Poster: Hardware Trojans – Detect and React?

Adrian Dabrowski, Peter Fejes, Johanna Ullrich, Katharina Krombholz, Heidelinde Hobel, Edgar Weippl
SBA Research, Vienna, Austria
E-Mail: (firstletterfirstname)(lastname)@sba-research.org

***Abstract*—Hardware Trojans are a serious threat. In comparison to their software counterparts, appropriate detection measures are still missing. The main reason is that there are no malware implementations to develop and test against. To solve this, we implemented a *Hardware Trojan Kit* (HTK) that enables the modular construction of Hardware Trojans based on the attributes *activation*, *covert communication*, *payload* and *detection*. We included invasive *detection* methods (i.e. inserted during the design phase to support the detection of modifications in post-production) as it will allow to test attack- and defense methods in a modular way. Then we analyzed these implementations for typical hardware structures. We identified multiple such structures that can serve as a warning signal. They will allow the development of more accurate detection methods.**

I. Introduction

Hardware Trojans are referred to as malicious changes of hardware that may result in functional changes of the respective device [1]. Their impact is underestimated by end-users, vendors, manufacturers and even the security community. As *Integrated Circuits* are an essential part of our everyday life – from household devices like washing machines to infrastructure facilities like power plants, from industrial to military applications – a malfunction has a significant impact on society and economy. Thus their successful detection and prevention is an issue of utter importance. However, hardware Trojans are introduced in the production process which distinguishes them from their software counterparts. For economic reasons, parts of the production chain are often outsourced to external contractors in different countries, yielding an untrustworthy production process.

During the design of a trusted hardware development process, we encountered a lack of real-world examples and implementations. We consider that vendors refrain from disclosure of information to protect intellectual property as well as contracts with customers, and to avoid negative impact on their reputation. Descriptions of Hardware Trojans are mainly found in academic publications [2, Ch. 2] and competitions like the Embedded Systems Challenge [3]. Nevertheless, the scientific community needs comprehensive samples for further investigation. Therefore we implemented them in a very modular way to facilitate construction and analysis. The kit consists of building blocks allowing the flexible generation of manifold Trojans. Thereby Trojans with different characteristics are created to evaluate newly developed detection methods.

The non-triviality of Trojan detection lies mainly in the complexity of hardware implementations. Additionally, such Trojans are targeted attacks tailored to a specific victim and a specific product. This is a crucial difference to typically untargeted (dragnet) software Trojans aiming at a mass of unspecific systems. In contrast, their hardware counterparts are adapted to their victim, exacerbating detection.

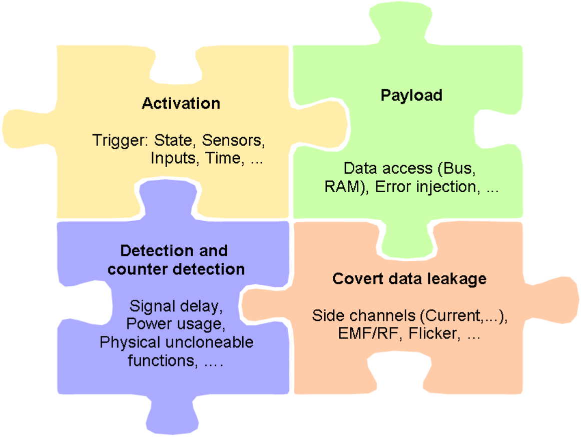
Activation
Trigger: State, Sensors, Inputs, Time, ...
Payload
Data access (Bus, RAM), Error injection, ...
Detection and counter detection
Signal delay, Power usage, Physical uncloneable functions, ...
Covert data leakage
Side channels (Current,...), EMF/RF, Flicker, ...

Fig. 1. Classification of Modules

The main contributions of this work are the development and providence of a modularized Hardware Trojans Kit based on existing descriptions, enabling the construction of manifold Trojans. This allowed us to identify typical design-time and net-list hardware Trojan structures and the location of threats in complex designs. Furthermore, these Trojans allow the evaluation of attack- and defense methods in a standardized fashion.

II. Methodology

The kit is assembled in a modular way and based on four characteristics: *activation*, *covert communication*, *payload* and *detection*, see Figure 1.

Activation: Hardware Trojans use trigger circuits to activate themselves and their payload. Normally triggers are rare events, e. g. specific data patterns or environment properties, and active Trojans are easier to detect than inactive ones. The implemented activation modules are *Thermal Trigger* (based on temperature dependent MOSFET characteristics), *Synchronous Counter*, *Asynchronous Counter*, *Hybrid Counter*, *Modified UART Parity Error*, *Character Counter*, *Character Finite State Machine* (FSM) and *ADC Trigger*.

Covert Communication: Confidential data, e. g. cryptographic keys, is secretly leaked over covert channels, i. e. ways not thus intended. Third parties that know of the channel characteristics are able to extract and decode the leaked information. The implemented modules are *AM radio transmitter*, *Modified UART Idle State*, *Modified UART Character Encoding*, *LED Transmission* and *Power Consumption Side Channel*.

This work was supported by the FIT-IT program (project number 835922) and the COMET K1 program by FFG (Austrian Research Funding Agency).

Payload: A malicious payload modifies/damages original hardware functionality and aims at inhibiting or disrupting normal operation. The implemented modules encompass *Modified FSM*, *UART with Modified Reset*, *UART Modified Sent Data*, *Clock Division Modification*, *Modified Carry Lookahead Adder*, *Modified Memory Enable Signal* and *Modified Memory Content*.

Detection: We consider a systematic view as incomplete without the inclusion of detection methods and other known countermeasures built into the Integrated Circuit - as they form one unit and may lead to false positives. Within this work, we focused on invasive methods which encompass special circuits for the detection of unauthorized modifications and implemented *Ring Oscillators*, *Physically Unclonable Functions* and *Shadow Circuits*.

III. Results

A. Malware Structures

We analyzed implementations on HDL/RTL level and on netlist level. We found the following characteristic structures being of interest for Trojan detection: *Asynchronous latches* are flip-flops which are either un-clocked, self-clocked or clocked by an external event/another gate, and not as usually tied to a global clock. An attacker might need such asynchronous elements to generate signals, hide information leakage or manipulation. *Gated wire/output signals* mean wires/outputs controlled by a gate influencing the signal, e. g. a controllable inverter forwarding false values to the succeeding gates by means of an XOR. A *ring oscillator* is a combinatorial loop without a constant frequency and an appealing sealing method. *Unused pins or bond wires* are convenient for covert channel dissemination. The data is then leaked by directly sending serial data or using the wire as an antenna.

Hardware implementations are frequently realized as finite-state machines. An attacker may desire to add *additional hidden states*. The *one-hot encoding* scheme realizes every additional state with an extra state bit, making it easy to modifications. However, with *sequential encoding* the number of bits stays the same as long as the state space is kept below the next power-of-2 boundary. *Gated reset signals* bring a flip-flop back to its initial state, which is suspicious in case of independence from the global reset. *Local or gated clocks* are independent from the global clock, which is in contrast to typically globally clocked logic. Such changes are feasible by means of a low number of changed or added gates.

B. Appropriate Detection Measures

Table I provides an overview of typical structures existing in the modules for activation, covert communication, payload and detection. However, some of these characteristics have a legitimate use as well. For example, ring oscillators used in some detection methods are similar to the possibly malicious modifications they try to detect.

TABLE I. Malware Structures in Kit Modules

	Extra slices	**Extra LUTs**	**Specific elements**	**Async. latch**	**Gated wire/output**	**RO**	**Unused pin/bond**	**Hidden FSM state(s)**	**Latch/FF wo/gated reset**	**Local/gated clock**
Activation										
Thermal trigger	101 (27+RO)	186 (7)	RO and a measurement circuit		x	x				
Synchronous counter	37/46	1								
Asyncronous counter	15x	6		x	x					(x)
Hybrid counter	21	14		x	x				x	
UART parity error	54	57	Extra comparator/gates		x					
Character counterx	10	10	Extra comparator			x				
Character FSMx	0	65	Extra FSM					x		
ADC trigger	30	45	FSM /monitoring circuit					x		
Covert channel										
AM radio	292	521	Unused bond				x			

Modified UART idle	6(tx)
	14(rx)
		8 / 25	Mod. FSM, mod. baud rate					x		

Modified UART character	9(tx)
	2(rx)
		16 / 2	Mod. RS232 character, extra shift-register		x			x		

LED transmission	85	83	Blinkingx LED	x		(x)				
Power side channel			Measurement device	x						
Payload										
Mod. FSM	0	0	Extra state					x		
UART with mod. reset	10	10	Gated reset						x	
UART with mod. tx data	0	65						x		
Clock division mod.	2	1	Local clock							x
Mod. carry lookahead adder	0	2	Gated signal		x					
Mod. memory enable signal	1	6	Extra comparator/gates	x						
Mod. memory content	8	5	Mod. latch	x						
Mod. sync. divider	3	6								
Mod. *case*-divider	5	7	Extra state					x		
Mod. combinatorial dividerx	0	0	XOR instead of OR		x					
Detection										
Ring Oscillator						x				
Physically Unclonable Function						x				
Shadow Circuit			Exact copy of the circuit							

IV. Discussion and Conclusion

In Section I, we argued that hardware Trojans are a severe threat. However, comprehensive knowledge on this topic is missing. Detection methods rarely exist due to the hardware implementations’ complexity and the fact that hardware Trojans are typically targeted attacks. Furthermore, we discussed that hardware Trojan implementations are missing.

We proposed a Hardware Trojan Kit to create hardware based malware in a modular way. The modularization is based on the four key characteristics *activation*, *covert communication*, *payload* and *intrusive detection methods*. The analysis revealed typical malware structures which may serve as a strong indicator to reveal malicious hardware. Real-world hardware Trojans will cover more than one of the above-mentioned aspects; multiple indicators also reduce false positives. The effectiveness of this approach is currently under evaluation. Additionally, we see our work as a step towards enabling others to evaluate their detection and countermeasures against a variety of Trojans, thus accelerating the development of secure hardware.

References

[1] X. Wang, S. Narasimhan, A. R. Krishna, T. Mal-Sarkar, and S. Bhunia, “Sequential hardware trojan: Side-channel aware design and placement,” in *2011 IEEE 29th International Conference on Computer Design (ICCD)*, 2011, pp. 297–300.

[2] C. Krieg, A. Dabrowski, H. Hobel, K. Krombholz, and E. Weippl, “Hardware malware,” *Synthesis Lectures on Information Security, Privacy, and Trust*, vol. 4, no. 2, pp. 1–115, 2013. [Online]. Available: http://www.morganclaypool.com/doi/abs/10.2200/S00530ED1V01Y201308SPT006

[3] “Embedded Systems Challenge,” http://isis.poly.edu/esc/.