Remote Activation of Hardware Trojans via a Covert Temperature Channel

Priyabrat Dash, Chris Perkins, and Ryan M. Gerdes(✉)

Utah State University, Logan, UT 84322, USA
priyabrat.dash@aggiemail.usu.edu, perkinsck@gmail.com,
ryan.gerdes@usu.edu

Abstract. A hardware trojan (HT) is produced through the malicious tampering of an integrated circuit design. Depending on its placement and purpose, an HT may cause data leakage or corruption, computational errors, reduced system performance, and temporary or permanent denial-of-service through the disabling or destruction of the chip. The varied geographic locales involved in designing, fabricating, and testing a design allow an attacker ample opportunity to insert an HT. In this paper we propose a method to enable the remote activation of HT, via a covert temperature channel, across a network. Through experimentation, our activation method is shown to be feasible on modern computers. In addition, its design is tolerant of process variation to ensure that it can be reliably fabricated. The design was validated using industry standard STMicroelectronics 65 nm technology and shown to be undetectable against present detection techniques. We discuss the major challenges associated with such HT and future research needs to address them.

Keywords: Hardware Trojan · Remote activation · Covert channel · Detection

1 Introduction

Electronic devices are integral to almost every aspect of our lives, but the emergence of hardware specific threats has led some to reconsider the trustworthiness of the hardware used for information processing [2]. Globalization and cut-throat competition in the electronics industry has led to the outsourcing of integrated circuit (IC) manufacturing to untrustworthy foundries [23]. Because chip designers are no longer in control of the production of ICs, affordable, yet unreliable, third-party fabricators make it possible for attackers to make malicious modifications to a circuit before it is fabricated. Additionally, attackers could modify designs through the compromise of the computer aided design (CAD) tools used by designers; malicious circuitry may also already exist in the blackbox intellectual property (IP) modules commonly used in IC design. These malicious and unintended additions to ICs are called hardware Trojans (HT), and they are of particular concern to military, financial and industrial sectors as they can lead to functionality errors, performance reduction, denial-of-service, or information leakage [9].

B. Thuraisingham et al. (Eds.): SecureComm 2015, LNICST 164, pp. 294–310, 2015.
DOI: 10.1007/978-3-319-28865-9_16

In an attempt to evade detection, HT are often composed of two parts: the payload, and the trigger [22]. The payload is the circuitry designed to effect the goal of the attacker through interaction with the targeted IC. The trigger is intended to keep the trojan stealthy by activating the payload only after some attacker-defined event has taken place.

In this paper, we devise a remotely activated *analog* HT trigger that enables the payload when a certain temperature is reached at the core of the infected circuit. Our target is a chip residing on, or connected to, the logic board in a computer connected to the network. The trigger is influenced via a covert temperature channel, wherein an attacker is able the raise the computer's temperature remotely by sending a large number of requests to the target computer over the network.

A temperature sensitive trigger is ideal from an attacker's point of view for two reasons. Firstly, the possibility of remote activation allows an attacker to achieve their ends without physical access to a target device. Secondly, an analog temperature switch-based trigger is much smaller and quieter—i.e. its area and static and dynamic power draw are lower—than the combinatorial and sequential circuits conventionally used to trigger trojans. As will be shown in Section 6, our trigger, which consists of a temperature switch (tuned to respond to a specific temperature) along with a simple temperature sensor, uses much less power than can be detected using current power-based side channel detection techniques [4]. It should be noted that, rather than introducing an additional temperature sensor, it would be possible to hijack the signal from a temperature sensor that may already be present in an IC (such additions are extremely common for monitoring), making our trojan trigger even stealthier. Because of its analog nature and lack of interaction with the digital circuitry of the IC, our trojan is also able to evade parametric detection techniques such as path delay [14].

Our threat model is detailed and validated in Section 2, while we discuss our trigger design in Section 3. In Section 4, we present a modification to our design that ensures it is tolerant to the process variation inherent in modern fabrication; the design is verified via simulation in Section 5. Section 6 demonstrates that our trigger is able to evade current detection approaches. Related trojans/triggers are discussed in Section 7. Finally, Section 8 concludes the paper.

2 Threat Model

We consider an attacker who has implanted a gate-level trojan into a component of interest that is later installed into a computer with a network interface offering some service (e.g. web or database server; Figure 1). It is assumed that the attacker knows the proximity of the infected component to the CPU in the computer, and that the computer itself is in an environment with a steady, easily predicted ambient temperature, such as a temperature controlled data center. The trojan is composed of an internal trigger and payload; through remote interaction with the computer the attacker is able to induce an internal state that triggers the payload. Specifically, the internal trigger is designed to activate the payload when the temperature of the component exceeds a pre-defined threshold; i.e. we utilize a temperature-based covert channel [29]. Remote activation

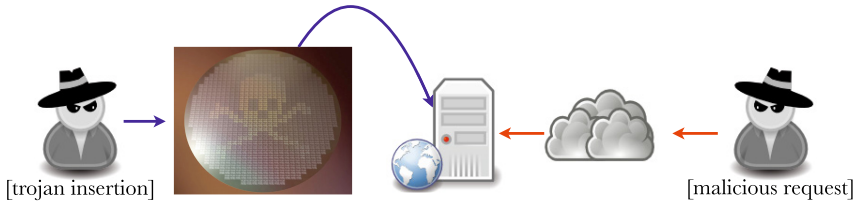

Fig. 1. Threat model: an attacker implants a temperature-triggered hardware trojan into component that is installed in a server. The attacker activates the trojan by sending it spurious requests that cause utilization to increase, causing a rise in temperature that triggers the trojan.

is achieved by sending requests to the service running on the computer at a rate sufficient to increase the CPU utilization, and hence the temperature of the computer [18]. For example, in the case of a webserver an attacker in control of a botnet could initiate new connections or issue page requests to cause excess resource consumption. If the network interface is Internet/externally facing the attacker could initiate the attack remotely, otherwise they would require access to the private network on which the computer resides.

The payload will depend on the goals of the attacker and the component in which the trojan is implanted. For example, data corruption or leakage could be effected if the trojan is located in a southbridge-like component, controller for the data storage device, memory controller, or even peripheral component (e.g. Ethernet card). The attacker could also opt to simply disable the computer by disconnecting the supply voltage of one of these components during the duration of the attack, or even permanently by creating a short inside the component that results in burnout.

Validation of Threat Model

While the existence of temperature-based covert channels is well established, existing work has focused on either coarse-grained case temperature or CPU temperature measurement [18,29]. Our threat model specifies that the attacker knows only the proximity to the CPU. Thus, to establish the temperature threshold at which to trigger the payload, the attacker must know how CPU utilization will affect different regions of the computer.

To this end, we measured the temperature inside a Dell Optiplex 960 at four different locations (Figure 2), using calibrated Texas Instruments LM35A temperature sensors (accuracy $\pm 0.2^{\circ}$ [25]), at different CPU utilization levels. The sampling rate was 100 S/s; every second the last 100 samples would be averaged to obtain the temperature for the previous second. The `cpulimit` program was used to control the utilization of the resource consumption `busy` program, which spawns a specified number of threads that each execute an infinite loop [1]. Data was collected for several utilization levels (e.g. 0,10,100,400%) and profiles (cycling between different utilization levels) over the course of many days,

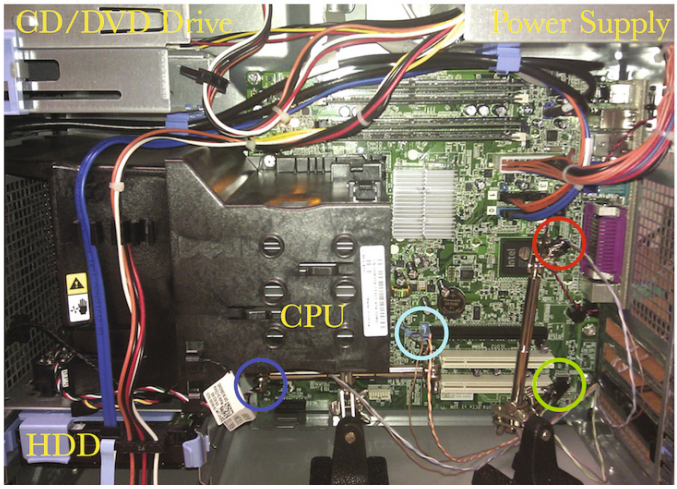

Fig. 2. The experimental setup used to validate the threat model. Sensor locations are located at circles. The case was closed during experiments

of which Figure 3 is representative. In this experiment we varied the utilization level periodically: one hour at 400% and one hour at idle (the processor in the computer is dual core with Hyperthreading enabled). We see that sensors closest to the CPU experience the greatest increase in temperature, but that in each case at least 1° increase is observed.

Thus, in this instance, it is feasible for an attacker to design a temperature trigger based on the crossing of a threshold temperature and still have it be effective at differing positions in the computer. Additionally, we note that the maximum temperature achieved through our testing predictably occurred after 25 minutes of 400% utilization (Figure 3), a condition, given the underutilization of most datacenter servers [6,7], that is unlikely to occur during regular or even heavy usage. This helps to minimize the chance that a trojan will be activated by benign workloads. Finally, it is possible that such increased, unscheduled CPU activity could set off CPU workload monitor alarms. The attacker would need to be aware of this possibility and initiate an attack at times when the targeted machine is untended (e.g. during the nighttime) or set a target temperature that can be attained before intervention can be performed.

3 Hardware Trojan Trigger Design

An effective HT trigger needs to be accurate and stealthy in its operation. An accurate trigger will only switch the payload on after a specific event defined by an attacker has taken place. Our trigger consists of an analog temperature sensor circuit along with a multistage inverter designed to switch on at a certain triggering temperature using a voltage signal provided by the sensor (Figure 4). The output of the switching circuitry is connected to the gate of a MOSFET, whose

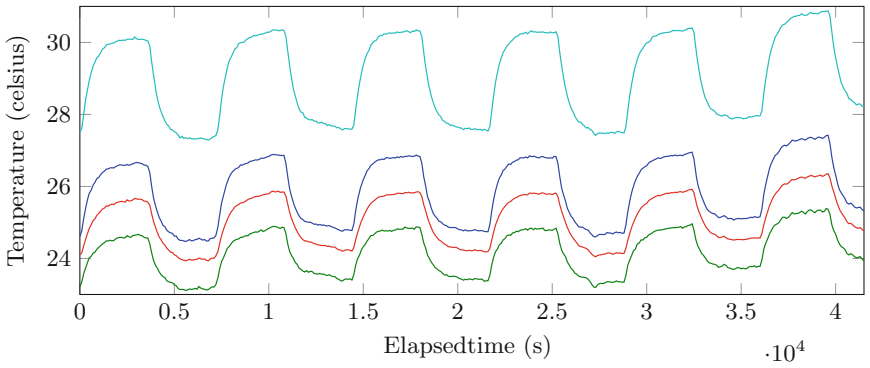

Fig. 3. Temperature at different locations inside the computer with a varying utilization level (0% to 400% periodic). The line color corresponds to the sensor locations given in Figure 2.

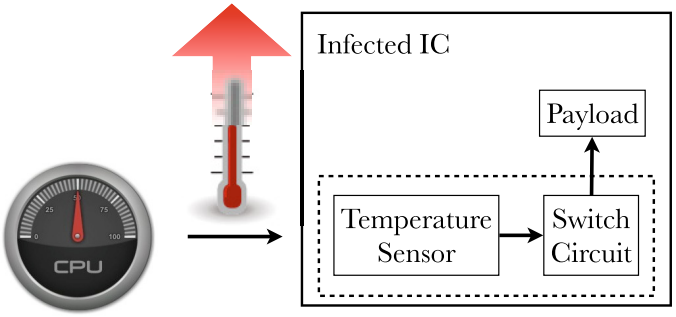

Fig. 4. The proposed method for triggering the payload of a HT: Increased CPU utilization causes a rise in the temperature of an infected IC. The trojan trigger utilizes a low-power temperature sensor to feed switching circuitry designed to activate the payload when the IC temperature exceeds a given threshold.

drain and source are connected to the payload ground line and the IC ground, respectively. When the switching output is logic high the MOSFET switch is closed, allowing power to flow to the payload. The transition temperature at which the switching occurs is predetermined by the attacker before insertion, as per Section 2. We now highlight the design of the sensing and switching components of the trigger.

3.1 Temperature Sensor

The temperature sensor circuit used for our simulations is a BiCMOS design based on the work of [27]. It consists of a cascaded configuration of p-type MOSFETs and pnp BJTs (Figure 9, Appendix). The output of the sensor V_{sen} is the sum of emitter-base voltage V_{eb} of the BJT $Q2$ and source-gate voltage

V_{sg} of the pMOSFET $M2$ ($M1$ is cascaded to the emitter of $Q2$ to obtain the summed output). Under stable biasing both V_{eb} and V_{sg} exhibit negative temperature dependence. As V_{eb} and V_{sg} exhibit complimentary non-linearity against temperature, the linear combination of both voltages results in a high linearity temperature sensor. The pMOSFET portion of the current reference circuit ($M4$ and $M6$) provides stable biasing for $M2$. $Q1$ and $Q2$ are placed in a current mirror configuration ensuring that the collector currents in both $Q1$ and $Q2$ remain the same. This configuration acts as bias for the pnp BJT $Q2$, thereby reducing power consumption and augmenting the decrease in nonlinear deviations in V_{eb}.

The temperature sensor was selected based on the following criteria:

1. *Sensitivity*: This refers to the amount that the output voltage of the sensor changes per degree Celsius. Typical values for solid state temperature sensors range from ±0.5mV° to ±9mV° [10]. The higher the sensitivity the fewer the number of stages needed in the switching circuitry, which reduces overall power consumption (discussed in Section 3.2). The nominal sensitivity of our sensor is -3.4mV° with a linearity of 99.96% [27]. We increased the sensitivity of the circuit to -10.12mV° by increasing the width of the $M1$ transistor by 5.3 m.
2. *Power*: The power of the overall design should be minimized. This circuit used 56.31nW at 1.2V, the lowest power circuit we could find.
3. *Positive or negative voltage correlation*: Refers to whether the output voltage of the sensor increases or decreases with an increase/decrease in temperature. Again, this affects our switching circuitry: a positive correlation requires an even number of switching stages (four or six) to ensure that output of the final stage is high to activate our payload power switch. Our sensor has a negative correlation, therefore the switching circuit can use an odd number of stages, which results in fewer inverters and hence less power.

3.2 Switching Circuitry

The overriding concern for our trigger is that the voltage present at the gate of the payload power switch is high at exactly the temperature selected by the attacker but not before. That is, the transition region between switch on and off should be very small. To accomplish this we designed switching circuitry (Figure 10, Appendix) consisting of five common source amplifier stages in sequence [10].

The first stage has its amplifying transistor's gate attached to the output of the temperature sensor. The basic operation is that the final stage T_{out} will remain at a low voltage until the infected IC heats up to the trigger temperature and then it will quickly transition to the logic high value of the payload supply switch. The logic level of the switch circuit should remain low until the trigger temperature is reached to avoid false triggers. The transition point must reside at an input voltage that corresponds to a temperature just below the maximum temperature for swift response.

The transition from low logic to high logic has to be without any delay for smooth payload activation. Therefore the sensitivity of the switch circuit

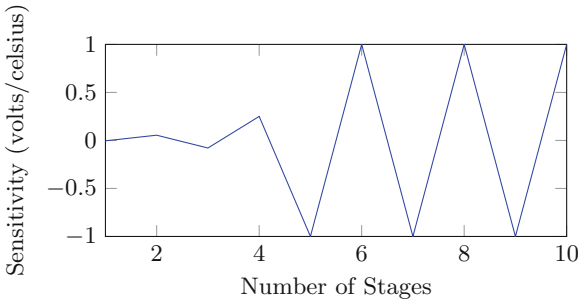

Fig. 5. The effective sensitivity of our temperature sensor at the output of our switching circuitry versus the number of stages in the switch.

is increased by adding additional inverter stages for sharp edge transition. In Figure 5 the sensitivity of individual stages is plotted. The sensitivity values are positive for an even number of stages and negative for odd stages. It can be seen that after the initial increase from stage one to stage five the value remains constant. That is, for a negative correlation sensor, adding additional stages will only increase the power consumption without any improvement in the sensitivity. Thus, the ideal number of stages for our switching circuitry is five, in terms of both minimizing power consumption and providing a fast switching response.

The transition point of the switch circuit—i.e. the temperature at which payload is activated—can be selected by changing the bias voltage of the transistors in a particular stage. For an n-stage circuit, the relationship between the output of a stage can be formulated with respect to the output of the previous stage as:

$$V_i = VDD - \frac{\beta n}{2} \left(V_{i-1} - V_{th}\right)^2 R_{pmos} \tag{1}$$

where V_i is the output of the i-th stage, VDD is the supply voltage, β the transistor's gain, V_{th} the threshold voltage of the transistor, and R_{pmos} the derivative of the drain current I_D with respect to drain-to-source voltage V_{DS} of the pmos transistor. The above equation is for the case of an ideal MOSFET; however, the exact values obtained from simulation are close to the values obtained by the above equation (error range of 11.82mV to -2.72mV).

Unfortunately, this switching architecture is extremely sensitive to changes in threshold voltage. For example, a change in the threshold voltage by 1mV results in a change in the transition point by far more than 1mV. This is quite a significant problem with regards to the accuracy of the attack temperature, as threshold voltage is deeply affected by process variation. That is, a significant change in V_{th} will cause the payload to be powered at a temperature other than that stipulated by the attacker. In varying the parameters that affect V_{th} in a Monte Carlo simulation, we observed a standard deviation of the threshold voltage of 45.62mV. Therefore we have need of a process tolerant switching circuitry to ensure robust and accurate trigger operation.

4 A Process Invariant Design

Continuous advancement in transistor dimensions scaling has led to rampant variations in process parameters affecting the operations of integrated circuits. Variability in channel length (L), oxide thickness (T_{ox}), and transistor threshold voltage (V_{th}) increase drastically in the nanometer technology [8]. Process variation in nanometer technology is categorized into random variations and systematic variations [3]. The systematic variations are caused by device manufacturing process variation such as chemical-mechanical planarization (CMP) [13]. Random variations, mainly caused by geometrical abnormalities, are considered to be the major contributors in 65 nm technology, with L and V_{th} being the most significant contributors to the random component [30]. In the case of our trigger, random parametric variations (threshold voltage and geometrical abnormalities), over which the attacker has no control, lead to unreliable triggering at different temperatures. Therefore the need for process tolerant circuitry, with a small area overhead, is unavoidable.

In this section we present a self-tuning inverter comparator circuit, based on the work of [21], in combination with the parallel gates technique [12], to achieve significant improvement in the circuit level variation tolerance. Our HT trigger of Figure 4 is thus slightly modified to incorporate this new circuitry: the output of the temperature sensor is connected to a self-tuning inverter circuit which in turn is fed to parallel gate inverter chains, replacing our original switching circuitry, to obtain a highly reliable temperature dependent trigger switch. Incorporating a self-tuning inverter comparator circuit gives complete controllability to the attacker to choose the temperature at which the trojan is activated. The process and temperature variation tolerant architecture provides error free targeted trigger temperature operation.

4.1 Self-tuning Inverter Comparator Circuit

The inverter comparator circuit (Figure 11, though designed for PWM application [21], is ideal for use in our trojan circuit due its low voltage operating range, low power consumption, and small area. The basic purpose of the circuit is to set its inherent threshold voltage to a predetermined voltage, independent of process and temperature variation; this is achieved via the principle of negative feedback. The inverter comparator consists of master and slave sub-circuits (Figures 11(a) and 11(b)), with the transistors of both designed to be of the same dimensions. The transistors $MM2$ and $MM3$ act as inverters, generating the desired output voltage to tune the transistors $MM1$ and $MM4$ at the supply rails. The supply rail transistors act as variable resistors to balance the variation in the master input through a negative feedback loop (wire $w1$).

The inherent threshold voltage can be altered according to the attacker's need by changing the voltage at the master circuit's bias node (connection between $MM1$ and $MM4$ in Figure 11(a), Appendix) to the desired threshold value; i.e. the voltage at this node determines the inverter's threshold voltage. In the cases where changing the voltage at the bias node using a voltage source would be

undesirable or impossible, bias resistors can also be used to change the inherent threshold voltage value, to a limited extent. For example, the threshold voltage can be set at 0.5 VDD by using equal bias resistors ($R1$ and $R2$ in Figure 11(a)). A threshold voltage of 0.5 VDD equals to 600 mV at $VDD = 1.2$V, which causes the inverter logic in the master-slave circuit to switch at a temperature of 27°. The Miller capacitor C prevents tuning error and undesired oscillations by providing high DC gain and low AC gain, respectively. The output of the master-slave circuit, S_{out}, thus provides our cascaded inverter stages with an appropriately biased, self-tuning input.

4.2 Parallel Gate Inverter Circuit

Even with the inclusion of a self-tuning inverter comparator circuit, our simple cascade of inverter stages circuit is still subject to high output variability due to the sensitivity of the cascaded-inverter circuitry to threshold voltage variations of nMOS and pMOS transistors. This is because small process-induced variability present in early stages can be amplified until they lead to very large fluctuations in the trojan trigger temperature. Therefore, the need for variation tolerant inverter stages arises. The use of programmable threshold voltage inverters [20] would lead to larger inherent voltage deviation, and the power overhead associated with them is proportional to the bit count of digital circuits used to program the inherent voltage. Body biasing techniques [26] have also been proposed, but they are not suitable for our purpose as they are not efficient in tackling random variations and also increase the circuit complexity (area and power consumption).

The standard deviation of the threshold voltage of a transistor is inversely proportional to the square root of the width of the transistor[19]. We therefore increased the width of the transistors in the inverter circuit but failed to observe a drastic reduction in output variation. It wasn't until we also connected the transistors in parallel (effectively increasing the width of the entire inverter) that the random variations were reduced to an acceptable level. The efficacy of the parallel structure can be explained by noting that the variations in each transistor of an inverter is independent of the another transistor, hence using single gates lead to the amplification of the overall random variation in the circuit. Using a parallel gate structure as in Figure 12, Appendix, on the other hand, leads to nullification of independent V_{th} variations in the corresponding parallel transistors [12]. The parallel gate design also leads to a decrease in the input and output capacitance leading to marginally less dynamic power consumption. The higher area overhead is negated when the positive impact it has on suppressing the process variation is considered.

5 Simulation

For the trigger circuit design and Monte-Carlo simulations Cadence Virtuoso IC 6.1.5 with 65 nm, 1.2 V technology library CORE65LPSVT of STMicroelectronics was used. The circuit simulator involved was Spectre. The 65 nm technology

was part of an STMicroelectronics kit used for standard, industrial circuit design and simulations.

Initial simulation of the temperature sensor with the five stage inverter stage led to a sharp switch from low to high at the T_{out}. The operation for a single sweep maintains the switch transition at the required temperature levels. But the simulation results above do not take process variation into account. Therefore we ran a Monte-Carlo simulation with the STMicroelectronics kits inherent parameter variations values in the model files. The resultant output was very disappointing, as for only 10 iterations the standard deviation of the transition point was 26.26°. Large fluctuations from the desired trigger temperature can cause uncertainty in executing the attack. It was a challenge to maintain the low area and power consumption of the circuit. Numerous process variations tolerant circuit lead to very high area and power overhead. Therefore a trade-off between area, power and standard deviation was made improve the performance of the trojan trigger.

The simulation result using our process tolerant circuit was improved and displayed accurate functioning of the trojan. The self-tuning inverter circuit along with parallel gate structure when ran for 100 iterations of Monte-Carlo simulations resulted in a standard deviation of 1.48° (Figure 6), which is acceptable in a real attack scenario. We note that should an attacker be in a position to select which chips are shipped to end-users, then they could select for shipment only those chips that trigger most closely to the desired temperature.

The final total power consumption of our process tolerant trigger circuit was 72.34nW, with the temperature sensor occupying 18.36 m with a draw of 46.31nW, the self-tuning inverter 6.52 m and 17.67nW, and the parallel gates 1.52 m and 8.36nW. The pMOS transistors width for the parallel gate circuit

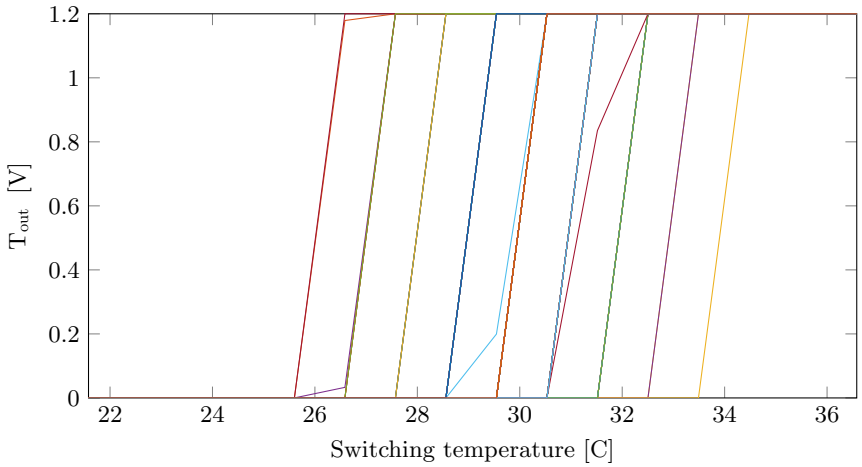

Fig. 6. The switching temperature for the final stage of the process resistant trojan circuitry for 100 Monte Carlo simulations. Target switching temperature was 30°. In 100 simulations a standard deviation of only 1.48° was observed.

were half that of five stage switch circuit. Therefore the area overhead was marginally more and the power consumption was also less considering a bigger circuit. The temperature sensor circuit used in our design can be directly used from one of the many target device motherboard temperature sensors. Utilizing on chip sensor will decrease the area requirement and power consumption of the trojan trigger making it even more difficult to detect.

6 Detectability of Design

Trojan detection is challenging because conventional post manufacturing test and validation processes are often incapable of discovering trojans with effective triggers. Assumption of trigger nodes in a basic benchmark circuit can itself lead to very large trigger sample space, making it almost impractical to test [5]. Therefore, deterministic and exhaustive testing approaches are infeasible. Many detection techniques have been proposed: activation techniques, which attempt to trigger trojans through various possible input combinations; side channel techniques, which monitor circuit side-channel parameters to discover abnormalities [4,17]; design for trust, which modifies the design process to make trojan insertion difficult [24]; and reverse engineering, which thoroughly examines the physical manifestation of the circuit [15]. While these techniques are effective, they are not comprehensive enough to eradicate trojans from modern circuitry. Additionally, these techniques commonly require a genuine, clean reference netlist to find any difference.

An ideal detection technique would be 1) able to detect small trojans, 2) non-destructive, 3) scalable, and 4) authenticate chips in a short time. We consider a few comprehensive detection techniques to evaluate our trigger's detectability. The analog nature of the circuit makes it difficult to analyze the mixed signal circuit as it does not cause any abnormal behavior during the target device simulation. Only after the implementation can it go active when the trigger condition is satisfied. Functional testing using automatic test pattern generation for a mixed signal would not be able to detect it, as it does not alter the functionality of a genuine IC. Similarly, the path delay analysis [14], which examines the propagation delay of the critical paths in a circuit and compares the values obtained with those of a non-infected IC, would be evaded as our trigger is not on a the functional path of the IC. Trojan detection and isolation using current transient current analysis [28] detects switching activity by measuring power consumption at different locations in the chip. Given the published data and our trigger power draw, the trojan circuitry could hide safely in the process variation. We therefore believe that a power-based side channel analysis, as exemplified by [4], to be the most likely detection method to succeed.

In [4] the Karhunen-Loeve (KL) expansion is used to differentiate the noise associated with process variation from the power consumed by the trojan. This approach has been shown to identify small trigger instances with area equivalent to 0.01% of the total size of the circuit, in the presence of random parameter variation as high as $\pm 7.5\%$. The method works by determining the eigenvalue

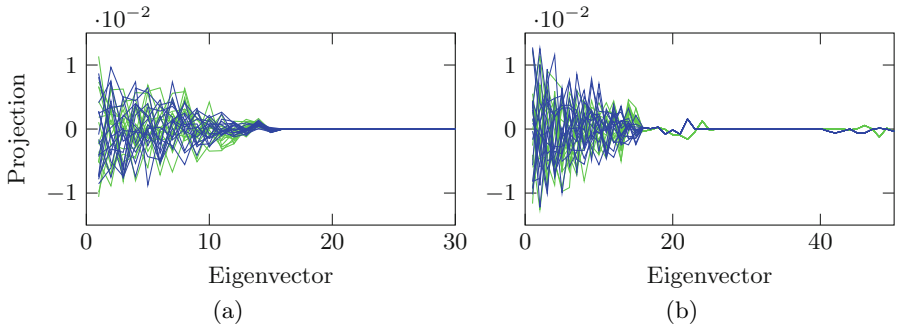

Fig. 7. The eigenvalue spectrum of the trojan (blue) and non-trojan (green) traces. (a) 30 and (b) 50 contiguous points of 16 traces for the non-trojan and trojan devices.

spectrum of the residuals of the signals after measurement noise and nominal power consumption are removed from a power trace for the IC. The spectrum of a trace from a non-trojan device will tend to zero as the number of eigenvectors increases; because of the additional signal (that of the trojan circuitry) the spectrum of a trojan trace will not approach zero at the same rate as the non-trojan.

We used a 256-bit AES circuit as the benchmark circuit to evaluate its detectability. The circuit was re-synthesized, flattened, power optimized, and then analyzed for power consumption, as outlined in [4]. Synopsys Core Synthesis Tools with 65nm, 1.2V technology library CORE65LPSVT of STMicroelectronics was used for the synthesis of the circuit with and without the Trojan. We used Cadence NCSim for the circuit simulation and switching activity analysis. Synopsys PrimeTime PX was used for power analysis and trace file generation. All the power traces were obtained by a time based power simulation with a 50 MHz clock frequency.

The process corner Synopsys library files from the CORE65LPSVT were used to create power traces with differing levels of process noise, as follows: Each sample point of a record would consist of a value picked at random in the range of the best and worst case. The nominal case was deemed to be the average of the best and the worst case. Process noise was recovered by subtracting the nominal case from the generated traces. Following the procedure outlined by [4], we performed the KL analysis on the portions of the trace with the *lowest* process noise, consisting of 30 and 50 contiguous points using 16 traces for each of the non-trojan and trojan traces (Figures 7(a) and 7(b)). We used the same points with 300 and 500 traces, respectively (Figures 8(a) and 8(b)). A worst case analysis, consisting of 600 points from low-noise regions of the trace, along with 6000 traces for the non-trojan and trojan was also undertaken. In all cases the eigenvalue spectrums do not separate at any sample point and are indistinguishable; thus, the trigger is undetectable.

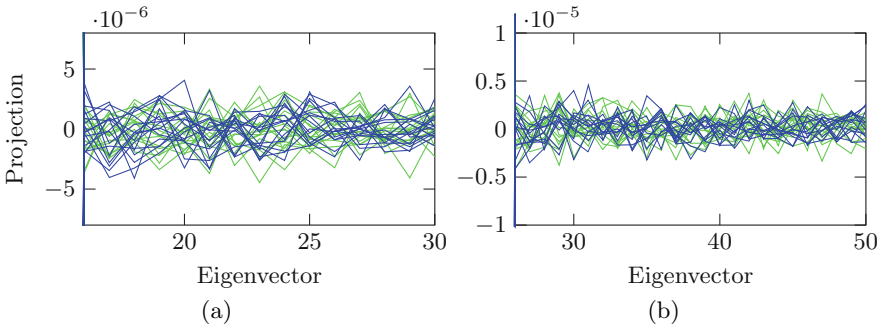

Fig. 8. The eigenvalue spectrum of high-order eigenvectors (where differences should be apparent) for the trojan (blue) and non-trojan (green) traces. (a) 30 and (b) 50 contiguous points of 300/500 traces for the non-trojan and trojan devices.

7 Related Work

There have been numerous active research papers on trojan detection techniques and implementations [9]. Different trojan implementations with effects on circuit parameters such as power, delay, and performance have been proposed [23]. Trojans can also lead to circuit degradation without affecting the overall functionality over a large period of time [9]. Most closely related to our work are trojans that leveraged thermal emissions to leak of data [16], and were triggered via a thermal process [11]. In this latter work, a thermally triggered trojan was implemented on a BASYS FPGA board. An increase in circuit activity increased the temperature causing the Trojan to trigger. Ring oscillators and counters were used to obtain the desired temperature level; i.e. the temperature was artificially increased through high power consuming hardware implanted by the attacker. Though the trigger was thermal based they needed physical access to the board; also, the use of registers needed to activate the trojan would lead to noticeable power consumption. Our low power trigger works in a mixed signal circuit without the need for any physical access to the target device.

8 Conclusion

In this paper, we have presented a fully functional hardware trojan trigger which targets computer logic boards but can also be extended to be maliciously included in any networked device. The trojan can be triggered remotely by increasing the core temperature of the targeted IC through increased network activity, which in turn leads to higher core utilization levels. The low power consumption, high process tolerant operation, and analog implementation mark the trigger as a very potent trojan example. Adversaries with such flexible, accurate and undetectable trojans pose a major threat to IC security. This new attack

vector points to the need for improved methods for detecting and preventing mixed signal hardware trojans.

Acknowledgements. The authors are grateful to Dr. Chris Winstead of Utah State University for providing technical guidance on the cascaded switching circuitry and use of his laboratory facilities.

References

1. CPU usage limiter for Linux (2015). https://github.com/opsengine/cpulimit
2. Abramovici, M., Bradley, P.: Integrated circuit security: new threats and solutions. In: Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, pp. 55. ACM (2009)
3. Agarwal, K., Nassif, S.: Characterizing process variation in nanometer CMOS. In: 44th ACM/IEEE Design Automation Conference, DAC 2007, pp. 396–399. IEEE (2007)
4. Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., Sunar, B.: Trojan detection using IC fingerprinting. In: IEEE Symposium on Security and Privacy, SP 2007, pp. 296–310. IEEE (2007)
5. Banga, M., Chandrasekar, M., Fang, L., Hsiao, M.S.: Guided test generation for isolation and detection of embedded trojans in ICs. In: Proceedings of the 18th ACM Great Lakes symposium on VLSI, pp. 363–366. ACM (2008)
6. Barroso, L.A., Clidaras, J., Hölzle, U.: The datacenter as a computer: An introduction to the design of warehouse-scale machines. Synthesis Lectures on Computer Architecture **8**(3), 1–154 (2013)
7. Benik, A., Ventures, B.: The sorry state of server utilization and the impending post-hypervisor era (2013). https://gigaom.com/2013/11/30/the-sorry-state-of-server-utilization-and-the-impending-post-hypervisor-era/
8. Bernstein, K., Frank, D.J., Gattiker, A.E., Haensch, W., Ji, B.L., Nassif, S.R., Nowak, E.J., Pearson, D.J., Rohrer, N.J.: High-performance CMOS variability in the 65-nm regime and beyond. IBM Journal of Research and Development **50**(4.5), 433–449 (2006)
9. Chakraborty, R.S., Narasimhan, S., Bhunia, S.: Hardware trojan: threats and emerging solutions. In: IEEE International High Level Design Validation and Test Workshop, HLDVT 2009, pp. 166–171. IEEE (2009)
10. Chang, M.H., Liu, C.P., Huang, H.P.: Chip implementation with combined temperature sensor and reference devices based on DZTC principle. Electronics Letters **46**(13), 919–921 (2010)
11. Chen, Z., Guo, X., Nagesh, R., Reddy, A., Gora, M., Maiti, A.: Hardware trojan designs on BASYS FPGA board. Embedded system challenge contest in cyber security awareness week-CSAW (2008)
12. Garg, R., Khatri, S.P.: A variation tolerant circuit design approach using parallel gates
13. He, L., Kahng, A., Tam, K.H., Xiong, J.: Simultaneous buffer insertion and wire sizing considering systematic CMP variation and random leff variation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **26**(5), 845–857 (2007)

14. Jin, Y., Makris, Y.: Hardware trojan detection using path delay fingerprint. In: IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2008, pp. 51–57. IEEE (2008)
15. Kash, J.A., Tsang, J.C., Knebel, D.R.: Method and apparatus for reverse engineering integrated circuits by monitoring optical emission (December 17, 2002), US Patent 6,496,022
16. Kiamilev, F., Hoover, R., Delvecchio, R., Waite, N., Janansky, S., McGee, R., Lange, C., Stamat, M.: Demonstration of hardware trojans. DEFCON, 16 (2008)
17. Lin, L., Burleson, W., Paar, C.: Moles: malicious off-chip leakage enabled by side-channels. In: Proceedings of the 2009 International Conference on Computer-Aided Design, pp. 117–122. ACM (2009)
18. Liu, H.: A measurement study of server utilization in public clouds. In: Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, DASC 2011, pp. 435–442. IEEE Computer Society, Washington, DC (2011). http://dx.doi.org/10.1109/DASC.2011.87
19. Orshansky, M., Nassif, S., Boning, D.: Design for manufacturability and statistical design: a constructive approach. Springer Science & Business Media (2007)
20. Segura, J., Rossello, J., Morra, J., Sigg, H.: A variable threshold voltage inverter for CMOS programmable logic circuits. IEEE Journal of Solid-State Circuits **33**(8), 1262–1265 (1998)
21. Tan, M.T., Chang, J.S., Tong, Y.C.: A process-and temperature-independent inverter-comparator for pulse width modulation applications. Analog Integrated Circuits and Signal Processing **27**(1–2), 95–107 (2001)
22. Tehranipoor, M., Wang, C.: Introduction to Hardware Security and Trust. SpringerLink: Bücher. Springer (2011). https://books.google.com/books?id=bNiw9448FeIC
23. Tehranipoor, M., Koushanfar, F.: A survey of hardware trojan taxonomy and detection (2010)
24. Tehranipoor, M., Salmani, H., Zhang, X., Wang, X., Karri, R., Rajendran, J., Rosenfeld, K.: Trustworthy hardware: Trojan detection and design-for-trust challenges. Computer **7**, 66–74 (2010)
25. Instruments, T.: LM35 Precision Centigrade Temperature Sensors. datasheet (2015)
26. Tschanz, J., Bowman, K., De, V.: Variation-tolerant circuits: circuit solutions and techniques. In: Proceedings of the 42nd Annual Design Automation Conference, pp. 762–763. ACM (2005)
27. Wang, R.L., Yu, C.W., Yu, C., Liu, T.H., Yeh, C.M., Lin, C.F., Tsai, H.H., Juang, Y.Z.: Temperature sensor using BJT-MOSFET pair. Electronics Letters **48**(9), 503–504 (2012)
28. Wang, X., Salmani, H., Tehranipoor, M., Plusquellic, J.: Hardware trojan detection and isolation using current integration and localized current analysis. In: IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, DFTVS 2008, pp. 87–95. IEEE (2008)
29. Zander, S., Branch, P., Armitage, G.: Capacity of temperature-based covert channels. Communications Letters, IEEE **15**(1), 82–84 (2011)
30. Zhao, W., Liu, F., Agarwal, K., Acharyya, D., Nassif, S.R., Nowka, K.J., Cao, Y.: Rigorous extraction of process variations for 65-nm CMOS design. IEEE Transactions on Semiconductor Manufacturing **22**(1), 196–203 (2009)

Appendix: Circuits Used in the Trojan Trigger Design

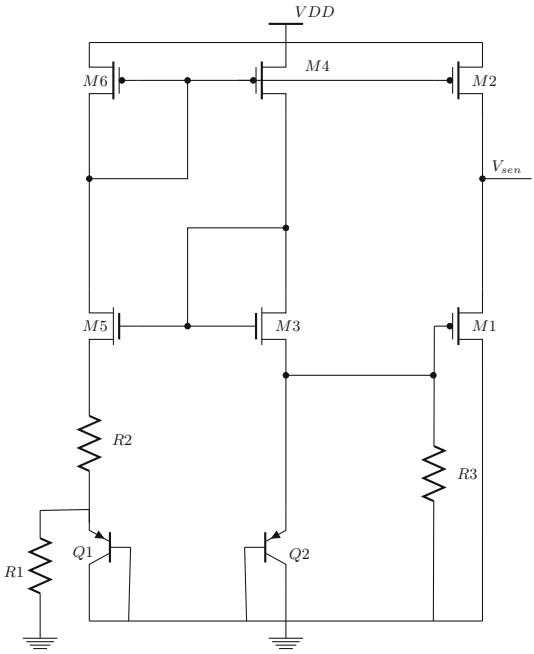

Fig. 9. The low power BiCMOS temperature sensor circuit used in our trojan trigger. Based on [27]. The temperature sensor circuit is the core of the trigger as it must accurately produce the voltage that corresponds to the selected triggering temperature. The output of the temperature sensor V_{sen} is given to the switch circuit.

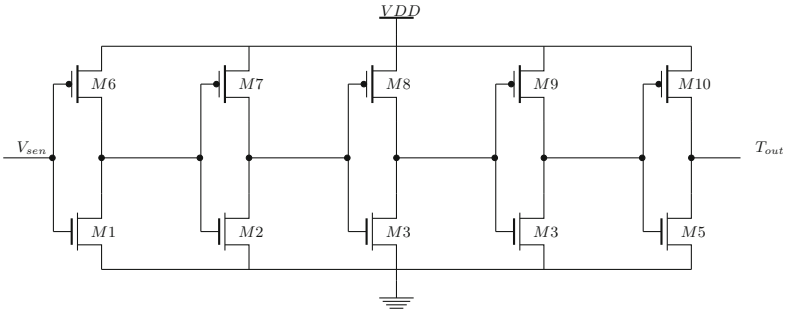

Fig. 10. A cascaded chain of inverters that switch from logic 0 to VDD at a particular temperature. The input to the inverter chain is V_{sen}, the output given from the temperature sensor circuit. When the output $T_{out} = VDD$ the trojan is triggered. Multiple states are added to get a sharp transition and correct inversion of the logic only at the specified temperature. Our switch circuit has an odd number of inverter stages as the temperature sensor has a negative sensitivity.

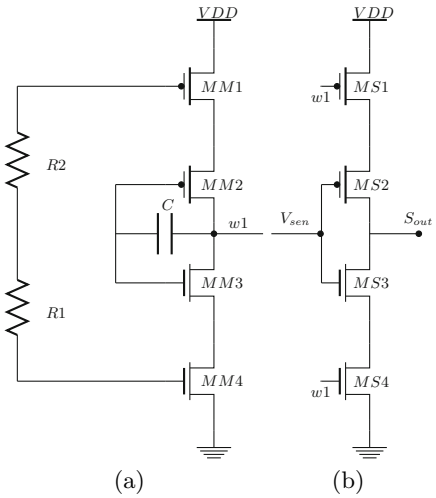

Fig. 11. The (a) master and (b) slave portions of our self-tuning comparator circuit. The purpose of the circuit is to ensure that the first stage of the cascaded inverters has a very low V_{th} variation, as switching variation caused by deviation in V_{th} is amplified through each stage of the inverter chain. The output S_{out} serves as the input to the cascade of parallel gates.

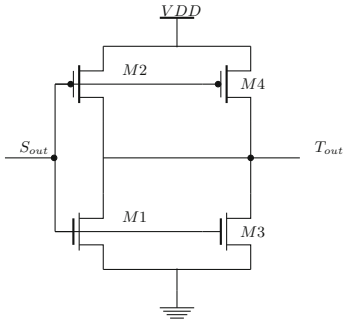

Fig. 12. The process invariant parallel-gate switching circuitry.