Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions

Xiaoxiao Wang and Mohammad Tehranipoor
ECE Department
University of Connecticut

Jim Plusquellic
ECE Department
University of New Mexico

ABSTRACT

This paper addresses a new threat to the security of integrated circuits (ICs) used in safety critical, security and military systems. The migration of IC fabrication to low-cost foundries has made ICs vulnerable to malicious alterations, that could, under specific conditions, result in functional changes and/or catastrophic failure of the system in which they are embedded. We refer to such malicious alternations and inclusions as Hardware Trojans. The modification(s) introduced by the Trojan depends on the application, with some designed to disable the system or degrade signal integrity, while others are designed to defeat hardware security and encryption to leak plain text information. This paper explores the wide range of malicious alternations of ICs that are possible and proposes a general framework for their classification. The taxonomy is essential for properly evaluating the effectiveness of methods designed to detect Trojans. The latter portion of the paper explores several Trojan detection strategies and the classes of Trojans each is most likely to detect.

1. INTRODUCTION

Chip design and fabrication is becoming increasingly vulnerable to malicious activities and alternations with globalization. This has raised serious concerns regarding possible threats to military systems, financial infrastructures and even household appliances. An adversary can introduce a Trojan designed to disable and/or destroy a system at some future time (we call it Time Bomb) or the Trojan may serve to leak confidential information covertly to the adversary. Trojans can be implemented as hardware modifications to application specific ICs (ASICs), commercial off the shelf (COTS) components, microprocessors, or digital signal processors (DSPs), or as firmware modifications, e.g., to field programmable gate arrays (FPGA) bitstreams [1][2].

Unfortunately, the detection of such inclusions is difficult for several reasons: 1) Nanometer IC feature sizes and system complexity make detection through physical inspection and destructive reverse engineering difficult and costly. Moreover, destructive reverse engineering does not guarantee that ICs not destructively inspected are Trojan-free. 2) Trojan circuits are by design activated under very specific conditions, which makes it difficult to activate and detect them using random stimuli. Moreover, existing automatic test pattern generation (ATPG) methods used in manufacturing test for detecting defects do so by operating on the netlist of the Trojan-free circuit specification. Therefore, existing ATPG algorithms cannot target Trojan activation/detection directly.

In order to develop methods designed to improve IC TRUST, it is essential to first define a taxonomy for Trojans. The Trojan classification scheme that we propose in Section 3 is derived from several fundamental characteristics of Trojans, including their physical, activation and action characteristics. Once a framework is established, we consider detection strategies in Section 4 and the metrics on which they can be evaluated, such as the complexity of the method and the amount of effort needed to establish trust.

2. TROJAN SECURITY IMPLICATIONS

A consequence of the proliferation of microelectronics is the increasingly important role it plays in the manipulation and communication of confidential information and in the management and control of equipment. This type of microelectronics-enabled automation also makes such systems vulnerable to attack. The software threat to security is well known and many techniques have been proposed and implemented to protect systems [8]. However, threats originating in the actual hardware are new and are disruptive to software security layers that run on the hardware. This is true because software security mechanisms can be easily bypassed by malicious hardware, and such hardware is extremely difficult to detect given the trends in system complexity and IC technology.

Below, we provide a list of applications that could potentially be targets for adversary to insert hardware Trojans. We acknowledge that this is not meant to be an exhaustive list of applications that are vulnerable to hardware-based security issues, but rather serves to highlight specific present and future systems that are likely to be targets for adversaries. The systems that are vulnerable to attack will continue to grow over time, which drives the need to develop more sophisticated methods for hardware authentication.

- **Military applications:** weapon control systems, battlefield communication systems, battlefield information collection and decision making systems, etc.

- **Aeronautic and astronautic applications:** space shuttle electronics, satellite electronics, aeroplane electronics, etc.
- **Civilian security-critical applications:** confidential business information management systems (banking systems, stock market information management systems, firewalls protecting company propriety information, etc.), confidential personal information management systems (medical record systems, personal financial record systems, etc.).
- **General purpose applications:** any electronic systems used to manage confidential information.
- **Secure IP:** fableless semiconductor industry, core developers.
- **Transportation security:** Secure search and information management systems, emergency awareness systems, etc.

3. TAXONOMY

Malicious alternations to the structure and function of a chip can take many forms. We decompose the Trojan taxonomy into three principle categories as shown in Figure 1, i.e., according to their physical, activation and action characteristics. The physical characteristics of a Trojan are further partitioned into four categories; type, size, distribution, and structure. Our proposed taxonomy, therefore, describes Trojans using six attributes, including four physical, one activation and one action attribute. Although it is possible for Trojans to be hybrids of this classification, e.g., have more than one activation characteristic, we believe this taxonomy captures the elemental characteristics of Trojans and will be useful for defining the capabilities of various detection strategies.

3.1. Trojan Physical Characteristics

The physical characteristics category describes the various hardware manifestations of Trojans (see Figure 2).

A. Type: The *type* category partitions Trojans into functional and parametric classes. The functional class includes Trojans that are physically realized through the addition or deletion of transistors or gates, while parametric refers to Trojans that are realized through modifications of existing wires and logic. The thinning of a wire, the weakening of a transistor or any modification of a physical geometry designed to sabotage reliability or increase the likelihood of a functional or performance failure are examples of the latter.

B. Size: The *size* category accounts for the number of components in the chip that have been added, deleted or compromised. Size of a Trojan can be an important factor during activation. A smaller Trojan has a higher probability for activation than a Trojan with larger number of inputs.

C. Distribution: The *distribution* category describes the location of the Trojan in the physical layout of the chip. For example, a *tight distribution* describes a Trojan whose components are topologically close in the layout while a *loose distribution* describes Trojans that are dispersed across the layout of the chip. Figure 2 shows some examples. Note that the distribution of Trojans depends on the availability of dead spaces on the layout. If very small dead spaces are available on the layout, then the adversary may be forced to place and route smaller portions of the Trojan in different dead spaces. Note that here we assume that the adversary may not change the physical layout dimension of the design.

D. Structure: If the adversary is forced to regenerate the layout to be able to insert the Trojan, then the chip dimensions change. This change could result in different placement for some or all the design components. Any changes in physical layout can change the delay and power characteristics of chip which will make it easier to detect the Trojan.

In order to minimize the probability of detection, an adversary is likely to adopt a strategy whereby the physical 'footprint' of the Trojan is as small as possible. We use the term *stealthy physical footprint* to describe the adversary's objective in this regard. For small, tightly coupled parametric Trojans, the goal is easily achieved because parametric Trojans can be introduced by changing the geometry of a single wire or transistor. For functional Trojans, size and distribution have significant impact on the physical footprint of the Trojan. For larger sizes, distributing the Trojan across the layout can improve the stealthy physical footprint criteria because detecting the Trojan based on, for example, an anomaly in a localized power or leakage signature, is more difficult. However, distributing the Trojan across the layout can actually worsen its physical footprint in other respects. For example, the length of the wires connecting the Trojan increases significantly, which changes the capacitance distribution of the Trojan-free chip and increases the chances that the Trojan will change the delay characteristics of the chip. For this reason, tightly coupled Trojans may be more attractive, particularly if power/leakage hiding techniques, such as power gating through transistor stacks, are used to reduce its footprint. Figure 3 shows six examples for Trojans with different physical characteristics. We will investigate this further in the context of detection strategies in Section 4.

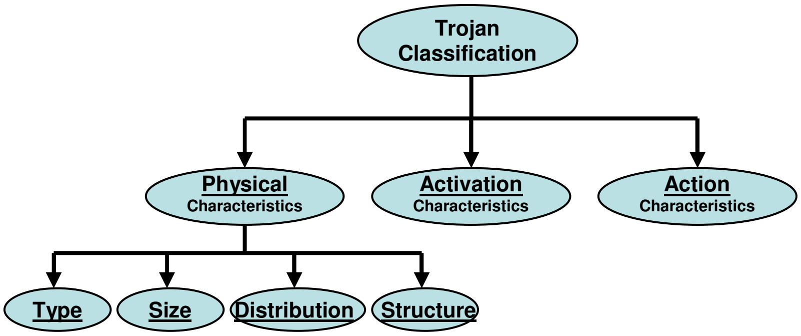

Figure 1. Taxonomy of Trojans

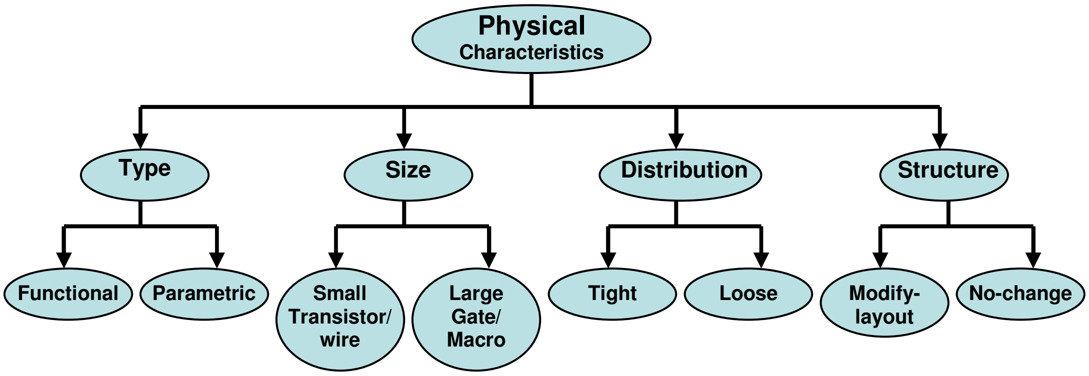

Figure 2. Physical characteristics of Trojans

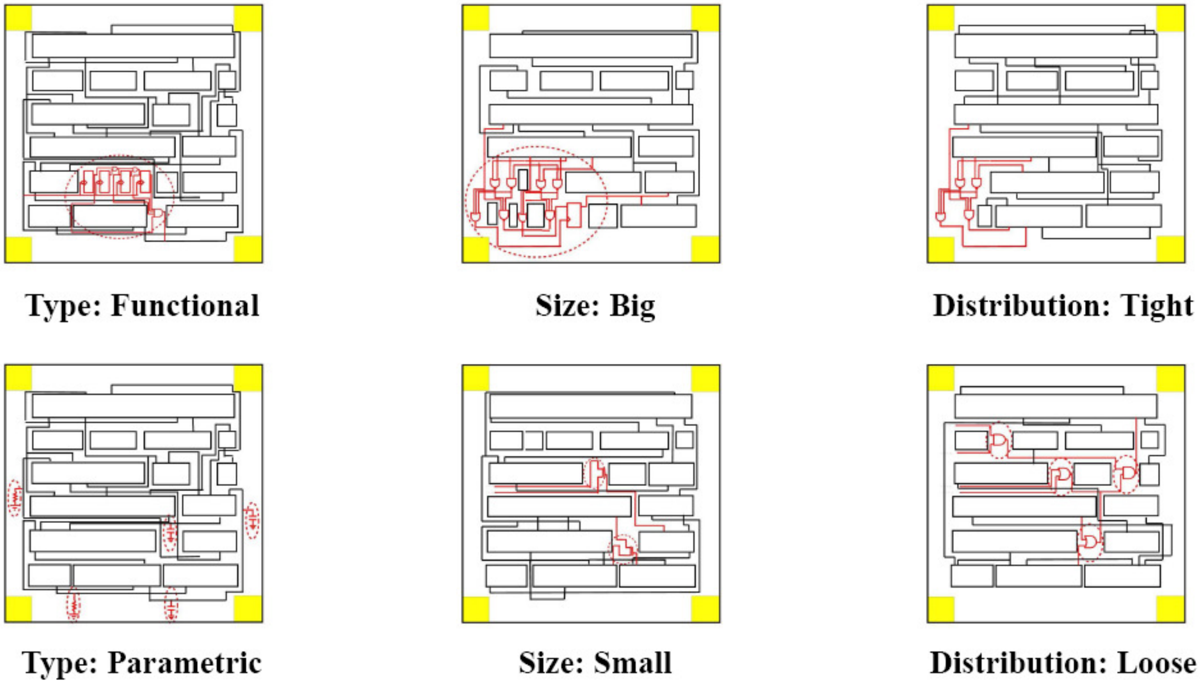

Figure 3. Examples for various Trojan physical characteristics

3.2. Trojan Activation Characteristics

Activation characteristics refer to the criteria that causes the Trojan to become active and carry out its disruptive function. The adversary who inserted the Trojan will make it difficult for the user of the chip to activate it, in an effort to prevent 'accidental' activation and detection during the testing phase(s) of the chip and system. Therefore, activation of a Trojan can be considered a 'rare event' from a statistical perspective.

We use the term *stealthy activation* to describe the adversary's objective in this regard. We partition Trojan activation characteristics into two sub-categories, labeled *Externally-activated* and *Internally-activated.* In Externally-activated category, the Trojan can be activated externally by adversary in his/her time of choosing. This can be done by embedding a receiver or antenna on chip and controlling it through external signals. This can also be done by accessing the internal registers and forcing them to specific date to extract secret keys.

The Internally-activated category is divided into two subclasses, labeled *Always-on* and *Condition-based* as shown in Figure 4. Always-on, as the name implies, indicates that the Trojan is always active and can disrupt the function of the chip at any time. This class covers Trojans that are implemented by modifying the geometries of the chip such that certain nodes or paths in the chip have a higher susceptibility to failure. We referred to these types of Trojans as 'parametric' in the *type* subclass of the physical characteristics class. In order for Always-on Trojans to meet the stealthy activation criteria, the adversary necessarily needs to insert them on nodes or paths that are rarely exercised. In the test community, such nodes and paths are referred to as *hard-to-detect faults* because the conditions needed to detect faults on them are difficult to determine and are statistically unlikely to occur using random and structural stimuli.

The Condition-based subclass includes Trojans that are 'inactive' until a specific condition is met. The activation condition can be based on the output of a sensor that monitors temperature, voltage or any type of external environmental condition, e.g., electro-magnetic interference (EMI), humidity, altitude, atmospheric pressure, etc. Or it can be based on an internal logic state, a particular input pattern or an internal counter value. The Trojan in these cases is implemented by adding logic gates and/or flip-flops to the chip, and therefore is represented as a combinational or sequential circuit.

We believe that fully activation for logic-based Trojans is an NP-complete problem. Fully activation of Trojans depends on the number of Trojan inputs and the states of the circuits the Trojan is observing. Section 4.2 describes the probability of fully detection for Trojans in more details. The partial activation of Trojans depends on the method(s) implemented to detecting and isolating Trojans. If part of a Trojan is activated, the power consumed by the gates included in that part will contribute to the total power consumption.

An important distinguishing characteristic between Always-on and Condition-based Trojans is the former is nearly 'invisible' when it is inactive while the latter is always visible to some degree when inactive. We define 'invisible' as undetectable when methods that measure the chip's digital and/or analog properties, including power consumption, are applied while the chip is undergoing some form of testing. The fact that Always-on Trojans are defined as subtle modifications to existing wire and transistor geometries, indicates that the chip that embeds them will behave identically to a Trojan-free chip, under the condition that the nodes or paths altered by the Always-on Trojan are not exercised by such tests, i.e., the Trojan remains inactive. In contrast, a Condition-based Trojan, which needs sensors or logic components to monitor for the activation condition, consumes power at some level and/or adds load to wires of the original circuit, which in turn changes the delay characteristics of the chip.

These subtle changes to the analog characteristics of the chip occur even while the Trojan remains inactive. This implies that detecting an Always-on Trojan will necessarily require its activation while detecting a Condition-based Trojan can be accomplished *without fully* activating it, in situations where the detection method incorporates, e.g., an analysis of the chip's power consumption characteristics. We will revisit this issue in Section 4.

3.3. Trojan Action Characteristics

Action characteristics identify the types of disruptive behavior introduced by the Trojan. The classification scheme shown in Figure 5 partitions Trojan actions into three categories; *Modify-function, Modify-specification,* and *Transmit-info* (see Figure 5). As the name implies, the Modify-function class refers to Trojans that change the chip's function through additional logic or by removing or bypassing existing logic. The Modify-specification class refers to Trojans that focus their attack on changing the chip's parametric properties, such as delay. The latter class represents parametric Trojans that modify wire and transistor geometries. Lastly, the Transmit-info class refers to Trojans that transmit key information from design mission mode to an adversary.

An important distinguishing characteristic between modify-function and modify-specification Trojans concerns their capabilities. The nature of modify-specification Trojans restricts their disruptive capabilities to actions that result in system failure. This is true because modify-specification Trojans are implemented as modifications to existing wires and transistors. Therefore, new capabilities are not possible. In contrast, the capabilities of modify-function Trojans are essentially limitless. As the examples illustrate, modify-function Trojans, once activated, can change virtually any characteristic of the chip or can introduce completely new functionality such as broadcasting confidential information over the power buss.

The type of malicious behavior introduced by the Trojan once activated depends greatly on the application. The following list gives some examples but is by no means an exhaustive list. The examples each list some of the attributes associated with the Trojan as they relate to the taxonomy discussed above. The labels following the slash give the subclass. The keyword *any* is used to indicate any class or subclass is assignable under that category.

- Trojan introduces a hard short or a delay fault in a rarely exercised section of the circuit.

Physical: parametric, small, tight, **Activation**: always-on, **Action**: modify-spec./*any*

- Trojan disrupts clock to shut down the chip or affect its synchronization.

Physical: functional, large, *any*, **Activation**: condition-based/*any*, **Action**: modify-function/add

- Trojan disables encryption.

Physical: functional, large, *any*, **Activation**: condition-based/*any*, **Action**: modify-function/bypass.

- Trojan changes functional output of the chip resulting in catastrophic failure.

Physical: functional, large, *any*, **Activation**: condition-based/*any*, **Action**: modify-function/*any*

- Trojan leaks information through side channels.

Physical: functional, large, *any*, **Activation**: condition-based/*any*, **Action**: modify-function/add

- Trojan destroys the operating environment of original circuit, such as shutting down power, generating noise to disrupt critical signals, or increasing the thermal gradients on the chip, possibly causing it to burn out.

Physical: functional, large, *any*, **Activation**: condition-based/*any*, **Action**: modify-function/add

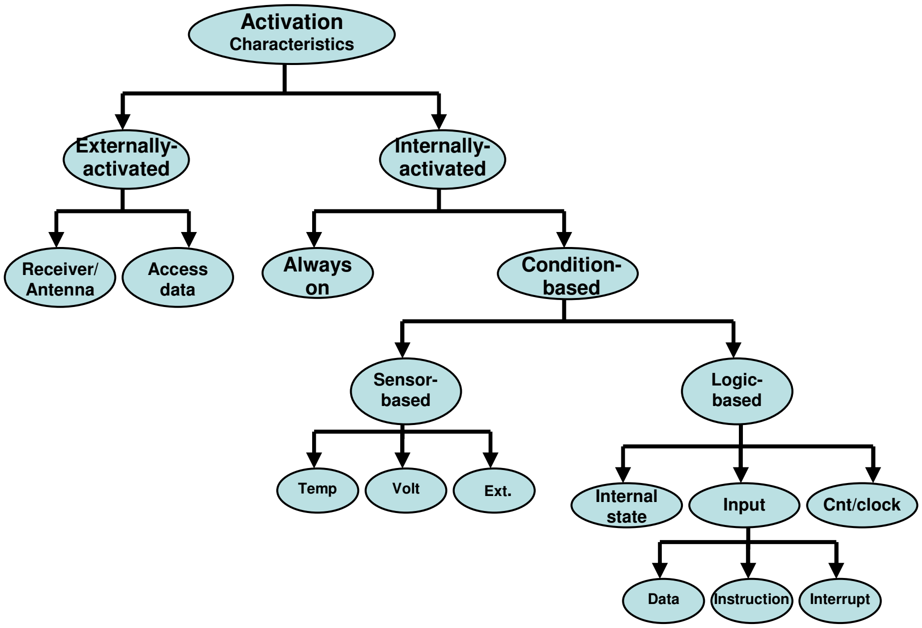

Figure 4. Trojan activation characteristics

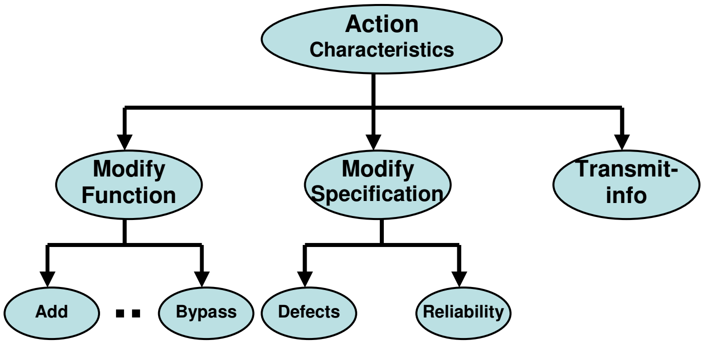

Figure 5. Trojan action characteristics

Although the modify-function class appear to be more attractive based on their action capabilities, they also are more 'risky' from the adversary's perspective. When describing the Trojan activation characteristics, we indicated that Trojans in the condition-based class are never completely invisible, because the additional logic they incorporate needs to continuously monitor for the activation condition. In order to accomplish this, the Trojans need to 'connect' to the existing logic. The connection(s) introduce capacitive load to the wires and power grid to which they connect, that in turn, changes the power consumption and thermal characteristics of the chip. Moreover, the logic added to determine activation needs to perform 'compare' operations that also consume power and introduce temperature variations. The power and temperature anomalies introduced by the Trojans in the modify-function class can be considered a *secondary* action characteristic of the Trojan. The secondary action characteristics of the Trojan are undesirable, from the adversary's perspective, but unavoidable, and their presence can be leveraged by certain Trojan detection methods described in the next section.

4. TROJAN DETECTION STRATEGIES

In this section, we outline the general approaches for detecting Trojans. Trojan detection methods can be applied immediately after the chip is returned to the customer, either as a die on a wafer or as a packaged chip, and/or they can be applied continuously during the lifetime of the system. For the latter case, board level support systems, such as trusted companions, are needed to carry out the monitoring. Although these types of approaches are of interest, the focus of this work is on 'time-zero' detection methods, i.e., methods applied before the chip is installed in the target system. We refer to this phase as Silicon Design Authentication that is done after manufacturing testing phase. In general, there are three basic approaches for detecting Trojans that we explain them in the following.

4.1. Failure Analysis-based Techniques

The first involves applying sophisticated failure analysis techniques such as scanning optical microscopy (SOM), scanning electron microscopy (SEM), pico-second imaging circuit analysis (PICA), voltage contrast imaging (VCI), light-induced voltage alternation (LIVA), charge-induced voltage alternation CIVA, etc. [3]. Although these techniques can be effective for authentication purposes, they are also extremely time consuming and expensive. Moreover, many require the sample (chip) to be prepared by backside thinning and de-processing operations. Obviously, this approach is not suited for applications in which every chip needs to be authenticated. Another drawback is that many of these techniques are becoming increasingly ineffective for technologies in the nanometer domain.

An important issue is that the adversary will most likely insert Trojans randomly in chips. Therefore, spending a large amount of time on each chip for authentication will be prohibitively expensive. As a result, new and efficient methods are required to detect Trojans with higher confidence level and minimum authentication time.

4.2. ATPG-based Trojan Detection Techniques

The second approach involves the use of 'standard' VLSI fault detection tools, such as automatic test pattern generation (ATPG). Detection of a Trojan is accomplished by applying a digital stimulus and inspecting the digital output of the chip. The digital stimulus is derived using the netlist of the chip. For Trojans of the parametric type as described in Section 3.1, the netlist of a chip is the same with and without the Trojan. This is true because parametric Trojans are introduced into the existing logic of the chip by violating design rules, i.e., thinning a wire, etc. Therefore, ATPG can be modified to target parametric Trojans. Given their stealthy activation criteria, ATPG directed to generate tests for nodes and paths that are hard-to-detect, i.e., difficult to control and/or observe, is likely to yield the best results for activation and detection of Trojans.

Unfortunately, ATPG is not effective for the functional Trojans, which are represented as inserted, additional logic. Without knowledge of this logic and how it is connected to the original logic in the chip, it is impossible for ATPG to perform a directed search for a vector or state that causes activation. Bear in mind, that if the activation criteria can be determined, then detection would be trivial in many cases, assuming the Trojan modifies the internal state or an output of the chip in some fashion. However, for Trojans that activate and leak information over side channels, e.g., the power supply, digital testing methods are not effective. Therefore, for functional Trojans, an ATPG approach is hindered by two problems, one that deals with activation and another that deals with detection. A third approach that can potentially solve these problems involves the measurement and analysis of the chip's side-channel signals [4][5][6]. For example, it is possible to stimulate the chip using digital stimuli and then measure the analog response signals of the chip, such as the transient or quiescent power supply current[1].

4.3. Side Channel Signals Analysis

Another possibility is to stimulate the power grid directly by driving it with a sine wave at one position and measuring its response at another. The analog nature of the side-channel response signals enables the use of highly sensitive detection techniques. Such techniques may be able to detect functional Trojans without activating them, i.e., through the measurement of their *secondary* action characteristics as described in Section 3.3. For example, we indicated that functional Trojans are never completely inactive because of the need to continuously monitor for the activation conditions. Consider a Trojan that activates based on a specific state of a data bus in the chip. The implementation of the Trojan, in this case, requires some type of comparator to be installed that monitors the wires of the data bus. The logic of the comparators, e.g., AND gates, switches as the data bus changes and therefore consumes power. Side-channel signal analysis can potentially detect the power anomaly introduced by the operation of the comparator. Other side-channels signals include electro-magnetic field variations, temperature variations, voltage variations, etc., that occur at various locations across the chip. New methods can be developed that use such signals to detect and isolate hardware Trojans.

Moreover, the highly sensitive nature of side-channel analysis techniques may allow detection of tightly coupled functional Trojans even without the application of a digital stimulus. The presence of the Trojan logic gates adds capacitance to the power grid. The presence of the additional capacitance changes in impulse response of the power grid. The impulse response of the power grid can be tested by injecting an analog stimulus onto the grid at one place and measuring the response at another.

The effectiveness of side-channel-based measurement and analysis techniques can be improved by adopting design-for-hardware-trust (DFHT) techniques, which, for example, add circuitry to support the measurement and analysis processes. On-chip voltage and temperature sensors can be installed to increase the level of sensitivity of side-channel measurement and analysis techniques by providing local observability at various positions across the 2-D layout of the chip. The DFHT strategy must also incorporate a validation strategy for the on-chip support circuits because of the potential of the adversary to sabotage the sensors.

4.4. Trojan Detection Challenges

Depending upon the method used for Trojan detection, there seem to be extremely difficult challenges associated with the method.

The taxonomy and discussion presented above suggest that detection strategies that depend on activating condition-based Trojans through the application of test patterns and detecting them through an analysis of the circuit's logic response may not be effective. Considering an intelligent and determined adversary, the Trojans can be inserted such that the probability of accidental detection using test patterns (functional, structural, and random) will be extremely low. Assume a Trojan with n number of inputs. Also, assume that p_i is the probability of justifying a 0 or 1 on ith input of the Trojan circuit. If the Trojan is inserted deep into the circuit, p_i will be extremely low. The probability (P) of activating this n-input Trojan and propagating its effect using these patterns would be:

$$P = P(\text{activation}) \,.\, P(\text{propagation})$$

$$P(\text{activation}) = \prod\nolimits^{n}_{i=1} p_{i.}$$

Assume p_i=10^{-3} and n=10, then P=10^{-30} . Considering P(propagation), the probability of successful propagation of the Trojan's effect if the Trojans output is connected to the circuit (e.g. Modify-function Trojan), can also worsen P. Note that P(propagation) is circuit topology dependent. This clearly demonstrates that relying on input patterns for Trojan detection may not seem to be an effective solution.

When using side-channel signal analysis methods for detecting hardware Trojans, the circuit process variations will be a major bottleneck. For instance, process variations significantly impact circuit leakage currents therefore, using IDDQ like methods to detect a Trojan will suffer from inaccuracy. Trojan type and size also play an important role in detection sensitivity. Larger Trojans will consume more leakage power and are easier to be partially activated which will consume more switching current. Smaller Trojans are harder to be detected using leakage and switching current analysis since they contribute negligibly to the total power in the circuit but easier to be activated using functional or structural patterns.

Detection of Trojans based on switching current analysis can be effective only if efficient patterns are generated and applied. The challenge here is to generate patterns that cause maximum switching in a small region in the circuit and minimum switching in other regions. Considering a small number of primary inputs in large and complex designs, this would seem to be a challenging task. The scan flip-flops can be used to facilitate the problem, however, the patterns must be shifted into the scan chain which makes the process significantly slow. The main advantage of using scan is in its significantly increased controllability and observability. This would cause increased switching in the circuit. New techniques must be developed to generate localized switching in the circuit to increase detection and isolation accuracy.

An inserted Trojan in the circuit can in fact impact the circuit delay characteristics. Delay test methods can be used to detect such Trojans however the deficiency of current transition delay ATPG methods will challenge its efficiency. When a Trojan is inserted into a circuit, equivalent to the gate capacitance will be added to the total capacitance of the path the Trojan is taping the signal from. The amount of delay is small therefore novel methods must be developed to detect such small delay in the circuit induced by Trojans. We acknowledge that process variations can also cause small delay to the circuit and in turn cause uncertainty during detection. The Trojan can however cause a multi-path small delay injection which could potentially be detected using efficient delay testing.

Also, note that depending on the type of the Trojan, we need to devise appropriate detection strategy. Some Trojans are easier to be detected using power analysis methods and some others are easier to be detected using delay analysis. Since the type and size of Trojans are not known to us, it is recommended to use both methods to target Trojans during silicon design authentication phase to increase the probability of detection.

Another issue that must be addressed during Trojan detection is the time taken to verify the authenticity of each chip. A reasonable assumption is that the adversary will most likely insert Trojans randomly in a large batch of chips. Therefore, the authentication time will be significantly important for large volume of chips fabricated in an untrusted foundry.

4.5. Metrics for Evaluating Detection Methods

In this section, we propose a set of metrics on which the effectiveness of a Trojan detection method can be evaluated. The metrics are partitioned in primary and secondary, to reflect their relative importance and applicability. The primary metrics apply to all techniques while some secondary metrics may not be relevant to a given technique. This list is not meant to be exhaustive, but rather is a sample of the criteria one might consider for the evaluation of their detection strategy.

Primary Metrics:

- What Trojan classes, as defined by Figure 1, are detectable using the method?
- How complex is the method, i.e., how long does it take to establish trust, how much data is processed, what type of data?
- Is it possible to defeat the method, i.e., what are its weaknesses given the advisory knows about it?
- Can the method detect the inclusion with non-activating inputs applied?

Secondary Metrics:

- How invasive is the method, what impact does it have on performance, design time, power consumption?
- Does the method require support (design-for-security) circuity? Is it possible to sabotage the support circuitry?
- How robust is the method to process variations?

5. AKNOWLEDGEMENT

The work of Xiaoxiao Wang and Mohammad Tehranipoor was supported in part by NSF grant CNS-0716535. The work of Jim Plusquellic was supported in part by NSF grant CNS-0716559.

6. CONCLUSIONS

A Trojan classification scheme is presented in this paper that partitions Trojans according to their physical, activation and action characteristics. The taxonomy can be used in conjunction with the Trojan detection methods outlined to help define their effectiveness and capabilities.

REFERENCES

[1] http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf

[2] http://www.darpa.mil/mto/solicitations/baa07-24/index.html

[3] J. Soden, R. Anderson and C. Henderson, “IC Failure Analysis Tools and Techniques -- Magic, Mystery, and Science”, International Test Conference, Lecture Series II “Practical Aspects of IC Diagnosis and Failure Analysis: A Walk through the Process”, 1996, pp. 1-11.

[4] A. Germida, Z. Yan, J. Plusquellic and F. Muradali, “Defect Detection using Power Supply Transient Signal Analysis”, International Test Conference, Sept. 1999, pp. 67-76.

[5] J. Plusquellic, D. Acharyya, A. Singh, M. Tehranipoor and C. Patel, “Quiescent-Signal Analysis: A Multiple Supply Pad IDDQ Method”, Design and Test of Computers, Volume 23, Issue 4, April 2006, pp. 278-293.

[6] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, “Trojan Detection using IC Fingerprinting”, Symposium on Security and Privacy, 2007, pp. 296 - 310.

[7] J. Viega and G. McGraw, Building Secure Software, Addison-Wiley, 2002.