Malware in Hardware Infrastructure Components

Christian Krieg, Edgar Weippl

SBA Research, Favoritenstraße 16, 1040 Wien, {ckrieg,eweippl}@sba-research.org

Abstract

Malicious hardware is a fairly new research topic that has attracted the interest of the scientific community. Therefore, numerous approaches have been proposed in the last years to counter the threat of so-called *hardware Trojans*. This chapter describes malicious hardware in the context of the security of hardware infrastructure components. Network infrastructure plays a vital role in our everyday lives, since many services depend on reliable and secure connections. In the following, we briefly introduce the topic of hardware Trojans. After describing their basic components, we give some insights into how hardware Trojans can be used maliciously in infrastructure devices. Furthermore, we outline the evolution of hardware Trojans and measures to counter them.

1 Introduction

Hardware infrastructure components are becoming increasingly vital in our everyday lives. Sensitive data, such as credit card numbers, medical data, private communications, and banking information are transmitted over communication channels, in most cases without the users being aware of it. It is, therefore, of utmost relevance that communication channels are secure against adversaries who try to gain access to sensitive data. Encryption and security protocols aim to secure communication from a sending device to a receiving device. Cryptography is applied within these devices, which means that data are available in plaintext. Moreover, cryptographic keys are available in such devices, which make them a potential point of interest for attackers who aim to compromise data confidentiality. One way to gain knowledge of cryptographic keys would be to tamper with the device in a way that allows the attacker to read out the keys from memory [5]. Other possibilities include fault injection [25] combined with analysis methods such as side-channel analysis [25]. In this contribution, we will consider a fairly new attack vector against the security of hardware infrastructure components. We will examine how malicious hardware inclusions – so-called *hardware Trojans* – affect the overall security of infrastructure components, focusing on data confidentiality. We outline the evolution of hardware Trojans and countermeasures in the literature.

When talking about malware in hardware infrastructure components, we focus on functions of a system that are implemented in hardware but have not been specified. It is the hardware itself that is malicious.

Malware can be introduced into hardware in many ways. For example, a malicious designer can inject an unspecified functionality into a design by adding just a few lines of hardware description code [45]. Furthermore, a synthesis tool can be modified by adversaries is that the hardware to be synthesized is altered in an unspecified manner [41]. And even if it is expensive and resource-intensive, a malicious chip producer can reverse-engineer a design to include circuitry at the physical level [2].

Hardware Trojans will be the topic of this chapter. Section 2 will briefly classify hardware Trojans. Section 3 presents some ideas on how malicious hardware can subvert an infrastructure and how sensitive data can be leaked by hardware Trojans. In Section 4, we outline the historical development of hardware Trojans, always being aware of defense methods to counter Trojans.

2 Components of Hardware Trojans

As hardware Trojans have to pass functional tests without being detected, they have two basic mechanisms: a *trigger* and a *payload* mechanism [50].

A trigger should activate the payload upon a certain condition, such as the occurrence of a rare event (e.g., a bit pattern 0x3745 on a data line), the lapse of a certain time interval (e.g., 10,000 seconds), or a condition that is met by the environment (e.g., temperature is 65°C). The most essential requirement of a trigger condition is that it is not met during functional tests, which are integral parts of the hardware production process. Otherwise, it could activate the Trojan during the test, which would make it more detectable.

The payload mechanism implements the effective function of a Trojan. For instance, this could be a kill switch (i.e., the permanent deactivation of a hardware system), the interception of sensitive data such as a cryptographic key, or the remote control of a hardware system (which corresponds to opening a hardware backdoor).

3 Hardware Trojans in Infrastructure Components

Section 4 provides a comprehensive overview of the approaches to Trojan detection. It shows that the problem is manifold and illustrates the wide range of attack vectors.

Although there have been no reports of an actual attack so far, recent research has addressed many threats to infrastructure components. For example, Jin and Makris show that it is possible to leak the cryptographic key of a wireless device over the wireless channel [23]. Depending on each bit of the key, the wireless signal is altered within the permitted tolerances. That way, an attacker only has to reside within the range of the wireless device, record the signal, and perform a statistical analysis to obtain the key. Subsequently, the attacker can use the key for authentication and use the device

as legitimate user, which enables her to subvert the entire infrastructure of which the device is a part.

Similarly, Lin et al. show how data can be leaked over a secret channel [32]. By modulating the power supply signal of a device, sensitive data can be leaked stealthily. It is hard to detect the covert transmission of data, since the signal is modulated in the code division multiplex, i.e., spread spectrum technology is used. Therefore, without knowledge of the correct code, the covert signal cannot be detected, as it is indistinguishable from noise. To obtain sensitive data (such as a cryptographic key), an attacker has to probe the power supply signal of the device under attack and demodulate it by correlating it with the proper code.

Likewise, King et al. designed a malicious processor, which has hidden hardware implemented that allows an attacker to perform extensive attacks on the software layer [27]. The *Illinois Malicious Processor* provides mechanisms to illegitimately login to an overlying operating system as administrative user without providing a password. This way, an attacker can get broad access over an infrastructure component. If such a processor is deployed, e.g., in a router, the infrastructure itself could be modified, therefore serving as a base for further attacks on the network layer.

4 Evolution of Hardware Trojans and Their Countermeasures

Hardware Trojans have become a serious problem for the security of IT systems. In the following, we outline the historical evolution of Trojans, which is accompanied by the evolution of countermeasures to fight Trojans. We describe the development from 2005, when the US Department of Defense published a report on the supply of semiconductors [19], to the present. To maintain logical association, we have grouped the approaches into subsections.

4.1 Raising Political Awareness

In 2005, the US Department of Defense released a report about the security of supply of high-performance integrated circuits [19]. In this report, they investigated the concept of a vertical business model for compliance with the demand for secure and authentic hardware. The report stated that the manufacturing of microchips had been relocated to low-wage countries for financial reasons. Therefore, the risk that chip manufacturers could add additional functions during the production process appeared possible. The report described trustworthiness as follows:

> "*Trustworthiness* includes confidence that classified or mission critical information contained in chip designs is not compromised, *reliability* is not degraded or unintended design elements inserted in chips as a result of design or fabrication in conditions open to adversary agents. Trust cannot be added to integrated circuits after fabrication; electrical testing and *reverse engineering* cannot be relied upon to detect undesired alterations in military

integrated circuits." [1]

As the production consistently had been transferred to potential enemies, the US Department of Defense did not believe that the supply of necessary semiconductors could be ensured in the event of war.

For this reason, the research project TRUST in Integrated Circuits (TIC) was started in 2007 by the Defense Advanced Research Projects Agency (DARPA) [17]. TIC is intended to develop technologies that can provide trust for circuits in the absence of a trusted foundry. It only considers technical efforts that address the fabrication of Application-Specific Integrated Circuits (ASICs) by non-trusted foundries and software implementation of configurable hardware, such as Field-Programmable Gate Arrays (FPGAs).

Adee [1] triggered a veritable flood of publications with his article on the threat of hardware Trojans.

2008 can be clearly identified as the year in which this topic gained academic interest. The Australian Department of Defence picked up the topic and published a report about the battle against hardware Trojans, evaluating its effectiveness [4].

4.2 Introducing Side-Channel Analysis

In the same year, Agrawal et al. [2] published their work on a method to detect secretly added functionality through side-channel analysis. A device under test is monitored with regard to physical side channels, such as supply voltage or timing. This work can definitely be seen as the starting point for numerous publications on this topic.

The focus at the time was clearly on side-channel analysis [7, 10, 11, 24, 31, 38, 47], but other methods in the area of logic tests [15] were proposed as well. To augment the detection rate, some approaches for increasing the activation of hardware Trojans were proposed [40, 22].

4.3 Malicious Computer Systems

King et al. [27] were the first to publish a comprehensive combined hardware/software attack. In this attack, a hardware Trojan serves as the basis of an extensive attack by allowing an attacker to sign on to the operating system with root privileges with the help of a hardware backdoor.

The University of New York held a competition where the goal was to implement hardware Trojans. The criteria for the competition were to insert malicious circuits into the original layout as unnoticeably as possible; the extraction of information without being noticed was also part of the position in the final ranking. Work submitted to the competition can be found in [16, 12].

[1] [19], p. 3

4.4 Increasing Trojan Activation

Many approaches have been presented for increasing the chance for Trojan activation, which should help improve the detection rate during functional tests. *Toggle minimization* is used to reduce the overall activity of a circuit to be able to measure the (partial) activity of a Trojan, if one is present [9].

Inverting the supply voltage of the logic gates in a circuit causes the logic states of the gates to be inverted as well. This measure causes an inversion of the detectability – a Trojan that was previously hard to spot can now be easy to detect [8].

Generating optimal testing patterns should increase the detection rate for logic tests. Chakraborty et al. [14] present an approach for initiating rare logic states multiple times in order to help trigger a potential trigger condition. Rare states are identified by a statistical method.

Salmani et al. [42] increase the chance of state changes ("toggles") by inserting dummy flip-flops into the original design. Dummy flip-flops are realized as scan flip-flops to preserve the original functionality.

Banga and Hsiao [6] present an approach that first determines signals that are easy to activate during a functional test. These signals are then ignored when testing for the presence of Trojans that are hard to identify. Using the remaining signals, a formal verification is carried out. Any detected Trojans are subsequently isolated.

4.5 Applying Gate-Level Characterization to Trojan Detection

Generally, side-channel analysis should detect deviations from expected behavior caused by hardware Trojans. Because Trojans strive to be hard to detect during a functional test, the impact of a Trojan is assumed to be as small as possible compared to overall circuit activity.

This is a problem for their detection, because the effect of process variations will be of almost the same order of magnitude as Trojan impact.

The approach of GLC (Gate-Level Characterisation) tries to characterize each gate of an IC. Here, the values performance, switching power and leakage current are used for characterization. Scaling factors are calculated to account for process variations that cannot be avoided in the manufacturing process. During a functional test, scaling factors are measured with the help of side-channel analysis. If the testing results of an IC differ too much from the calculated characteristics, a Trojan may have been implemented [36, 37].

4.6 Proposing Trojan-Resistant Bus Architectures

Trojans that have been inserted into complex hardware can also be detected with the help of the operating system. Bloom et al. [13] propose an approach where a simple hardware guard monitors accesses from the CPU to the memory data bus and performs liveness checks. A watchdog timer is started each time the monitor observes a certain pseudorandom memory access procedure, which is initiated by the operating system. If the watchdog times out, a Denial-of-Service (DoS)-attack is detected. The operating

system also periodically checks if memory protection is activated to prevent privilege escalation attacks.

Kim et al. [26] propose a Trojan-resistant bus architecture for Systems-on-Chip (SoCs). The architecture is able to detect unauthorized bus accesses. To prevent DoS attacks, permanent bus allocation to one bus node is blocked by limiting maximum bus allocation time.

4.7 Stealthy Trojan Communication

A very interesting new class of Trojans is introduced in Lin et al. [32], where they present a new technology called *Malicious Off-Chip Leakage Enabled by Side-Channels (MOLES)*, which allows the extraction of sensitive data with the help of spread-spectrum technology. Because the signal of the extracted information completely disappears in noise, a detection of the hidden data transfer is hardly ever possible. Lin et al. [33] describe how data can be transmitted by modulating the power supply signal using spread-spectrum techniques. The implementation makes use of high capacitances that draw current while charging. Depending on whether a zero or one is to be transmitted, a capacitor is charged or not. The charging current – encoded via spread-spectrum technique – can be analyzed by performing a side-channel analysis on the power supply.

4.8 Securing Multi-Core Architectures

Another approach to detect Trojans in multi-core systems is proposed by Mclntyre et al. [34]. In this approach, software to be executed is varied while keeping functional equivalence. This can be accomplished by using different compilations or alternative algorithms. The variants of the software are executed on multiple cores. If one variant of the software matches the trigger condition of an injected Trojan – thus activating it – the results of two calculations will differ. This way, a Trojan can be detected and isolated at runtime.

4.9 Introducing Run-Time Detection

The *BlueChip* approach introduced by Hicks et al. [21] relies on additional hardware modules. It is designed to render hardware Trojans injected at design time harmless at runtime.

Trojans are isolated by replacing suspicious circuits by software emulation. Suspicious circuits are identified with Unused Circuit Identification (UCI), which is a method that monitors the activity of a circuit during the functional test. If a part of a circuit remains unused during the entire testing period, it is considered to be assigned to a Trojan circuit (which should not be detected during functional tests and would, therefore, remain silent).

Waksman and Sethumadhavan [45] present an approach to combat Trojans especially in microprocessors at runtime. The assumption is that malicious functionality is injected during the design phase by malicious designers. The following constraints are defined:

1. The number of malicious designers is low, 2. the activity of malicious designers

remains unnoticed, 3. the attackers need few resources to inject a backdoor, 4. the backdoor is activated by a trigger, 5. the Read-Only Memories (ROMs) written during the design phase contain correct data (microcode).

Two types of backdoors serve as a Trojan model: *emitter* (send data) and *corruptor* backdoors (modify data). The latter are very difficult to detect, since their operations can be hard to distinguish from normal, legitimate operations. The proposed measure to prevent the hardware backdoors is an on-chip monitoring system that consists of four parts: predictor, reactor, target and monitor.

A Trojan is discovered if the result of the monitored unit does not match the predicted outcome of the predictor. The detection principle is based on the assumption that the monitored unit never communicates with the monitoring instance – therefore, the designer of a malicious unit X cannot corrupt the monitor of X.

4.10 Improving Side-Channel Analysis

Du et al. [20] propose another approach based on side-channel analysis to detect hardware Trojans. The power consumption of a particular region of an IC is compared to the power consumption of the same region of another IC. If the power consumptions differ greatly, the reason could be a Trojan. This technique is called *self referencing*.

Narasimhan et al. [35] partition a layout where the different partitions are stimulated through appropriate test patterns. Transient current (I_{DDT}) und maximum frequency (f_{max}) are determined via side-channel analysis. Because I_{DDT} and f_{max} are linearly dependent and f_{max} is unalterable, a Trojan can be detected by observing an increase of I_{DDT}.

To determine the smallest detectable Trojan, Rad et al. [39] examine the sensitivity of a transient analysis of power supply signals. The smallest detectable Trojan consists of a single logic gate under laboratory conditions if the Trojan responds to a test pattern. If the measurement is characterized by a Signal-to-Noise-Ratio (SNR) of 10 dB, the size of the smallest recognizable Trojan increases to seven gates.

4.11 Trojan Localization

Salmani et al. [43] rearrange scan chains to increase the detection of hardware Trojans. The appliance used to test integrated circuits is usually not bound to a specific layout on the chip. The approach suggests a layout that rearranges scan chains over the entire chip area, so that certain areas can be specifically activated or deactivated during the functional test.

This should make the activity of a Trojan visible. Experimental tests show a partial amplification of Trojan activity by a factor of 30.

4.12 Enhanced Gate-Level Characterization

Wei et al. [49] introduce a way of detecting Trojans using the method of GLC with thermal conditioning. Thermal conditioning means that an IC is intentionally heated

unevenly. This method exploits the fact that the leakage power increases exponentially with temperature.

The aim is to eliminate correlations when measuring leakage power that are caused by dependencies of gates with other gates. By heating correlated gates differently, more variability is introduced into the calculation results. The entire process is calculated using a simulation model. Simulation results can be used to obtain scaling factors in order to calibrate the measurement procedure to minimize measurement variations caused by process variations. The advantage of applying this method is the full characterization of all gates of an IC.

The detection of Trojans by GLC with thermal conditioning is not suitable for large circuits, because properties are determined for the entire circuit. Attackers can exploit this fact and inject ultra-small Trojans, whose impact will disappear in measurement noise [48]. To make the process scalable and thus useful for the analysis of large ICs, Wei and Potkonjak [48] extend the process by adding a preceding step of segmentation, which decomposes a large circuit into many small sub-circuits.

The segmentation criteria are chosen in a way that ensures that the results of the following GLC are as accurate as possible. The segmentation process itself is accomplished by varying an amount of primary input vectors and, at the same time, 'freezing' the other input vectors. The circuit part obtained by segmentation is now seen as an independent part of the circuit (i.e., a segment). A GLC with thermal conditioning, which is applied to this segment, provides information about the presence of Trojan. A subsequent identification mechanism based on the principle of 'guess and verify' should provide information about the type and the input pins of any existing Trojan.

4.13 Data Leakage by Trojans

Jin and Makris [23] present an attack that aims to leak an Advanced Encryption Standard (AES) key. This is achieved by manipulating the transmission signal of a wireless link within its tolerances. This work is the first presented attack in the analog domain.

Using an external guardian core, Das et al. [18] propose an approach to prevent data leaks via the data bus caused by hardware Trojans. The watchdog monitors the access behavior to the main memory by comparing each memory access with an emulated version of it. If the memory accesses match, it is approved by the guard, otherwise it is discarded. The emulation of the memory accesses is done in the software applications that are executed on the system.

4.14 Defining Threat Models

A new class of attacks against cryptographic algorithms is presented by Ali et al. [3]. These so-called multi-level attacks are based on the interaction of multiple people involved in the hardware design and production process.

The authors present an example of such an attack, where the secret key of a hardware implementation of the AES algorithm is leaked through a hidden channel of the power supply.

The authors assume a link between the developer and the operator of cryptographic hardware. The developer inserts the malicious circuit into the original circuit. After manufacturing, the secret key can be read by the operator. Collaboration between the two parties is necessary because otherwise, without knowledge of the technology used, the operator would not be able to read the key.

4.15 Combining Different Approaches to Side-Channel Analysis

A first approach for combining the possibilities of different methods of side-channel analysis is presented by Koushanfar and Mirhoseini [28], which is an advancement of [29].

The framework allows analysis by different side channels and the use of different evaluation methods, such as quiescent current, leakage current, and delay.

The mathematical analysis of the measurement results is based on GLC and a subsequent statistical analysis. Here, a new objective function is defined for the linear program that takes into account the *submodularity* of the problem: The impact of a Trojan on a side channel is greater the smaller the analyzed circuit is.

After GLC, the deviation of the measurement results (obtained by the various side-channel analyses) from the expected values is calculated for each gate. A sensitivity analysis is performed so that possible malicious circuits can be detected. The design of the Trojan determines the effect on the side channels. Some Trojans are more likely to have an impact on power consumption, while others will affect the performance. The measurement results of the different analyses (*multimodal*) are combined to achieve a higher detection rate. Experiments show that if the Trojans are inserted into areas with adequate sensitivity, the detection rate is 100%. The reverse is true as well: This method can also be used to find areas in which Trojans are most difficult to detect.

Lamech et al. [30] also evaluate the effectiveness of the combined results from the analyses of various side channels. In contrast to [28], however, they present no general model in which the results of different side-channel analyses could be combined. They show that by combining transient power and performance, followed by regression analysis, higher detection rates can be achieved in contrast to using each side channel on its own. Experiments show detection rates of up to 80% when they are not calibrated. After calibration, a detection rate of 100% can be achieved.

4.16 Increase Trojan Activation by Additional Flip-Flops

To increase the probabilities of state transitions in circuits during functional tests, Salmani et al. [44, 42] present an approach to insert dummy scan flip-flops into the original circuit layout. The consideration is that Trojans will be activated completely or in part and, thereby, have an impact on side channels. For example, the gates of a Trojan could switch – the additional energy consumption could be visible in the course of a side-channel analysis. The biggest motivation is to shorten the authentication process of an IC.

The method is carried out as follows: first, a threshold for the switching probability is determined that takes into account technical and economic considerations. Then, the switching probabilities of each sub-net are determined and the nets are divided into two groups, one with high and one with low switching probabilities. A dummy scan flip-flop is connected to nets with low switching probability to increase the probability of a transition. A net that tends to logic 0 is succeeded by a scan flip-flop that draws the output of the net (in a testing setup) to logic 1 if needed. The opposite is true for nets that tend to logic 1.

4.17 Avoiding Trojans

Waksman and Sethumadhavan [46] present an approach to avoid Trojans which is intended to prevent the occurrence of the trigger condition of digital, deterministic triggers.

Untrusted data is not monitored and manipulated within functional groups, but rather on their inputs and outputs. The idea is that data is scrambled and obscured in a controlled manner, so that a trigger cannot find a valid trigger condition and, therefore, a Trojan will never become active. The following types of triggers are considered:

1. ticking timebomb, 2. single-shot cheat code and 3. sequence cheat code.

A ticking timebomb is a time-controlled trigger that is activated when a predetermined number of clock cycles N has elapsed. The clock cycles that have already passed are usually determined using a counter. If this counter is always reset before reaching state N, a Trojan will never become active. This can be achieved by periodically resetting the entire digital system (*power reset*). The reset interval must be smaller than the testing period T that the obligatory functional test requires. If an attacker wants to achieve that a ticking timebomb will be activated, this must occur within N clock cycles. However, if $N < T$, the Trojan will be activated and detected during the functional tests.

Data-based triggers can be separated into single-shot cheat code and sequence cheat code triggers. Single-shot cheat code triggers are activated when a certain rare value is applied to the monitored interface. To prevent a rare value from being knowingly applied to an input of a compromised functional unit by an attacker, it is obfuscated by encryption in such a way that it no longer meets the trigger condition. Simple encryption methods are, for example, XOR, PUF, or random values. This approach is valid for non-computational units such as memory, etc. In order to protect computational units (e.g., ALU), *homomorphic* functions are used. Homomorphic functions obey the following rule: $f(g(x), g(y)) = g(f(x, y))$. An example of a homomorphic function is: $x^2y^2 = (xy)^2$. If we assume that the computational function is the squaring, a non-trustworthy value x that is to be processed is multiplied by a random value y before it is squared. To obtain a valid result, the result on the output of the functional unit must be divided by y^2.

The last class of triggers, sequence cheat codes, is countered by scrambling and inserting dummy loads. The scrambling is achieved by simple reordering. If this is not possible, dummy loads can be inserted into the data stream. A maximum number n of bits must be defined which is then allowed as a valid sequence. After n processed bits, a dummy load is inserted to avoid execution.

4.18 Using Ring Oscillators for Trojan Detection

Zhang and Tehranipoor [51] use a network of ring oscillators to detect injected Trojans. A ring oscillator is a simple circuit for generating oscillations, consisting of an odd number of similar gates.

The principle of detection is based on the fact that the frequency of a ring oscillator is influenced by physical parameters. Accordingly, the frequency also depends on the supply voltage V_{DD}. If V_{DD} drops, the propagation delay of the gates increases. This in turn means that the delay of the entire ring oscillator and, therefore, its cycle duration increases, which is equivalent to a drop in frequency.

A drop in V_{DD} occurs when a gate draws current. If the use of CMOS is assumed, this is the case with each switch of a transistor of a gate, hence, with every state change. If a Trojan is implemented in an IC, adjacent ring oscillators will see a stronger decrease in V_{DD} and, consequently, a frequency drop compared to ICs without a Trojan.

To achieve the best possible coverage, ring oscillators are distributed over the entire chip area. Using statistical methods, in which the frequencies of the built-in ring oscillators are evaluated (simple outlier analysis, principal component analysis and advanced outlier analysis), a detection rate of 100% is achieved. The validation using an FPGA provides detection rates between 80% and 100%. The robustness of the approach against direct attacks is considered very high, because the manipulations have direct impact on the frequency of the ring oscillators and, therefore, become visible immediately in functional tests.

5 Conclusion

Malicious hardware presents a serious threat to infrastructure components. Taking into account that powerful embedded systems, such as smartphones, are widely used today, a gloomy picture can be painted for the future. As they are extensively connected to the internet, a broad class of targets is available.

In this chapter, we showed how malicious circuits could be used to subvert an infrastructure by taking over its components. Further, we presented the basic components of hardware Trojans, i.e., trigger and payload mechanisms. We also outlined how hardware Trojans have evolved over time, as well as the methods to counter them.

References

[1] S. Adee. The Hunt For The Kill Switch. *Spectrum, IEEE*, 45(5):34 –39, may. 2008. ISSN 0018-9235. doi: 10.1109/MSPEC.2008.4505310.

[2] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan Detection using IC Fingerprinting. In *Security and Privacy, 2007. SP '07. IEEE Symposium on*, pages 296 –310, may. 2007. doi: 10.1109/SP.2007.36.

[3] Sk. Subidh Ali, Rajat Subhra Chakraborty, Debdeep Mukhopadhyay, and Swarup Bhunia. Multi-level attacks: An emerging security concern for cryptographic hardware. In *Proc. Design, Automation & Test in Europe Conf. & Exhibition (DATE)*, pages 1–4, 2011. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5763307.

[4] M. S. Anderson, C. J. G. North, and K. K. Yiu. Towards Countering the Rise of the Silicon Trojan. Technical report, 12 2008. URL http://dspace.dsto.defence.gov.au/dspace/bitstream/1947/9736/1/DSTO-TR-2220%20PR.pdf.

[5] Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2001. ISBN 0471389226. URL http://www.cl.cam.ac.uk/~rja14/Papers/SE-14.pdf.

[6] M. Banga and M. S. Hsiao. Trusted RTL: Trojan detection methodology in pre-silicon designs. In *Proc. IEEE Int Hardware-Oriented Security and Trust (HOST) Symp*, pages 56–59, 2010. doi: 10.1109/HST.2010.5513114.

[7] M. Banga and M.S. Hsiao. A region based approach for the identification of hardware Trojans. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 40 –47, jun. 2008. doi: 10.1109/HST.2008.4559047.

[8] M. Banga and M.S. Hsiao. VITAMIN: Voltage inversion technique to ascertain malicious insertions in ICs. In *Hardware-Oriented Security and Trust, 2009. HOST '09. IEEE International Workshop on*, pages 104 –107, 2009. doi: 10.1109/HST.2009.5224960.

[9] M. Banga and M.S. Hsiao. A Novel Sustained Vector Technique for the Detection of Hardware Trojans. In *VLSI Design, 2009 22nd International Conference on*, pages 327 –332, 2009. doi: 10.1109/VLSI.Design.2009.22.

[10] Mainak Banga. Partition based Approaches for the Isolation and Detection of Embedded Trojans in ICs. Master's thesis, Faculty of Virginia Polytechnic Institute and State University, 09 2008. URL http://scholar.lib.vt.edu/theses/available/etd-09042008-155719/unrestricted/MS_Thesis_Mainak.pdf.

[11] Mainak Banga, Maheshwar Chandrasekar, Lei Fang, and Michael S. Hsiao. Guided Test Generation for Isolation and Detection of Embedded Trojans in ICs. In *GLSVLSI '08: Proceedings of the 18th ACM Great Lakes symposium on VLSI*, pages 363–366, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-999-9. doi: http://doi.acm.org/10.1145/1366110.1366196.

[12] Alex Baumgarten, Michael Steffen, Matthew Clausman, and Joseph Zambreno. A case study in hardware Trojan design and implementation. *International Journal of Information Security*, 10:1–14, 2010. ISSN 1615-5262. URL http://dx.doi.org/10.1007/s10207-010-0115-0.

[13] G. Bloom, R. Simha, and B. Narahari. OS support for detecting Trojan circuit attacks. In *Hardware-Oriented Security and Trust, 2009. HOST '09. IEEE International Workshop on*, pages 100 –103, 2009. doi: 10.1109/HST.2009.5224959.

[14] Rajat Chakraborty, Francis Wolff, Somnath Paul, Christos Papachristou, and Swarup Bhunia. MERO: A Statistical Approach for Hardware Trojan Detection. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 396–410. Springer Berlin / Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-642-04138-9_28.

[15] R.S. Chakraborty, S. Paul, and S. Bhunia. On-demand transparency for improving hardware Trojan detectability. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 48 –50, jun. 2008. doi: 10.1109/HST.2008.4559048.

[16] Z. Chen, X. Guo, A. Nagesh, M. Reddy, and A. Maiti. Hardware Trojan Designs on BASYS FPGA Board. http://filebox.vt.edu/users/xuguo/homepage/publications/csaw08.pdf, 2008. URL http://filebox.vt.edu/users/xuguo/homepage/publications/csaw08.pdf.

[17] DARPA. Trust in Integrated circuits (TIC). http://www.darpa.mil/MTO/solicitations/baa07-24/index.html, Mar 2007. URL http://www.darpa.mil/MTO/solicitations/baa07-24/index.html.

[18] A. Das, G. Memik, J. Zambreno, and A. Choudhary. Detecting/preventing information leakage on the memory bus due to malicious hardware. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010*, pages 861 –866, mar. 2010. URL http://portal.acm.org/citation.cfm?id=1871135.

[19] Defense Science Board, Department of Defense, U.S.A. High Performance Microchip supply. http://www.cra.org/govaffairs/images/2005-02-HPMS_Report_Final.pdf, 02 2005. URL http://www.cra.org/govaffairs/images/2005-02-HPMS_Report_Final.pdf.

[20] Dongdong Du, Seetharam Narasimhan, Rajat Chakraborty, and Swarup Bhunia. Self-referencing: A Scalable Side-Channel Approach for Hardware Trojan Detection. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 173–187. Springer Berlin / Heidelberg, 2010. URL http://dx.doi.org/10.1007/978-3-642-15031-9_12.

[21] Matthew Hicks, Murph Finnicum, Samuel T. King, Milo M. K. Martin, and Jonathan M. Smith. Overcoming an Untrusted Computing Base: Detecting and Removing Malicious Hardware Automatically. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 159 –172, May 2010. doi: 10.1109/SP.2010.18.

[22] S. Jha and S.K. Jha. Randomization Based Probabilistic Approach to Detect Trojan Circuits. In *High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE*, pages 117 –124, 2008. doi: 10.1109/HASE.2008.37.

[23] Y. Jin and Y. Makris. Hardware Trojans in Wireless Cryptographic ICs. *Design Test of Computers, IEEE*, 27(1):26 –35, jan. 2010. ISSN 0740-7475. doi: 10.1109/MDT.2010.21.

[24] Yier Jin and Y. Makris. Hardware Trojan detection using path delay fingerprint. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 51 –57, 2008. doi: 10.1109/HST.2008.4559049.

[25] Chong Hee Kim and J.-J. Quisquater. Faults, injection methods, and fault attacks. *IEEE Design & Test of Computers*, 24(6): 544–545, 2007. doi: 10.1109/MDT.2007.186.

[26] Lok-Won Kim, J.D. Villasenor, and C.K. Koc. A Trojan-resistant system-on-chip bus architecture. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1 –6, 2009. doi: 10.1109/MILCOM.2009.5379966.

[27] Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou. Designing and implementing malicious hardware. In *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–8, Berkeley, CA, USA, 2008. USENIX Association. URL http://portal.acm.org/citation.cfm?id=1387709.1387714.

[28] F. Koushanfar and A. Mirhoseini. A Unified Framework for Multimodal Submodular Integrated Circuits Trojan Detection. 6 (1):162–174, 2011. doi: 10.1109/TIFS.2010.2096811.

[29] Farinaz Koushanfar, Azalia Mirhoseini, and Yousra Alkabani. A Unified Submodular Framework for Multimodal IC Trojan Detection. In Rainer Böhme, Philip Fong, and Reihaneh Safavi-Naini, editors, *Information Hiding*, volume 6387 of *Lecture Notes in Computer Science*, pages 17–32. Springer Berlin / Heidelberg, 2010. URL http://dx.doi.org/10.1007/978-3-642-16435-4_2.

[30] C. Lamech, R. Rad, M. Tehrani, and J. Plusquellic. An Experimental Analysis of Power and Delay Signal-to-Noise Requirements for Detecting Trojans and Methods for Achieving the Required Detection Sensitivities. (99), 2011. doi: 10.1109/TIFS.2011.2136339. Early Access.

[31] Jie Li and J. Lach. At-speed delay characterization for IC authentication and Trojan Horse detection. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 8 –14, 2008. doi: 10.1109/HST.2008.4559038.

[32] Lang Lin, W. Burleson, and C. Paar. MOLES: Malicious off-chip leakage enabled by side-channels. In *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, pages 117 –122, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5361303.

[33] Lang Lin, Markus Kasper, Tim Güneysu, Christof Paar, and Wayne Burleson. Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 382–395. Springer Berlin / Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-642-04138-9_27.

[34] D. Mclntyre, F. Wolff, C. Papachristou, S. Bhunia, and D. Weyer. Dynamic evaluation of hardware trust. In *Hardware-Oriented Security and Trust, 2009. HOST '09. IEEE International Workshop on*, pages 108 –111, 2009. doi: 10.1109/HST.2009.5224990.

[35] S. Narasimhan, Dongdong Du, R.S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, and S. Bhunia. Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach. In *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, pages 13 –18, 2010. doi: 10.1109/HST.2010.5513122.

[36] Michael Nelson, Ani Nahapetian, Farinaz Koushanfar, and Miodrag Potkonjak. SVD-Based Ghost Circuitry Detection. In Stefan Katzenbeisser and Ahmad-Reza Sadeghi, editors, *Information Hiding*, volume 5806 of *Lecture Notes in Computer Science*, pages 221–234. Springer Berlin / Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-642-04431-1_16. 10.1007/978-3-642-04431-1_16.

[37] Miodrag Potkonjak, Ani Nahapetian, Michael Nelson, and Tammara Massey. Hardware Trojan horse detection using gate-level characterization. In *DAC '09: Proceedings of the 46th Annual Design Automation Conference*, pages 688–693, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-497-3. doi: http://doi.acm.org/10.1145/1629911.1630091.

[38] R. Rad, J. Plusquellic, and M. Tehranipoor. Sensitivity analysis to hardware Trojans using power supply transient signals. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 3 –7, jun. 2008. doi: 10.1109/HST.2008.4559037.

[39] R. Rad, J. Plusquellic, and M. Tehranipoor. A Sensitivity Analysis of Power Signal Methods for Detecting Hardware Trojans Under Real Process and Environmental Conditions. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18 (12):1735 –1744, 2010. ISSN 1063-8210. doi: 10.1109/TVLSI.2009.2029117.

[40] R.M. Rad, Xiaoxiao Wang, M. Tehranipoor, and J. Plusquellic. Power supply signal calibration techniques for improving detection resolution to hardware Trojans. In *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, pages 632 –639, 11 2008. doi: 10.1109/ICCAD.2008.4681643.

[41] J. A. Roy, F. Koushanfar, and I. L. Markov. Extended abstract: Circuit CAD tools as a security threat. In *Proc. IEEE Int. Workshop Hardware-Oriented Security and Trust HOST 2008*, pages 65–66, 2008. doi: 10.1109/HST.2008.4559052.

[42] H. Salmani, M. Tehranipoor, and J. Plusquellic. New design strategy for improving hardware Trojan detection and reducing Trojan activation time. In *Hardware-Oriented Security and Trust, 2009. HOST '09. IEEE International Workshop on*, pages 66 –73, 2009. doi: 10.1109/HST.2009.5224968.

[43] H. Salmani, M. Tehranipoor, and J. Plusquellic. A layout-aware approach for improving localized switching to detect hardware Trojans in integrated circuits. In *Proc. IEEE Int Information Forensics and Security (WIFS) Workshop*, pages 1–6, 2010. doi: 10.1109/WIFS.2010.5711438.

[44] H. Salmani, M. Tehranipoor, and J. Plusquellic. A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time. (99), 2011. doi: 10.1109/TVLSI.2010.2093547. Early Access.

[45] Adam Waksman and Simha Sethumadhavan. Tamper Evident Microprocessors. In *SP '10 Proceedings of the 2010 IEEE Symposium on Security and Privacy*, pages 173 –188, may. 2010. doi: 10.1109/SP.2010.19.

[46] Adam Waksman and Simha Sethumadhavan. Silencing Hardware Backdoors. In *Proc. IEEE Symp. Security and Privacy (SP)*, pages 49–63, 2011. doi: 10.1109/SP.2011.27. URL http://www.cs.columbia.edu/~simha/preprint_oakland11.pdf.

[47] Xiaoxiao Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic. Hardware Trojan Detection and Isolation Using Current Integration and Localized Current Analysis. In *Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS '08. IEEE International Symposium on*, pages 87 –95, 2008. doi: 10.1109/DFT.2008.61.

[48] Sheng Wei and M. Potkonjak. Scalable segmentation-based malicious circuitry detection and diagnosis. In *Proc. MayAugust*, pages 483–486, 2010. doi: 10.1109/ICCAD.2010.5653770.

[49] Sheng Wei, Saro Meguerdichian, and Miodrag Potkonjak. Gate-level characterization: Foundations and hardware security applications. In *Proc. 47th ACM/IEEE Design Automation Conf. (DAC)*, pages 222–227, 2010. URL http://ieeexplore.ieee.org/ielx5/5510861/5522347/05522644.pdf?tp=&arnumber=5522644&isnumber=5522347.

[50] F. Wolff, C. Papachristou, S. Bhunia, and R.S. Chakraborty. Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme. In *Design, Automation and Test in Europe, 2008. DATE '08*, pages 1362 –1365, mar. 2008. doi: 10.1109/DATE.2008.4484928.

[51] Xuehui Zhang and Mohammad Tehranipoor. RON: An on-chip ring oscillator network for hardware Trojan detection. In *Proc. Design, Automation & Test in Europe Conf. & Exhibition (DATE)*, pages 1–6, 2011. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5763260.