Third Party IP Trust Verification Using Information Flow Analysis

Dr. Jason Oberg
Co-founder and CEO
Tortuga Logic
GOMAC 2019

***Keywords*—Hardware Security; Third Party IP; Verification; Trust**

I. OVERVIEW

In today's system design landscape, many hardware designers are utilizing cores from third-party vendors in their designs. This practice has many benefits such as reduction in costs, shortened time-to-market, and allows for a larger range of design choices. Unfortunately, there has been an increase in security concerns due to integration of these third-party vendors' cores and their potential for either being the inadvertent cause of a security vulnerability, or exhibiting malicious behavior that is unknown to the hardware designer. The types of vulnerabilities can include trojans that either leak information, disable the system, or violate integrity properties.

Because of this, hardware designers are in need of viable solutions to help them analyze the security of the cores that they integrate into their systems. The current toolbox includes manual inspection of code, review of design documents, and functional verification tools, none of which are adequate in reliably finding undocumented behavior that may pose as a security threat in these third-party cores, malicious or not. Information Flow Analysis is an advanced technique that allows for these types of security violations to be identified alongside functional verification, and is specifically tailored to find security vulnerabilities.

In this poster, we discuss how third-party cores can introduce system-level security vulnerabilities into the design. Next, we will outline common security concerns that need to be taken into consideration when integrating third-party cores. We then discuss common hardware security verification techniques, as well as their benefits and drawbacks. Next, we will present the state-of-the-art techniques and methodologies, namely Information Flow Analysis, for analyzing third-party cores for security vulnerabilities, and how these techniques can be employed across the entire design lifecycle. Lastly, we will ground the discussion by discussing experimental results from using these techniques to find microarchitectural side-channel vulnerabilities in a RISC-V Rocket core processor, as well as find security vulnerabilities and unintended behavior in both open-source communication and encryption cores.