Boot time Bitstream Authentication for FPGAs

Ali Shuja Siddiqui, Yutian Gui and Fareena Saqib
University of North Carolina at Charlotte
asiddiq6@uncc.edu, ygui@uncc.edu, and fsaqib@uncc.edu

1 Hardware Demo Objectives

Major commercial Field Programmable Gate Arrays (FPGAs) vendors provide encryption and authentication for programmable logic fabric (PL) bitstream using AES and RSA respectively. They are limited in scope of security that they provide and have proven to be vulnerable to different attacks. As-such, in-field deployed devices are susceptible to attacks where either a configuration bitstream, application software or dynamically reconfigurable bitstreams can be maliciously replaced. This hardware demo presents a framework for secure boot and runtime authentication for FPGAs. The presented system employs on-board cryptographic mechanisms and third-party established architectures such as Trusted Platform Module (TPM). The scope of this hardware demo is of systems level.

2 Introduction

The use of Field Programmable Gate Arrays (FPGAs) is widespread in the commercial hardware space e.g. in Internet-of-Things (IoT) [1], on-board automotive electronics [2], wireless sensor networks and power distribution networks. Depending on the application, once a device has been deployed in the field, it may be accessible by unfavorable actors, either physically or through networked means. Such a deployed can fall a prey to firmware modification or bitstream modification attacks, where an attacker maliciously modifies the firmware running or even the bitstream running on the system.

In this work, we present a framework for securing hardware bitstream at boot and for securing dynamic partial reconfiguration bitstreams at runtime. The presented framework uses Trusted Platform Module (TPM) as a trust anchor. Once the system has been booted up and the bitstream has been loaded, the system uses on-board crypto functions to implement runtime system attestation. This work is novel compared to existing architectures since it includes integration of TPM in the first stage boot loader, that no current FPGA vendor is providing, additionally it is providing security for partial reconfiguration blocks at runtime.

3 Attack Model

In the software domain, Secure Boot can be used to ensure that no maliciously modified firmware or software may execute on the system. However, there has been limited work done for the security of the bitstream. Major FPGA vendors, e.g. Xilinx and Altera, have provided on-board encryption and authentication support using AES and RSA. However, for both these devices, the key storage provided is either using one-time programmable efuses or battery backed RAM (BBRAM). Keys stored in the NVM like efuses can be extracted [3]. Whereas, BBRAM requires a guarantee for power for the entirety of its operation. Furthermore, attacks against Xilinx provided secure boot has been documented where attackers were able to modify the second stage boot loader so that it could load an unauthorized binary [4]. An alternative to using vendor supplied security is using self-authentication using physical unclonable functions (PUFs) [5]. Key generated by the PUF is used to encrypt the application software and the boot loaders. A limitation of this approach is that it does not authenticate the content provider. Therefore, an attacker with the correct encryption key can update the node with malicious code.

Partial reconfigurable blocks on the FPGA are areas of logic that can be reconfigured on the fly. They allow for hardware functions to be replaced while the system is operational. A partial reconfigurable block is susceptible to be reprogrammed by an attacker through software exploits or using hardware trojans.

4 Experimental Results

Block diagram of the secure node architecture is given in Figure 1. The node fabric is divided into Processing System (PS) and the Programmable Logic (PL). The PS is equipped with a Trusted Execution Environment (TEE). To provide a trust anchor for a node, we are using a Trusted Platform Module (TPM). TPM provides cryptographic functions such as key generation, encryption, digital signing and data sealing. TPM also provides tamper resistant non-volatile memory. This memory is used for storing keys and secure object. This work adds security extensions to the First Stage Bool Loader provided by Xilinx We consider bitstream provider as a server that pushes bitstream and software updates to a node in the field. This work incorporates digital certificates to establish the identity of the bitstream provider. Once the identity has been established, the integrity of the bitstream is verified with the hash provided by the bitstream provider. If both operations complete successfully, the bitstream is loaded on the fabric. This process is repeated for the second stage boot loader and the application software.

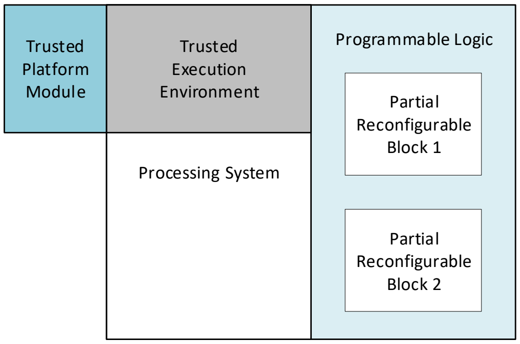
Trusted Platform Module
Trusted Execution Environment
Processing System
Programmable Logic
Partial Reconfigurable Block 1
Partial Reconfigurable Block 2

Figure 1: Secure Boot and Runtime FPGA Architecture.

5 Key Observations and Outcomes

With the help of the experimental setup, we can demonstrate a successful secure boot operation, consisting of secure loading of the bitstream, system software and the application software. Once the system is in normal operational state, secure loading process of partial bitstream can also be demonstrated.

To demonstrate the attacks on this framework, an unverifiable bitstream or a software image is loaded onto to the system. In such a case, the system alerts the user of the discrepancy.

6 List of Equipment

To present the framework, we are using Xilinx Zedboard connected to Infineon TPM SLB9670[5]. To program the board, we are using vendor provided software as well as Xilinx Vivado v2017.4 and Xilinx SDK 2017.4. Figure 2 shows the experimental setup.

7 References

[1] Intel. IoT (Internet of Things) SoC and FPGA Solutions - Intel® FPGA. [online] Available at: https://www.intel.com/content/www/us/en/internet-of-things/products/programmable/overview.html [Accessed 14 Jan. 2019].

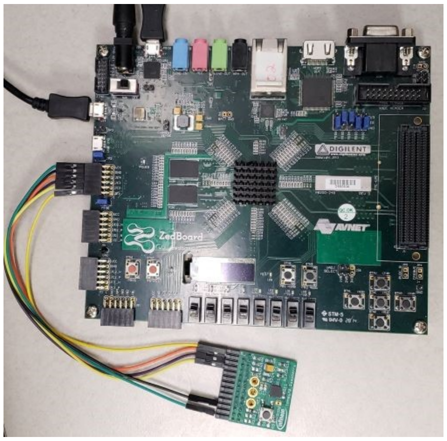

Figure 2: The experimental setup.

[2] Xilinx.com. (n.d.). Automotive Driver Assist. [online] Available at: https://www.xilinx.com/applications/megatrends/automotive-driver-assist.html [Accessed 14 Jan. 2019].
In this section, include references to your attack(s) and countermeasure(s).
[3]"Chip Design » Why Anti-Fuse is The Only Secure Choice for Encryption Key Storage by David Hsu, Kilopass Technology Inc.", Chipdesignmag.com, 2011. [Online]. Available: http://chipdesignmag.com/display.php?articleId=5045. [Accessed: 15- Jan- 2019].
[4] N. Jacob, J. Heyszl, A. Zankl, C. Rolfes and G. Sigl, "How to Break Secure Boot on FPGA SoCs Through Malicious Hardware", Lecture Notes in Computer Science, pp. 425-442, 2017. Available: 10.1007/978-3-319-66787-4_21 [Accessed 15 January 2019].
[5] "OPTIGA™ TPM SLB 9670VQ2.0 - Infineon Technologies", Infineon.com. [Online]. Available: https://www.infineon.com/cms/en/product/security-smart-card-solutions/optiga-embedded-security-solutions/optiga-tpm/slb-9670vq2.0/. [Accessed: 15- Jan- 2019].