Post-Deployment Trust Evaluation in Wireless Cryptographic ICs

Yier Jin*, Dzmitry Maliuk* and Yiorgos Makris†
*Department of Electrical Engineering, Yale University
†Department of Electrical Engineering, The University of Texas at Dallas
{yier.jin@yale.edu, dzmitry.maliuk@yale.edu, yiorgos.makris@utdallas.edu}

***Abstract*—The use of side-channel parametric measurements along with statistical analysis methods for detecting hardware Trojans in fabricated integrated circuits has been studied extensively in recent years, initially for digital designs but recently also for their analog/RF counterparts. Such post-fabrication trust evaluation methods, however, are unable to detect dormant hardware Trojans which are activated after a circuit is deployed in its field of operation. For the latter, an on-chip trust evaluation method is required. To this end, we present a general architecture for post-deployment trust evaluation based on on-chip classifiers. Specifically, we discuss the design of an on-chip analog neural network which can be trained to distinguish trusted from untrusted circuit functionality based on simple measurements obtained via on-chip measurement acquisition sensors. The proposed method is demonstrated using a Trojan-free and two Trojan infested variants of a wireless cryptographic IC design, as well as a fabricated programmable neural network experimentation chip. As corroborated by the obtained experimental results, two current measurements suffice for the on-chip classifier to effectively assess trustworthiness and, thereby, detect hardware Trojans that are activated after chip deployment.**

I. Introduction

Malicious modifications to integrated circuits (ICs), commonly referred to as *hardware Trojans*, have been the subject of intense study in recent years. Such modifications, which are purportedly done without the knowledge of the designer or end-user of a chip, may provide additional functionality that can be exploited by a perpetrator to cause erroneous results, steal sensitive information or incapacitate a chip. Evidently, given the range of applications where ICs are deployed, the impact of such hardware Trojans can be catastrophic.

Accordingly, various Trojan detection methods have been proposed to date, largely falling in two categories: *enhanced functional testing* and *side-channel fingerprint generation and checking*. The former are based on the assumption that infrequently occurring events will be employed by attackers to trigger the hardware Trojan, and therefore aim to include such events in the test plan [1], [2]. The latter assume that a hardware Trojan will not alter the functionality but rather only the parametric profile of a chip. Therefore, they rely on a fingerprint constructed from parameters such as global power consumption [3], path delays [4], or currents on power grids [5], [6], along with a trusted fingerprint region which is statistically learned from genuine circuits (golden models), in order to differentiate Trojan-infested from Trojan-free chips.

Since the aforementioned methods are typically applied prior to chip deployment, a possible attack strategy to evade them is to design hardware Trojans that are dormant at test time and are only activated later in the field of operation. This can be easily achieved through a lapsed time or pre-specified input trigger [7]. Therefore, continuing to evaluate trustworthiness after chip deployment becomes equally important. To this end, in this paper we propose a general post-deployment trust evaluation architecture, which is based on on-chip measurement acquisition and classification, and we demonstrate its effectiveness on a wireless cryptographic IC.

The remainder of this paper is organized as follows: in section II, we describe the proposed post-deployment trust evaluation architecture. In section III, we provide details of a wireless cryptographic IC and a programmable analog neural network chip, which we use as an experimental platform to demonstrate this method. Finally, in section IV, we present experimental results which corroborate the effectiveness of the proposed post-deployment trust evaluation method.

II. Proposed Trust Evaluation Architecture

The proposed architecture for post-deployment trust evaluation is shown in Figure 1. 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 parametric 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.

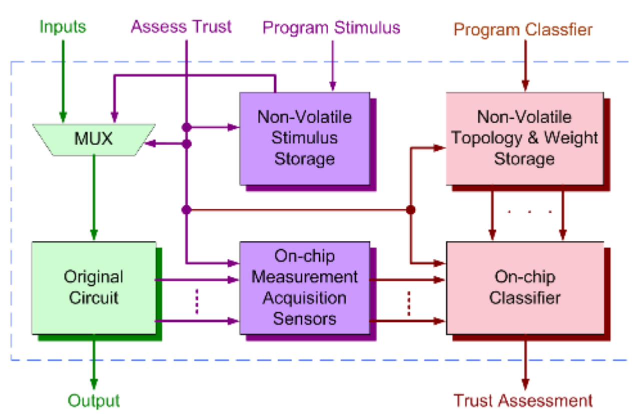

Fig. 1. Proposed post-deployment trust evaluation architecture

978-3-9810801-8-6/DATE12/©2012 EDAA

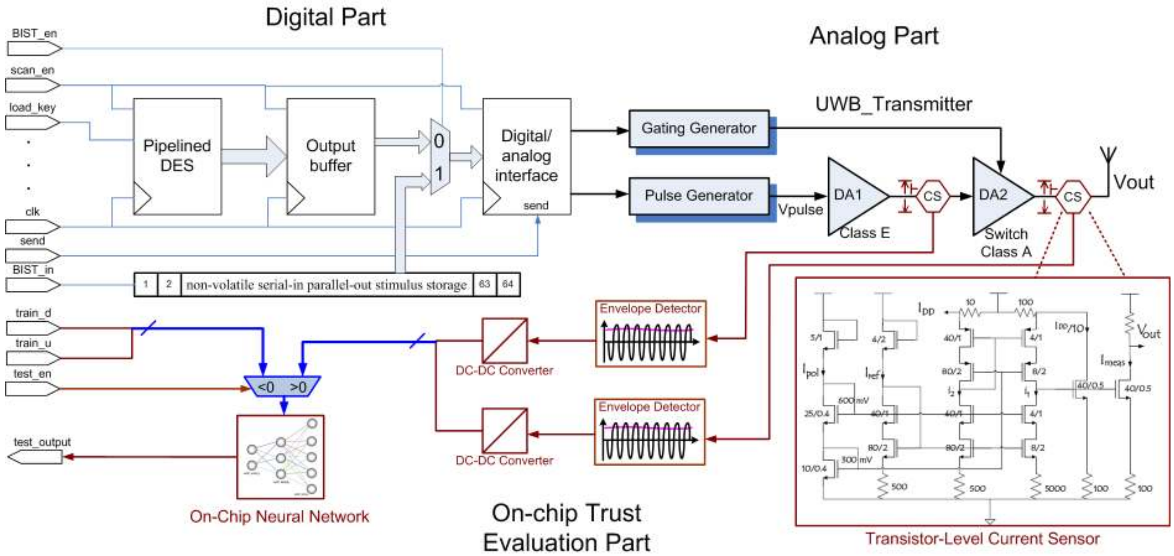

Fig. 2. Architecture of wireless cryptographic IC experimental platform

- 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.

We point out that 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).

III. Experimental Platform

A. Target Circuit

The experimental platform which we use to demonstrate the effectiveness of the proposed post-deployment Trojan detection method is based on our previously introduced mixed-signal wireless cryptographic IC [8]. 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 2 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 trust evaluation resources, which we added for the purpose of this work. 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. Our 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, so that we can demonstrate *in silicon* the ability to detect hardware Trojans.

B. 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 (SIPO) shift register, as shown in Figure 2. 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 stored in the SIPO register is sent to the UWB transmitter, in order to perform trust evaluation.

In this work, we use two current measurements obtained from the UWB transmitter 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 [9]. The transistor-level structure of this current sensor can be seen in the blow-out part of Figure 2. The output of the BICS is a high frequency signal which we convert to a DC voltage through a CMOS envelope detector [10]. Both the current sensor and 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).

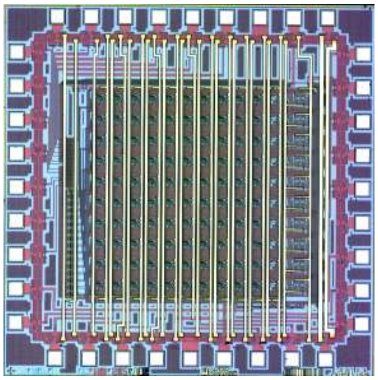

Fig. 3. Micrograph of Analog Neural Network Chip

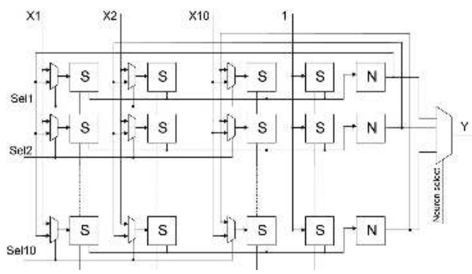

Fig. 4. Reconfigurable neural network architecture

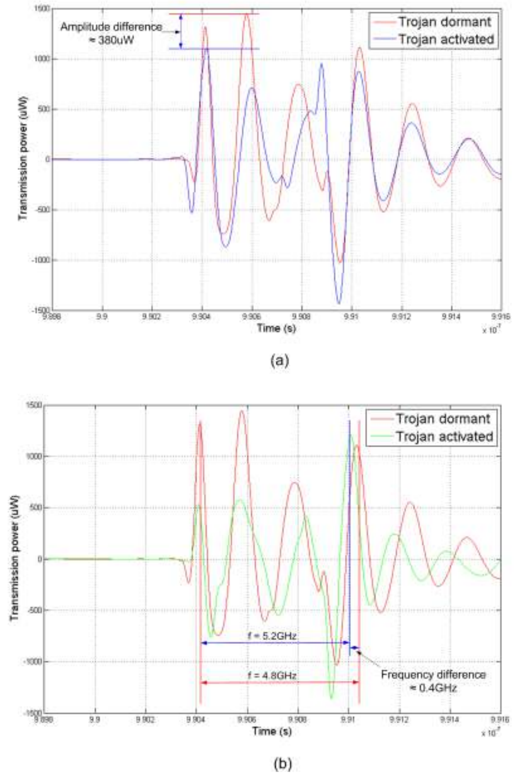

Fig. 5. (a) Difference in Type-I Trojan-infested circuit transmission when Trojan is dormant and activated, (b) Difference in Type-II Trojan-infested circuit transmission when Trojan is dormant and activated

C. 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, we employ an analog neural network experimentation chip. Using this programmable chip, we implement artificial neural networks, which we then train to learn (through a training set of chips) the mapping between the current measurements obtained from the two BICS which we 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. We 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 3 shows the stand-alone version of the programmable analog neural network chip which we used in our platform. This chip serves as a flexible platform for our experiments by virtue of two properties: trainability, which allows it to learn 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 4 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 [11].

D. 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 we introduced in [8] with the exception that both Trojans are dormant during the testing stage and are only activated after deployment[1]. Through simple modifications on only the digital portion of the chip, they leak the encryption key by hiding it in the wireless transmission amplitude (Type-I) or frequency (Type-II) 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 5(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

[1]We use a counter as the time-lapse trigger for each Trojan but other types of Trojan triggers may also be used. Trojan triggering is outside the scope of this paper and we refer the interested reader to [7] for a relevant discussion.

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. 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 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.

IV. Experimental Results

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

A. Dataset Generation

Using Spice-level Monte-Carlo simulation with $\pm 7.5\%$ process variations on all circuit parameters, we generated 1K chip instances of the Trojan-free circuit. Similarly, we also generated 1K chip instances of the Type-I Trojan-infested circuit and 1K chip instances of the Type-II Trojan-infested circuit. For each of the Trojan-free chip instances, we measured the transmission power when a logic '1' is transmitted. In addition, we collected the measurements of the two current sensors when a pre-selected 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.

B. Observations

Before we proceed with classification results, we point out the following observations on the collected dataset:

- 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 6(a)-(c), where we depict the transmission power 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

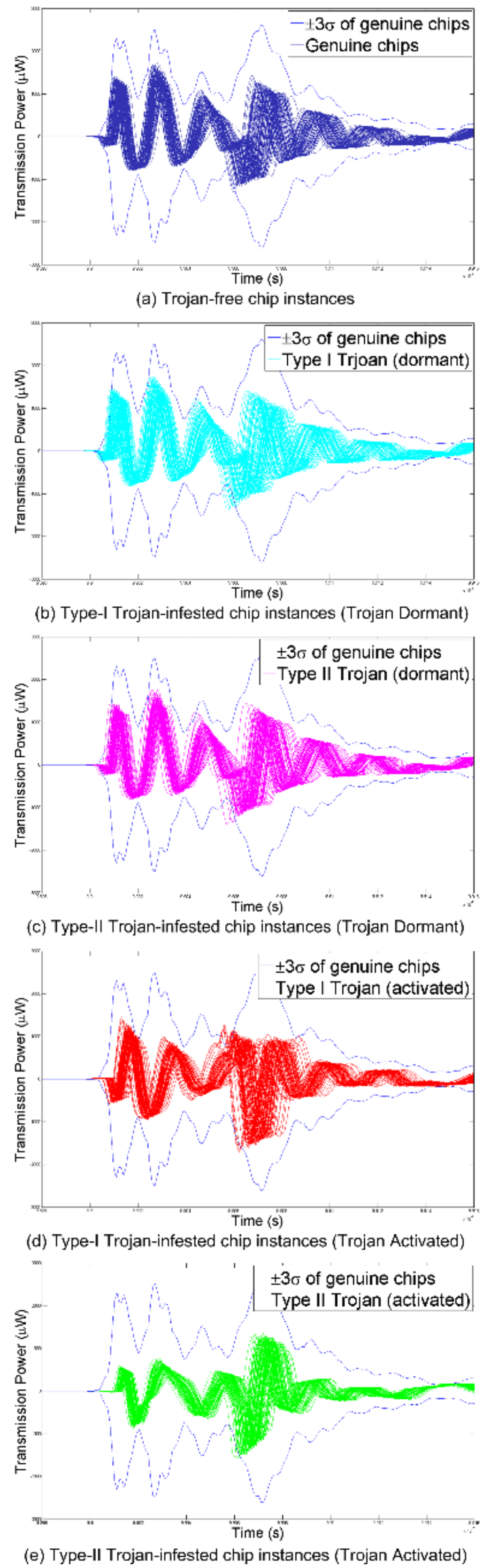

Fig. 6. $\pm 3\sigma$ transmission power envelope of Trojan-free chip instances enclosing the various chip populations in our dataset

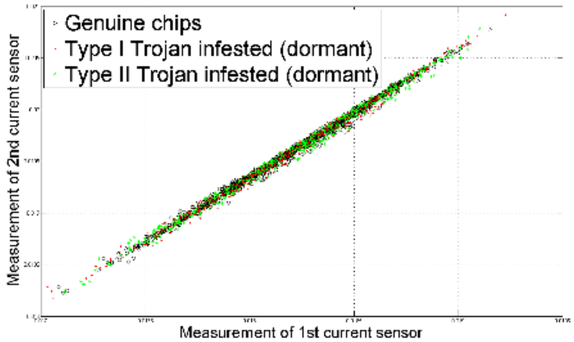

Fig. 7. Current sensor measurements with Trojans dormant

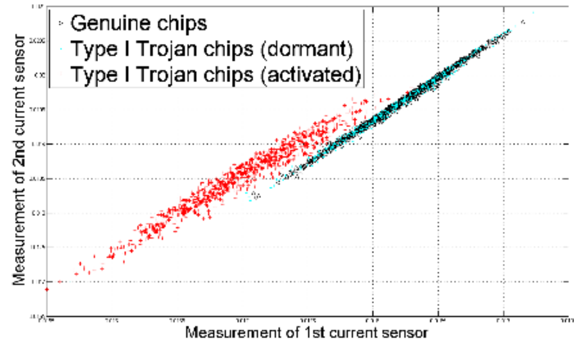

Fig. 8. Current sensor measurements for Type-I Trojan-infested chips

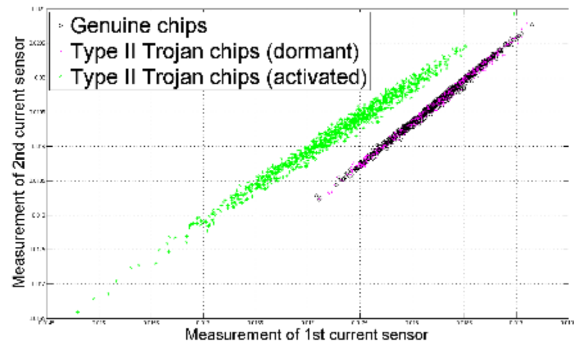

Fig. 9. Current sensor measurements for Type-II Trojan-infested chips

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 Figures 6(d)-(e). This is consistent with the results reported in [8] 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 7, where we depict 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 Figures 8 and 9 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.

C. On-Chip Classifier Construction and Training

Using the reconfigurable neural network experimentation platform chip described in Section III, we can emulate classifiers involving a range of neurons and various different topologies. We note that, in order to train an on-chip classifier to distinguish trusted from untrusted functionality, one 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, one does 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 our experiments we only use the data (i.e. the two current sensor measurements) from the 1K Trojan-free chip instances to train our classifier. In other words, we are solving a 1-class classification problem, where our objective is to train a classifier to enclose the region of acceptable (trusted) functionality without any data of unacceptable (untrusted) functionality. To this end, we employ the 1-class classification training algorithm described in [12]. As can be observed in Figure 10, 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 10 is the actual boundary learned by the trained on-chip neural network, which we approximated via a fine-grained sweeping of the two inputs of the neural network (current sensor measurements). As a point of reference, we also show the boundary learned by the software version of the selected neural network. Evidently, the boundary learned in hardware is essentially identical to the one learned in software.

D. On-Chip Trust Evaluating Effectiveness

Having trained the on-chip classifier with the data from the Trojan-free chip instances, we proceed to assess its effectiveness in correctly classifying the two types of Trojan-infested chip populations. In order to obtain a global picture, we present the trained classifier with the data from both when the Trojan is dormant and when the Trojan is activated. The former will allow us 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

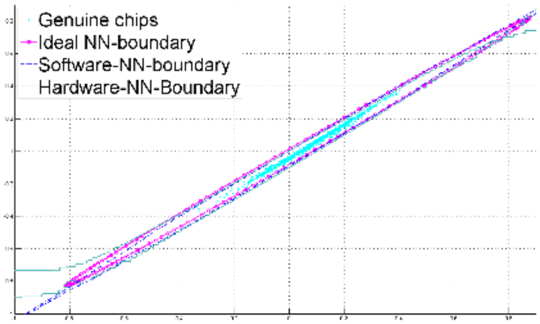

Fig. 10. The trained boundary learned through software NN and hardware NN for Trojan-free chips

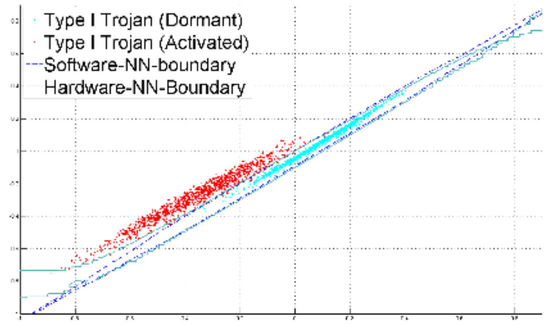

Fig. 11. Ability of boundary learned through software and hardware NN to correctly classify dormant and activated Type-I Trojan-infested chips

the Trojan is active). Figure 11 depicts the learned boundary, along with the footprints of the Type-I Trojan-infested chip instances with the Trojan dormant and active. Similarly, Figure 12 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 I and II 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 providing an effective post-deployment trust evaluation capability.

TABLE I
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 II
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%

V. CONCLUSIONS

Existing hardware Trojan detection methods, which are typically applied prior to deployment of an integrated chip, are unable to detect dormant hardware Trojans which are activated later in the operation field. Towards alleviating the threat of such dormant hardware Trojans, we introduced a general architecture for post-deployment trust evaluation, based on on-chip stimulus generation, parametric measurement acquisition, and classification into trusted and untrusted operation regions. Using a Trojan-free and two Trojan infested variants of a wireless cryptographic IC, along with a reconfigurable analog neural network experimentation chip, we demonstrated that the proposed method results in negligible false alarms and can effectively perform post-deployment trust evaluation.

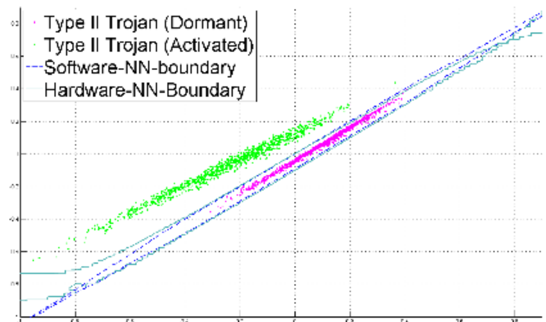

Fig. 12. Ability of boundary learned through software and hardware NN to correctly classify dormant and activated Type-II Trojan-infested chips

REFERENCES

[1] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty, "Towards Trojan-free trusted ICs: Problem analysis and detection scheme," in *IEEE Design Automation and Test in Europe*, 2008, pp. 1362–1365.

[2] H. Salmani, M. Tehranipoor, and J. Plusquellic, "New design strategy for improving hardware Trojan detection and reducing Trojan activation time," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2009, pp. 66–73.

[3] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *IEEE Symposium on Security and Privacy*, 2007, pp. 296–310.

[4] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 51–57.

[5] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic, "Power supply signal calibration techniques for improving detection resolution to hardware Trojans," in *IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 632–639.

[6] R. Rad, J. Plusquellic, and M. Tehranipoor, "Sensitivity analysis to hardware Trojans using power supply transient signals," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 3–7.

[7] Y. Jin, N. Kupp, and Y. Makris, "Experiences in hardware Trojan design and implementation," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2009, pp. 50–57.

[8] Y. Jin and Y. Makris, "Hardware Trojans in wireless cryptographic ICs," *IEEE Design and Test of Computers*, vol. 27, pp. 26–35, 2010.

[9] M. Cimino, H. Lapuyade, M. De Matos, T. Taris, Y. Deval, and J. Begueret, "A robust 130nm-cmos built-in current sensor dedicated to RF applications," in *IEEE European Test Symposium (ETS)*, 2006, pp. 151–158.

[10] L. Abdallah, H.-G. Stratigopoulos, C. Kelma, and S. Mir, "Sensors for built-in alternate RF test," in *IEEE European Test Symposium (ETS)*, 2010, pp. 49–54.

[11] D. Maliuk, H. Stratigopoulos, H. Huang, and Y. Makris, "Analog neural network design for RF built-in self-test," in *Proceedings of the IEEE International Test Conference (ITC)*, 2010, pp. 23.2.1–23.2.10.

[12] A. Skabar, "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*, 2003.