Trojan Detection using IC Fingerprinting[*]

Dakshi Agrawal[1] Selçuk Baktır[1,2,†] Deniz Karakoyunlu[2,†] Pankaj Rohatgi[1] Berk Sunar[2,†]

[1] IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598

[2] Electrical & Computer Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts, 01609

Abstract

Hardware manufacturers are increasingly outsourcing their IC fabrication work overseas due to their much lower cost structure. This poses a significant security risk for ICs used for critical military and business applications. Attackers can exploit this loss of control to substitute Trojan ICs for genuine ones or insert a Trojan circuit into the design or mask used for fabrication. We show that a technique borrowed from side-channel cryptanalysis can be used to mitigate this problem. Our approach uses noise modeling to construct a set of fingerprints *for an IC family utilizing side-channel information such as power, temperature, and electromagnetic (EM) profiles. The set of fingerprints can be developed using a few ICs from a batch and only these ICs would have to be invasively tested to ensure that they were all authentic. The remaining ICs are verified using statistical tests against the* fingerprints*. We describe the theoretical framework and present preliminary experimental results to show that this approach is viable by presenting results obtained by using power simulations performed on representative circuits with several different Trojan circuitry. These results show that Trojans that are 3–4 orders of magnitude smaller than the main circuit can be detected by signal processing techniques. While scaling our technique to detect even smaller Trojans in complex ICs with tens or hundreds of millions of transistors would require certain modifications to the IC design process, our results provide a starting point to address this important problem.*

1. Introduction

1.1. Problem Statement

Economic and market forces have driven most hardware manufacturers to outsource their IC fabrication to ever cheaper fabrication facilities abroad. As a result, the majority of the ICs available today are being manufactured at fabrication facilities in low-cost countries around the globe. While outsourcing of IC fabrication reduces the cost significantly, it also makes it much easier for an attacker to compromise the IC supply chain for sensitive commercial and defense applications. For example, the attacker could substitute Trojan ICs for genuine ICs during transit or subvert the fabrication process itself by implanting additional *Trojan circuitry* into the IC mask.

Such Trojans could be designed to be hard (or nearly impossible) to detect by purely functional testing, yet be capable of inflicting catastrophic damage. For example, a *Trojan circuit* could be designed so that it monitors for a specific but rare trigger condition, e.g., a specific bit pattern in received data packet or on a bus, or until a timer reaches a particular value. Once triggered the Trojan could take actions such as disabling the circuit, leaking secrets or creating glitches to compromise the integrity and security of the larger system to which the IC belongs. For example, a simple yet destructive Trojan in an RSA [24] circuit could wait for a trigger condition and then insert a fault in the CRT inversion step of an RSA signature computation leading to the compromise of the RSA key [6].

While this threat to the integrity of the IC supply is already a cause for alarm within defense circles in some countries [19, 9, 1], we believe that it should also be a cause for concern for vendors and consumers of commercial grade cryptographic and security critical hardware. Compounding this problem is the fact that currently there are no good, long-term solutions to this problem. While individual ICs

[*] This work was supported in part by a DARPA seedling grant under contract: Wyle Labs 19041.0C.25-111S, Stable IC Sensors and IC Fingerprinting.

[†] These authors' work was supported in part by the NSF-CAREER Award ANI-0133297.

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE

could be destructively reverse-engineered to check for fidelity to the original design, this does not guarantee that ICs not subjected to such testing are free of Trojans. For this reason, some governments have been subsidizing the operations of a few local, high-cost and economically unviable "trusted" fabrication plants for manufacturing military ICs. However most developing countries and commercial vendors cannot afford this expensive option. Another option suggested in [9] is camouflaging or obfuscation, where some critical IC designs are requested to be manufactured along with several non-critical ICs, or the true function of an IC is buried in confusing logic. However, this will not deter a motivated attacker who is willing to spend some effort in determining whether or not a critical IC is being manufactured, and subsequently in subverting the design. Apart from a few special cases, general circuit obfuscation has been shown to be impossible to achieve [4, 29, 14]. Secondly, an attacker may be able to implant a destructive Trojan without having to understand the details of its operation. For example, in the RSA Fault injection Trojan example given above, *almost any* disruption to one of the CRT exponentiations will do. We note here that *once IC manufacturing is secured*, subsequent risk of IC substitution in the supply-chain could be mitigated using existing techniques. For example, security critical chips could be designed so that an on-chip key gets created e.g, by using a silicon physically random, uncloneable functions [12], gets certified upon manufacture and subsequently protected using anti-tamper countermeasures.

1.2. Trojan Detection: A New Technique

While completely eliminating the threat posed by a compromised IC supply chain appears to be a daunting "grand challenge", in this paper, we propose a novel side-channel based approach that can be used for detecting the presence of Trojan circuitry in ICs that are practically impossible to detect using purely functional testing since they would be activated by a trigger condition such as a match between the signal on a bus or register and a certain bit pattern which would occur with very low probability during functional testing. Our technique requires just a few ICs to be destructively tested while permitting the rest of the ICs to be non-destructive validated using side-channel analysis for the absence of any significantly-sized (3–4 orders of magnitude smaller) Trojans. We believe that our technique would be part of any comprehensive approach that would be developed to deal with this threat.

Several side channels [16, 17, 22, 11, 3] and side-channel analysis techniques have emerged over the past decade and have proved to be highly effective in extracting information about the internal operations of embedded devices from their timing, power consumption and electromagnetic (EM) emanation profiles. Typically, an attacker tries to deduce critical information such as the encryption key using the leakage from these side-channels. An important aspect of side channel attack techniques is that these techniques are effective even though the information present within the side-channels could be masked by various types of noise, including measurement noise, ambient noise, and other random signal variations that manifest themselves during the circuit operation. While the initial attack techniques such as Differential Power/EM Analysis dealt with the problem of noise by averaging it out over multiple samples, later work on template attacks and its variants [7, 23] actually build statistical models for the noise and used them to classify individual noisy signals. In particular, these attacks use noise models built from one IC to attack another IC from the same mask. The success reported with these techniques provided the initial motivation to pursue side-channel analysis to detect Trojans ICs—the problem of Trojan detection essentially reduces to detecting a Trojan signal hiding in the IC *process noise*, i.e., the small, random, physical and side-channel differences among different ICs produced from the same process.

In this paper, we propose the following side-channel based *fingerprinting* methodology for detecting Trojan ICs. This initial approach does not require any changes to current processes and practices regarding the design and fabrication of ICs, and in particular, it does not require trusted fabs. However, it does require an additional IC fingerprint generation and validation step to be carried out by a trustworthy IC testing facility to gain assurance that the chances of Trojans being present in the validated ICs have been significantly lowered. The same testing strategy could even be used to increase the assurance of ICs manufactured from a trusted fab. The *fingerprinting* methodology consists of the following steps:

1. Select a few ICs at random from a family of ICs (i.e., ICs with the same mask and manufactured in the same fab).
2. Run sufficient I/O tests multiple times on the selected ICs so as to exercise all of their expected circuitry and collect one or more side-channel signals (power, EM, thermal emissions etc.) from the ICs during these tests.
3. Use these side-channel signals to build a "side-channel fingerprint" for the IC family.
4. Destructively test the selected ICs to validate that they are compliant to the original specifications.
5. All other ICs from the same family are non-destructively validated by subjecting them to the same I/O tests and validating that their side-channel signals are consistent with the "side-channel fingerprint" of the family.

In the second step, the challenge is to isolate a small and non-redundant set of tests that provide sufficient coverage of the IC's functionality. It is critical that the overall behavior of the IC in both the data and control paths is captured during these tests. In the third step, building the fingerprint requires characterizing the signal(s) and noise on different side-channels for the IC family for different inputs. The challenge here is to develop a characterization which is as comprehensive as possible without being impractical, and which is capable of distinguishing most Trojans from genuine ICs. We will provide details on this fingerprint generation in Section 3. The fourth step could utilize techniques such as demasking, delayering and layer-by-layer comparison of X-ray scans with the original mask. This step is likely to be expensive, but since it is only done for a few selected ICs from an entire family, the cost when amortized over all the ICs that are tested from that family may still be acceptable. Also, the fifth step should only be carried out if the destructive tests in the fourth step do not identify any problems with the ICs used to build the fingerprint.

We note that there has been some related work on using side-channels for testing and trapdoor detection. For example, power supply current profile monitoring is a common technique in testing ICs and identifying (non-adversarial) defects [15, 5, 13]. Lee, Jung and Lim [18] propose to use timing and power analysis techniques to detect hidden trapdoors in smartcards. Specifically, they consider power analysis to identify undisclosed instructions built into the system by the manufacturer for reprogramming (and recycling) wrongly issued smartcards. These works however do not address the problem of Trojan detection.

In this work, we will use power signals as the side channel and analyze the effectiveness of our fingerprinting methodology for detecting Trojans by using power simulations from sample cryptographic circuits implementing the Advanced Encryption Standard (AES) [2] and RSA algorithm [24] and different types and sizes of Trojans, including Trojans triggered by timing/clock counting and Trojans triggered by a synchronous/asynchronous comparator, with sizes ranging from 10% to 0.01% of the total IC size and for different levels of noise introduced by process variations (+/- 2%, 5%, 7.5%).

We would like to note that while it may be theoretically possible for an adversary to hide a Trojan from our tests by ensuring that its signal is so similar to process noise that it cannot be distinguished from it, this is likely to be extremely difficult and costly to accomplish. Firstly the adversary does not know what tests, side channels and localization techniques will be used for testing and cannot easily predict the process dependent noise that will be introduced in these channels. Channels such as EM are very hard to model and the adversary would have to resort to trial and error involving manufacturing actual ICs and testing them using a variety of tests and side-channels to make sure that these do not reveal the Trojans's presence. This should be contrasted with the ease with which one can put in a Trojan circuit that is hidden from functional testing.

The rest of the paper is structured as follows. In Section 2, we discuss the impact that Trojans have on the power side-channel and some simple tests that can be used to detect a large class of Trojans. In Section 3, we present the theoretical framework for detecting Trojans, describe advanced techniques that can be used in the presence of overwhelming process noise, and present our results in this setting. Section 4 describes the architecture for the circuits and Trojans used in our experiments and the setup used to perform our simulations. In Section 5 we present our experimental results. Finally, in Section 6 we present our conclusions and future work.

2. Trojans and their Side-channel Leakage

There are several types of Trojan circuits that could infest ICs, however most Trojan circuits share some behavioral characteristics that make them useful for the attacker. All Trojan circuits need to be *stealthy*, i.e., hard to detect either from the physical appearance of the IC or during its testing and normal use. This means that the Trojan IC has to have the same physical form-factor, pin-out and *very similar* input/output behavior, i.e., for most inputs, the output of an IC with a Trojan circuit should be indistinguishable from the output of a genuine IC. In particular, if the output is a deterministic function of the input, then the Trojan IC has to output the same function for most inputs[1]. For a deterministic circuit, this essentially means that the Trojan circuit needs to monitor inputs, intermediate results, or some clock/time circuitry and wait for a trigger condition before altering the output behavior either by producing incorrect results or by causing other failures. The trigger condition has to occur with very low probability during testing or normal usage, but could be invoked more frequently by the attacker. The trigger condition may also be chosen to occur after a certain time has elapsed. For non-deterministic circuits, e.g., those involving the use of IC generated randomness, the Trojan circuit could more easily encode information in the output without detection but still needs to be very selective (possibly trigger based) in producing detectably incorrect results or causing failure.

From the perspective of an attacker, it is fairly easy to manufacture elaborate complex Trojan ICs that look like the genuine ICs and have similar input/output behavior during testing and normal use. Modern IC manufacturing techniques leave a lot of room for inserting large, complex Trojan circuitry within the main circuit without impacting die

[1] This study did not consider analog Trojan ICs which get activated, operated, or communicated to from the outside by means of analog signals.

size or pin-outs and appropriate trigger conditions are easy to identify and implement. In Section 3, we will show that by using side-channel fingerprints, it is not too difficult to identify the presence of a Trojan. But first we will mention much simpler methods for detecting Trojans.

2.1. Trojan Detection via Simple Side-channel Analysis

Even simple side-channel analysis can detect many types of Trojan circuits. For example, a Trojan IC whose timing is different for test inputs will get detected by timing analysis, or a Trojan IC which demonstrates a significantly different behavior compared to a genuine IC at any time during the entire computation on the test data will get detected using a *Simple Power Analysis* (SPA)/*Simple Electromagnetic Analysis* (SEMA) [22, 11, 3] like technique on a single power/EM trace.

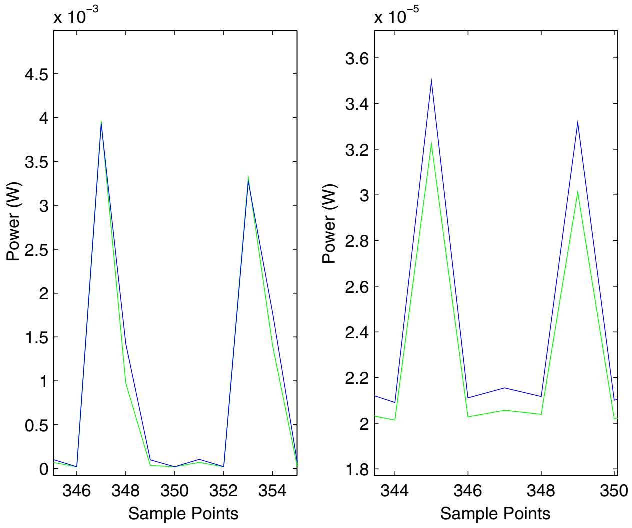

Figure 1. Genuine (green/grey) and Trojan (blue/black) AES signals at 100MHz (left) and 500Khz (right).

Another example of Trojan ICs that can be detected by simple side-channel analysis are the ICs that have a relatively large Trojan circuitry even if the circuitry remains mostly inactive on the test data. This is because of the fact that the total power consumption in a digital circuit comprises the dynamic power consumption and the leakage power consumption whose individual contributions are given by the following equation [8]:

$$P = \underbrace{(\frac{1}{2} \cdot C \cdot V_{DD}^2 + Q_{se} \cdot V_{DD}) \cdot f \cdot N}_{dynamic\ \ power} \quad + \quad \underbrace{I_{leak} \cdot V_{DD}}_{leakage\ \ power}$$

where C, Q_{se} and V_{DD} are technology dependent parameters, N represents the switching activity and f is the clock frequency. Note here that the leakage current, I_{leak}, depends only on the number of gates in the circuit and the fabrication technology. Since the dynamic power is linearly dependent on the clock frequency and the switching activity, and the leakage power depends only on the circuit area, one can discover a large Trojan circuit simply by running the ICs at a very low clock frequency. For example, Figure 1 shows the details of the power signals from a non-Trojan AES [2] circuit (green or grey) with an equivalent area of 4302 2-input NAND gates and a Trojan AES circuit (blue or black) with a 10-bit counter as the Trojan which has an equivalent area of 247 2-input NAND gates. On the left the circuit is clocked at 100 MHz and sampled at 1 ns intervals and on the right the circuit is clocked at 500 KHz and sampled at 200 ns intervals. The Trojan in this case is roughly 5.6% of the total circuit size. At 100 MHz it is difficult to distinguish between the baseline power consumption of the Trojan and non-Trojan ICs, however, the difference between their baseline power consumption is obvious at 500 KHz. We would like to note that the functionality of a dynamic circuit depends on a minimum clock frequency which is required to refresh the storage nodes. This may be a limitation for testing a circuit at extremely low clock frequencies (below the minimum available clock frequency) unless one utilizes special design techniques for refreshing the storage nodes.

In fact the only types of Trojan ICs that can survive the simple side-channel tests described above are those which contain a small Trojan circuit and perform computations essentially very similar to that done by the genuine IC to produce the expected results and to maintain the side channel signal shape on test inputs. The additional computation on the test data performed by such Trojan ICs may be relatively simple such as testing for a trigger condition, monitoring or otherwise storing information in preparation for changing I/O behavior if activated.

Since the additional computation performed by a Trojan IC on the test data has to be simple to avoid simple side-channel tests, one may expect that distinguishing power/EM signatures for such Trojan ICs could be hidden within the signal measurement noise. However, signal measurement noise can be easily eliminated by averaging over many signals obtained with the same input. Using averages over many power/EM traces, even small power/EM contributions from the Trojan circuit relative to the main circuit can be picked up. Figure 2 shows a simulation of how an average power signal would look like for the Trojan and genuine AES circuits running at 100MHz. The signal from the genuine AES circuit is shown in green (or grey) and the *additional signal* introduced by the Trojan circuit is shown in black. This additional signal riding on top of the expected

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE

non-Trojan signal is so large that it will easily stand out, once signal measurement noise is reduced.

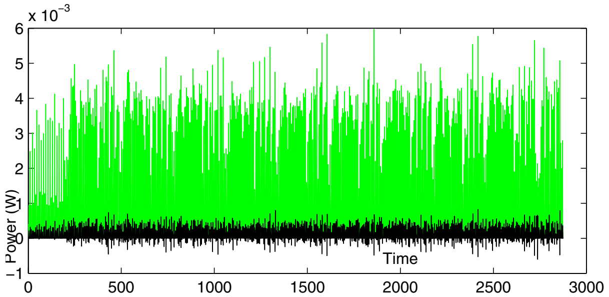

Figure 2. Genuine AES signal in green/grey overlayed with Trojan contribution in black.

The effectiveness of the averaging based detection technique above is only limited by the effect of process noise on side-channel signals: No two ICs are exactly identical, and process variations between different ICs manufactured from the same specification manifest as slight differences in the average side-channel signals for the same computation. The process noise is likely to be much smaller than the measurement noise and the Trojan ICs that have a minimal Power/EM footprint on the test data may escape detection using signal averaging tests. However, as we will show in the next section, just having a Trojan IC with a power/EM profile during testing that is comparable to or less than the process noise will not suffice to protect a Trojan IC against more sophisticated testing and side channel analysis.

3. Trojan Detection Theory

Consider an IC I, that executes a calculation C. Also, consider a power measurement M done on I when it is executing the computation C. The power trace obtained in this measurement, $r(t; I; C; M)$, can be modeled as consisting of four components: (a) the mean power consumption $p(t; C)$ (the mean is computed over several measurements done on several ICs from the same family during multiple executions of the calculation C), (b) process noise $n_p(t; I; C)$, (c) measurement noise $n_m(t; M)$, and (d) possibly an extra power leakage $\tau(t; I; C)$ due to a Trojan circuit in I. Note that in our model, the process noise $n_p(t; I; C)$ and the extra signal injected by the Trojan circuit $\tau(t; I; C)$ may depend on the particular IC I, and the executed calculation C. Thus in our model the power trace of a genuine IC is given by

$$r_G(t; I; C; M) = p(t; C) + n_p(t; I; C) + n_m(t; M) \ ,$$

and the Trojan IC adds an additional component to give

$$r_T(t; I; C; M) = p(t; C) + n_p(t; I; C) + n_m(t; M) \\ + \tau(t; I; C) \ .$$

Typically, the measurement noise $n_m(t; M)$ is a random noise that varies on each measurement. Since none of the other components of the power signal depends on the measurement M, the random noise $n_m(t; M)$ can be eliminated by averaging over a large number of measurements taken from the same IC. Therefore, in the rest of this paper, we will ignore the measurement noise, and work with the following models of the power traces for genuine and Trojan ICs, respectively:

$$\begin{aligned} r_G(t; I; C) &= p(t; C) + n_p(t; I; C), \\ r_T(t; I; C) &= p(t; C) + n_p(t; I; C) + \tau(t; I; C) \ . \end{aligned}$$

If we assume that we have access to multiple genuine ICs, then we can compute the mean of $r_G(t; I; C)$ over several genuine ICs to eliminate the process noise $n_p(t; I; C)$ and calculate the mean power consumption $p(t; C)$ that occurs during the calculation C. Since the mean power consumption $p(t; C)$ is common among the power traces obtained from both the genuine and the Trojan ICs, it can also be subtracted out from all the power traces, and hence for our analysis, we can assume that,

$$\begin{aligned} r_G(t; I; C) &= n_p(t; I; C), \\ r_T(t; I; C) &= n_p(t; I; C) + \tau(t; I; C) \ . \end{aligned}$$

We model the Trojan detection problem as follows.

Definition 1 *Trojan Detection Problem. Given K genuine ICs I_1, I_2, $\ldots$, I_K and the process noise signals $n_p(t; I_1; C)$, $n_p(t; I_2; C)$, $\ldots$, $n_p(t; I_K; C)$ generated by the ICs I_1, I_2, $\ldots$, I_K, respectively, during the execution of the calculation C, and given an IC I_{K+1} with a mean power trace $r(t; I_{K+1}; C)$ (mean taken over multiple executions of calculation C with the average $p(t; C)$ subtracted), how can we determine if the IC I_{K+1} contains a Trojan circuit?*

In other words, we have the following two hypothesis for genuine and Trojan ICs, respectively, and our goal is to detect the correct hypothesis.

1. $H_G : r(t; I_{K+1}; C) = n_p(t; I_{K+1}; C)$
2. $H_T : r(t; I_{K+1}; C) = n_p(t; I_{K+1}; C) + \tau(t; I_{K+1}; C)$

The Trojan detection problem above can be viewed as a signal characterization problem. We need to characterize the process noise $n_p(t; I; C)$ and check if the signal under hypothesis testing differs from the process noise. One

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE

powerful technique to find such characteristics is *subspace projection*, where the signal $r(t; I_{K+1}; C)$ and the process noise signals $n_p(t; I_1; C)$, $n_p(t; I_2; C)$, $\ldots$, $n_p(t; I_K; C)$ from known genuine ICs are projected in a signal subspace where signals from Trojan and genuine ICs are likely to have different characteristics. The main obstacle in this analysis is that we do not know the Trojan circuit or what precisely it may be trying to accomplish. The Trojan IC may be monitoring the clock, contents of a register, or transitions on a bus. The power consumed by the Trojan may be correlated with the clock, input or output data, result of some intermediate calculation, etc. In absence of this knowledge apriori, it may seem that nothing short of a full characterization of the process noise would work.

However, in our initial experiments with simulated Trojan ICs, we could easily find signal subspaces where characteristics of the genuine and Trojan ICs differed considerably without having to perform a full characterization of the process noise. For example, consider an otherwise small Trojan whose power consumption does not fall when the genuine IC's power consumption falls. This may be the case because the process noise within a signal trace is correlated to the signal amplitude and thus drops off when the IC is not consuming much power. An analysis performed on the power traces at such points in time will easily pick up the Trojan.

As an illustration, Figure 3 shows an RSA computation (in green or gray) with the process noise (in red or dark grey) and the Trojan signal (in black) simulated via a $\pm 5\%$ random variation in cell libraries across processes. The RSA signal shows periods of high power consumption corresponding to each modular multiplication operation, separated by a short time-interval of low power consumption in between these multiplications. The Trojan power signature in this case is much smaller than the process noise in general. However, in contrast, note that the process noise is much smaller than the Trojan power signature and is largely zero in between the modular multiplications, while the Trojan signal displays a regular structure both during and in between the modular multiplications since it is counting clocks. Figure 4 shows the Trojan power signature and process noise in one such region of low activity in between two modular multiplications— in such regions, with its relatively much larger magnitude the Trojan contribution to the signal (black) stands out compared to the process noise (green or grey).

Even when the IC's power consumption, and therefore the correlated process noise, does not fall relative to the Trojan at any point in the computation, the Trojan ICs can be detected by using advanced signal processing techniques. In the rest of this paper, we will demonstrate the use of Karhunen-Loève (KL) expansion [28, 10, 21] to detect Trojan ICs. Using this technique we were able to determine

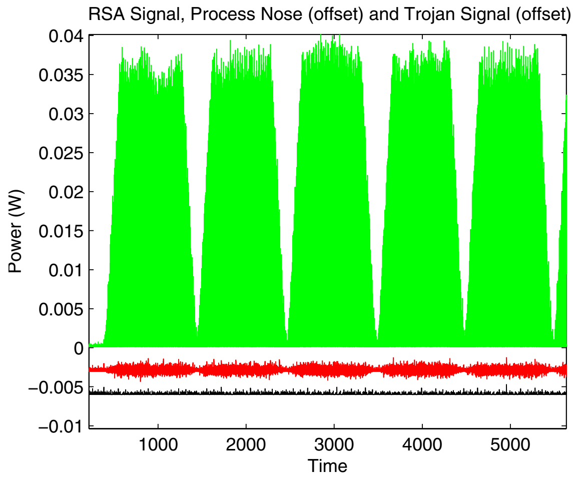

Figure 3. Genuine RSA signal (top: green or grey), process noise(middle: red or dark grey) and Trojan contribution (bottom: black).

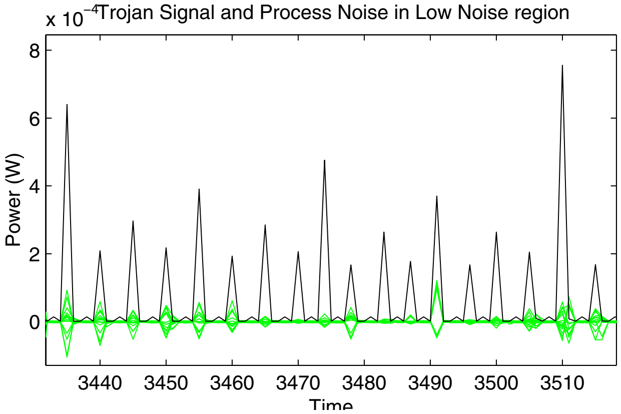

Figure 4. Trojan (black) vs. process noise (green or grey) in between two modular multiplications.

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE
IEEE COMPUTER SOCIETY

all Trojans introduced into our RSA circuit under many different process-noise assumptions. Before we go into the results obtained by using the KL expansion, we will briefly describe its mathematical foundation.

Theorem 1 *Suppose that* $\{Y_t, t \in [0,T]\}$ *is a zero-mean second-order random process with autocovariance function* $C_Y(t,u)$ *that is continuous on* $[0,T]^2$*, then* C_Y *can be expanded in the uniformly and absolutely convergent series*

$$C_Y(t,u) = \sum_{k=1}^{\infty} \lambda_k \psi_k(t)\psi_k(u), \qquad (t,u) \in [0,T]^2, \quad (1)$$

where λ_k *and* ψ_k*, for* $k = 1, \ldots, \infty$ *are* eigenvalues *and corresponding orthonormal eigenfunctions of* C_Y*. Furthermore,* $\{Y_t, t \in [0,T]\}$ *can be represented by the following mean-square convergent series:*

$$Y_t = \sum_{k=1}^{\infty} Z_k \psi_k(t), 0 \geq t \geq T. \qquad (2)$$

where $\{Z_k\}_{k=1}^{\infty}$ *is referred to as the* KL coefficients *of the sample* Y_t*.*

While a discussion of the mathematical technicalities in the above theorem is clearly outside the scope of this paper; we note that the conditions under which the above theorem holds are very mild and easy to satisfy [21].

The KL expansion essentially provides a separation of the randomness and the time-variation of a random process: the sequence $\{Z_k\}_{k=1}^{\infty}$, loosely referred to as the eigenvalue spectrum of a sample, varies from sample to sample, and has no time dependency, while $\psi_k(t)$, eigenvectors of the random process, are fixed from sample to sample, but vary with time. Since we are interested in a characterization of the randomness of the signals, we can just focus on $\{Z_k\}_{k=1}^{\infty}$.

The variance of the random variable Z_k is given by λ_k. Thus, if for a random process Y_t, eigenvectors ψ_k are arranged such that their corresponding eigenvalues λ_k are monotonically decreasing in k, then for a sufficiently large positive integer K, the sequence, $Z_1, \ldots, Z_K$, captures most of the randomness contained in the process. Furthermore, for $k > K$, Z_k is close to zero (and has close to zero variance) for any sample from the random process Y_t. In other words, the random signal Y_t "lives" in a signal subspace spanned by $\psi_1(t), \ldots, \psi_K(t)$, and it is absent from the signal subspaces spanned by $\psi_{K+1}(t), \psi_{K+2}(t), \ldots$.

We can exploit this fact to find Trojan ICs. By using the KL expansion, we can find a signal subspace from which the process noise is absent. Unless the Trojan signals completely live in the same subspace as the process noise, the projection from the samples of Trojan signals in this subspace (say given by $Z_{K+1}, Z_{K+2}, \ldots$) would be non-zero and show large variability in contrast to similar projection from the samples of process noise which will be close to zero and show close to zero variance (see Sections 5.1, 5.2, and 5.3).

If the Trojan signal was *very* small, there is a possibility that the Trojan signal may completely live in the same subspace as the the process noise. However, it is unlikely that its spectrum characteristics would be *exactly* the same as the process noise, e.g. in terms of mean and variance of the eigenvalue spectrum. Such differences can be captured by a statistical analysis of the spectrum (see Section 5.4). Here we note a strong analogy of our technique to the emissions spectroscopy technique used to detect trace amounts of metals in a material. Each metal has a unique emission spectrum, and by comparing these spectrums emissions spectroscopy can be used to detect minutes amounts of trace metals (as small as 10 ppb) in a material.

We will now describe our experimental setup in Section 4, followed by a discussion of results for different Trojan circuits and process noises in Section 5.

4. Experimental Setup

4.1. ICs Used in Our Analysis

We used synthesized RSA circuits [24] for the analysis presented in the rest of this paper. RSA circuits are used in many systems and they are of high value to the attackers. The Trojan added to these circuits was a simple counter or comparator. In a counter based Trojan, the Trojan circuit counts clock cycles and disables the IC after a threshold is reached. In the case of a comparator based Trojan, the Trojan circuit compares the data in a bus or a register against a fixed value and alters the computation if there is a match.

4.1.1 RSA Circuits

Our RSA design employs the left-to-right binary square and multiply exponentiation algorithm. We employ a scalable, pipelined and high radix Montgomery Multiplier (MM) architecture to realize square or multiply operations. The operand length, word size, and pipeline depth in the MM circuit are parameterized. All simulation results in this paper were obtained for a pipeline depth of 8 and word size of 8 bits. We chose these values of pipeline depth and word size to have reasonable circuit area and speed that are in appropriate ranges for real life applications. The memories to hold operands, exponent and modulus, and the FIFO memory necessary for the pipeline structure are omitted from the synthesized RSA circuit. Our RSA circuit design is shown in the Appendix.

In order to generate a power trace, the circuit was resynthesized, flattened, power optimized, mapped, and then

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE

finally analyzed for power. Moreover, each step requires simulation for back-annotation. The execution time of the square and multiply algorithm is $O(n)$, and the execution time of MM algorithm is $O(n^2)$. This leads to an overall execution time of $O(n^3)$. Therefore, the simulation time as well as the size of the intermediate files used to capture the switching activity grows very quickly. For instance, for 512-bit operands and a 16-bit short exponent, the generation of a single power trace takes about 4 hours on a high-end workstation. Due to these constraints, in our simulations we used RSA circuits for only 256-bit or 512-bit operands rather than the typical sizes of 1024-bit or more. The 256-bit RSA circuit we used has an equivalent area of 27909 2-input NAND gates and average power consumption of 2.239 mW, whereas the 512-bit RSA circuit has an equivalent area of 27914 2-input NAND gates and average power consumption of 3.001 mW. Both the 256-bit and 512-bit RSA circuits had a maximum clock frequency of 617 MHz.

4.1.2 Trojan Circuits

In our experiments, we used three different Trojan circuits. The first Trojan circuit was a 16-bit counter with an equivalent area of 406 2-input NAND gates which occupies roughly 1.4% of the total circuit area of the RSA circuits described earlier. The second Trojan circuit was a simple 8-bit *sequential* comparator with an equivalent area of 33 2-input NAND gates. As a final attempt, in order to test the limits of our technique, we used an even simpler 3-bit *combinational* comparator with an equivalent area of only 3 2-input NAND gates. Figures 5 and 6 show the VHDL code for the simple 8-bit sequential and 3-bit combinational Trojan comparator circuits. Note that the area of Trojan circuits used in our experiments goes from 406 gates to 33 gates to 3 gates, roughly an order of magnitude decrease at each step.

4.2. Testbed Used for Circuit Simulation and Power Trace Generation

We used Synopsys Core Synthesis Tools [25] with the 0.13 μm, 1.0 V technology library *tcb0131vhptc* of Taiwan Semiconductor Manufacturing Company (TSMC) for the synthesis of the RSA circuits with and without the Trojan. We also used ModelSim SE/PE 5.7g [20] for simulation and switching activity analysis and Synopsys PrimePower StandAlone [26] for power analysis. Then we conducted simulations and obtained power traces for different scenarios. We ran our simulations at 50 MHz clock frequency.

4.3. Modeling Ambient Noise and Process Variations

Since no two ICs are identical even if they use the same masks and go through the same fabrication process, their side-channel signals differ even for the same input. We call this variation *process noise*. In our experiments, we modeled the process noise by randomly altering the parameters of the TSMC technology library in the range of ±2%, ±5% or ±7.5%. Each different variation of parameter values represents a different physical circuit manufactured through the same process.

5. Experimental Results

5.1. Experiment 1: 512-Bit RSA Circuit with a 16-Bit Counter Based Trojan and with ±2% Parameter Variations

For this experiment, we used the 512-bit RSA circuit with the 16-bit counter based Trojan. Recall that the 16-bit counter based Trojan has an equivalent area of 407 2-input NAND gates, and it occupies roughly 1.4% of the total circuit area.

To emulate the process noise, we introduced ±2% random variations in the library parameters to obtain 15 new libraries and compiled them using the Synopsys Library Compiler [27]. We then used each compiled library (typical library as well as 15 new libraries) to synthesize 16 genuine ICs and 16 ICs with Trojan. We then conducted power simulations and obtained 16 traces for RSA and 16 traces for the Trojaned RSA.

Figure 7 shows the eigenvalue spectrum of the signals (40 contiguous sample points) taken in the middle of a modular multiplication operation where the process noise is the highest in amplitude. The spectrum for ICs with Trojans are plotted in blue (or black) and the spectrum for the genuine ICs are plotted in green (or grey). As we can see in Figure 7, even though we analyzed a very short trace segment, the eigenvalue spectrums of the genuine ICs and ICs with Trojans stand apart on the eigenvectors 12 and 14, yielding a simple test to detect Trojan ICs.

Note that the power-traces used to derive the results given above were obtained by using a single key value. We have verified that the value of the used key does not change our results.

5.2. Experiment 2: 256-Bit RSA Circuit with the 16-Bit Counter Based Trojan and with ±5% Parameter Variations

The main goal of this experiment is to emulate larger process variations. To expedite power-trace simulations, we used the 256-bit RSA circuit instead of using the 512-bit RSA circuit in this experiment. This circuit has approximately the same area as the 512-bit circuit, thus preserving the ratio of Trojan area to total circuit area.

```
% Create garbage data.
constant wdata_z         : std_logic_vector(WORD_SIZE-1 downto 0) := (others =>'Z');
process (clk, reset)
begin
        if reset = '0' then
             Trojan <= '0';                          % On Reset set Trojan register bit to 0
        elsif (clk'event and clk = '1') then
        % On clock do 8-bit comparison of bus with fixed value and set Trojan bit if there is a match
           Trojan <= (((d_from_fifo(0) nand '1')) and (d_from_fifo(1) nand '1') and ((d_from_fifo(2)
           nand '1')) and (d_from_fifo(3) and '1')  and ((d_from_fifo(4) and '1'))  and (d_from_fifo(5)
           nand '1')  and ((d_from_fifo(6) and '1'))  and (d_from_fifo(7) and '1')) ;
        end if;
 end process;
 % If Trojan register bit is set then output garbage results instead of actual results
 with Trojan select wdata   <= wdata_z           when '1',
                        wdata_actual    when others;
```

Figure 5. VHDL code for the 8-bit sequential Trojan comparator circuit.

```
% Trojan: output of combinatorial circuit comparing 3 data/exponent bits with fixed value

Trojan    <= ((d_from_fifo(0) nand '1') and (e_data(1) and '1') and (e_data(0) and '1')) ;

% Trojan output ORed into last bit of data used in calculation

wdata   <= wdata_actual(7 downto 1) & (wdata_actual(0) or Trojan);
```

Figure 6. VHDL code for the 3-bit combinational Trojan comparator circuit.

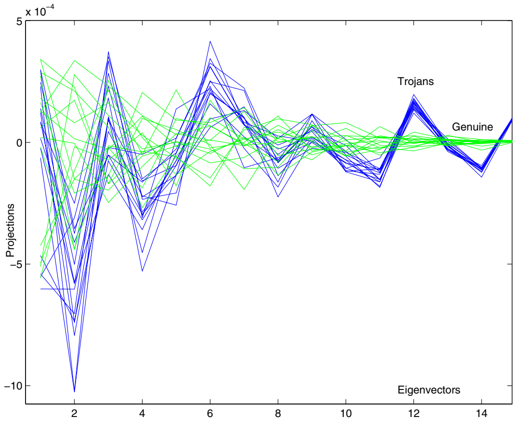

Figure 7. Projections of power traces from genuine and Trojan ICs on process noise eigenvectors, Experiment 1.

To emulate larger process noise, we increased the range of random library-parameter variations to $\pm 5\%$, and compiled them using the Synopsys Library Compiler [27]. We then used each compiled library (typical library as well as 15 new libraries) to synthesize 16 genuine ICs and 16 ICs with Trojan just as was done for the first experiment. For each synthesized circuit, we conducted power-trace simulations.

However, this time, for both the genuine and Trojan RSA circuits, 4 out of 32 power traces, corresponding to two specific technology library variations, resulted in abnormal behavior at exactly the same region of the power traces. We believe this behavior is caused by the increased amount of random variation in the technology library that may break the piecewise linear model for some VLSI cells. In turn, that may lead to anomalous signals being produced during simulations. In our analysis, we ignored the anomalous power traces, and proceeded with only 28 power traces.

Figure 8 shows the eigenvalue spectrum of the power-trace signals (40 contiguous sample points) taken in the middle of a modular multiplication operation where the process noise is the highest in amplitude. With the larger parameter variations, in this case, the process noise has roughly the same magnitude as the extra power leakage caused by the Trojan. However, even in this case, the traces from genuine and Trojan circuits clearly differ in the 12-th and 13-th eigenvectors. Thus once again, the eigenvalue spectrum yields a simple test to distinguish Trojan ICs from the genuine ones.

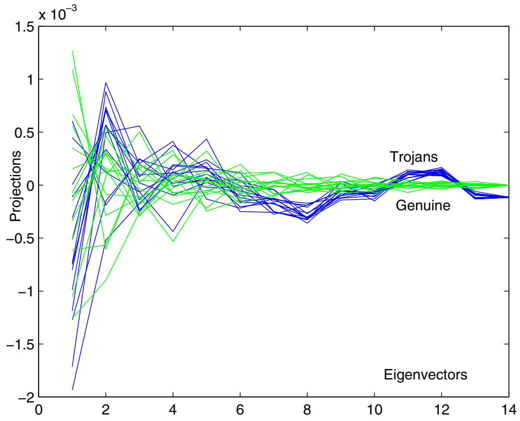

Figure 8. Projections of power traces from genuine and Trojan ICs on process noise eigenvectors, Experiment 2.

5.3. Experiment 3: 256-bit RSA Circuit with the 8-bit Sequential Comparator Based Trojan and with ±5% Parameter Variations

The goal of this experiment is to push the limits of our technique even harder by decreasing the size of the Trojan circuit by an order of magnitude. The 8-bit *sequential* comparator based Trojan used in this experiment has an equivalent area of 33 2-input NAND gates constituting only 0.12% of the total circuit area. We kept the parameter variations in the same range of ±5% as used in the second experiment. It turns out that with the smaller Trojan circuit, the power-trace contribution of the Trojan is now much smaller in magnitude than the process noise (see Figure 9).

In the resulting power traces, we found that either the Trojan signals are completely enveloped inside the larger process-noise signals (Figure 10) or the Trojan signals step out of the process-noise envelop at certain sample points (Figure 10). We found that in either case, it is possible to detect the Trojan signals by using the KL analysis. It was interesting to note that Trojan detection is easier when the Trojan signals step out of the process-noise envelope. Figure 11 shows the eigenvalue spectrums of the genuine ICs and the ICs with Trojan obtained by taking 30 contiguous sample points in time. Note that in the first case, when Trojan signals are completely within the process-noise envelope, the traces from the genuine and Trojan circuits clearly separate in the 14-th and 15-th eigenvectors. In contrast, in the second case, when Trojan signals step out of the process-noise envelope, the traces start separating earlier in the 8-th eigenvector and separate more often, e.g. at the 11-th, 13-th, 15-th, 17-th ,18-th and 20-th eigenvectors.

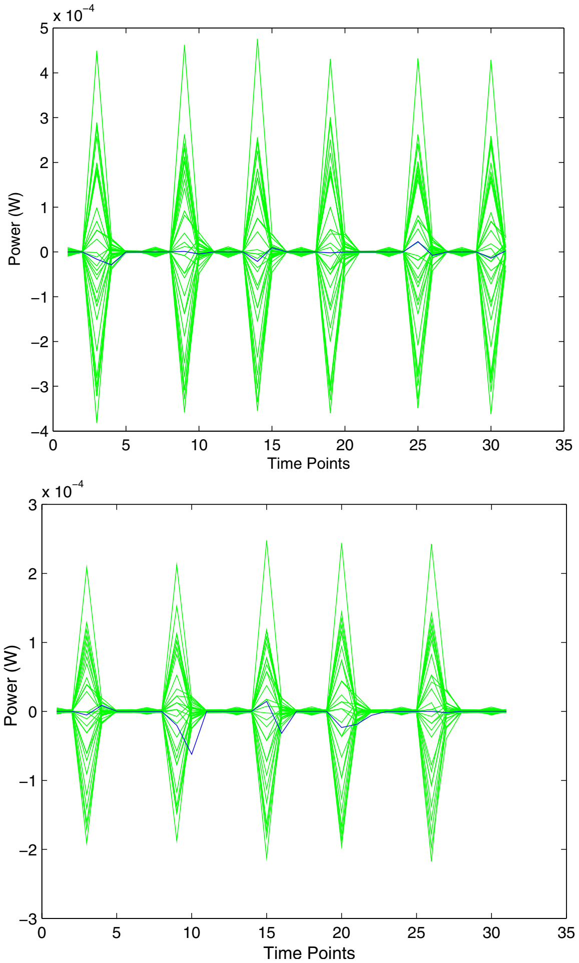

Figure 10. Trojan signals (blue or black) inside (top figure) and outside (bottom figure) the process noise envelopes (green or grey), Experiment 3.

5.4. Experiment 4: 256-bit RSA Circuit with the 3-bit Combinational Comparator Based Trojan and with ±7.5% Parameter Variations

Finally, in order to explore the limit of the proposed technique, we shrunk the Trojan size by another order of magnitude and increased the range of parameter variations to ±7.5%. We used the simple 3-bit *combinational* comparator based Trojan which has an equivalent area of only 3 2-input NAND gates constituting only 0.01% of the total circuit area. As shown in Figure 12, in this case, the process noise completely overwhelms the Trojan signal.

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE

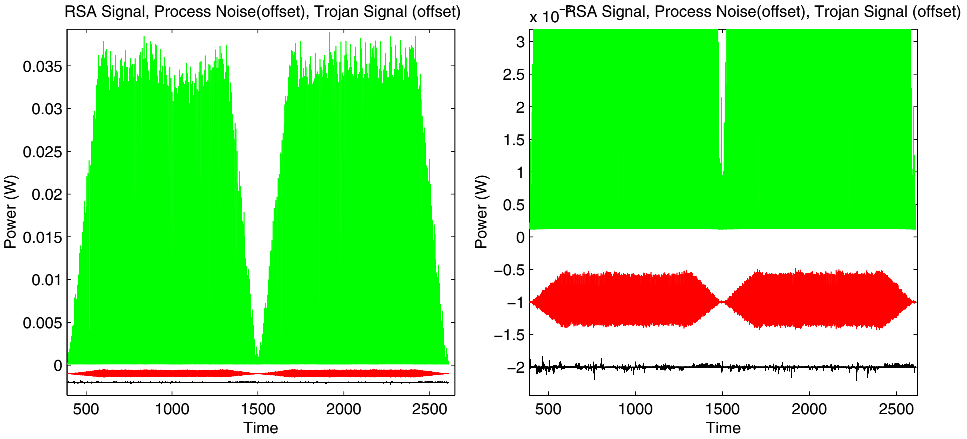

Figure 9. Genuine RSA (top:green or grey), process noise (offset, middle: red or dark grey) and Trojan (offset, bottom: black), zoomed in on the right.

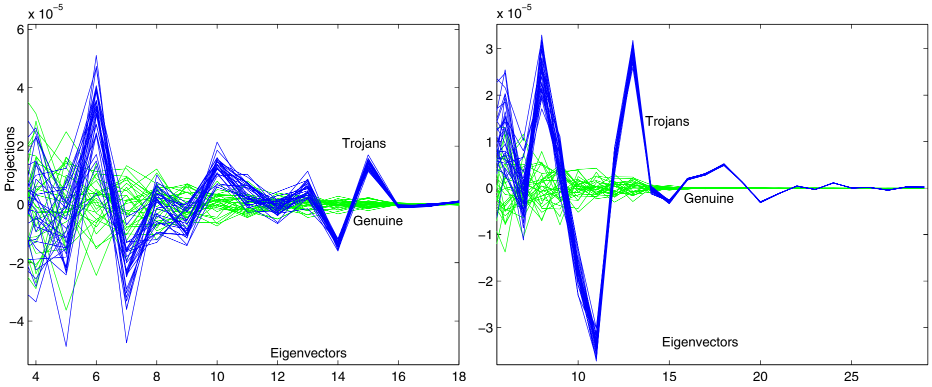

Figure 11. Projections of power traces from genuine (green or grey) and Trojan (blue or black) ICs on process noise eigenvectors when Trojan signal is hidden inside (left figure) and stepping out of (right figure) the process noise envelopes, Experiment 3: Case 1.

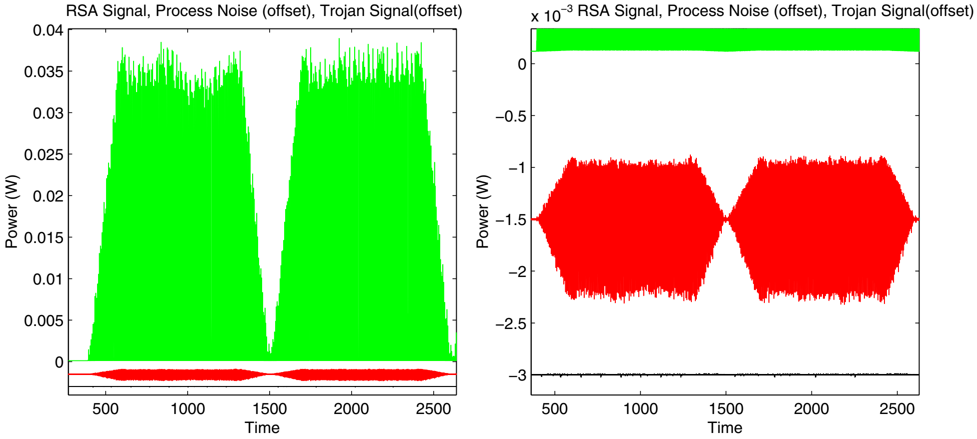

Figure 12. Genuine RSA (top: green or grey) with process noise (offset, middle: red or grey) and Trojan (offset, bottom: black) on the left and a zoomed in version on the right.

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE

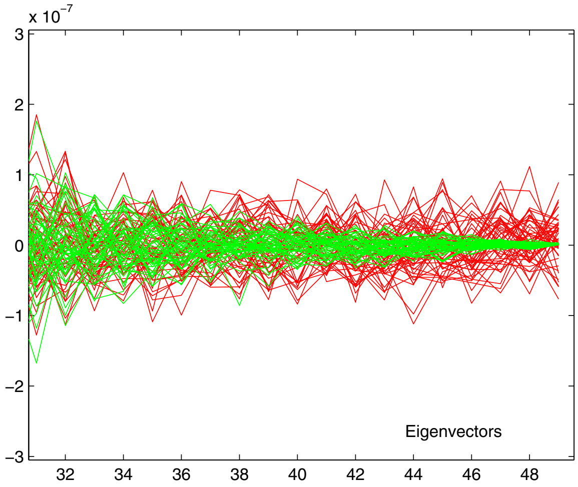

Figure 13. Eigenvalue spectrums of signals from genuine (green or grey) and Trojan (red or dark grey) ICs, Experiment 4.

Figure 13 shows the eigenvalue spectrums of the power-traces obtained from genuine ICs and ICs with Trojan, obtained by taking 50 contiguous sample points. Note that in this case the eigenvalue spectrum from the genuine and Trojan ICs do not separate and are not distinguishable. So Trojan detection finally seems to be a challenge. We tried the following two ad-hoc approaches to remedy the situation.

5.4.1 Approach 1. Statistical Analysis of the Eigenvalue Spectrums

In this approach, we first partition all signals into disjoint, contiguous time windows. Then we perform the KL analysis for each window by finding the eigenvectors of the process-noise using only the non-Trojan signals. Then we project the non-Trojan signals on to these eigenvectors and determine the mean μ and the standard deviation σ of the spectrum for each eigenvector. To check whether any given signal belongs to a genuine IC or an IC with Trojan, we find its eigenvalue spectrum by projecting it on the process-noise eigenvectors. If the eigenvalue spectrum is outside the $\mu \pm 4\sigma$ envelope for any eigenvector and for any time window, then we make the assertion that the signal is coming from an IC with Trojan. Likewise, if the whole spectrum stays inside the $\mu \pm 4\sigma$ envelope for all time windows, then we make the assertion that the signal is coming from a genuine IC (see Figure 14).

Using this statistical approach, we tested 49 ICs with Trojan using 70 time windows and achieved 100% success rate in detecting them as Trojan. We also tested 49 genuine ICs using the same 70 time windows and only one of them was falsely detected as Trojan resulting in a 2% false Trojan assertion rate. This single false Trojan case can be seen on the right of Figure 14. Figure 15 provides an alternative perspective on the efficacy of this approach. Clearly, our statistical method for detecting very small Trojans requires improvements and we need to develop the theory in this area more.

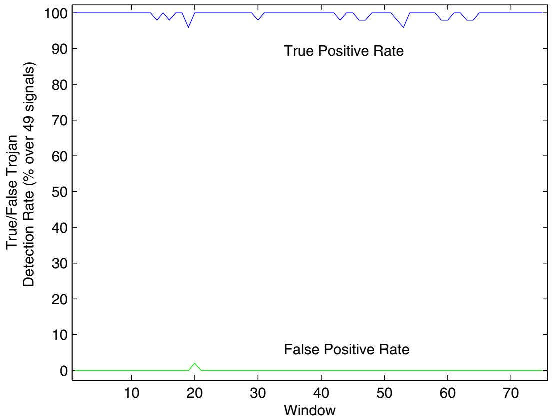

Figure 15. Percentage rates of true (blue or black) and false (green or grey) Trojan detection over adjacent time-windows, Experiment 4: Approach 1.

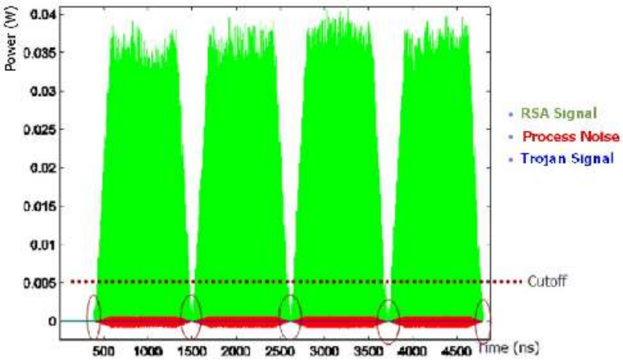

Figure 16. Experiment 4, Approach 2: Filtering and using only the data in low process noise regions.

5.4.2 Approach 2. Filtering the Signals and Focusing on Low Noise Regions

Another idea for Trojan detection for such high process-noise is to go back to the original KL analysis and focus on

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE

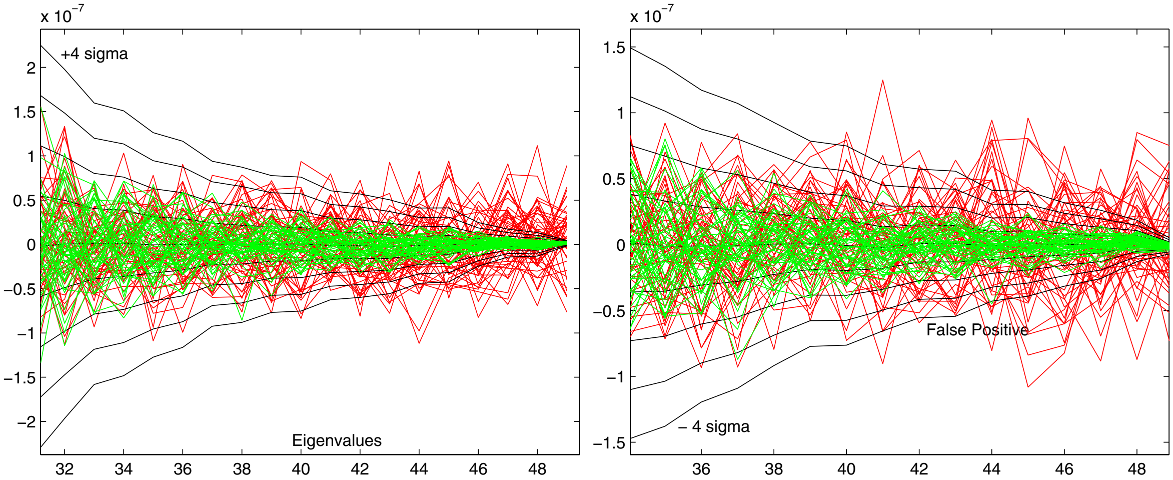

Figure 14. Eigenvalue spectrums of genuine (green or grey) and Trojan (red or dark grey) ICs with $\mu + i \times \sigma$ envelopes in black (left figure). False Trojan detection (right figure). Experiment 4: Approach 1.

the regions that have lower process-noise (see Figures 16 and 17). Note that unlike the earlier case described in Section 2.1, where in low noise regions the Trojan overwhelmed the process noise, in this case, it is the other way around—process noise is overwhelming. However, in the KL analysis, we are able to obtain a clear separation between the eigenvalue spectrums of the signals from the genuine and Trojan ICs at multiple eigenvectors. In particular, Figure 18 shows that for our experiments the eigenvalue spectrums of the genuine ICs and ICs with Trojans stand apart on the eigenvectors 43, 46 and 48.

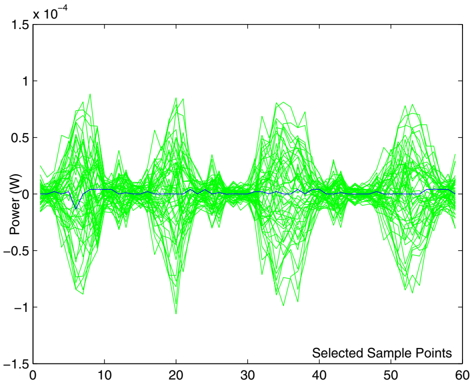

Figure 17. Process noise (green or grey) and Trojan signal (blue or black) observed in low noise regions, Experiment 4: Approach 2.

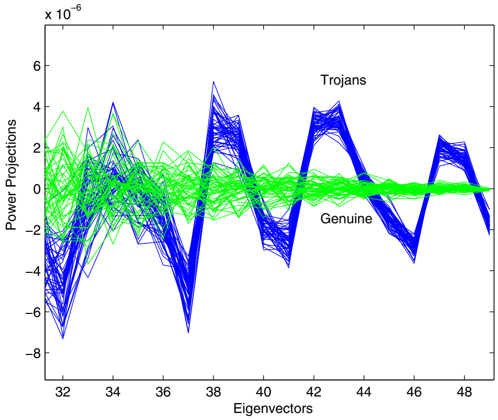

Figure 18. KL Analysis at low noise regions, Experiment 4: Approach 2.

6. Conclusions

In this paper, we demonstrated the feasibility of building effective fingerprints for an IC family to detect Trojan ICs. We designed and synthesized an RSA circuit and three different Trojan circuits. We used the power traces obtained from the simulations of these circuits to built the IC fingerprints. We modeled three sets of process variations by creating random variations (up to $\pm 2\%$, $\pm 5\%$ and $\pm 7.5\%$) in the cell libraries that were used to synthesize the designs. In all cases, fairly simple analysis of the power signals could distinguish genuine ICs from those containing Trojan circuits down to 0.01% of the size of the main circuit.

We showed that in general it is difficult to hide signal distortions introduced by a Trojan circuit as a Trojan circuit leaks signal in signal subspaces that are not present in genuine ICs. For many Trojans, these subspaces are easy to find, for example, the signal from a Trojan circuit may leak when the rest of the circuit is not active. We showed that even when Trojan signal is well-hidden within the variations of the signals generated by the process-noise, it can be detected by signal processing techniques.

As discussed in the introduction, we believe that even an adversary with general knowledge of this fingerprinting technique will incur great difficulty and cost in manufacturing a Trojan that can survive these tests since, apriori, the adversary doesn't know the side-channels or the parameters of the testing process (e.g. clock frequency) being considered. Furthermore, side-channels such as EM, consist of multiple sub-channels due to spatial and non-linear effects [22, 11, 3], and the full characterization of the Trojan signal and the process noise can only become clear when the ICs get manufactured. Thus the adversary has a difficult task of designing a Trojan circuit that would not leak in any of the side channels for any of the test parameters.

For our future work, we would like to make two improvements to the methodology presented in this paper. First, instead of working with simulated ICs, we would like to work with real fabricated ICs. Second, we would like to work with much larger ICs to cover general purpose microprocessor architectures. Both of these improvements require a significant investment of capital. A goal of this research work was to convince ourselves (and our sponsors) that the approach of using signal processing techniques has a potential to detect minute Trojan circuits without resorting to destructive testing.

We also plan to widen the scope of our study to include more side-channels, specifically EM emissions, to pick up the localized spatial distortions introduced by a Trojan circuit and also explore other signal processing techniques to evaluate their relative efficacy in detecting Trojan ICs.

References

[1] Website. DARPA BAA06-40, TRUST for Integrated Circuits. `http://www.darpa.mil/BAA/BAA06-40mod1.html`.

[2] Advanced encryption standard (AES). Website, Nov 2001. `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`.

[3] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM side-channel(s). In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 29–45. Springer Verlag.

[4] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer.

[5] I. Baturone, J. Huertas, S. Sánchez-Solano, and A. Richardson. Supply current monitoring for testing CMOS analog circuits. In *Proc. XI Conference on Design of Circuits and Integrated Systems (DCIS), Sitges*, pages 231–236, 1996.

[6] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults. In W. Fumy, editor, *Advances in Cryptology - EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer Verlag, 1997.

[7] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 12–28. Springer Verlag, 2002.

[8] S. Devadas and S. Malik. A survey of optimization techniques targeting low power VLSI circuits. In *Proceedings of the 32nd ACM/IEEE Conference on Design Automation*, pages 242–247, 1995.

[9] Defense Science Board Task Force. High performance microchip supply. Website, Feburary 2005. `http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf`.

[10] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic, New York, 1990.

2007 IEEE Symposium on Security and Privacy(SP'07)
0-7695-2848-1/07 $20.00 © 2007 IEEE

[11] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer.

[12] Blaise Gassend, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In V. Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 148–160. ACM, 2002.

[13] Amy Germida, Zheng Yan, James F. Plusquellic, and Fidel Muradali. Defect detection using power supply transient signal analysis. In *International Test Conference*, pages 67–76, September 1999.

[14] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *FOCS*, pages 553–562. IEEE Computer Society, 2005.

[15] C. F. Hawkins, J. M. Soden, R. R. Fritzemeter, and L. K. Horning. Quiescent power supply current measurement for CMOS IC defect detection. In *Proceedings of IEEE Transactions On Industrial Electronics*, pages 211–218, 1989.

[16] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In N. Koblitz, editor, *CRYPTO 1996*, volume 1109 of *Lecture Notes in Computer Science*, pp 104–113. Springer.

[17] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In M. J. Wiener, editor, *CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer.

[18] Jung Youp Lee, Seok Won Jung, and Jongin Lim. Detecting trapdoors in smart cards using timing and power analysis. In F. Khendek and R. Dssouli, editors, *Proceedings of TestCom 2005*, volume 3502 of *Lecture Notes in Computer Science*, pages 275–288. Springer Verlag.

[19] Joseph I. Lieberman. White paper: National security aspects of the global migration of the U.S. semiconductor industry. Website, June 2003. `http://lieberman.senate.gov/documents/whitepapers/semiconductor.pdf`.

[20] Model Technolgy Inc., Oregon. *ModelSim SE User's Manual*, version 5.5f edition, Sep 2001.

[21] H. Vincent Poor *An introductin to Signal Detection and Estimation* Second Edition, Springer-Verlag.

[22] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (EMA): Measures and countermeasures for smart cards. In I. Attali and T. P. Jensen, editors, *E-smart 2001, Proceedings*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer Verlag.

[23] Christian Rechberger and Elisabeth Oswald. Practical template attacks. In C. H. Lim and M. Yung, editors, *Information Security Applications, 5th International Workshop, WISA 2004, Revised Papers*, volume 3325 of *Lecture Notes in Computer Science*, pages 443–457. Springer Verlag.

[24] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[25] Synopsys Inc. *Design Compiler User Guide*, version 2002.05 edition, Jun 2002.

[26] Synopsys Inc. *Prime Power Manual*, version 2002.05 edition, Sep 2002.

[27] Synopsys Inc. *Library Compiler Reference Manual: Technology and Symbol Libraries*, version x-2005.09 edition, December 2005. This volume provides information on synthesis, test, and power tools.

[28] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I.* John Wiley & Sons, New York, 1968.

[29] Hoeteck Wee. On obfuscating point functions. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 523–532. ACM, 2005.

A. RSA circuit block diagram

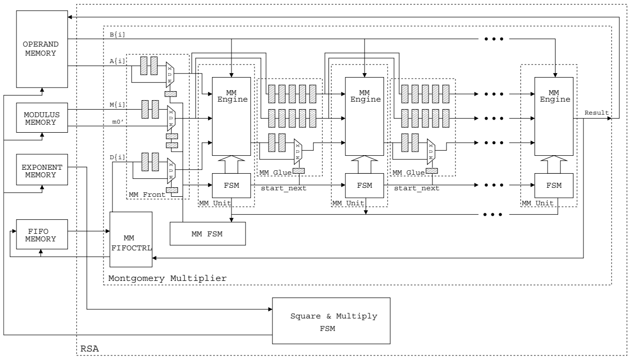

Figure 19. Block diagram of the RSA circuit.