2023 International Conference on Field Programmable Technology (ICFPT) | 979-8-3503-5911-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICFPT59805.2023.00056

OpenTitan based Multi-Level Security in FPGA System-on-Chips

Sujan Kumar Saha, Abigail N. Butka, Muhammed Kawser Ahmed, and Christophe Bobda
Department of Electrical and Computer Engineering,
University of Florida, Gainesville, FL 32603
Email: {sujansaha, butkaa, muhammed.kawsera}@ufl.edu, and cbobda@ece.ufl.edu

***Abstract*—This article proposes a design methodology for integrating the OpenTitan-based security subsystem that enables Multi-level Security in FPGA SoCs. OpenTitan uses a light version o f RISC-V and a set of crypto-accelerators. We extend the capabilities of OpenTitan to host the Mandatory Access Control (MAC) based policies as well as enforce the policies in hardware IPs in the SoCs. We implemented a hardware policy server, integrated it into the OpenTitan subsystem using the provided toolchain, and integrated the subsystem into an FPGA SoC. The experimental analysis shows the use case scenario of this security extension and the corresponding low overheads.**

***Index Terms*—System-on-Chip, Security, Threats, Attack, Defense**

I. Introduction

Modern cyber-physical systems (CPS) depend electronic components the reliability of which is currently being impacted by increases in *cyber attacks*. To cope with unforeseen or malicious run-time events, critical CPS must be highly adaptive to fend off cyber attacks to achieve mission goals. Field Programmable Gate Arrays (FPGAs) provide the required technology to achieve performance through hardware/software co-design and adaptivity through run-time reconfiguration. To cope with the security requirements in CPS, FPGA-based System-on-Chips (SoCs) must provide the architectural layer access enforcement needed to defend against cyber attacks. Existing solutions such as TrustZone [1] provides extension for custom IPs in FPGA SoCs, but it still lacks in fine-grain access control. Saha et al. proposed the extension of Security-Enhanced Linux (SELinux) down to hardware in a distributed manner to enforce fine-grained access control [2]. But it also does not clearly describe the security of the access enforcement component, which may be compromised by malicious software. To combat this, a security subsystem based solution is inevitable to enable further isolation. In this work, We present a Root-of-Trust-based architecture that expands the functionality of a given SoC to handle security in FPGA-based SoC. The security infrastructure integrates the Multi-Level Security (MLS) paradigm with the OpenSource framework OpenTitan [3], an emerging and promising solution in the field of secure system design.

II. Proposed Method

A. Threat Model

This work aims to address potential attack scenarios on hardware IPs based on access control in systems where untrusted software may exist. We focus on two primary attack scenarios that are deemed most likely to occur. **(1)** malicious software can directly access an accelerator IP and illicitly extract sensitive data by accessing the driver module. **(2)** Attackers may exploit their knowledge of the address range and register offsets of a hardware IP to either read/write secret information from the hardware IP while another application is using it.

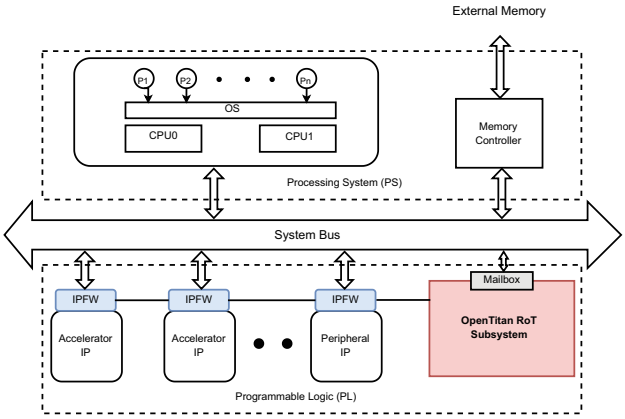

Fig. 1. Secure SoC with OpenTitan subsystem and integrated Multi-level Security

B. OpenTitan subsystem based SoC

Our goal is to create a security subsystem based SoC to isolate the security components from the rest of the System. The motivation for this isolation is to provide a separate execution of security-related firmware outside of the main memory. We target the OpenTitan Root-of-Trust SoC as security subsystem, because **(1)** OpenTitan uses a open-source light-weight RISC-V which can be highly customised, and **(2)** a good range of crypto and security primitive IPs are available in IP repo. We integrate the OpenTitan-based subsystem into the SoC by using mailbox communication as shown in Figure 1.

C. MLS Integration into SoC

One of the major goals of this work is to integrate Multi-Level Security into the SoC. But the OpenTitan processing system does not inherently support this feature. To enable the MLS at the hardware level, we need two major hardware com-

979-8-3503-5911-4/23/$31.00 ©2023 IEEE
DOI 10.1109/ICFPT59805.2023.00056

TABLE I
FPGA RESOURCE UTILIZATION OF SECURITY COMPONENTS

	Resource	Utilization	Utilization %
RoT Subsystem	LUT	13794	25.92
	FF	20017	18.81
IPFW	LUT	149	0.28
	FF	334	0.31
Policy Server	LUT	56	0.1
	FF	170	0.15

ponents (a hardware policy server and an access enforcement wrapper named IP Firewall) and a secure firmware.

1) Policy Server: The Policy Server is a memory-based component that can host access control security policies. It is integrated into the OpenTitan subsystem. The Policy Server has three major components: the memory, policy loading, and policy fetching unit. In FPGA SoCs, the block RAM can be used as the memory component. The policy loading unit is responsible for initial policy loading and run-time policy updates. The policy fetching unit extracts the security policies from the memory unit if there is a policy read request by the IPFW.

2) Secure Firmware: Each security policy is a 32-bit binary values (16-bit process ID+14-bit IP Identifier+ 2-bit R/W permission). These binary values are loaded in to the policy server by using a secure firmware. The firmware is written in C language. These firmware C codes run the RISC-V processor of the OpenTitan subsystem. Another part of this secure firmware is to create read/write driver APIs for the security IPs in the subsystem by abstracting the IP specific I/O register base addresses and offsets. These APIs are used by the software developer to access the corresponding security IPs.

3) IP Firewall: The IP Firewall (IPFW) is a security wrapper that enforces access control over the IPs. It has two major parts: An access vector cache (AVC) and policy lookup module (PLM). The AVC is a cache component that keeps the processes' most recently used security context (IP Identifier and permission). The PLM checks the access permission of a request to access the IP. When an access request comes to the IPFW of an IP, first, the PLM checks the IP core ID of that request, then checks the IP identifier and permission. If the access permission is granted, the data is forwarded to the IP for acceleration. Otherwise, the PLM sends a policy fetching request to the policy server to get the corresponding policy.

III. EXPERIMENTATION

A prototype secure SoC is developed using Xilinx Zynq Z-7020 FPGA SoC and Vivado tool. The processing side has ARM cortex CPU and PL side has OpenTitan subsystem, a RSA crypto module, and MLS components. We run the bare metal applications with 100MHz frequency.

A. Case Studies

Two case studies were performed against the two attack scenarios mentioned in the threat model. One firmware application was developed to load the initial binary access control policies into the policy server. Then, for attack scenario **(1)**, one software application is run on the Zynq processor that tries to access the RSA module with the corresponding correct context ID and permissions. In this case, the IPFW allowed the application to send input data to the RSA and receive the correct cipher value. Then, the context ID has been modified to mimic the application as a malicious software. The IPFW successfully denied the access permission. For attack scenario **(2)**, two applications are run on two ARM cores that try to access the RSA IP simultaneously. However, as the IPFW is in a busy state while one application is reading/writing, preventing the second one, potentially malicious application from sending any read or write requests to the RSA.

TABLE II
EXECUTION DELAY OF IPFW AND POLICY SERVER

Component	Operation	Cycle Count
IPFW	AVC Hit	3
	AVC Miss	7
Policy Server	Policy Firmware Load	1 cycle per policy
	Policy Fetching	3

B. Overhead Analysis

We take the resource utilization results after generating the bitstream in Vivado. As reported in Table I, we observe that the overall LUT usage by the RoT subsystem is 25.92% for all the components in the PL which consists of the RISC-V processor, Mailbox, policy server, IPFW, and the RSA crypto module. Here, we observe that RoT Subsystem takes most of the LUT and FF usage, whereas the MLS enforcement components (IPFW and Policy Server) are consuming fewer area resources. Table II shows the execution delay in cycle counts. Here, we observe 3 cycles are required for IPFW hit case and the policy server fetching operation. In the worst case scenario, up to 7 cycles are required. Hence, the proposed approach adds limited area and delay overhead.

IV. CONCLUSION

The goal of this paper is to develop a opensource subsystem based secure SoC in FPGA. We leverage the OpenTitan Root-of-Trust SoC components to create the proposed security subsystem. We also integrate the MLS into the security subsystem to achieve access enforcement at hardware level. A prototype secure SoC is designed to perform the case studies which shows the correct implementation of the target security goals. The area and execution cycle overhead is also minimal.

ACKNOWLEDGMENT

This work was supported by Defense Advanced Research Projects Agency (DARPA) through the Automated Integration of Secure Silicon (AISS) project.

REFERENCES

[1] B. Ngabonziza, D. Martin, A. Bailey, H. Cho, and S. Martin, "Trustzone explained: Architectural features and use cases," in *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2016, pp. 445–451.

[2] S. K. Saha and C. Bobda, "Fpga accelerated embedded system security through hardware isolation," in *2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2020, pp. 1–6.

[3] OpenTitan, "Opentitan root-of-trust silicon chip," https://opentitan.org/.