Hardware Trojan Detection in Analog/RF Integrated Circuits

Yier Jin, Dzmitry Maliuk, and Yiorgos Makris

Abstract Globalization of semiconductor manufacturing has brought about increasing concerns regarding possible infiltration of the Integrated Circuit (IC) supply chain by skilled and resourceful adversaries, with the intention of introducing malicious modifications (a.k.a hardware Trojans) which can be exploited to cause incorrect results, steal sensitive data, or even incapacitate a chip. While numerous prevention and detection solutions have been introduced in the recent, past, the vast majority of these efforts target digital circuits. Analog/RF ICs, however, are equally vulnerable and potentially even more attractive as attack targets, due to their wireless communication capabilities. Accordingly, in this chapter, we review existing research efforts in hardware Trojan detection in Analog/RF ICs. Specifically, using a wireless cryptographic IC as an experimentation platform, we demonstrate the effectiveness of side-channel fingerprinting along with advanced statistical analysis and machine learning methods in detecting hardware Trojans both after its manufacturing and after its deployment in its field of operation.

1 Introduction

The problem of maliciously intended modifications (a.k.a. hardware Trojans) in manufactured integrated circuits (ICs) has recently become of interest not only to academic researchers but also to governmental agencies and industrial entities [4]. Partly because of design outsourcing and migration of fabrication foundries to low-cost areas across the globe, and partly because of increased reliance on external

Yier Jin
University of Central Florida, Orlando, FL, USA, e-mail: yier.jin@eecs.ucf.edu

Dzmitry Maliuk
Yale University, New Haven, CT, USA, e-mail: dzmitry.maliuk@gmail.com

Yiorgos Makris
University of Texas at Dallas, Richardson, TX, USA, e-mail: yiorgos.makris@utdallas.edu

hardware intellectual property (IP) and Electronic Design Automation (EDA) software from various vendors, the integrated circuit supply chain is now considered far more vulnerable to such malicious modifications than ever before. Fears that skillful and resourceful adversaries may be able to compromise some stage of IC design and/or fabrication and insert Trojan hardware are becoming increasingly intense, as rumors about actual occurrence of such cases surface [4]. In essence, the fundamental concern is that hardware Trojan-infested chips may be capable of additional functionality which is unknown to the designer/vendor/customer and which can be exploited by the perpetrator after chip deployment. Evidently, depending on the field of application, the consequences of such attacks may range from minor inconvenience to major catastrophic events, especially since the intended target of such dubious ICs will most likely be a sensitive domain, such as financial, military, or other vital infrastructure.

While the severity of the potential implications of such a threat has fueled several research efforts towards better understanding and dealing with hardware Trojans both at the pre-silicon [39, 24, 22, 27, 18, 16, 15, 14] and at the post-silicon [34, 33, 40, 35, 20, 26, 25, 8, 10, 7, 6, 12, 9, 31, 32, 36] stage, the vast majority of these efforts target traditional digital circuits. In contrast, this chapter will focus on the problem of hardware Trojans in the analog/RF domain and will also introduce hardware Trojan detection methods for wireless ICs. Similar to digital circuits, analog/RF ICs are now prevalent in electronic systems, facilitating industrial control and wireless communication and becoming an inseparable part of modern everyday activities. At the same time, analog/RF ICs (and, by extension, the integrated systems containing analog/RF modules) are particularly vulnerable and constitute a very appealing target for hardware Trojan attacks; indeed, since such circuits receive and transmit information over public wireless channels, the attacker does not need to obtain physical access to their input/output space, making such attacks far more realistic. Moreover, most modern communication systems employ some form of encryption in order to protect the privacy of the information that is communicated over the 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 (e.g. the encryption key) is stored on these devices. Therefore, development of pertinent Trojan hardware mitigation methods for analog/RF ICs is equally (if not more) critical as with their digital counterparts.

Toward this end, this chapter studies the threat of hardware Trojans specifically within the context of analog/RF ICs and examines remedies to ensure their trustworthiness both during the manufacturing testing process and after their deployment in their field of operation. Through the material introduced in this chapter we seek to achieve the following objectives:

- Delineate the threat and potential impact of hardware Trojans in analog/RF ICs. Specifically, we will focus on vulnerability introduced by the margins that are typically allowed in the transmission parameters in order to deal with process variations and we will show that these margins can be exploited in order to gain control of a chip and/or leak sensitive information. The trade-off between the level of harm that these hardware Trojans may incur and the impact on

area/power/performance, which is strongly correlated to their detection susceptibility, will also be investigated.
- Elucidate the shortcomings of existing test methods in exposing hardware Trojans in the analog/RF IC domains. Since analog/RF Trojans do not change the functionality of the chip, they are very difficult to be detected by traditional manufacturing testing methods. The effectiveness of existing hardware Trojan detection methods introduced in the digital domain will also be investigated.
- Devise efficient hardware Trojan detection methods based on statistical analysis and machine learning, specifically for analog/RF ICs. The effect of a carefully designed hardware Trojan is expected to be hidden within the parametric design margins, making side channel information of a Trojan-infested chip appear perfectly legitimate if examined in isolation. However, for the hardware Trojan to be of utility to the attacker, it needs to impose some form of structure in the transmission signals and/or other side-channel signals, through which remote commands will be issued or secret information will be leaked. Statistical analysis methods can therefore be used to detect the existence of this added structure and machine learning (i.e. trained classifiers) can be used to distinguish between Trojan-free and Trojan-infested chip populations.

2 Hardware Trojans in Wireless ICs

Using as an experimentation vehicle a simple wireless cryptographic circuit and two example hardware Trojans which were specifically designed to attack wireless ICs [21], we will demonstrate the following three key findings:

- **Attack Complexity:** Minor modifications to a wireless cryptographic chip suffice to leak secret information. The vulnerability of such chips stems partly from the fact that they transmit over a public wireless channel. Their true Achilles heel, however, is the fundamentally analog nature of a wireless transmission, which entails several continuous parameters (e.g. amplitude, frequency, phase, etc.). In order to tolerate variations due to fabrication process and/or operating conditions, specifications for these parameters are defined as windows of acceptable performances rather than exact values. As a result, a hardware Trojan can hide additional information within the tolerance margins of such continuous entities and secretly transmit it. While such transmissions abide by all specifications and appear to be perfectly legitimate, an adversary who knows the structure of the additional information will be able to extract it.
- **Detection Difficulty:** Evading detection by traditional manufacturing test methods is trivial. The functionality of the digital part of the chip in normal operation mode and in test mode can be preserved despite the addition of the hardware Trojan; hence, no structural (i.e. scan-based) or functional tests (or even enhanced functional tests for hardware Trojan-detection) will fail in a fault-free but Trojan-infested chip. Similarly, since the analog functionality of the chip is left intact,

all analog/RF specification tests will pass. Furthermore, since the leaked information is hidden within the allowed transmission specification margins, system-level functional tests will also pass. Existing side-channel fingerprint generation and checking methods, at least in their original form, also fall short in detecting hardware Trojans in wireless cryptographic ICs.

- **Possible Solution:** Despite the fact that hardware Trojans can be hidden within the process variation margins of a wireless cryptographic chip and may not be exposed through any of the above methods, it may still be possible to detect them. Effective hardware Trojans must impose a specific structure on the transmission parameters, which the attacker leverages to snoop the secret key. While this structure is not known to the defender, advanced statistical analysis of these parameters may be sufficient to reveal its existence and, thereby, expose the hardware Trojan. Since the attacker does not know what data will be collected or how it will be analyzed, this method is difficult to evade. In other words, the element of surprise by the attacker, who picks the structure of the hidden data, is counteracted by a similar element of surprise by the defender, who picks the measurements and the statistical analysis method.

3 Pre-Deployment Hardware Trojan Detection

The most common threat model adopted in hardware Trojan research assumes that the culprit is either at the foundry or at design houses where third party intellectual property (IP) is acquired from. In either case, once silicon is obtained and before it is shipped to customers, it is essential to test not only for manufacturing defects (which is the objective of VLSI testing) but also for hardware Trojans. Therefore, we first discuss the problem of pre-deployment hardware Trojan detection in analog/RF ICs, wherein we can exercise the device under test in a controlled environment with pre-specified stimuli.

3.1 Experimentation Vehicle

The experimentation vehicle used to elucidate the problem of hardware Trojans in analog/RF ICs is shown in Figure 1. This is a mixed-signal wireless cryptographic IC, capable of encrypting and broadcasting data, which can be used in secure data transmission over open channels. The digital part includes a pipelined Digital Encryption Standard (DES) core [2], an output buffer and a serializer, which serves as the interface between the digital and the analog part. The analog part is an Ultra-Wide-Band (UWB) transmitter.

The DES core in the chip is a performance-optimized design with 16 encryption blocks in a pipeline structure. Each block can independently run the Feistel function f, which is the central part of the DES algorithm. A fully pipelined key

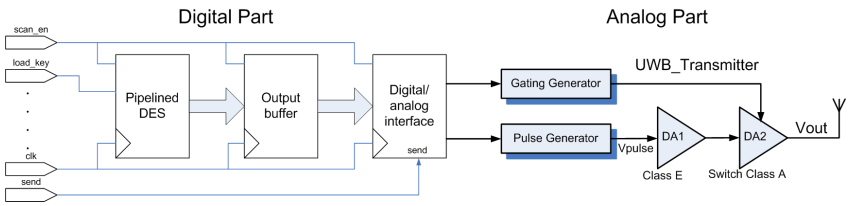


Fig. 1 Block diagram of example wireless cryptographic integrated circuit

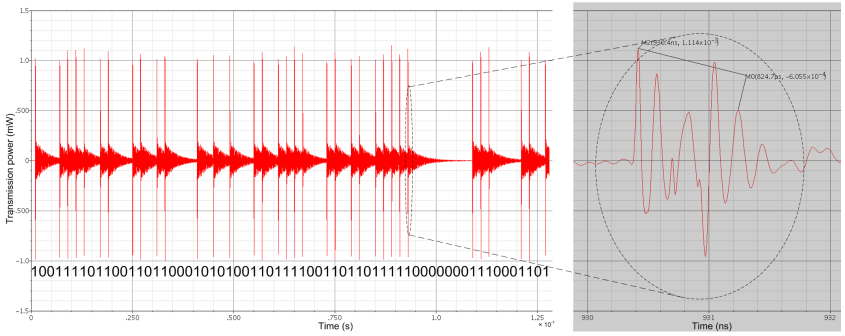


Fig. 2 Example of 64-bit ciphertext block transmission

generation module is designed to operate in parallel with these encryption blocks. In order to achieve high operating frequency, the initial permutation and inverse initial permutation of the plaintext are handled through hard-wiring, with no logic circuitry involved. The widths of the input and output data are both 64 bits, which is the length of a plaintext/ciphertext block. The output buffer is a First-In First-Out (FIFO) structure of 64-bit words, which supports reading and writing speeds commensurate with the performance of the pipelined DES core. The digital/analog interface converts the 64-bit data block from the buffer into a serial bit stream and passes it on to the UWB transmitter. The interface also adjusts the data-sending frequency to ensure signal integrity in this mixed-signal design. A pulse on the `send` primary input passes the contents of the output buffer to the interface and finally to the UWB transmitter for broadcasting. The UWB transmitter [41] consists of a pulse signal generator, a gating signal generator and two driver amplifiers (DAs) and can transmit data over a wide spectrum of frequency bands with very low power consumption and high data rate. The UWB transmitter is in active mode and transmits a high frequency signal when the information bit to be transmitted is '1', otherwise it is in idle mode.

The chip is designed in TSMC CL013G .13μm CMOS technology process [1]. The digital part runs at a frequency of 75MHz and the UWB transmitter has a data rate that exceeds 50Mbps. Tests for the digital part cover both stuck-at and delay faults using a full-scan chain of Enhanced Scan Flip-Flops [17]. For the analog part,

besides the traditional specification tests, the spectrum of the output pulse sequence of the DA chain at a data transmission rate of 50Mbps is also measured [41]. System-level functional tests involve randomly generated patterns which are encrypted and broadcasted by the UWB transmitter. A receiver decrypts the ciphertext and compares to the expected plaintext, in order to detect any discrepancies.

Figure 2 shows a simulation example of a typical transmission of a 64-bit block of ciphertext and a magnified view of the transmission signal when a '1' bit is broadcasted. UWB specification calls for a transmission frequency between 3.1GHz and 10.6GHz. The specifications for this particular implementation define its frequency between 4GHz and 6GHz. Transmitting a '1' bit involves between 5 and 7 peaks of amplitude over 300uW with at least one of them over 900uW. The actual performances of each individual chip will vary, depending on the fabrication process variations. For example, the response of the circuit instance shown in the figure, which was randomly picked from a population of 200 chips generated through a Monte Carlo Spice simulation with 5% process variation on all transistor parameters, operates at a frequency of 4.8GHz and involves 5 peaks of amplitude over 300uW with the largest measuring at 1114uW.

3.2 Hardware Trojans

Two hardware Trojans are designed which, through minute modifications, are capable of leaking the encryption key by hiding it in the wireless transmission parameter (i.e. amplitude or frequency) margins allowed in the design specifications in order to deal with process variations. Thus, they ensure that the circuit continues to comply to all of its functional specifications. The working principle of these Trojans is simple: extract one bit at a a time from the 56-bit encryption key, which is stored in the DES core, and leak it by hiding it in one 64-bit block of transmitted data. After 56 ciphertext blocks are transmitted, the entire key will have been broadcasted.

Implementation Details: Each hardware Trojan involves two modifications. The first modification, which is shown in Figure 3(a), is common to both hardware Trojans and aims to extract the encryption key from the DES core. The second modification, which is shown in Figure 3(b), is different for each of the two hardware Trojan and aims to manipulate the transmission amplitude or frequency in order to leak the key through the wireless channel.

The key extraction modification exploits the ability of Enhanced Scan Flip-Flops to store two bits, one in the D flip-flop and one in the follow-up latch, so that back-to-back vectors can be applied for the purpose of detecting delay faults when the circuit is in test mode [17]. During normal operation, however, the latches are transparent, essentially holding the same information as the D flip-flops. In the example circuit, the 56-bit encryption key is stored in a sequence of 56 Enhanced Scan Flip Flops which are serially connected in a scan chain, as shown in the top part of Figure 3(a). The basic idea for extracting the secret key is to store it only in the latches of the Enhanced Scan Flip Flops and reuse the D flip-flops to create a 56-bit rota-

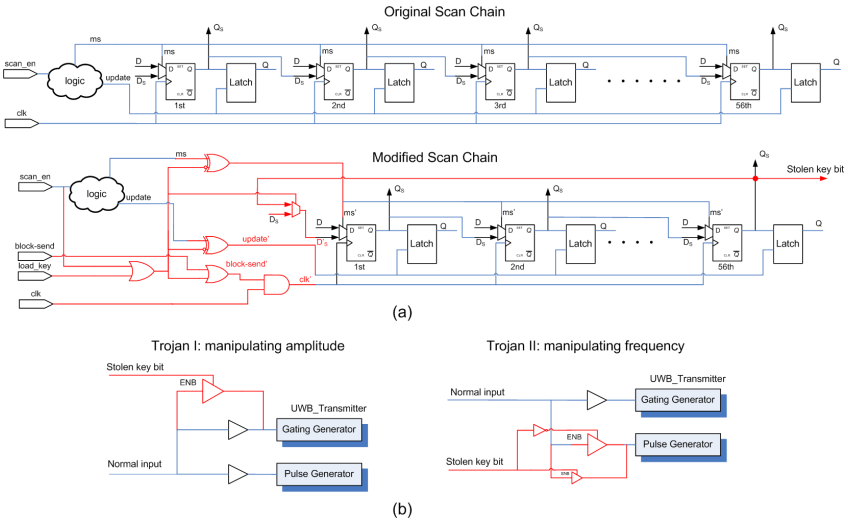


Fig. 3 (a) Extracting the key bitwise, through a rotator made out of the 56 enhanced scan flip-flops where it is stored, (b) Broadcasting the stolen key bit by manipulating the amplitude or the frequency of the UWB transmission

tor. Initially, when the key is loaded by the user, both the flip-flops and the latches hold the correct bits. Then, every time a data block is transmitted, the last bit of this rotator is extracted and hidden in the transmission, while the rotator shifts its contents by one position. Only the D flip-flops of the Enhanced Scan Flip Flops hold a rotated version of the key, while the follow-up latches continue to hold the correct version, so that the ciphertext is correctly produced. Simple control logic consisting of a few gates, shown in red color in the bottom part of Figure 3(a), suffices for this purpose.

The key transmission modification receives the stolen bit and based on its value modifies the transmission signal in one of two ways. The first option (Type-I), shown on the left side of Figure 3(b), manipulates the transmission amplitude; when the stolen key bit is '1', an additional driver strengthens the legitimate transmission signal before it reaches the gating generator, thereby slightly increasing the transmission amplitude. Figure 4(a) shows the corresponding impact on the signal transmitted by the example circuit instance used in Figure 2. In this case, the amplitude increases from 1114uW to 1235uW, but the frequency remains at 4.8GHz. The second option (Type-II), shown on the right side of Figure 3(b), manipulates the transmission frequency; when the stolen key bit is '1', the original buffer is bypassed and an alternative buffer is used to delay the output of the pulse generator, thereby slightly increasing the transmission frequency. Figure 4(b) shows the corresponding impact on the signal transmitted by the example circuit instance used in Figure 2. In this case, the frequency increases from 4.8GHz to 5.2GHz but the amplitude remains at 1105uW. In both cases, when the stolen key bit is '0', no change occurs in the transmitted signal.

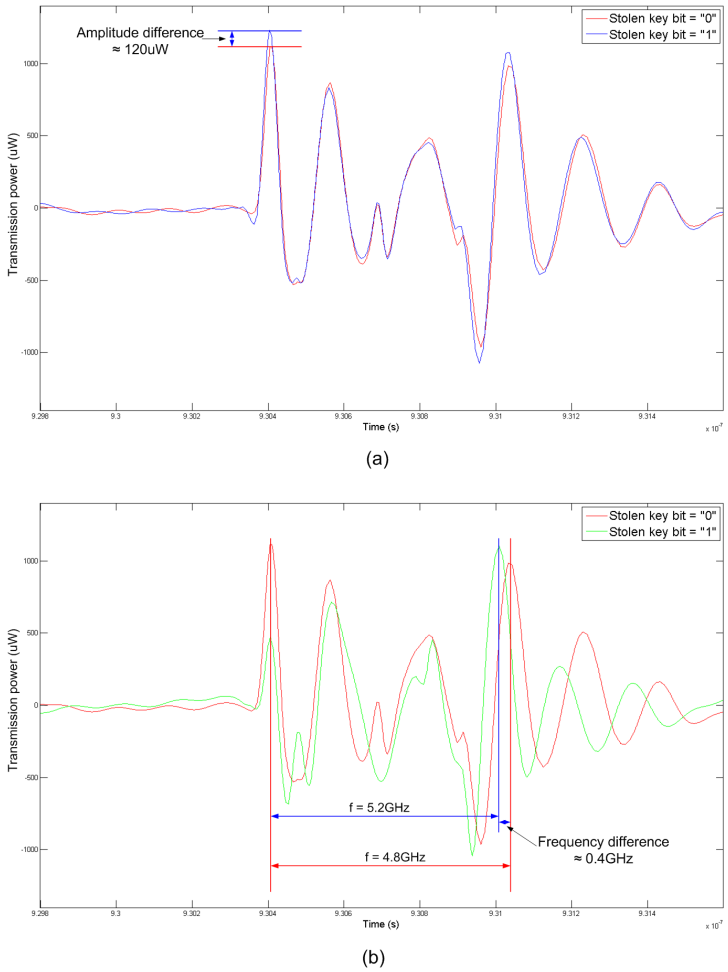


Fig. 4 (a) Difference in Type-I Trojan-infested circuit transmission depending on value of stolen key bit, (b) Difference in Type-II Trojan-infested circuit transmission depending on value of stolen key bit

The overall area overhead incurred by each of the above Trojans is around 0.02% of the digital part of the chip. This figure assumes that the storage elements holding the secret key are Enhanced Scan Flip Flops which are connected in sequence. If this is not the case and a separate 56-bit rotator needs to be added, the area overhead still remains well below 0.4% of the digital part of the chip.

Secret Information Extraction: Figures 4(a) and (b) show the transmission power waveform of a Type-I and a Type-II Trojan-infested chip, respectively, when the stolen key bit transmitted along with the legitimate signal is '1', as well as when it is '0'. Evidently, in the Type-I Trojan-infested chip, the difference in the stolen

key bit value is reflected as a difference of 120uW in the maximum amplitude. Similarly, in the Type-II Trojan-infested chip, the difference in the stolen key bit value is reflected as a 0.4GHz difference in the frequency. Both of these differences are well within the margins allowed for process variations and operating condition fluctuations and would not raise any suspicion. While the attacker does not know a priori the exact amplitude or frequency levels in each of the two cases, the fact that this difference is always present suffices for extracting the secret key. All the attacker needs to do is listen to the wireless channel to observe these two different amplitude or frequency levels, which correspond to a stolen key bit of '1' and a stolen key bit of '0', respectively. Once these two levels are known, listening to 56 consecutive transmission blocks reveals a rotated version of the 56 bits of the encryption key. Using this information, the attacker needs at most 56 attempts (i.e. all rotations of the extracted 56 bits) to decrypt the transmitted ciphertext.

3.3 Evaluation of Existing Test and Trojan Detection Methods

The mechanism through which the two hardware Trojan examples leak the secret information over the wireless channel allows them to evade detection not only by traditional manufacturing testing but also from previously proposed Trojan detection methods.

Functional, Structural, and Enhanced Testing: The hardware Trojan examples do not alter the functionality of the digital part of the circuit. In normal operation, the enhanced scan flip-flops that hold the key bits are loaded appropriately. Numerous randomly generated functional test vectors are simulated to verify the correctness of the produced ciphertext. In test mode, the scan chain also operates as expected. To demonstrate that structural tests do not detect these hardware Trojans, a standard industrial ATPG tool is used to generate test vectors for all stuck-at and delay faults in the Trojan-free circuit. These tests are simulated on the two Trojan-infested circuits. As expected, all tests passed. Enhancing the test set with further vectors that exercise rare events [40, 35] is also ineffective, since the hardware Trojans do not affect the digital functionality. The analog portion is not modified and, therefore, it also passes the traditional specification-based analog/RF test.

System-Level Testing: System-level tests examining the parameters of the wireless transmission also fail to expose the hardware Trojans, since the structure added by the leaked information is hidden within the margins allowed for process variations. To demonstrate this, we measured the transmission power of 200 genuine (i.e. Trojan-free) chips, 100 chips infested with a Type-I hardware Trojan and 100 chips infested with a Type-II hardware Trojan, which we generated using Monte Carlo Spice-level simulation assuming 5% process variations on all circuit parameters. Figure 5(a) plots the transmission power when a '1' is transmitted by half of these chips, as well as the $\mu \pm 3\sigma$ envelope of the transmission power when a '1' is transmitted by the other half of these chips. Figures 12(b) and (c) plot the transmission power when a '1' is transmitted by the Type-I and Type-II Trojan infested

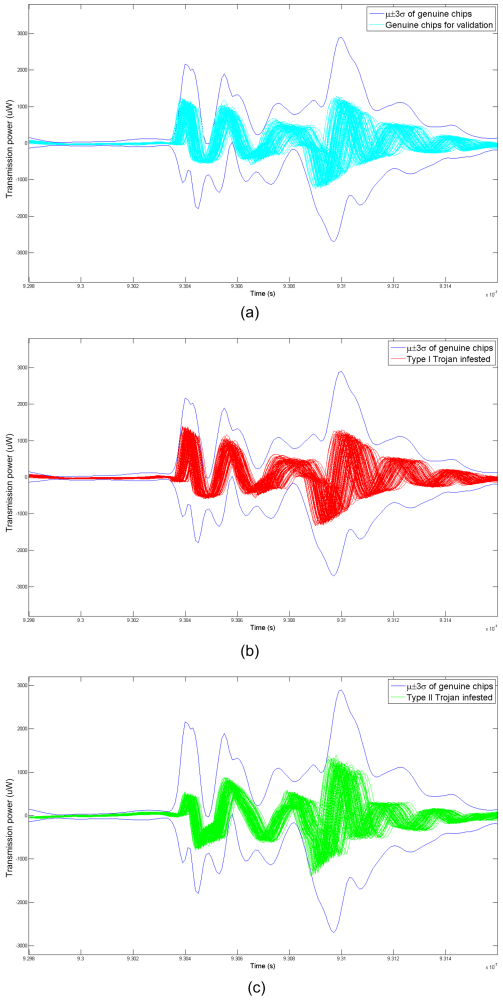


Fig. 5 (a) $\mu \pm 3\sigma$ transmission power envelope of 100 Trojan-free chips and transmission power of another 100 Trojan-free chips, (b) Transmission power of 100 Type-I Trojan-infested chips, (c) Transmission power of 100 Type-II Trojan-infested chips

chips, respectively. Evidently, given any one of these transmission power plots, it is not possible to distinguish whether it comes from a Trojan-free or a Trojan-infested chip.

Local Current Traces: An interesting hardware Trojan detection method based on local current traces is presented in [33, 34]. This test strategy detects anomalies introduced by the Trojan in the currents measured at the power ports and takes into account process and operating conditions variations. The authors demonstrate that their method can detect Trojans of size as small as 2% of the power grid. In order

to implement this method in the design, the chip needs to be divided into at least 20 power grids with at least 30 uniformly located power ports. The availability of these power ports is a serious obstacle to implementing this method. Furthermore, a capable attacker would probably observe the existence of these power ports and could possibly invent countermeasures to prevent the injected hardware Trojans from becoming visible through these ports.

Global Power Traces: In [5], the authors use global power consumption traces to distinguish between Trojan-free and Trojan-infested chips. The method employs statistical analysis of the Eigenvalue spectrum and can effectively detect hardware Trojans occupying 0.12% of the total circuit area, assuming process variation in the order of 5%. But when the hardware Trojan area is reduced to only 0.01% and the process variation is increased to 7.5%, false alarms start to appear. Considering the very low area overhead of the hardware Trojans (i.e. 0.02%) and based on the limitations outlined in [5], it is unlikely that statistical analysis of the total power consumption will expose them. Indeed, even when this method is applied to the power traces of the digital part only[1], wherein the hardware Trojans are hidden, it was not possible to effectively distinguish between Trojan-free and Trojan-infested chips in any Eigenvalue sub-space. Nevertheless, as mentioned in [5], other parameters may still prove effective. In fact, the solution used in the following section employs a similar statistical analysis of the wireless transmission power.

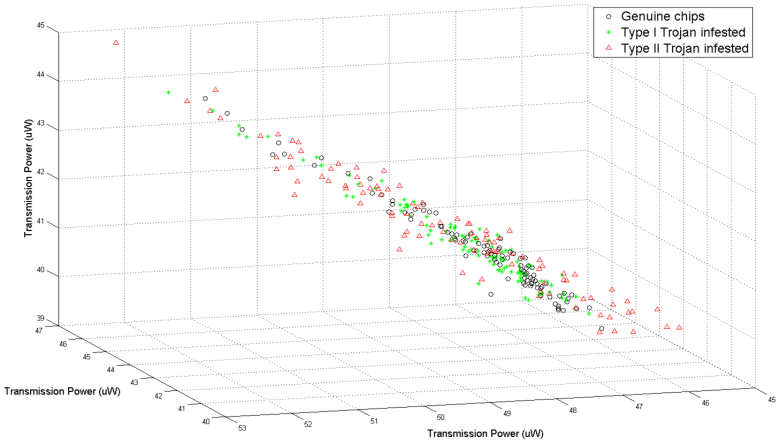
Path Delay Traces: A similar statistical method proposed in [20] utilizes path delay fingerprints to differentiate Trojan-free from Trojan-infested chips. While the hardware Trojan examples add some delay to a small number of paths in the digital part of the circuit, the impact is too small to be observed. Even if those paths related to the encryption key are checked, the complexity of the pipelined encryption circuitry provides enough margin to hide the added delay. To verify this, the path delay based Trojan detection method was applied assuming process variations in the range of 5% but it was unable to identify the existence of hardware Trojans.

3.4 Statistical Analysis to the Rescue

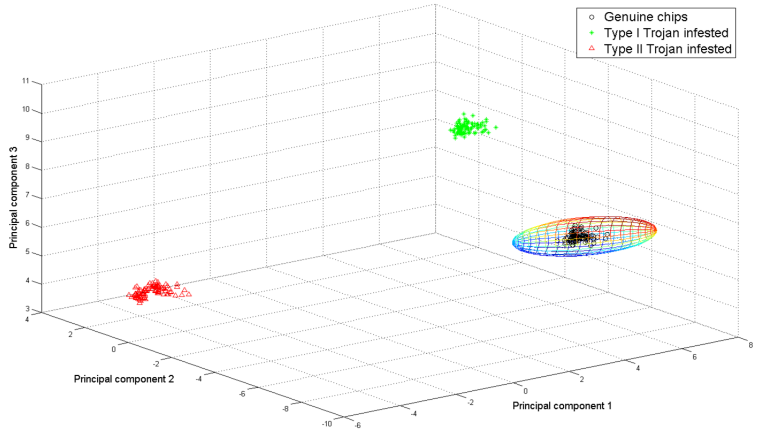
While the structure added to the transmitted signal for the attacker to extract the stolen key leaves individual transmissions within the acceptable specification boundaries, it enables the possibility that such hardware Trojans can be exposed through statistical analysis of the transmission parameters.

To demonstrate this principle, a measurement the total transmission power is used for broadcasting one block of data (i.e. 64-bits). For 100 Type-I Trojan-infected, 100 Type-II Trojan-infected, and half of the 200 Trojan-free circuit instances which are generated via Monte Carlo simulation with 5% process variations, the total transmission power is measured when transmitting each of six randomly selected blocks (the same for all circuits). Of course, the Trojan-infested chips also leak one key

[1] Mixed-signal SoCs typically have separate power ports for the analog and the digital parts.



(a)



(b)

Fig. 6 (a) Projection of genuine and Trojan-infested chip populations on three out of six transmission power measurement, (b) Projection of genuine and Trojan-infested chip populations on three principal components of six transmission power measurements

bit during each of the six transmissions, half of which are set to '1'. All six measurements for all genuine and all Trojan-infested chips are within the acceptable specification range. Even when the three chip populations are projected on the six-dimensional space of these measurements, it is impossible to distinguish them since they fall upon each other. Figure 6(a) shows a projection of the three populations on three of these dimensions. Evidently, separating the genuine from the Trojan-

infested populations in this space is not possible. The situation is similar for any other subset of three measurements.

However, running a Principal Component Analysis (PCA) on these measurements reveals that the structure of the genuine chip data is different than the structure of the Trojan-infested chip data. Figure 6(b) shows a projection of the three populations on the three principal components of the data, clearly revealing that they are separable in this space. Therefore, the trusted boundary is defined as a simple minimum volume enclosing ellipsoid (MVEE [30]) which encompasses the genuine population. Then, any chip whose footprint on the space of the selected three principal components does not fall within the trusted boundary will be discarded as suspicious. In the example, this method detects all Type-I and Type-II Trojan-infested chips without inadvertently discarding any genuine chips.

Given the small number of transmission parameters (or combinations thereof) wherein the attacker can hide the added structure, as well as the large number of measurements that the defender can utilize to identify statistical discrepancies, the defender can easily detect the inserted hardware Trojan. Finally, similar statistical analysis and machine learning-based methods involving parametric measurements have been previously employed successfully for the purpose of manufacturing testing [38] and radiometric fingerprinting [11] of analog/RF circuits. However, this is the first attempt to apply such methods towards hardware Trojan detection in wireless cryptographic ICs or aanalog/RF ICs in general.

4 Post-Deployment Hardware Trojan Detection

While the aforementioned side-channel fingerprinting method can be very effective in detecting hardware Trojans prior to IC deployment, it relies on the assumption that the Trojan is active at test time. Hence, it fails to detect dormant hardware Trojans which are activated only *after* an IC is deployed in its field of operation, through a lapsed-time counter or an external trigger [19]. Therefore, continuing to evaluate trustworthiness after deployment through on-chip support for hardware Trojan detection is equally important. To this end, in this section we introduce a general post-deployment hardware Trojan detection architecture [23], which is based on on-chip measurement acquisition and classification, and we demonstrate its effectiveness on the wireless cryptographic IC experimentation vehicle.

4.1 Proposed Trust Evaluation Architecture

The proposed architecture for post-deployment trust evaluation is shown in Figure 7. The overall idea is fairly straightforward: after the circuit is deployed, the end-user can trigger the trust evaluation procedure at any time; during trust evaluation, on-chip resources are used to apply a known stimulus to the circuit and to obtain para-

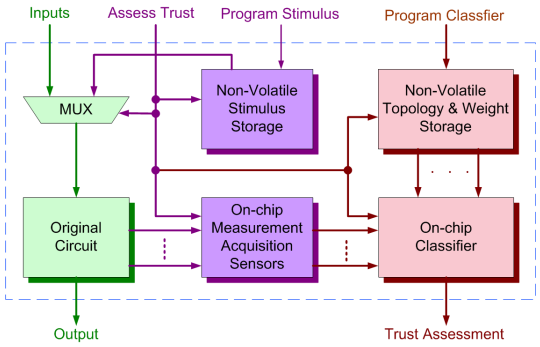


Fig. 7 Proposed post-deployment trust evaluation architecture

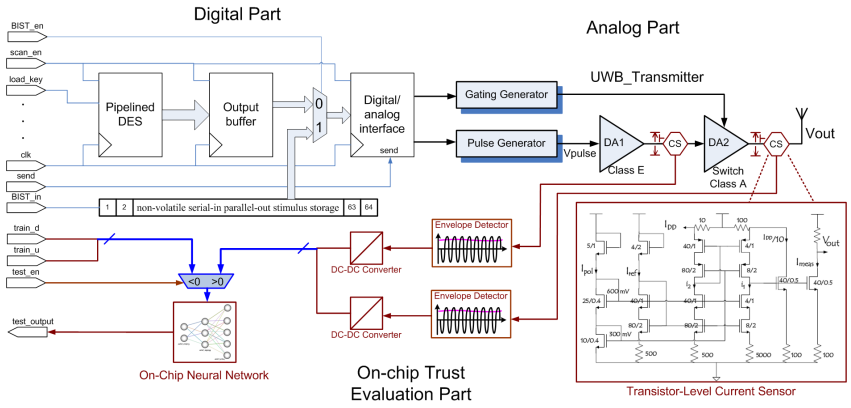


Fig. 8 Architecture of wireless cryptographic IC experimental platform

metric measurements, which are subsequently assessed on-chip to decide whether the circuit is operating within a trusted region. To this end, several components are added to the chip, along with the original circuit:

- A programmable on-chip non-volatile stimulus storage component (i.e., Flash, EEPROM, or OTPROM) and a multiplexer through which the known necessary excitation stimulus is provided to the circuit.
- Measurement acquisition sensors, to obtain the parametric signature of the circuit in response to the known stimulus.
- An on-chip classifier, to assess the parametric signature obtained via the sensors and to decide whether the circuit operation is trusted or not.
- Programmable on-chip non-volatile storage for programming the topology and the weights that define the region accepted as trusted by the classifier.

The programmability and non-volatility are required, so that the actual stimulus, the topology of the classifier, and the region accepted as trusted are stored on the chip only after it is fabricated. Thereby, a potential attacker is not privy to this information. While the attacker may be able to understand what parameters are being

measured, without knowledge of the stimulus, the actual structure of the classifier and the definition of the trusted region, it will be very difficult to design a hardware Trojan that evades detection. In essence, the proposed architecture counteracts the element of surprise possessed by the attacker (i.e., the ability to choose the location, functionality, and time of activation of the hardware Trojan) by a similar element of surprise possessed by the defender (i.e., the ability to choose the type of parametric signature, the method and bounds for assessing its trustworthiness, and the time of trust evaluation).

4.2 Experimentation Vehicle

4.2.1 Target Circuit

The experimental platform which is used to demonstrate the effectiveness of the proposed post-deployment Trojan detection method is an extension of the mixed-signal wireless cryptographic IC [21] which was used in the previous section. We remind that this chip takes plain-text at its input, encrypts it using an on-chip stored key, and then transmits the cipher-text on a public wireless channel. Figure 8 shows the basic architecture of the entire platform, which is divided into three parts: (i) the digital part, which includes a pipelined Digital Encryption Standard (DES) core, an output buffer, and a serializer serving as the interface between the digital and analog parts, (ii) the analog part, which is an ultrawide-band (UWB) transmitter, and (iii) the on-chip resources, which are added for the purpose of post-deployment hardware Trojan detection. These include an on-chip non-volatile serial-in parallel-out 64-bit register to hold the trust evaluation stimulus, two current sensors along with envelop detectors and DC-DC converters to obtain the side-channel fingerprint of the chip, and a neural network to classify it as trusted or untrusted. The current experimentation platform consists of SPICE-level simulation models for all components, except for the neural classifier. The latter is emulated through a programmable analog neural network experimentation chip to demonstrate, in silicon, the ability to detect hardware Trojans.

4.2.2 On-Chip Trust Evaluation Resources

The on-chip trust evaluation part performs two tasks, namely parametric measurement acquisition and data classification. Parametric measurements are obtained via on-chip sensors in response to a known stimulus, which is also stored on-chip using a non-volatile serial-in parallel-out shift (SIPO) register, as shown in Figure 8. The `BIST_in` signal is used to fill in the 64-bit wide register with a value *after* fabrication and prior to deployment. Another `BIST_en` signal controls the data flow to the digital/analog interface. When `BIST_en` is '0', the input of the interface is the ciphertext to be sent by the UWB transmitter while when it is '1', the pattern

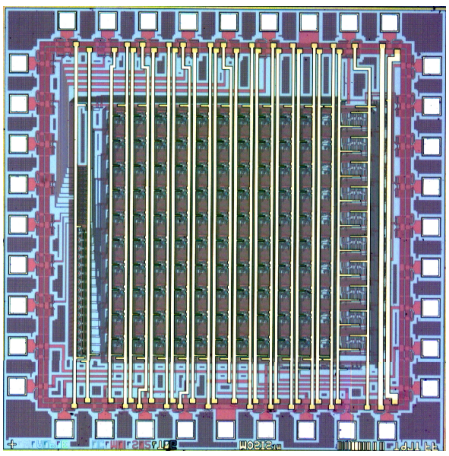


Fig. 9 Micrograph of Analog Neural Network Chip

stored in the SIPO register is sent to the UWB transmitter, in order to perform trust evaluation.

In this platform, two current measurements obtained from the UWB transmitter are used for trust evaluation. In order to lower area overhead and increase accuracy/stability of the measured currents, a robust CMOS built-in current sensor (BICS) is implemented [13]. The transistor-level structure of this current sensor can be seen in the blow-out part of Figure 8. The output of the BICS is a high frequency signal which is then converted to a DC voltage through a CMOS envelope detector [3]. Both the current sensor and the envelope detector are CMOS designs so that they are compatible with other parts of the circuit. A DC-DC converter is then used to match the measurement to the input range of the circuit that will perform data classification (i.e. the on-chip neural network).

4.2.3 On-Chip Classifier

To demonstrate in silicon that an on-chip classifier can, indeed, detect a hardware Trojan upon its activation in the operation field, an analog neural network experimentation chip is employed [28]. Using this programmable chip, artificial neural networks are implemented, which are then trained to learn (through a training set of chips) the mapping between the current measurements obtained from the two BICS integrated inside the UWB transmitter, and the trusted operation region. The trained neural networks can then be evaluated with respect to their capability to detect Trojan-infested chips using a validation set. Note that an analog VLSI implementation of the neural classifier is necessary in order to contain the area and power overhead of the proposed trust evaluation.

Figure 9 shows the stand-alone version of the programmable analog neural network chip which is used in the platform. This chip serves as a flexible platform for the experiments by virtue of two properties: *trainability*, which allows it to learn

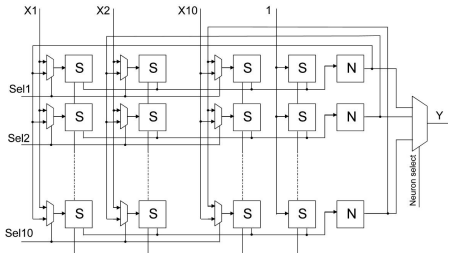


Fig. 10 Reconfigurable neural network architecture

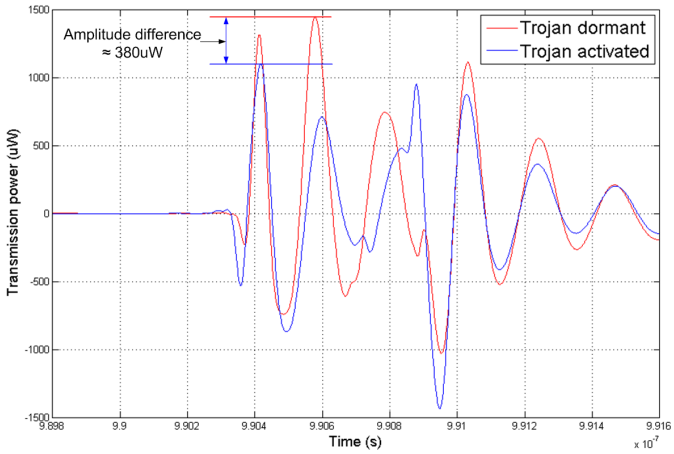
complex boundaries from the training set, and *reconfigurability*, which is used to adjust the number of hidden neurons to match the complexity of the target task. The possible topologies include all 2-layer networks within the available number of on-chip synapses and neurons. As will be shown later, network topologies with very small number of hidden neurons are sufficient to meet both the accuracy requirements to differentiate Trojan-infested chips from genuine chips and the low overhead requirements. Figure 10 illustrates the block-level schematic of the circuit implementation in the neural network chip. The circuit consists of a matrix of synaptic blocks (S) and neurons (N). The synapses represent mixed-signal devices, in the sense that they conduct all computations in analog form while their weights are implemented as digital words stored in a local memory. The results of synapse multiplication are summed and fed to the corresponding neuron, which performs a squashing function and produces an output either to the next layer or the primary output. The architecture is very modular and can easily be expanded to any number of neurons and inputs within the available silicon area [29].

4.2.4 Hardware Trojans

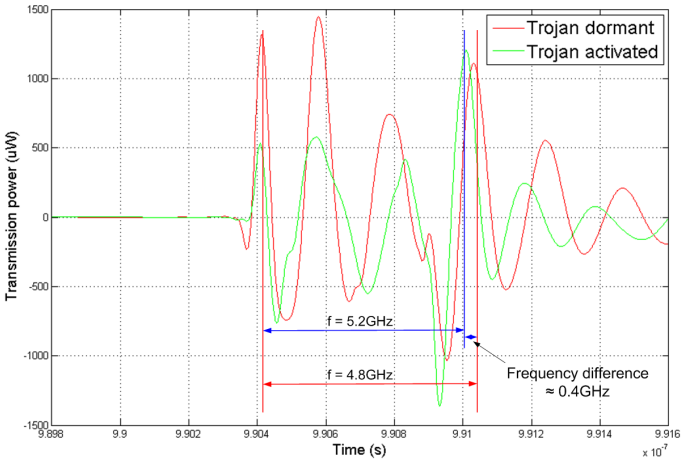
In addition to the Trojan-free circuit, two alternative hardware Trojan-infested variants of the wireless cryptographic IC are also designed. These are of similar structure and working principle to the Trojans introduced in the previous section, with the exception that both Trojans are dormant during the testing stage and are only activated after deployment[2]. As before, through simple modifications, when activated these hardware Trojans leak the encryption key by hiding it in the wireless transmission amplitude or frequency margins allowed due to process variations; thus, they ensure that the circuit continues to comply to all of its functional specifications and, thereby, evade testing both on the digital and on the analog side.

Figures 11(a) and (b) show the transmission power waveform of a Type-I and a Type-II Trojan-infested chip, respectively, when the Trojan is activated and the stolen bit is ‘1’, as well as when the Trojan is dormant (in which case, the stolen bit value is irrelevant). Evidently, in the Type-I Trojan-infested chip, the activation of the Trojan will alter the maximum amplitude by as much as 380uW from which attackers can differentiate a logic ‘1’ or logic ‘0’ value for the stolen key bit. Sim-

[2] Interested readers are referred to [19] for a relevant discussion on Trojan triggering.



(a)



(b)

Fig. 11 (a) Difference in Type-I Trojan-infested circuit transmission when Trojan is dormant and activated, (b) Difference in Type-II Trojan-infested circuit transmission when Trojan is dormant and activated

ilarly, in the Type-II Trojan-infested chip, the difference in the stolen key bit value is reflected as a 0.4GHz difference in the frequency when the Trojan is activated. Both of these differences are well within the margins allowed for process variations and operating condition fluctuations and would not raise any suspicion. While the attacker does not know a priori the exact amplitude or frequency levels in each of the two cases, the fact that this difference is always present suffices for extracting the secret key. All the attacker needs to do is listen to the wireless channel to observe

these two different amplitude or frequency levels, which correspond to a stolen key bit of '1' and a stolen key bit of '0', respectively, after the Trojan is activated. Once these two levels are known, listening to 56 consecutive transmission blocks reveals a rotated version of the 56 bits of the encryption key. Using this information, the attacker needs at most 56 attempts (i.e. all possible rotations of the extracted 56 bits) to decrypt the transmitted ciphertext.

4.3 Experimental Results

In order to assess the effectiveness of the proposed post-deployment trust evaluation method, measurements are collected from multiple instances of the wireless cryptographic IC described. These measurements are then processed in silicon through an on-chip classifier implemented on the reconfigurable neural network experimentation platform.

4.3.1 Dataset Generation

Using Spice-level Monte-Carlo simulation with $\pm 7.5\%$ process variations on all circuit parameters, 1K chip instances of the Trojan-free circuit are generated. Similarly, 1K chip instances of the Type-I Trojan-infested circuit and 1K chip instances of the Type-II Trojan-infested circuit are also generated. For each of the Trojan-free chip instances, the transmission power when a logic '1' is transmitted is measured. In addition, the measurements of the two current sensors are collected when a preselected 64-bit block (i.e. alternating 0s and 1s) is transmitted. The same measurements are also collected for the 1K Type-I Trojan-infested chips and 1K Type-II Trojan-infested chips, with the Trojan first dormant and then activated.

4.3.2 Observations

The following observations are made before further analysis of the collected dataset is performed:

- The transmission power profile of the Trojan-free chip-instances is indistinguishable from the transmission power profile of the Type-I Trojan-infested and Type-II Trojan infested chip instances with the Trojan *dormant*. This is demonstrated in Figures 12(a)-(c), where the transmission power is depicted for the chip instances of each of these three populations, enclosed within the $\pm 3\sigma$ boundary of the Trojan-free chip population. As may be observed, given any one of these transmission waveforms, it is impossible to definitively place it to one of the three populations. Even more interestingly, the transmission power profile of the Type-I Trojan-infested and Type-II Trojan infested chip instances with the Trojan *active* is also indistinguishable from the aforementioned populations, as shown in

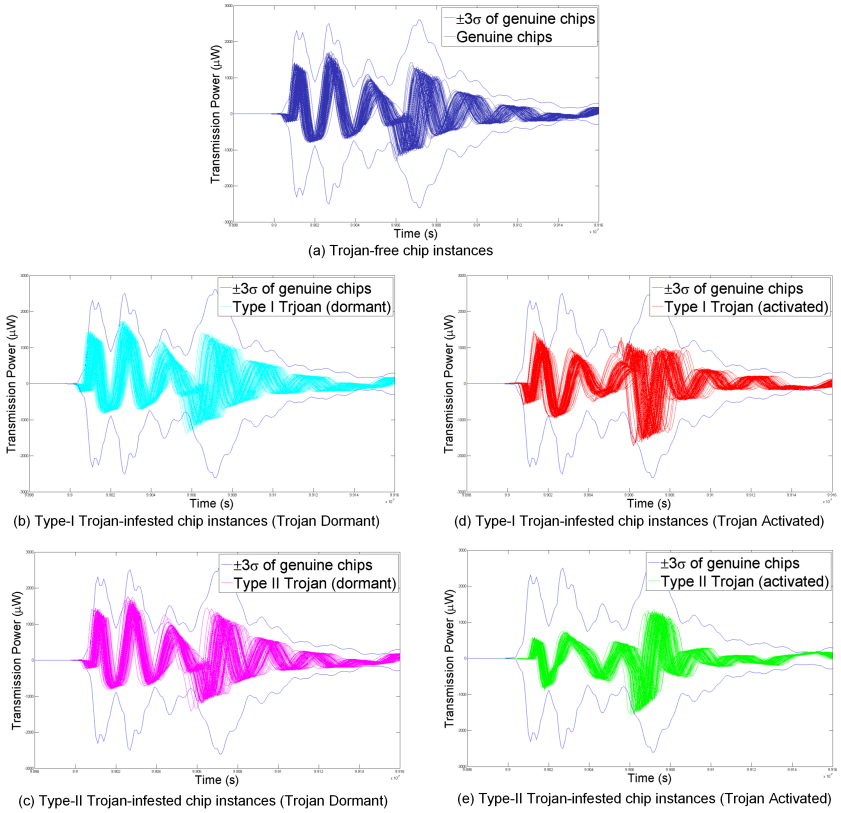


Fig. 12 3σ transmission power envelope of Trojan-free chip instances enclosing the various chip populations in the dataset

Figures 12(d)-(e). This is consistent with the results reported in [21] and affirms that the hardware Trojans do not violate the circuit specifications. In other words, a transmission of a Trojan-infested circuit with the Trojan activated appears to be perfectly legitimate and within the margins allowed for process variations and operational conditions fluctuation, hence the Trojans evade detection.

- The current sensor measurements of the Trojan-free chip instances are indistinguishable from the current sensor measurements of the Type-I Trojan-infested and Type-II Trojan infested chip instances with the Trojan *dormant*. This is demonstrated in Figure 13 which depicts the three chip populations on the two-dimensional space of the current measurements. Evidently, the three populations fall upon each other, attesting to the inadequacy of pre-deployment methods in detecting dormant Trojans.
- The current sensor measurements of the Trojan-infested chip instances with the Trojan activated are distinguishable from the current sensor measurements of the Trojan-infested chip instances with the Trojan dormant. This is demonstrated in

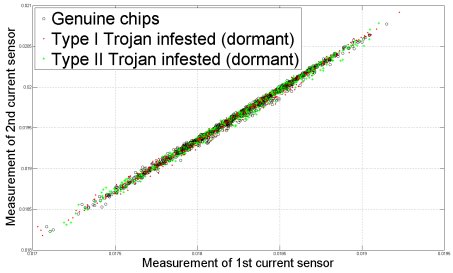


Fig. 13 Current sensor measurements with Trojans dormant

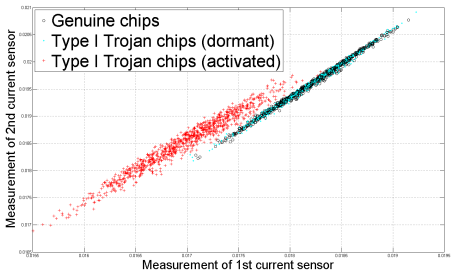


Fig. 14 Current sensor measurements for Type-I Trojan-infested chips

Figures 14 and 15 for each of the two Trojan types. As may be observed, while each current sensor measurement by itself is insufficient to separate the Trojan-active and Trojan-dormant populations, their combination provides adequate information to do so. Therefore, it is possible that a trained on-chip classifier will be able to pick up the difference in the current sensor measurements when the Trojan is activated post-deployment and, thereby, alert of untrusted circuit operation, as aimed by the proposed methodology.

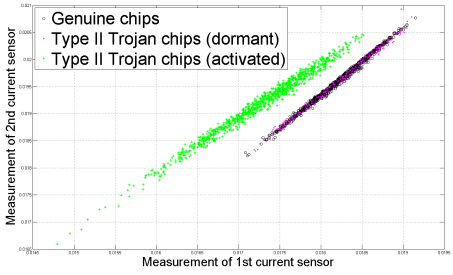


Fig. 15 Current sensor measurements for Type-II Trojan-infested chips

4.3.3 On-Chip Classifier Construction and Training

The reconfigurable neural network experimentation platform chip [28] described in Section 4.2 provides classifiers involving a range of neurons and various different topologies. In order to train an on-chip classifier to distinguish trusted from untrusted functionality, the testers should only rely on information from Trojan-free chips (or Trojan infested chips with the Trojan dormant, if Trojan-free chips are unavailable). This is important because, in a realistic scenario, the testers do not have advance knowledge of the various different types of Trojans and their potential impact, which will only appear after deployment of the chip. Therefore, in the experiments only the data (i.e. the two current sensor measurements) from the 1K Trojan-free chip instances to train the classifier is used. In other words, it is a 1-class classification problem, where the objective is to train a classifier to enclose the region of acceptable (trusted) functionality without any data of unacceptable (untrusted) functionality. To this end, the 1-class classification training algorithm described in [37] is employed. As can be observed in Figure 16, the boundary enclosing the trusted behavior is an ellipsoid, which can be approximated through a fairly simple two-layer neural network topology involving 4 neurons. The boundary shown in Figure 16 is the actual boundary learned by the trained on-chip neural network. As a point of reference, the boundary learned by the software version of the selected neural network is also showed. Evidently, the boundary learned in hardware is essentially identical to the one learned in software.

4.3.4 On-Chip Trust Evaluating Effectiveness

After training, the on-chip classifier with the data from Trojan-free chip instances is assessed for its effectiveness in correctly classifying the two types of Trojan-infested chip populations. In order to obtain a global picture, the trained classifier is presented with the data from both when the Trojan is dormant and when the Trojan is activated. The former will allow the testers to evaluate the false positive rate (i.e. incorrectly rejecting a chip when the Trojan is dormant) and the false negative rate (i.e. incorrectly accepting a chip when the Trojan is active). Figures 17 depicts the learned boundary, along with the footprints of the Type-I Trojan-infested chip instances with the Trojan dormant and active. Similarly, Figure 18 depicts the learned boundary, along with the footprints of the Type-II Trojan-infested chip instances with the Trojan dormant and active. As may be observed, the trained classifier performs extremely well and almost perfectly encapsulates the chip populations when the Trojan is dormant, while almost perfectly excluding the chip populations when the Trojan is activated. Tables 1 and 2 report the confusion matrices for the Type-I and Type-II Trojan-infested chip populations, respectively. For comparison, the effectiveness of the software version of the classifier is also reported, demonstrating that the error due to the hardware implementation is minimal.

While not zero, the false positive and false negative rates are very low, indicating that the proposed on-chip classifier-based methodology has the potential of provid-

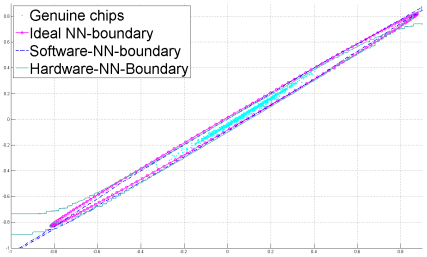


Fig. 16 The trained boundary learned through software NN and hardware NN for Trojan-free chips

Table 1 Type I Trojan Classification

		Classified by hardware		Classified by software	
		Dormant	Activated	Dormant	Activated
Actual	Dormant	99.9%	0.1%	100%	0%
	Activated	2.8%	97.2%	1.3%	98.7%

Table 2 Type II Trojan Classification

		Classified by hardware		Classified by software	
		Dormant	Activated	Dormant	Activated
Actual	Dormant	99.8%	0.2%	100%	0%
	Activated	0%	100%	0%	100%

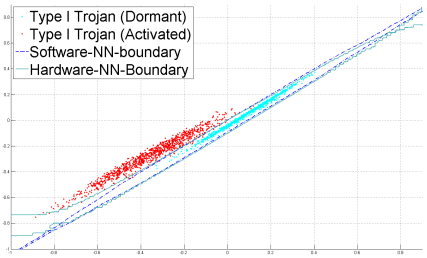


Fig. 17 Ability of boundary learned through software and hardware NN to correctly classify dormant and activated Type-I Trojan-infested chips

ing an effective post-deployment trust evaluation capability. Further research is still required towards achieving zero misclassification rate to ensure trustworthiness of deployed circuits.

5 Conclusion

The threat of hardware Trojans has fueled recent research in evaluating trustworthiness of fabricated ICs, both in the digital and in the analog/RF domains. In this chapter, current hardware Trojan detection methods in the analog/RF domain and,

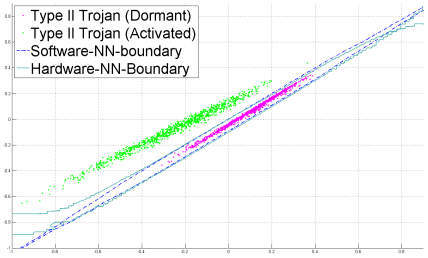


Fig. 18 Ability of boundary learned through software and hardware NN to correctly classify dormant and activated Type-II Trojan-infested chips

more specifically, in wireless cryptographic ICs was introduced. Using a Trojan-free and two Trojan-infested versions of a DES encryption core and a UWB transmitter, we demonstrated (i) the simplicity of a hardware Trojan attack and the ease with which it can leak sensitive information such as the encryption key, (ii) the inability of existing manufacturing test and hardware Trojan detection methods developed for digital circuit to detect such hardware Trojans in the analog/RF domain, and (iii) the power of side-channel fingerprinting in detecting such Trojans using statistical analysis of the transmission power waveforms and trained classifiers to distinguish between hardware Trojan-free an hardware Trojan-infested ICs. Furthermore, the problem of dormant hardware Trojans which are inactive during pre-deployment test and are only activated after deployment was discussed, along with the new challenges that it presents. Accordingly, a post-deployment trust evaluation architecture was introduced, wherein on-chip resources including sensors and a classifier are added to the IC, in order to support hardware Trojan detection in the field of operation. While these solutions provide an excellent initial step towards thwarting hardware Trojans in wireless cryptographic circuits, there remain plenty of challenges and opportunities for further research towards developing an arsenal of hardware Trojan prevention and detection methods for analog/RF ICs.

References

1. http://www.mosis.com/products/fab/vendors/tsmc/tsmc013-cl
2. http://www.opencores.org/projects.cgi/web/des/overview
3. Abdallah, L., Stratigopoulos, H.G., Kelma, C., Mir, S.: Sensors for built-in alternate rf test. In: Test Symposium (ETS), 2010 15th IEEE European, pp. 49–54 (2010)
4. Adee, S.: The hunt for the kill switch. IEEE Spectrum **45**(5), 34–39 (2008)
5. Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., Sunar, B.: Trojan detection using IC fingerprinting. In: IEEE Symposium on Security and Privacy, pp. 296–310 (2007)
6. Banga, M., Chandrasekar, M., Fang, L., Hsiao, M.S.: Guided test generation for isolation and detection of embedded Trojans in ICs. In: Proceedings of the 18th ACM Great Lakes symposium on VLSI, pp. 363–366 (2008)
7. Banga, M., Hsiao, M.: A novel sustained vector technique for the detection of hardware Trojans. In: 22nd International Conference on VLSI Design, pp. 327–332 (2009)

8. Banga, M., Hsiao, M.: VITAMIN: Voltage inversion technique to asertain malicious insertion in ICs. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 104–107 (2009)
9. Bloom, G., Narahari, B., Simha, R., Zambreno, J.: Providing secure execution environments with a last line of defense against Trojan circuit attacks. Computers & Security **28**(7), 660–669 (2009)
10. Bloom, G., Simha, R., Narahari, B.: OS support for detecting Trojan circuit attacks. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 100–103 (2009)
11. Candore, A., Kocabas, O., Koushanfar, F.: Robust stable radiometric fingerprinting for frequency reconfigurable devices. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 43–49 (2009)
12. Chakraborty, R., Wolff, F., Paul, S., Papachristou, C., Bhunia, S.: MERO: A statistical approach for hardware Trojan detection. In: Cryptographic Hardware and Embedded Systems, *Lecture Notes in Computer Science*, vol. 5747, pp. 396–410 (2009)
13. Cimino, M., Lapuyade, H., De Matos, M., Taris, T., Deval, Y., Begueret, J.: A robust 130nm-cmos built-in current sensor dedicated to rf applications. In: Test Symposium, 2006. ETS '06. Eleventh IEEE European, pp. 151–158 (2006)
14. Drzevitzky, S., Kastens, U., Platzner, M.: Proof-carrying hardware: Towards runtime verification of reconfigurable modules. In: International Conference on Reconfigurable Computing and FPGAs, pp. 189–194 (2009)
15. Drzevitzky, S., Platzner, M.: Achieving hardware security for reconfigurable systems on chip by a proof-carrying code approach. In: 6th International Workshop on Reconfigurable Communication-centric Systems-on-Chip, pp. 1–8 (2011)
16. Hicks, M., Finnicum, M., King, S.T., Martin, M.M.K., Smith, J.M.: Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 159–172 (2010)
17. Jha, N., Gupta, S.: Testing of Digital Systems. Cambridge University Press (2003)
18. Jin, Y., Kupp, N., Makris, M.: DFTT: Design for Trojan test. In: IEEE International Conference on Electronics Circuits and Systems, pp. 1175–1178 (2010)
19. Jin, Y., Kupp, N., Makris, Y.: Experiences in hardware Trojan design and implementation. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 50–57 (2009)
20. Jin, Y., Makris, Y.: Hardware Trojan detection using path delay fingerprint. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 51–57 (2008)
21. Jin, Y., Makris, Y.: Hardware Trojans in wireless cryptographic ICs. IEEE Design and Test of Computers **27**, 26–35 (2010)
22. Jin, Y., Makris, Y.: Proof carrying-based information flow tracking for data secrecy protection and hardware trust. In: IEEE 30th VLSI Test Symposium (VTS), pp. 252–257 (2012)
23. Jin, Y., Maliuk, D., Makris, Y.: Post-deployment trust evaluation in wireless cryptographic ICs. In: Design, Automation Test in Europe Conference DATE), pp. 965–970 (2012)
24. Jin, Y., Yang, B., Makris, Y.: Cycle-accurate information assurance by proof-carrying based signal sensitivity tracing. In: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 99–106 (2013)
25. Lin, L., Burleson, W., Paar, C.: MOLES: Malicious off-chip leakage enabled by side-channels. In: ICCAD '09: Proceedings of the 2009 International Conference on Computer-Aided Design, pp. 117–122. ACM (2009)
26. Lin, L., Kasper, M., Guneysu, T., Paar, C., Burleson, W.: Trojan side-channels: Lightweight hardware Trojans through side-channel engineering. In: Cryptographic Hardware and Embedded Systems, *LNCS*, vol. 5747, pp. 382–395. Springer-Verlag Berlin (2009)
27. Love, E., Jin, Y., Makris, Y.: Proof-carrying hardware intellectual property: A pathway to trusted module acquisition. IEEE Transactions on Information Forensics and Security **7**(1), 25–40 (2012)
28. Maliuk, D., Makris, Y.: A dual-mode weight storage analog neural network platform for on-chip applications. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2889–2892 (2012)

29. Maliuk, D., Stratigopoulos, H., Huang, H., Makris, Y.: Analog neural network design for RF built-in self-test. In: Proceedings of the IEEE International Test Conference (ITC), pp. 23.2.1–23.2.10 (2010)
30. Moshtagh, N.: Minimum volume enclosing ellipsoid. In: GRASP Laboratory, University of Pennsylvania, http://www.seas.upenn.edu/ñima/papers/Mim_vol_ellipse.pdf (2005)
31. Nelson, M., Nahapetian, A., Koushanfar, F., Potkonjak, M.: SVD-based ghost circuitry detection. In: Information Hiding, *Lecture Notes in Computer Science*, vol. 5806, pp. 221–234 (2009)
32. Potkonjak, M., Nahapetian, A., Nelson, M., Massey, T.: Hardware Trojan horse detection using gate-level characterization. In: DAC '09: Proceedings of the 46th Annual Design Automation Conference, pp. 688–693 (2009)
33. Rad, R., Plusquellic, J., Tehranipoor, M.: Sensitivity analysis to hardware Trojans using power supply transient signals. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 3–7 (2008)
34. Rad, R.M., Wang, X., Tehranipoor, M., Plusquellic, J.: Power supply signal calibration techniques for improving detection resolution to hardware Trojans. In: IEEE/ACM International Conference on Computer-Aided Design, pp. 632–639 (2008)
35. Salmani, H., Tehranipoor, M., Plusquellic, J.: New design strategy for improving hardware Trojan detection and reducing Trojan activation time. In: IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 66–73 (2009)
36. Sinanoglu, O., Karimi, N., Rajendran, J., Karri, R., Jin, Y., Huang, K., Makris, Y.: Reconciling the IC test and security dichotomy. In: 18th IEEE European Test Symposium (ETS), pp. 1–6 (2013)
37. Skabar, A.: Single-class classifier learning using neural networks: An application to the prediction of mineral deposits. In: The 2nd International Conference on Machine Learning and Cybernetics, pp. 2127–2132 (2003)
38. Stratigopoulos, H.G., Makris, Y.: Error moderation in low-cost machine-learning-based analog/RF testing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **27**(2), 339–351 (2008)
39. Waksman, A., Suozzo, M., Sethumadhavan, S.: FANCI: Identification of stealthy malicious logic using boolean functional analysis. In: Proceedings of the ACM SIGSAC Conference on Computer & Communications Security, CCS '13, pp. 697–708 (2013)
40. Wolff, F., Papachristou, C., Bhunia, S., Chakraborty, R.S.: Towards Trojan-free trusted ICs: Problem analysis and detection scheme. In: IEEE Design Automation and Test in Europe, pp. 1362–1365 (2008)
41. Yuan, T., Zheng, Y., Ang, C., Li, L.: A fully integrated CMOS transmitter for ultra-wideband applications. In: IEEE Radio Frequency Integrated Circuits Symposium, pp. 39–42 (2007)