ADobf: Obfuscated Detection Method against Analog Trojans on I^{2}C Master-Slave Interface

(Invited Paper)

Mezanur Rahman Monjur, Sandeep Sunkavilli and Qiaoyan Yu
Department of Electrical and Computer Engineering, University of New Hampshire
Durham, NH 03824, USA

***Abstract*—[1] Hardware Trojan war is expanding from digital world to analog domain. Although hardware Trojans in digital integrated circuits have been extensively investigated, there still lacks study on the Trojans crossing the boundary between digital and analog worlds. This work uses Inter-integrated Circuit (I^{2}C) as an example to demonstrate the potential security threats on its master-slave interface. Furthermore, an obfuscated Trojan detection method is proposed to monitor the abnormal behaviors induced by analog Trojans on the I^{2}C interface. Experimental results confirm that the proposed method has a high sensitivity to the compromised clock signal and can mitigate the clock mute attack with a success rate of over 98%.**

***Index Terms*—Hardware Trojan, analog Trojan, Trojan detection, I^{2}C, obfuscation.**

I. Introduction

Fabless design and outsourcing fabrication have become a prevalent business model to help the semiconductor companies to maximize their profits. However, the globalized business model raises a serious concern on the trustworthiness of outsourced electronic devices. For instance, malicious modification, a.k.a hardware Trojan (HT), could be placed in the original design to alter logic function or leak information at various phases of the long circuits and systems supply chain [1]. Although there are comprehensive surveys on digital Trojans [2], the investigation of analog Trojans is still in its youth. Indeed, the Trojan war does occur not only in the digital domain; it is sprawling to the analog domain, too. Recent literature [3]–[6] calls attention to analog-circuit triggered Trojans or Trojans in analog/Radio Frequency (RF) circuits. Along the line of that call, this work studies hardware Trojans crossing the digital and analog domains.

Different from common digital Trojans, analog Trojans use analog triggering mechanisms or/and analog circuit based payloads [4]. Because of the small size and analog characteristics, it is challenging to detect analog Trojans by scrutinizing hardware footprint or side-channel signals. The work [7] uses a logic ring oscillator and a multi-purpose controller to detect A2 analog Trojans [5]. The sensor sensitivity mostly depends on the Trojan switching frequency. The run-time Trojan detection R2D2 method presented in [6] identifies a set of concerning signals and then initiates a hardware interrupt when there is abnormal toggling on these guarded signals.

[1]This work is partially supported by the NSF grants No.1652474 and No.1717130.

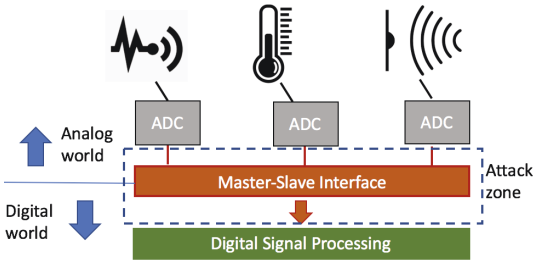


Fig. 1: Attack surface between analog and digital worlds.

Complementary to the general studies on analog Trojans, this work investigates and then mitigates the security threats on the communication interface between digital master and analog slave devices shown in Fig. 1. More specifically, Inter-Integrated Circuit (I^{2}C) interface is adopted as our study subject. Since the clock line plays a critical role in I^{2}C communication, it is imperative to mitigate attacks on the clock line (SCL) and improve the attack resilience of the data line (SDA) against analog Trojans. The related work [8] introduces a frequency sensor to detect the tampered clock period. However, that method is effective only when the attacks mute the clock signal for a period of time, rather than a single clock cycle. In addition, no evidence is provided in [8] to demonstrate if the frequency sensor is able to handle clock split attacks. Once the additive detection module is known in public, attackers could compensate clock cycles to disguise the attack.

Aiming for the hardware Trojans across Analog and Digital domains, we propose an obfuscated Trojan attack detection method named *ADobf* to provide a high sensitivity on the compromised clock cycles and protect the detection unit itself with an obfuscation key. Our method strengthens I^{2}C communication channels with minor overhead.

II. Attacks Crossing Analog and Digital Domains

Attacks implemented on the boundary of analog and digital worlds are different than those in the digital domain with regard to adversary, attack means and cost, visibility, effective period, and challenges posed on attack detection. In this work, we use I^{2}C interface as an example to analyze the attack across the analog and digital domains.

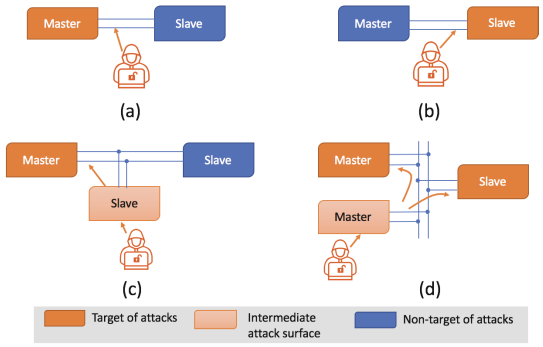


Fig. 2: Models for the attack on master-slave communication channels. (a) Direct attack on master, (b) direct attack on slave, (c) indirect attack on master via a compromised slave, and (d) indirect attack on master and slave via another compromised master.

A. Attack Models

The goal of crossing domain attacks is either manipulating the signal from the slave (in analog domain) to disturb the master (in digital domain), or impersonating the master to mislead the slave. Figure 2 depicts four scenarios. The top two cases in Fig. 2 are direct attacks, which cause the signal on the master-salve interface to lose its integrity. In contrast, the bottom two cases in Fig. 2 show an indirect (sophisticate) attack, where adversary first compromises another slave or master and then impersonates that device to inject malicious signals. The case (d) in Fig. 2 indicates that an indirect attack will potentially affect more devices than a direct attack.

B. Demonstration of Attack Examples

1) Attack on I²C Data Line SDA: We continue to use the I²C master-slave interface to demonstrate practical attacks. As shown in Fig. 3, three analog Trojans are inserted on the data link between a master and a slave. The Trojan is implemented in a format of pull-up or pull-down resistor R_{HT} and capacitor C_{HT}. The Trojan will be activated by an internal or external trigger signal. A pull-down resistor R_{HT} could mute the valid bit 'high', as shown in Fig. 4(a). In contrast, the capacitor based analog Trojan could lead the valid bit 'low' to go 'high', as shown in Fig. 4(b). Both Trojan cases in Fig. 4 demonstrate the analog characteristics of Trojans appeared in the I²C communication channel.

2) Attack on I²C Clock Line SCL: As the clock line plays a critical role in the master-slave communication protocol, it could be a primary target of analog Trojans. Since the slave is an off-chip device, the clock line spans both digital and analog domains. If there is no specific clock regulation available on board, the master-slave interface is vulnerable to clock attacks, including clock mute (i.e., no clock switching) and clock split (i.e., multiple switching transitions in one clock cycle). A simple resistor or capacitor shown in Fig. 3 will be sufficient to execute clock attacks. As I²C data transmission heavily relies on the clock line, a clock split attack will sabotage the data frame as shown in Fig. 5. If carefully crafted, the compromised

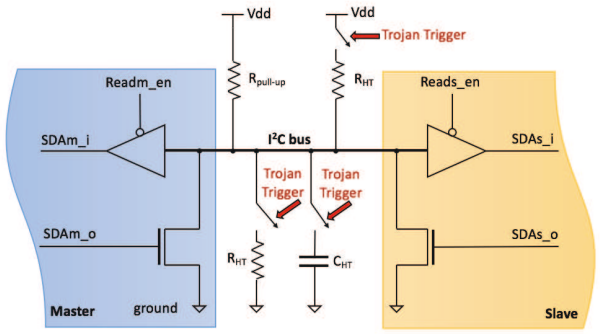


Fig. 3: Trojan attack in I²C communication channel.

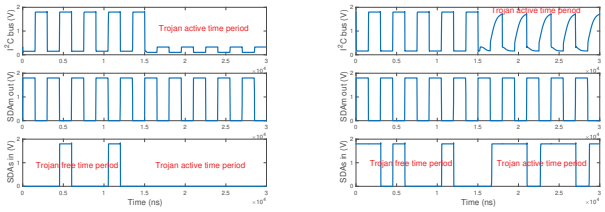


Fig. 4: Impact of (a) resistor and (b) capacitor based analog Trojans on I²C data line.

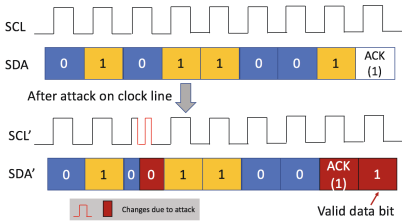


Fig. 5: Impact of a clock split attack on a data frame in I²C communication.

data frame will be acknowledged as if normal. It is easy to implement a clock mute attack, as well. Figure 6 illustrates the result of a clock mute attack performed in the master-slave interface between a Xilinx FPGA chip and an off-chip temperature sensor. The muted clock cycle successfully leads to the modification of the most significant bit of a data frame, which represents the sensed ambient temperature.

C. Challenges on Analog Attack Detection

Recent literature [4] reports that an analog Trojan formed by a few transistors is powerful enough to alter the system priority. Analog Trojans also appear in a phase locked loop and sensor controllers [9]. Compared with digital attacks, analog attacks can be performed on more surfaces and cost less hardware. As a result, analog Trojans pose more challenges on attack detection than digital Trojans. The main reasons are as follows: (1) more trigger mechanisms could be exploited to activate analog Trojans, (2) analog threshold for the attack determination makes the detection less reliable, (3) more ambient parameters would cause false positive, and (4) more asynchronous signals involved in the system would need more

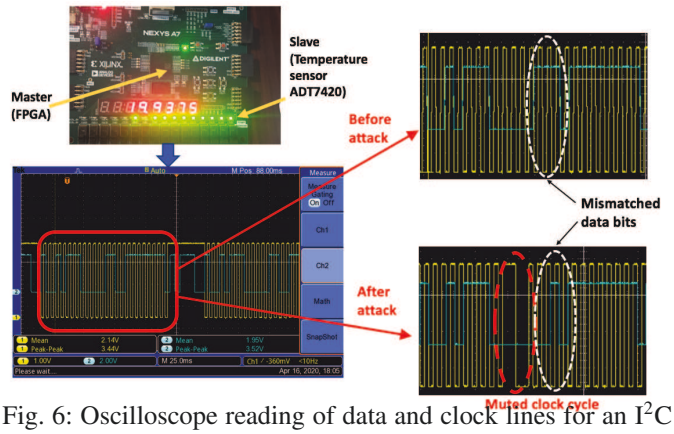


Fig. 6: Oscilloscope reading of data and clock lines for an I^{2}C interface in the scenario of clock mute attack.

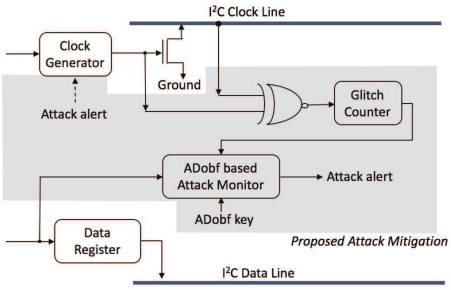


Fig. 7: Proposed architecture diagram.

precise sampling for the process of attack detection. Overall, the balance of false positive/negative and the sensitivity of attack detection is extremely challenging for the analog world.

III. Proposed ADobf Method for Analog Trojans

We propose an obfuscated Trojan detection method, *ADobf*, to thwart clock attacks in I^{2}C communication. Our ADobf method provides a high sensitivity to the compromised clock signal and meanwhile protects the attack detection module itself via obfuscation. The architectural diagram of ADobf is depicted in Fig. 7. The proposed ADobf compares the clock signal from the clock generator in the master device with the voltage from the I^{2}C clock line. Due to the switching delay induced by the open-drain transistor, there will be one (and only one) voltage glitch if the clock signal is propagated normally. The *Glitch Counter* counts the voltage glitches continuously and assists the *ADobf based Attack Monitor* to determine the presence of attacks. An attack alert will notify the clock generator and stop the data line from accepting malicious data frames.

More details of the ADobf based attack monitor are described in Algorithm 1. On each rising edge of the clock glitch, the glitch counter increases by 1. The counter content is compared with the clock threshold *Obf.threshold*, which is the correct number of clock glitches without any attacks. The fact that the glitch counter captures fewer glitches than *Obf.threshold* indicates some clock cycles being muted. On the contrary, if more glitches are obtained, a clock split attack could happen in the past measurement period. The value of

Algorithm 1: Proposed ADobf detection method against attacks on master-slave interface.

Data: clock line, data line, cmd instruction, Obf.key
Result: Attack alert
CLK_{glitch} = CLK_{gen} XNOR $CLK_{I^2C bus}$;
Obf.threshold = I^2C_cmd_decode (cmd.instr, Obf.key, cmd.start, cmd.stop);
Initialize Glitch.counter;
while CLK_{glitch} *rising edge* **do**
  Glitch.counter++;
  if *cmd.stop* **then**
    if *(Glitch.counter* $<$ *Obf.threshold)* **then**
      Clock mute attack detected;
    else
      if *(Glitch.counter* $>$ *Obf.threshold)* **then**
        Clock split attack detected;
      else
        Normal operation;
      end
    end
    Reset Glitch.counter;
  else
    Glitch.counter++;
  end
end
function I2C_cmd_decode(*);
  Calculate no. clock cycles (cmd.instr, cmd.start, cmd.stop) $\rightarrow$ NumCycle;
  NumCycle & Barrel_shifter(Obf.key) $\rightarrow$ Obf.threshold;
  return Obf.threshold;

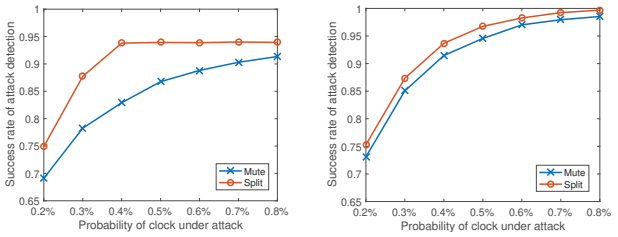
Obf.threshold is a run-time and obfuscated parameter, which is not available for attackers who do not have the obfuscation key *Obf.key*. Although attackers could calculate the number of clock cycles that each data frame takes to transfer over the I^{2}C interface, they will not be able to predict the key-dependent *Obf.threshold*. For simplicity, we use a barrel shifter to rotate the obfuscation key and then perform a bitwise AND logic with the number of clock cycles *NumCycle* in the attack-free scenario. Since only the master user knows *Obf.key*, our attack detection method is capable of resisting clock attacks originated from someone having access to the I^{2}C bus.

IV. Experimental Results

We evaluated the proposed ADobf in an I^{2}C interface. Our ADobf was implemented in VerilogHDL and the master module was synthesized with a 45nm FreePDK technology.

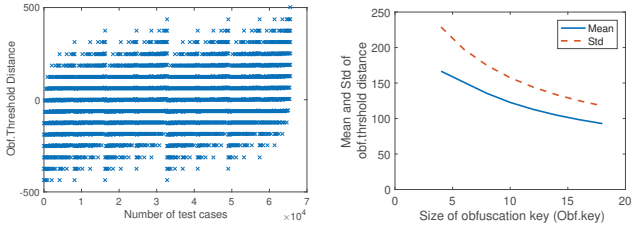
A. Effectiveness of Obfuscated Attack Detection

1) Success Rate of Detection: We transmitted 1000 data bits from an I^{2}C slave to a master. We set 7 and 8 respectively to the number of the address bits for the slave device and the size of each data frame. The clock line for the I^{2}C channel was randomly tampered so that some clock cycles were muted or split. Each compromised clock cycle led to the loss of one valid data bit. We varied the number of clock cycles under attack from 2 to 8. As shown in Fig. 8, the success rate of attack detection of the proposed method increases with the number of clock cycles under attack and the size of the obfuscation key. When the probability of clock tampering is 0.2%, our detection against the clock mute attacks is around 0.7. As the clock line was attacked with a higher frequency (0.8%), our method achieves a detection rate of above 0.95. Figures 8(a) and (b) also indicate that it is easier to detect clock



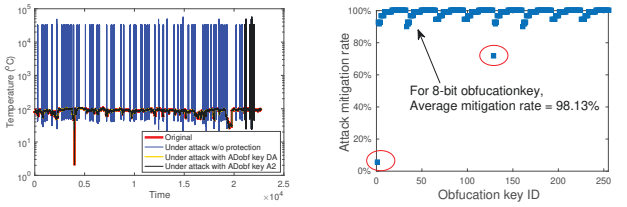
(a) (b)

Fig. 8: The success rate of proposed ADobf attack detection with the key size of (a) 4 and (b) 16.



(a) (b)

Fig. 9: Unpredictability of proposed ADobf. (a) Obf.threshold distance between the 8-bit correct key and wrong keys. (b) Statistics of Obf.threshold for various key sizes.



(a) (b)

Fig. 10: Attack mitigation achieved by proposed method. (a) A temperature benchmark from NAB transmitted over a compromised I^{2}C interface, and (b) attack mitigation rate for different obfuscation keys.

split attack than clock mute attack. A longer obfuscation key will further improve the success rate of clock attack detection.

2) Unpredictability on Obf.threshold: The strength of our ADobf method relies on the unpredictability of *Obf.threshold*. If the comparison threshold is known, adversary may find a way to bypass the countermeasure. Given a 16-bit obfuscation key, we randomly selected one key vector and used 1000 clock cycles to generate the *Obf.threshold* for all possible wrong keys. Figure 9(a) shows that the difference between the *Obf.threshold* obtained from correct and incorrect key cases is in a wide range of -500 to 500. Figure 9(b) further confirms that the average distance and the standard deviation are large for different key sizes. Thus, we conclude that our ADobf is capable of providing a high unpredictability and can thwart the attack from reverse engineering on our countermeasure.

3) Attack Mitigation Rate: We used an I^{2}C interface to transmit the Numenta Anomaly Benchmark (NAB) [10], *machine temperature system failure*. Each floating number for the sensed temperature was represented by 32 bits, 16 for integer and another 16 for fraction. We introduced 100 clock glitches in the NAB transmission. As shown in Fig. 10(a), the original NAB carries one abnormal data point (below 10 degrees); however, the clock mute attack increases the number of abnormal temperatures to close 100. Our ADobf method significantly alleviates the clock mute attack. The mitigation effect varies with different obfuscation keys. We examined the mitigation rate for all possible 8-bit obfuscation keys. As shown in Fig.10(b), our method achieves an average attack mitigation rate of 98%. Only two cases out of 256 are below the mitigation rate of 80%.

TABLE I: Comparison of Performance and Overhead.

Metric	Area (μm^2)	Dynamic Power (μw)	Leakage Power (μw)	Critical Path Delay(ns)
Baseline Master	2320.22	12.39	11.20	0.93
ADobf Master 4	2525.77	14.88	12.23	0.94
ADobf Master 8	2696.12	17.99	13.41	0.94

B. Hardware Cost

The proposed attack detection and mitigation method bring in moderate hardware overhead as reported in Table I. The ADobf module costs 8.9% and 16.2% more area for 4-bit and 8-bit keys, respectively. The power consumption increases accordingly. As our method runs in parallel with normal I^{2}C operation, the critical path delay is only increased by 1%.

V. Conclusion

Complementary to the studies on general digital and analog Trojans, this work focuses on the security threats on the I^{2}C interface between digital master and analog slave devices. Unlike the sensor-based abnormal clock detection, the proposed ADobf algorithm achieves a high attack detection (close to 1) against 0.8% attack injection rate. Our case study with NAB benchmark shows that the proposed method can mitigate over 98% clock mute attacks.

References

[1] J. Dofe and Q. Yu, "Novel dynamic state-deflection method for gate-level design obfuscation," *TCAD*, vol. 37, no. 2, pp. 273–285, 2018.

[2] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *D&T*, vol. 27, pp. 10–25, Jan 2010.

[3] K. Subramani, G. Volanis, M. Bidmeshki, A. Antonopoulos, and Y. Makris, "Trusted and secure design of analog/RF ICs: Recent developments," in *Proc. IOLTS*, pp. 125–128, 2019.

[4] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: Analog malicious hardware," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 18–37, 2016.

[5] Y. Hou, H. He, K. Shamsi, Y. Jin, D. Wu, and H. Wu, "On-chip analog trojan detection framework for microprocessor trustworthiness," *TCAD*, vol. 38, pp. 1820–1830, Oct 2019.

[6] Y. Hou, H. He, K. Shamsi, Y. Jin, D. Wu, and H. Wu, "R2D2: Runtime reassurance and detection of A2 Trojan," in *Proc. HOST*, pp. 195–200, 2018.

[7] D. Deng, Y. Wang, and Y. Guo, "Novel design strategy towards a2 trojan detection based on built-in acceleration structure," *TCAD*, pp. 1–1, 2020.

[8] R. Jiménez-Naharro, F. Gómez-Bravo, J. Medina-García, M. Sánchez-Raya, and J. A. Gómez-Galán, "A smart sensor for defending against clock glitching attacks on the I2C protocol in robotic applications," *Sensors*, vol. 17, no. 4, p. 677, 2017.

[9] V. V. Rao and I. Savidis, "Protecting analog circuits with parameter biasing obfuscation," in *Proc. LATS*, pp. 1–6, 2017.

[10] "The numenta anomaly benchmark." https://github.com/numenta/NAB.