrate the proposed AM STS HT attack we use the SDR bladeRF board from Nuand™ [33]. This board contains three main chips: an RF transceiver LMS6002 from Lime Microsystems™, a Field-Programmable Gate Array (FPGA) Cyclone IV from Intel™ (formerly ALTERA™), and a USB 3.0 peripheral controller FX3 from Cypress™. The RF transceiver has on-chip baseband and RF loopback modes allowing us to perform measurements using the same board. For our measurements we employ the baseband loopback mode and we model the communication channel with Additive White Gaussian Noise (AWGN). Fig. 8 shows a block diagram architecture of the test setup used for the experiments. It shows the interaction between the baseband DSP and the AFE, as well as the loopback modes. The HT attack mechanism at the transmitter side is placed in the preamble generation depicted by a Trojan horse. The defense mechanisms at the receiver side were implemented in the baseband at the synchronization process during runtime or at test time and are depicted by a crossed-out Trojan horse.

5.2 Transparency to legitimate communication: choice of α

The AM STS HT attack leaks data through the STS, thus it affects the coarse CFO estimation. However, the standard uses the LTS, which is left intact by the AM STS HT attack, for fine CFO estimation and correction. The hypothesis is that there exists a value of α below which the introduced CFO due to the AM STS HT attack can still be compensated, thus making the AM STS HT attack transparent to the legitimate communication. In essence, while the inconspicuous legitimate receiver successfully synchronizes and thereafter discards the preamble, the rogue receiver who knows the leaking mechanism processes further the preamble to retrieve the leaked data.

To demonstrate that our hypothesis is valid, we studied the impact of different α values on the constellation diagram of the received decoded payload signal and on the BER performance. We performed these measurements

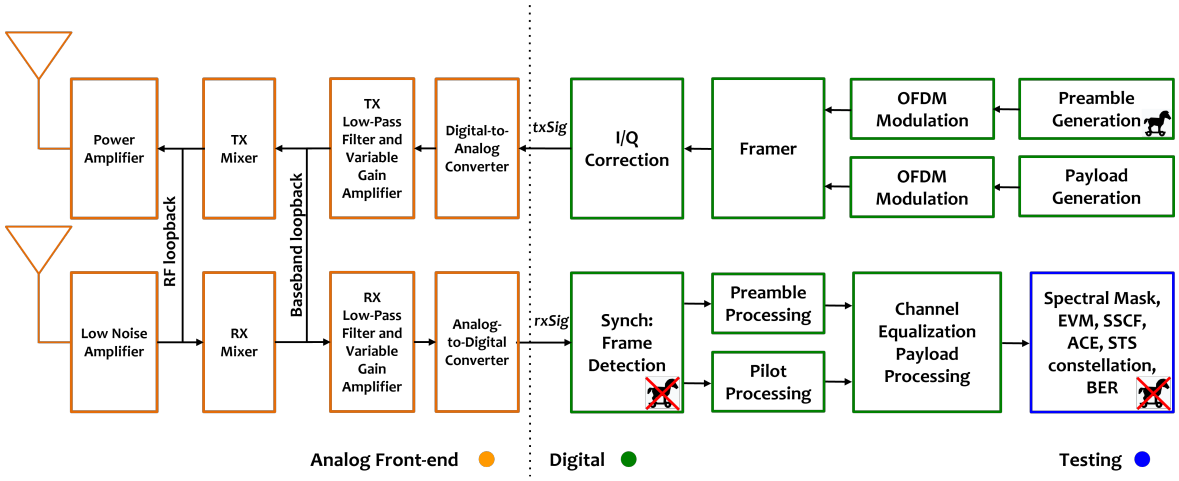
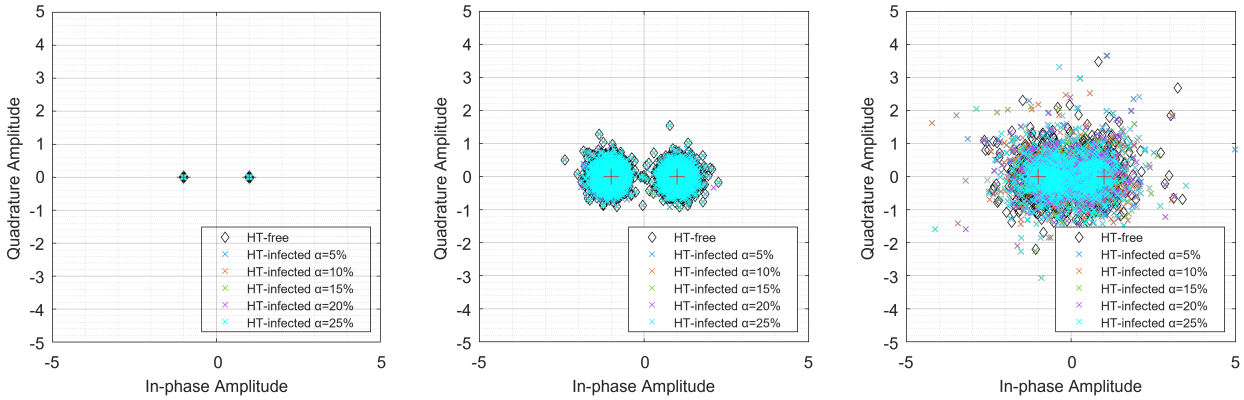


Fig. 8. RF transceiver test setup using the loopback mode. The Trojan horse shows the stage of the attack and the crossed-out Trojan horse show the stages of the defense mechanisms.



(a) SNR=40 dB. (b) SNR=10 dB. (c) SNR=2 dB.

Fig. 9. Constellation diagrams of the decoded payload of the received OFDM-BPSK signal for different SNR values. The result is shown for an HT-free device and an AM STS HT-infected device using different values of α.

using an OFDM transmission using Binary Phase Shift Keying (BPSK) modulation for different values of SNR for an HT-free device and an AM STS HT-infected device using different values of α.

Fig. 9 shows the constellation diagrams of the received payload. Visually they are indistinguishable between an HT-free and an AM STS HT-infected device regardless of the value of α. Fig. 10, however, shows that for α greater than 15% the AM STS HT-infected device presents a degraded BER for SNR greater than 2 dB that distinguishes it from the HT-free device. In conclusion, the proposed AM STS HT attack does not have any impact on either the synchronization or the performance of legitimate communication when the value of α is chosen to be less than 15%.

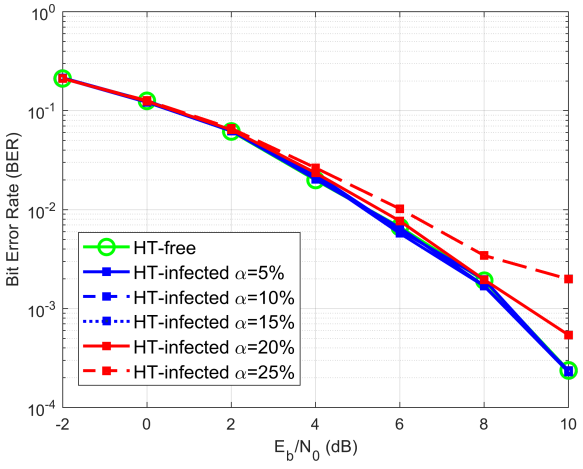


Fig. 10. Measured BER of the received OFDM-BPSK signal using an HT-free device and an AM STS HT-infected device for different values of α.

5.3 Resilience to test-based and run-time defenses

The proposed AM STS HT attack was tested against known defences aiming at detecting HT-infected chips at post-manufacturing test or HT activity during run-time. For all the following experiments, unless explicitly mentioned, we use $\alpha = 10\%$.

5.3.1 *Spectral mask test*

We performed spectral measurements to analyze whether transmissions with an AM STS HT-infected device and

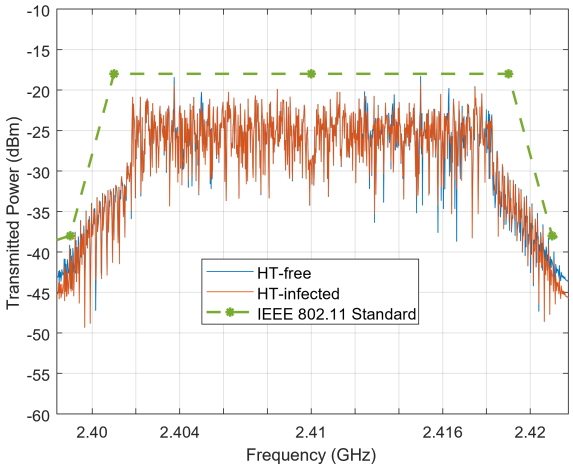


Fig. 11. Measured PSD of signals transmitted with an AM STS HT-infected device and an HT-free device, along with the IEEE 802.11 standard spectral mask specification [36].

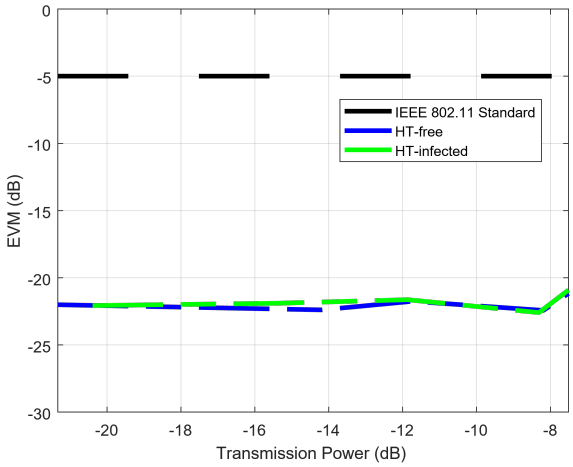


Fig. 12. Measured EVM of the transmitted BPSK signal from an AM STS HT-infected device and an HT-free device, along with the IEEE 802.11 standard EVM specification [36].

transmissions with an HT-free device were distinguishable. The Power Spectral Density (PSD) of the transmitted signals are shown in Fig. 11. The signals are centered at the carrier frequency $2.41\,\mathrm{GHz}$ and they occupy a $20\,\mathrm{MHz}$ bandwidth. Along with the signals, the spectral mask margins specified by the IEEE 802.11 standard are shown [36]. As it can be seen, the PSD of the HT-infected device is indistinguishable from that of the HT-free device and is standard-compliant. Note that there is no pre-processing that can be performed on the transmitted data to make the two PSD curves distinguishable. Any subtle difference is due to the channel noise and RF impairments and not due to the HT activity.

5.3.2 EVM test

Fig. 12 shows the EVM measurements at different transmission power levels for an OFDM-BPSK transmission with an AM STS HT-infected device and an HT-free device. As it can be seen, the EVM measurements are compliant with the IEEE 802.11 standard [36] and there is no impact on EVM due to HT insertion. It should be noted that the variability between transmissions from the HT-infected device and the HT-free device is due to electronic noise and hardware impairments in the SDR circuitry and not due to the HT activity. This is because the HT activity takes place in the preamble and the EVM test detects payload divergence from the ideal symbols of a target constellation.

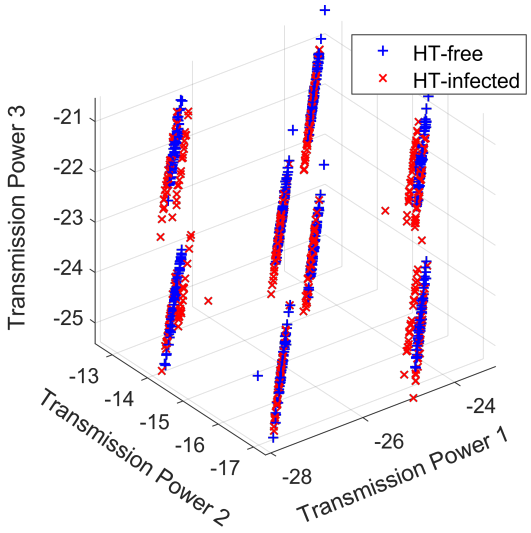
5.3.3 SSCF test

To detect the HT presence using SSCF, we generated two populations with 1000 HT-free devices and 1000 HT-infected devices using 2000 different random combinations of I/Q gain and phase imbalance. For the gain imbalance we used 5% variability and for the phase imbalance 10% variability. To vary I/Q imbalance we used the I/Q correction module in the DSP. Then, for each device we transmitted the same information at six different power levels in the range from $-27\,\mathrm{dBm}$ to $-14\,\mathrm{dBm}$ and considered as feature the total received power at the receiver through the baseband loopback of the transceiver. We used half of the HT-free devices to train a one-class Support Vector Machine (SVM) classifier and the other half were used as an independent validation set.

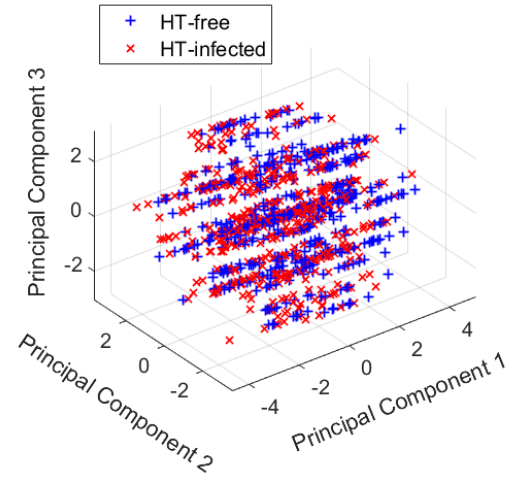
The SVM shows a very poor accuracy to correctly classify the HT-infected devices, with a misclassification rate of 44.5%, essentially pointing to a random decision. Fig. 13a shows a 3D plot of the first 3 transmission power levels and Fig. 13b shows a 3D plot of the first 3 principal components resulting from a Principal Component Analysis (PCA). In both cases, the 2 populations are overlapped and, thereby, the HT-infected devices cannot be screened out. Due to the total overlapping, there is no classifier that can separate the two classes, i.e., using other popular classifiers such as a decision tree or a deep feed-forward neural network produces the same result.

5.3.4 ACE test

The ACE test is capable of discriminating between signal variations due to channel impairments and HT activity only in the case where the HT activity is generated in the AFE. If the HT is infecting the baseband of the transmitter, i.e., the preamble of the signal, as is the case for the proposed AM STS HT attack, the ACE test will not detect the HT activity. To demonstrate this, Fig. 14 shows three ACE tests with the resulting post-channel equalization of the received payload. The signal amplitude is indicated by color variations, and the color bar on the right-hand side of the plots shows the color used to represent a given amplitude value. The x-axis shows the received payload divided in OFDM symbols and the y-axis shows the subcarrier indexes k. Fig. 14a shows an HT-free transmission where the received amplitude of the payload remains constant along the OFDM symbols. Fig. 14b presents an example of HT activity hidden in the amplitude modulation of the payload, like the approach presented in [30]. As it can be seen, the HT increases the amplitude of the received signal for a certain amount of time. This amplitude modulation corresponds to a leaked bit. In particular, the dark columns correspond to a leaked bit 1, whereas the lack of darkness corresponds to leaked bit 0. In this case, the ACE defense spots the leaked data, e.g. in Fig. 14b the leaked message is '01001011110'. In contrast,

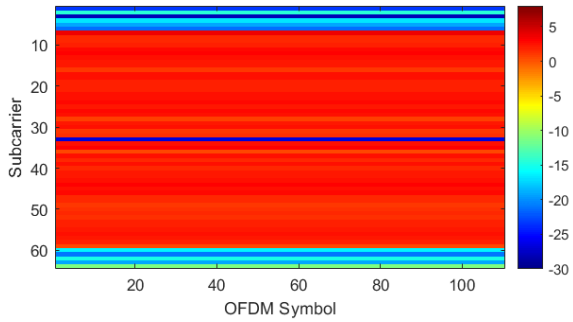


(a) Projection of HT-free and HT-infected devices onto a 3D space composed of the first 3 transmission power levels.

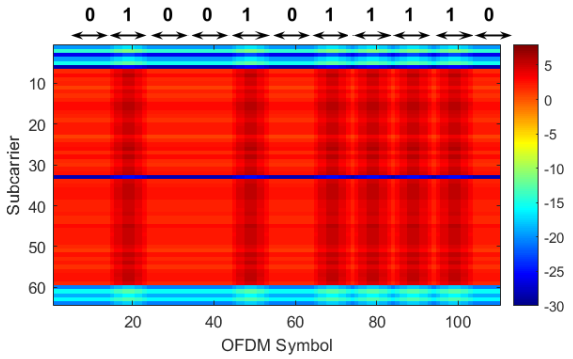


(b) Projection of HT-free and HT-infected devices onto a 3D space composed of the first three principal components.

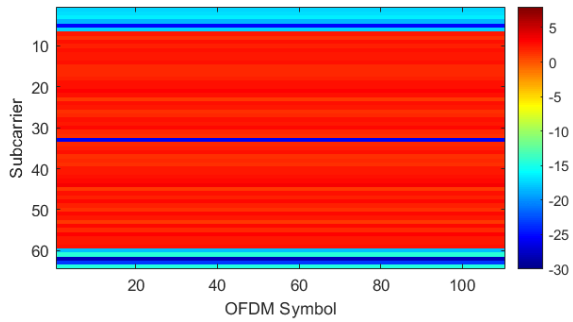
Fig. 13. Measurement results from the SSCF test.



(a) ACE test for an HT-free transmission.



(b) ACE test for an HT-infected transmission based on amplitude modulation at the AFE, like the approach presented in [30]. The HT activity on the payload is identified by the ACE test.



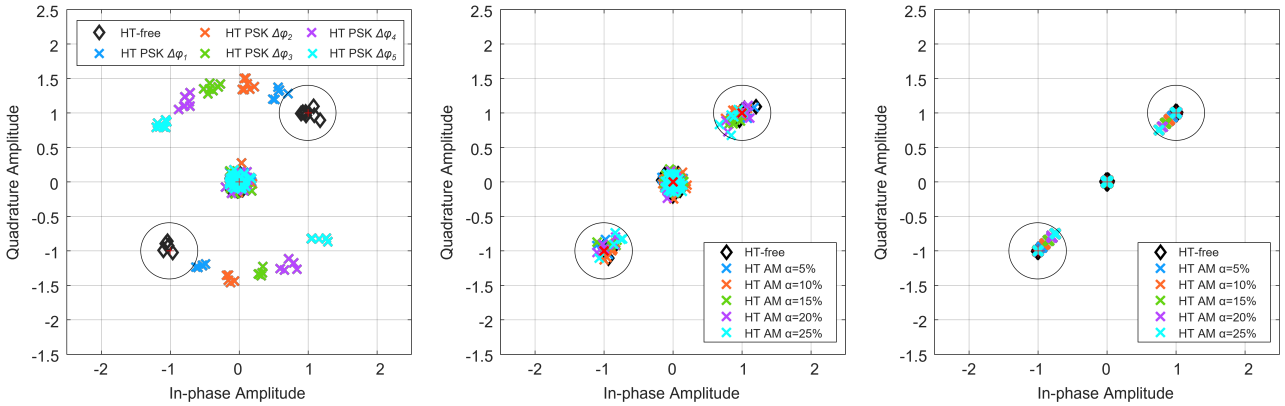
(c) ACE test for an HT-infected transmission with the AM STS HT. The HT activity on the STS cannot be identified by the ACE test.

Fig. 14. Measurement results from the post-channel equalization of the received payload.

the ACE test is unable to detect the HT activity in the case of the proposed AM STS HT attack, as shown in Fig. 14c. When the HT is hidden in the STS of the transmitted signal, Fig. 14c does not point to any HT activity despite the fact that the covert channel is enabled.

5.3.5 STS constellation test

Another possible defense for detecting an STS anomaly is to demodulate the preamble, extract the STS_F, and observe its constellation. In a HT-free transmission, the constellation is composed of only 3 symbols, namely {-1-j , 0, 1+j}. If the observed constellation points can be distinguished from these expected 3 symbols, then the attack is deemed detected. For this defense, we compare the proposed AM STS HT attack with the attack proposed in [24], which also acts upon the STS in the preamble but implements the information leakage differently. In particular, the STS (alternatively called STF in [24]) contains BPSK symbols that are shifted by 45°. In [24], the data are leaked by introducing an additional phase shift $\Delta\phi$ into all STS symbols. This attack proposed in [24] is called PSK STF HT attack. This defense detects the PSK STF HT attack, whereas the proposed AM STS HT attack bypasses it successfully. This is demonstrated in Fig. 15. More specifically, Fig. 15a shows the resulting constellation for a 20 dB SNR and for 5 different phase shifts using the PSK STF HT attack in [24], whereas Fig. 15b shows the resulting constellation for the proposed AM STS HT attack for a 20 dB SNR and for 5 different amplitude modulations from $\alpha = 5\%$ to $\alpha = 25\%$. The black circles around the coordinates -1-j and 1+j show where the STS symbols should be found ideally in the absence of a HT, along with the points at zero which are the subcarriers with zero amplitude value. In Fig. 15a most of the points are outside the circles making the PSK STF HT attack [24] easily noticeable, while



(a) Constellation of the STS_F preamble using the PSK STF HT attack [24].

(b) Constellation of the STS_F preamble using the proposed AM STS HT attack.

(c) Constellation of the STS_F preamble using the proposed AM STS HT attack at a high SNR scenario.

Fig. 15. Measurement results from the STS constellation test.

in Fig. 15b all the points are inside the circles even for a large amplitude modulation of $\alpha = 25\%$ making the AM STS HT attack practically undetectable. To detect the AM STS HT attack a very high SNR would be required and a close examination of the constellation. Fig. 15c shows the constellation diagram of the STS_F preamble in such a scenario having $40\,\mathrm{dB}$ of SNR. To avoid being detected, an attacker must choose lower α values.

It should be noted that this STS constellation analysis is not cost-effective to be implemented as a run-time defense in every wireless receiver since specialised equipment is needed to perform it.

5.3.6 *Single-branch STS_t correlation test*

We propose a new less complex and low-cost run-time defense compared to the STS constellation test, which relies on comparing each individual I/Q branch of the stored nominal STS_t against the corresponding I/Q branch of the received STS_t. This defense mechanism is placed in the synchronization process as depicted in Fig. 8 by a crossed-out Trojan horse in the synch block.

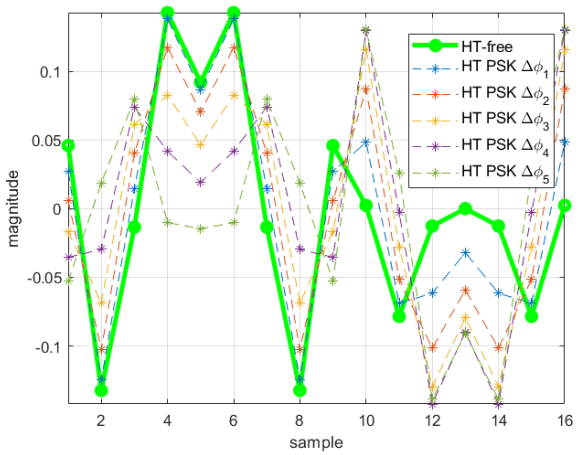
The wireless receiver has access to the STS_t digital I/Q samples with which it performs the correlation operation to search for the start of the frame. STS_t corruption as a result of phase shifting due to the PSK STF HT attack [24] causes the I/Q branches to be unbalanced. To visualize the effect of a phase shifting in the STS_t, Fig. 16a shows a time domain comparison of the I branch of the HT-free STS_t against 5 different phase shifts. As it can be seen, the samples differ from the HT-free STS_t. Therefore, this proposed defense mechanism consists of performing a correlation operation between the first 16 samples of the I branch of the received STS_t and the first 16 stored samples of the I branch of the ideal STS_t. If the maximum value of the correlation is not found in the index 16, then the frame is infected by a HT. To strengthen the defense, the same correlation analysis can be repeated for the Q branch, and if any of the two correlation operations fails then a HT is detected.

Although this low-cost and practical defense is capable of detecting the PSK STF HT attack in [24], as shown in Fig. 16a, the proposed AM STS HT attack still evades this defense, as shown in Fig. 16b for 5 different amplitude modulations.

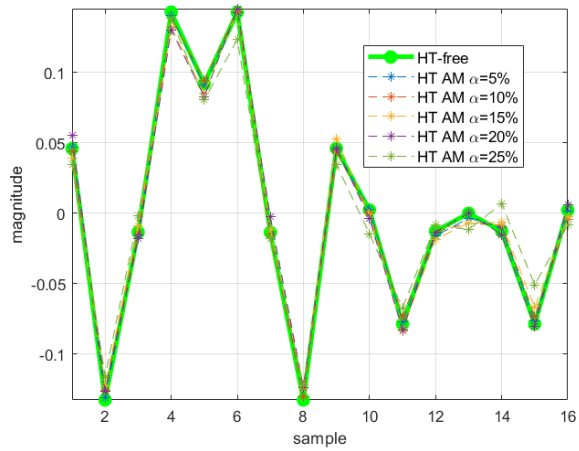
5.4 Demonstrator

To demonstrate the HT functionality from the attacker's perspective, we performed an encrypted transmission using the HT-infected transmitter of Fig. 4, where the leaked information is the encryption key from an Advanced Encryption Standard (AES) core, and we attempted to decrypt the payload using the recovered leaked secret key. As payload, we transmitted the well-known standard 512 x 512 pixel color version test image of a mandrill (a.k.a. baboon). The payload was processed, converted to plaintext, and finally encrypted using the AES algorithm with a 128-bit secret key. According to the threat model of Section 3.1, the demonstrator consists of Alice sending the encrypted image to Bob, while without her knowledge the encryption key is leaked via the corrupted subcarriers of the STS_F according to the proposed AM STS HT attack in Section 3.2. Eve who is the eavesdropper retrieves the stolen key from the HT-infected STS as described in Section 4.3 and tries to decrypt the cyphertext to reveal the message, i.e., the mandrill image in this case. As shown in Fig. 17, the received encrypted payload in Fig. 17a is decrypted correctly in Fig. 17b when applying the recovered leaked key. Section 5.5 details how the leaked key is recovered even at low SNR scenarios.

Considering a wireless IC, where the encryption key is stored in a Tamper-Proof Memory (TPM), there will be no key updates unless more keys are stored in the TPM, which increases the size and complexity of the TPM. In addition, AES is a symmetric cipher, that is, if the transmitter refreshes the key then the key at the receiver must also be updated, leading to an increase in the complexity of the communication system. Therefore, we consider that an update of the encryption key is very unlikely in our scenario. However, even in the scenario of a key update, the number of frames used to transmit a payload while the key is constant is greater than the number of frames required to extract the key from the preamble of those frames. Thus, the rogue receiver could store the frames encrypted by some secret key, de-embed the key from the preambles of those



(a) Real (I) part of the HT-free STS_t and the received STS_t for different phase shifts in the PSK STF HT attack [24].



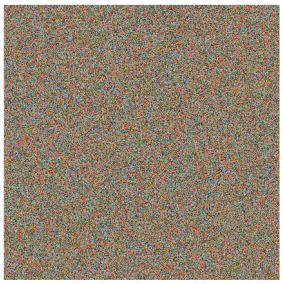
(b) Real (I) part of the HT-free STS_t and the received STS_t for different amplitude values in the proposed AM STS HT attack.

Fig. 16. Measurement results from the single-branch STS_t correlation test.

frames, and decrypt the payload. Since the HT mechanism continuously leaks the key, changes in the register used to store the key would be reflected in the preamble of the new frames to be transmitted, and the rogue receiver could repeat the procedure described above for a new secret key.

5.5 Reliability of the covert channel

Assuming a 128-bit secret key and choosing to leak one byte per frame, i.e., only 8 out of the 12 non-zero subcarriers of the STS_F are corrupted, then after 16 frames the secret key would be completely transmitted. Starting from the 17th frame, the secret key will be repeated for the duration of the transmission creating redundancy of the leaked data. This redundancy allows the attacker to apply an error correction algorithm to recover the secret key even under low SNR conditions. We conducted an experiment to evaluate the resilience of the proposed technique in various SNR scenarios and for various HT amplitude modulations α. We used a simple voting system in which, after having some redundancy, the value of each bit is chosen as the one that is repeated the most. For example, after 48 frames,



(a) Received encrypted payload.

(b) Decrypted payload using the recovered key leaked through the corrupted subcarriers of STS_F.

Fig. 17. Demonstration of the AM STS HT attack: stealing the cypher key and recovering the transmitted encrypted mandrill (a.k.a baboon) image.

the secret key has been repeated 3 times and we will have a redundancy of 3 for each of the 128 bits of the secret key. According to the voting system, the value of each bit is determined as the one that has been received at least twice. Then, after each key iteration we aimed at decrypting the ciphertext using the received key. If the ciphertext is decrypted correctly, then the recovered key is the correct one. Fig. 18 shows the results obtained from our experimentation. Each point is an average of 6 measurements to account for the channel noise. It can be observed that for larger α the number of iterations needed to recover the secret key without any error is lower. For example, for $\alpha = 10\%$, which is sufficient to thwart the single-branch STS_t correlation defense as shown from Fig. 16b, and for SNR values greater than 24 dB only one series of 16 frames is needed to obtain the secret key without errors, while for an SNR of 15 dB it is required to wait for 6 repetitions of the secret key, i.e., 96 frames in total. As it can be seen from Fig. 18, for the lowest unfavorable SNR of 15 dB and the lowest α of 5% that results in minute and imperceptible preamble deviations, 20 repetitions are needed. Finally, since only 8 out 12 subcarriers are corrupted, the attacker can use the other 4 not corrupted subcarriers to tune a threshold amplitude value to demodulate the leaked data and further reduce the error rate.

6 Related Prevention and Detection Defense Mechanisms

In Section 5.3 we demonstrated experimentally that the proposed HT evades all known detection defenses at test time or during run-time. The covert channel is undetected when the nominal receiver is post-processing the HT-infected signal. There exist, however, several other generic HT countermeasures based on insertion prevention and pre-silicon or post-silicon detection that are in principle applicable under different threat model scenarios. The proposed HT relies on performing small malicious modifications in the digital section of the RF transceiver, i.e., DSP and IP core from which the information is being leaked.

If the attack is staged by an EDA tool provider or if the digital section of the RF transceiver is a 3PIP core, then the owner of the RF transceiver can check for the presence of the HT prior to fabrication using: (a) functional verification

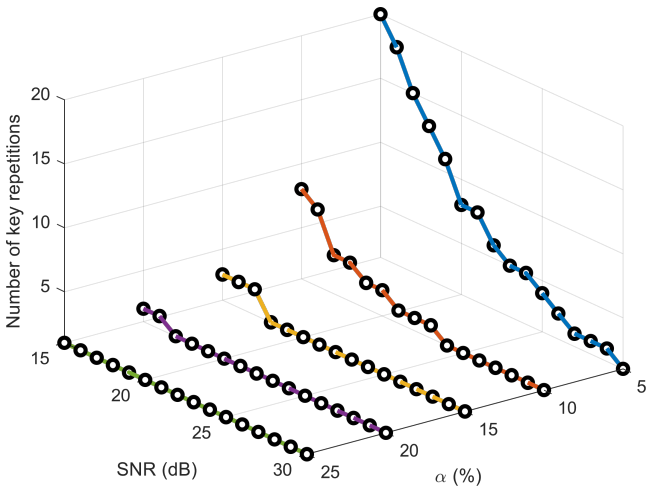


Fig. 18. Reliability test for five different modulation amplitudes α under different SNR scenarios.

of the 3PIP cores [39]; (b) structural analysis of Hardware Description Language (HDL) codes [39]; (c) logic testing tools [35], [40], [41], [42], [43], [44]; (d) specific simulation benches, i.e., performing aging simulations along with overclocking [45] or short-term aging [46] to magnify the effect of the HT without triggering it; (e) search methods for unused components during design-time verification, which thereafter can be removed as potentially suspicious [47]; and (f) Information Flow Tracking (IFT) methods that track the propagation of sensitive data and verify that they do not reach unauthorized sites in the design [11], [48], [49], [50], [51], [52]. In our case, IFT could be used to spot the connection between the register in the IP core where the information is stored and the preamble generation block in the DSP.

If the attack is staged by the foundry, pre-silicon prevention methods include: (a) filling in all unused spaces on the layout, which are most likely insertion areas for the HT, with functional filler cells and checking if those have changed [53]; and (b) design obfuscation, for example using locking [54], [55], [56], [57], [58], [59], [60], camouflaging [61], [62], [63], or split manufacturing [64], [65], aiming at obscuring the circuit functionality so as to make it difficult for the attacker to insert the HT.

The test-based and run-time defenses discussed in Section 5.3 are post-silicon HT detection methods. Other post-silicon HT detection methods include: (a) destructive reverse-engineering, which involves de-packaging and delayering the chip, imaging the chip's layers, and using software to stitch together the prepared images, thereby recovering the layout and netlist, which thereafter can be carefully examined to detect the presence of HTs [66], [67]; (b) non-destructive side-channel analysis to expose the HT location, for example using optical circuit analysis [68], electromagnetic emanation (EM) measurements [69], [70], [71], thermal map analysis [72], backscattering [73], and laser probing [74]; and (c) using on-chip monitors, i.e., current sensors [75], thermal sensors [76], and invariance checkers [77], for run-time HT detection.

All the aforementioned defenses should be evaluated in the context of the proposed HT and can be the subject of future work.

7 Conclusions

We proposed the novel AM STS HT attack for leaking sensitive data out of wireless ICs. The AM STS HT attack acts on the preamble of a transmission frame, hiding the data into the transmission part that is used only for system synchronization, thereby not affecting the communication. The HT mechanism itself is hidden inside the dense digital baseband of the transmitter having a tiny footprint with 0.109% ALUTs overhead for an FPGA implementation. The proposed attack is stealthy being completely transparent to the normal RF transceiver operation. The leaked data can be recovered only by the intended rogue receiver using an inverse operation that has prohibitively high-cost and is impractical to perform during run-time on every regular receiver. An indicative throughput of the established covert channel is 750 kbps. We demonstrated with hardware measurements using the SDR bladeRF board from Nuand™ that the proposed AM STS HT attack is capable of evading any previously reported defense either at run-time or based on performance testing. We also proposed a novel low-cost run-time correlation-based defense to detect HT activity hidden in the synchronization data. This defense also falls short in revealing the proposed AM STS HT attack. Our experiments demonstrate the strength of the proposed AM STS HT attack calling for the development of a specific practical defense so as to ensure the security of wireless communications. To this end, we discussed several known generic HT countermeasures that could be potentially applicable and should be further evaluated. Finally, we demonstrated the AM STS HT attack from the attacker's perspective where an encrypted message with a 128-bit secret key is being leaked. We analyzed the reliability of the covert channel and we demonstrated that the key can be successfully recovered after less than 10 key transmissions even in the most unfavorable SNR scenario.

Acknowledgements

This work was supported by the ANR STEALTH project under Grant N° ANR-17-CE24-0022-01. The work of A. R. Díaz-Rizo was supported by the Mexican National Council for Science and Technology (CONACYT) through Fellowship.

References

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

[2] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Computer*, vol. 43, no. 10, pp. 39–46, Oct. 2010.

[3] 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, Jul. 2014.

[4] 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. Autom. Electron. Syst.*, vol. 22, no. 1, pp. 6:1–6:23, Dec. 2016.

[5] S. Bhunia and M. M. Tehranipoor (Eds.), *The Hardware Trojan War: Attacks, Myths, and Defenses*, Springer International Publishing, 2018.

[6] A. Jain, Z. Zhou, and U. Guin, "Survey of recent developments for hardware trojan detection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021.

[7] Y. Shiyanovskii, F. Wolff, A. Rajendran, C. Papachristou, D. Weyer, and W. Clay, "Process reliability based trojans through NBTI and HCI effects," in *NASA/ESA Conf. Adapt. Hardw. Syst.*, Jun. 2010, pp. 215–222.

[8] L. Lin, T. Güneysu M. Kasper, C. Paar, and W. Burleson, *Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering*, Berlin, Germany: Springer, 2009.

[9] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware trojans: Extended version," *J. Cryptograph. Eng.*, vol. 4, no. 1, pp. 19–31, Apr. 2014.

[10] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: analog malicious hardware," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 18–37.

[11] X. Guo, H. Zhu, Y. Jin, and X. Zhang, "When capacitors attack: Formal method driven design and detection of charge-domain trojans," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 1727–1732.

[12] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors," in *ACM/IEEE Int. Symp. Comput. Archit.*, Jun. 2014, pp. 361–372.

[13] C. Kison, O. M. Awad, M. Fyrbiak, and C. Paar, "Security implications of intentional capacitive crosstalk," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3246–3258, Dec. 2019.

[14] K. Nagarajan, M. N. I. Khan, and S. Ghosh, "ENTT: A family of emerging NVM-based trojan triggers," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, May 2019.

[15] Z. Liu, Y. Li, Y. Duan, R. L. Geiger, and D. Chen, "Identification and break of positive feedback loops in trojan states vulnerable circuits," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 289–292.

[16] X. Cao, Q. Wang, R. L. Geiger, and D. J. Chen, "A hardware trojan embedded in the inverse widlar reference generator," in *Proc. IEEE 58th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2015.

[17] Q. Wang, R. L. Geiger, and D. Chen, "Hardware trojans embedded in the dynamic operation of analog and mixed-signal circuits," in *Proc. Nat. Aerosp. Electron. Conf. (NAECON)*, Jun. 2015, pp. 155–158.

[18] C. Cai and D. Chen, "Performance enhancement induced trojan states in op-amps, their detection and removal," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 3020–3023.

[19] Q. Wang, D. Chen, and R. L. Geiger, "Transparent side channel trigger mechanism on analog circuits with PAAST hardware trojans," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018.

[20] M. Elshamy, G. Di Natale, A. Pavlidis, M. Louërat, and H.-G. Stratigopoulos, "Hardware Trojan attacks in analog/mixed-signal ICs via the test access mechanism," in *Proc. IEEE Eur. Test Symp. (ETS)*, May 2020.

[21] M. Elshamy *et al.*, "Digital-to-analog hardware Trojan attacks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 2, pp. 573–586, Feb. 2022.

[22] 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.

[23] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "Secret agent radio: Covert communication through dirty constellations," in *Information Hiding*, M. Kirchner and D. Ghosal, Eds., Berlin, Heidelberg, 2013, pp. 160–175, Springer Berlin Heidelberg.

[24] 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.

[25] Z. Hijaz and V. S. Frost, "Exploiting OFDM systems for covert communication," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct./Nov. 2010, pp. 2149–2155.

[26] S. Grabski and K. Szczypiorski, "Steganography in OFDM symbols of fast IEEE 802.11n networks," in *Proc. IEEE Secur. Priv. Workshops*, May 2013, pp. 158–164.

[27] K. S. Subraman, A. Antonopoulos, A. A. Abotabl, A. Nosratinia, and Y. Makris, "Demonstrating and mitigating the risk of an FEC-based hardware trojan in wireless networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2720–2734, Feb. 2019.

[28] Y. Jin and Y. Makris, "Hardware trojans in wireless cryptographic ICs," *IEEE Design Test Comput.*, vol. 27, no. 1, pp. 26–35, Jan./Feb. 2010.

[29] 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.

[30] K. S. Subramani, N. Helal, A. Antonopoulos, A. Nosratinia, and Y. Makris, "Amplitude-modulating analog/RF hardware trojans in wireless networks: Risks and remedies," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3497–3510, Apr. 2020.

[31] S. Chang, G. Bhat, U. Ogras, B. Bakkaloglu, and S. Ozev, "Detection mechanisms for unauthorized wireless transmissions," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 23, no. 6, pp. 70:1–70:21, Nov. 2018.

[32] K. Sankhe *et al.*, "Impairment shift keying: Covert signaling by deep learning of controlled radio imperfections," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 598–603.

[33] Nuand, "SDR bladeRF 2.0 micro xA9," https://bit.ly/3z2QV1N, Online.

[34] C. Kapatsori, Y. Liu, A. Antonopoulos, and Y. Makris, "Hardware dithering: A run-time method for trojan neutralization in wireless cryptographic ICs," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct./Nov. 2018.

[35] M. Fyrbiak *et al.*, "HAL—the missing piece of the puzzle for hardware reverse engineering, trojan detection and insertion," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 3, pp. 498–510, Mar. 2019.

[36] IEEE, "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 Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016.

[37] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, Dec. 1997.

[38] Nuand, "Open-source ieee 802.11 compatible software defined radio vhdl modem (bladeRF-wiphy)," https://github.com/Nuand/bladeRF-wiphy/, Online.

[39] X. Zhang and M. Tehranipoor, "Case study: Detecting hardware trojans in third-party digital IP cores," in *IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, Jan. 2011, pp. 67–70.

[40] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, *MERO: A Statistical Approach for Hardware Trojan Detection*, Berlin, Germany: Springer, 2009.

[41] A. Waksman, M. Suozzo, and S. Sethumadhavan, "FANCI: Identification of stealthy malicious logic using boolean functional analysis," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, Nov. 2013, p. 697–708.

[42] H. Salmani, "COTD: Reference-free hardware trojan detection and recovery based on controllability and observability in gate-level netlist," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 338–350, Feb. 2017.

[43] M. A. Nourian, M. Fazeli, and D. Hely, "Hardware trojan detection using an advised genetic algorithm based logic testing," *J. Electron. Test.: Theory Appl.*, vol. 34, no. 4, pp. 461–470, Aug. 2018.

[44] S. K. Haider, C. Jin, M. Ahmad, D. M. Shila, O. Khan, and M. van Dijk, "Advancing the state-of-the-art in hardware trojans detection," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 18–32, Jan./Feb. 2019.

[45] V. R. Surabhi, P. Krishnamurthy, H. Amrouch, K. Basu, J. Henkel, R. Karri, and F. Khorrami, "Hardware trojan detection using controlled circuit aging," *IEEE Access*, vol. 8, pp. 77415–77434, Apr. 2020.

[46] V. R. Surabhi, P. Krishnamurthy, H. Amrouch, J. Henkel, R. Karri, and F. Khorrami, "Exposing hardware trojans in embedded platforms via short-term aging," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3519–3530, Nov. 2020.

[47] M. Hicks, M. Finnicum, S. T. King, M. M. K. Martin, and J. M. Smith, "Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 159–172.

[48] A. C. Myers and B. Liskov, "A decentralized model for information flow control," in *Proc. 16th ACM Symp. Operating Syst. Princ. (SOSP)*, Oct. 1997, p. 129–142.

[49] X. Li *et al.*, "Sapper: A language for hardware-level security policy enforcement," *Proc. 19th Int. Conf. Archit. Support Program. Lang. Operating Syst. (ASPLOS)*, vol. 42, no. 1, pp. 97–112, Feb. 2014.

[50] Y. Jin, X. Guo, R. G. Dutta, M.-M. Bidmeshki, and Y. Makris, "Data secrecy protection through information flow tracking in proof-carrying hardware IP—part I: Framework fundamentals," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2416–2429, Oct. 2017.

[51] M-M. Bidmeshki, X. Guo, R. G. Dutta, Y. Jin, and Y. Makris, "Data secrecy protection through information flow tracking in proof-carrying hardware IP—part II: Framework automation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2430–2443, Oct. 2017.

[52] M. M. Bidmeshki, A. Antonopoulos, and Y. Makris, "Proof-carrying hardware-based information flow tracking in analog/mixed-signal designs," *IEEE J. Emerging Selected Topics Circuits Syst.*, vol. 11, no. 2, pp. 415–427, Jun. 2021.

[53] K. Xiao, D. Forte, and M. Tehranipoor, "A novel built-in self-authentication technique to prevent inserting hardware trojans," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 12, pp. 1778–1791, Dec. 2014.

[54] J. A. Roy, F. Koushanfar, and I. L. Markov, "Ending piracy of integrated circuits," *Computer*, vol. 43, no. 10, pp. 30–38, Oct. 2010.

[55] K. Shamsi, M. Li, K. Plaks, S. Fazzari, D. Z. Pan, and Y. Jin, "IP protection and supply chain security through logic obfuscation: A systematic overview," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 24, no. 6, pp. 65:1–65:36, Sep. 2019.

[56] A. Chakraborty *et al.*, "Keynote: A disquisition on logic locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 1952–1972, Oct. 2020.

[57] J. Leonhard *et al.*, "Digitally-assisted mixed-signal circuit security," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 8, pp. 2449–2462, Aug. 2021.

[58] M. Elshamy, A. Sayed, M.-M. Louërat, H. Aboushady, and H.-G. Stratigopoulos, "Locking by untuning: A lock-less approach for analog and mixed-signal IC security," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 12, pp. 2130–2142, Dec. 2021.

[59] A. R. Díaz-Rizo, J. Leonhard, H. Aboushady, and H. Stratigopoulos, "RF transceiver security against piracy attacks," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 69, no. 7, pp. 3169–3173, Jul. 2022.

[60] A. R. Díaz-Rizo, H. Aboushady, and H.-G. Stratigopoulos, "Anti-piracy design of RF transceivers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, 2022, early access.

[61] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Nov. 2013, pp. 709–720.

[62] A. Vijayakumar, V. C. Patil, D. E. Holcomb, C. Paar, and S. Kundu, "Physical design obfuscation of hardware: A comprehensive investigation of device and logic-level techniques," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 64 – 77, Jan. 2017.

[63] J. Leonhard, A. Sayed, M.-M. Louërat, H. Aboushady, and H.-G. Stratigopoulos, "Analog and mixed-signal IC security via sizing camouflaging," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 5, pp. 822–835, Jul. 2021.

[64] Y. Wang, P. Chen, J. Hu, G. Li, and J. Rajendran, "The cat and mouse in split manufacturing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 5, pp. 805–817, May 2018.

[65] T. D. Perez and S. Pagliarini, "A survey on split manufacturing: Attacks, defenses, and challenges," *IEEE Access*, vol. 8, pp. 184013–184035, Oct. 2020.

[66] T. Sugawara *et al.*, "Reversing stealthy dopant-level circuits," *J. Cryptograph. Eng.*, vol. 5, no. 2, pp. 85–94, Jun. 2015.

[67] B. Lippmann *et al.*, "Integrated flow for reverse engineering of nanoscale technologies," in *Proc. 24th Asia and South Pacific Design Automat. Conf.*, Jan. 2019, p. 82–89.

[68] F. Stellari, P. Song, A. J. Weger, J. Culp, A. Herbert, and D. Pfeiffer, "Verification of untrusted chips using trusted layout and emission measurements," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, May 2014, pp. 19–24.

[69] O. Söll, T. Korak, M. Muehlberghuber, and M. Hutter, "EM-based detection of hardware trojans on FPGAs," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, May 2014, pp. 84–87.

[70] X. T. Ngo, Z. Najm, S. Bhasin, S. Guilley, and J.-L. Danger, "Method taking into account process dispersion to detect hardware trojan horse by side-channel analysis," *J. Cryptograph. Eng.*, vol. 6, no. 3, pp. 239–247, Sep. 2016.

[71] J. He, Y. Zhao, X. Guo, and Y. Jin, "Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2939–2948, Oct. 2017.

[72] Y. Tang, S. Li, L. Fang, X. Hu, and J. Chen, "Golden-chip-free hardware trojan detection through quiescent thermal maps," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2872–2883, Dec. 2019.

[73] L. N. Nguyen, C.-L. Cheng, M. Prvulovic, and A. Zajić, "Creating a backscattering side channel to enable detection of dormant hardware trojans," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 7, pp. 1561–1574, Jul. 2019.

[74] A. Stern, D. Mehta, S. Tajik, U. Guin, F. Farahmandi, and M. Tehranipoor, "SPARTA-COTS: A laser probing approach for sequential trojan detection in COTS integrated circuits," in *IEEE Phys. Assur. Insp. Electron. (PAINE)*, Dec. 2020.

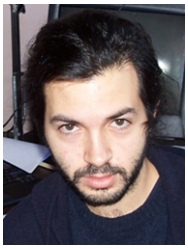
[75] S. Narasimhan, W. Yueh, X. Wang, S. Mukhopadhyay, and S. Bhunia, "Improving IC security against trojan attacks through integration of security monitors," *IEEE Design Test Comput.*, vol. 29, no. 5, pp. 37–46, Sep./Oct. 2012.

[76] D. Forte, C. Bao, and A. Srivastava, "Temperature tracking: An innovative run-time approach for hardware trojan detection," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2013, pp. 532–539.

[77] A. Pavlidis, E. Faehn, M.-M. M. Louërat, and H.-G. Stratigopoulos, "Run-time hardware trojan detection in analog and mixed-signal ICs," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2022.

Alán Rodrigo Díaz Rizo received the B.Sc. degree in electronics and communication engineering from the University of Guadalajara, Guadalajara, Mexico, in 2015, and the M.Sc. degree in electrical engineering from the Center for Research and Advanced Studies of the National Polytechnic Institute (Cinvestav), Mexico, in 2018. He is currently pursuing the Ph.D. degree with the Computer Science Laboratory (LIP6), Sorbonne Université, Paris, France. His research interests include hardware security, cognitive radio, and radio signal processing.

Hassan Aboushady (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from Cairo University, Egypt, in 1993, and the M.Sc. and Ph.D. degrees in electrical engineering and computer science from Sorbonne Université, Paris, France, in 1996 and 2002, respectively. He is currently an Associate Professor with Sorbonne Université. His research interests include sigma-delta modulation, analog/RF circuit design, analog-to-digital and digital-to-analog conversion, and security in analog and mixed-signal circuits. He is the author and coauthor of more than 70 publications in these areas. He is a member of the IEEE CIRCUITS AND SYSTEMS FOR COMMUNICATIONS COMMITTEE (CASCOM). He was a recipient of the Best Paper Award from the IEEE Design Automation and Test in Europe Conference in 2004 and a recipient and a co-recipient of the second and third Best Student Paper Awards from the IEEE Midwest Symposium on Circuits and Systems in 2000 and 2003, respectively. He also served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS. He is an IEEE-CAS Distinguished Lecturer.



Haralampos-G. Stratigopoulos (Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2001, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 2006. He was a Researcher at the TIMA Laboratory, French National Center for Scientific Research (CNRS), Université Grenoble Alpes, Grenoble, France. He is currently a Research Director at the LIP6 Laboratory, CNRS, Sorbonne Université, Paris, France. His main research interests include hardware security, neuromorphic computing, design-for-test for analog, mixed-signal, and RF circuits and systems. He was the General Chair of the 2015 IEEE International Mixed-Signal Testing Workshop (IMSTW) and the 2021 and 2022 AI Hardware: Test, Reliability and Security (AI-TREATS) Workshop and the Program Chair of the 2017 IEEE European Test Symposium (ETS). He has served on the Technical Program Committees for the Design, Automation, and Test in Europe Conference (DATE), Design Automation Conference (DAC), IEEE International Conference on Computer-Aided Design (ICCAD), IEEE European Test Symposium (ETS), IEEE International Test Conference (ITC), IEEE VLSI Test Symposium (VTS), and several others international conferences. He has also served as an Associate Editor for IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I:REGULAR PAPERS, IEEE DESIGN AND TEST, and Journal of Electronic Testing: Theory and Applications (Springer).