Symbolic Execution Based Test-patterns Generation Algorithm for Hardware Trojan Detection

Lixiang Shen , Dejun Mu , Guo Cao , Maoyuan Qin ,
Jeremy Blackstone , Ryan Kastner

PII: S0167-4048(18)30535-2
DOI: 10.1016/j.cose.2018.07.006
Reference: COSE 1368

To appear in: *Computers & Security*

Received date: 10 January 2018
Revised date: 1 June 2018
Accepted date: 9 July 2018

Please cite this article as: Lixiang Shen , Dejun Mu , Guo Cao , Maoyuan Qin , Jeremy Blackstone , Ryan Kastner , Symbolic Execution Based Test-patterns Generation Algorithm for Hardware Trojan Detection, *Computers & Security* (2018), doi: 10.1016/j.cose.2018.07.006

Symbolic Execution Based Test-patterns Generation Algorithm for Hardware Trojan Detection

Lixiang Shen[a,b,*], Dejun Mu[a], Guo Cao[c,d], Maoyuan Qin[a], Jeremy Blackstone[e], Ryan Kastner[e]

[a]School of Automation , Northwestern Polytechnical University, Xian 710072 , PR China

[b]School of Computer Information and Engineering, Changzhou Institute of Technology, Changzhou 213032, Jiangsu, PR China

[c]School of Management, Northwestern Polytechnical University, Xian 710072 , PR China

[d]School of Economics and Management, Changzhou Institute of Technology, Changzhou 213032, Jiangsu, PR China

[e]Department of Computer Science and Engineering at the University of California, San Diego,CA 92093-0404, United States

Abstract

Hardware Trojan detection is a very difficult challenge. However, the combination of symbolic execution and metamorphic testing is useful for detecting hardware Trojans in Verilog code. In this paper, symbolic execution and metamorphic testing were combined to detect internal conditionally triggered hardware Trojans in the register-transfer level design. First, control flow graphs of Verilog code were generated. Next, parallel symbolic execution and satisfiability modulo theories solver generated test patterns. Finally, metamorphic testing detected the hardware Trojans. The work used Trust-Hub benchmarks in experiments.

Key words

hardware Trojan; symbolic execution; Satisfiability Modulo Theory; metamorphic testing; control flow graph

1. Introduction

As modern embedded system design becomes more complex, malicious insiders have more opportunities to modify the hardware with hardware Trojans. Hardware Trojans are a type of malicious code that cause insertions, deletions and modifications to the original hardware design. They can threaten integrity, confidentiality and availability by altering the original function of the design, leaking sensitive information(Y Jin, 2010) and reducing the reliability of the hardware. Hardware Trojans can cause very serious security problems(Li et al., 2016) in many industries. Nissim(Nissim et al., 2017) described several USB hardware Trojans which installed backdoors, emulated a keyboard or mouse and exfiltrated data.

A hardware Trojan is usually composed of two parts: a trigger and payload. Triggers can activate payloads when a special condition is satisfied. The condition of a trigger is usually satisfied with very low-probability, so a payload can be activated with rare probability. When payload circuits are activated, malicious activities will occur. The aim of hardware Trojan detection is finding triggers and payloads. In different design objects, Trojans have different characteristics. A trigger may be classified as either an external trigger or an internal trigger. An internal-trigger uses an activation condition based on a particular input pattern, an internal logic state or counter value(Mohammad Tehranipoor, 2010). Time and data can be used as activation conditions of an internal-trigger. Hardware Trojans triggered with time are called time-bombs. Time-bombs cause serious threats to many high security areas because they are only affected by the internal system clock. System clocks don't need to be controlled by attackers who have to access to a hardware system. If a time-bomb is activated after a very long time, it will be very difficult to detect it because the testers may not have enough time to test all the code for time-bombs. Although formal validation techniques can verify all possible input values, it cannot prove that a time-bomb will never go off(Waksman and Sethumadhavan, 2011). Hardware Trojans triggered with data are called cheat codes. Cheat codes are the keys to identifying the payload of hardware Trojans. This work detects internal-trigger hardware Trojans.

Up to now, most literature focuses on post-fabrication detection which **analyses IC** chips, such as side channel technology. However, designers usually implement the functions at register-transfer level(RTL) code. Trojans inserted in the design at the register-transfer abstraction level or higher can be detected in RTL design. If Trojans are inserted in the gate-level netlist or later design stages, they can be detected by using equivalence checking tools for original RTL code(N Fern, 2016) .

The aim of this study was to generate efficient test patterns for detecting internally triggered hardware Trojans in RTL code. We proposed a test generation method for hardware Trojans in RTL code. First, the synthesizable Verilog code was analyzed to generate the control flow graphs(CFGs). Next, parallel symbolic execution was implemented and a satisfiability modulo