Detection of Malicious FPGA Bitstreams using CNN-Based Learning*

Jayeeta Chaudhuri and Krishnendu Chakrabarty
Department of Electrical and Computer Engineering, Duke University, Durham, NC

2022 IEEE European Test Symposium (ETS) | 978-1-6654-6706-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ETS54262.2022.9810438

Abstract—**Multi-tenant FPGAs are increasingly being used in cloud computing technologies. Users are able to access the FPGA fabric remotely to implement custom accelerators in the cloud. However, sharing FPGA resources by untrusted third-parties can lead to serious security threats. Attackers can configure a portion of the FPGA with a malicious bitstream. Such malicious use of the FPGA fabric may lead to severe voltage fluctuations and eventually crash the FPGA. Attackers can also use side-channel and fault attacks to extract secret information (e.g., secret key of an AES encryption module). We propose a convolutional neural network (CNN)-based defense mechanism to detect malicious circuits that are configured on an FPGA by learning features from the data-series representation of the bitstreams of malicious circuits. We use the classification accuracy, true-positive rate, and false-positive rate metrics to quantify the effectiveness of CNN-based classification of malicious bitstreams. Experimental results on Xilinx FPGAs demonstrate the effectiveness of the proposed method.**

I. Introduction

Field-programmable gate-arrays (FPGAs) are now ubiquitous in cloud computing infrastructures and reconfigurable system-on-chips (SoCs). The availability of FPGA in cloud data centers has opened up new opportunities for users to improve application performance by enabling them to implement customizable hardware accelerators directly on the FPGA fabric. FPGAs are therefore being incorporated today for specialized compute-intensive services in cloud data centers, e.g., by Amazon and Microsoft [1] [2].

As FPGAs are increasingly shared and remotely accessed by multiple users and third parties, they are a major reason for rising security concerns. Modules running on an FPGA may include circuits that induce voltage-based fault attacks and denial-of-service [3] [4].

In order to prevent the attacker from directly configuring the FPGA with an invalid or malicious bitstream, the defender (e.g., the FPGA vendor) can incorporate an on-chip bitstream checking mechanism that detects and blocks such bitstreams before loading them to the FPGA fabric [5]. An approach to analyze FPGA bitstreams using neural networks is presented in [6]. Machine learning-based approaches for Hardware Trojan detection have been proposed in [7] and [8].

The key contributions of this paper are as follows:

- Generation of RO variants and non-combinational oscillators;
- Visualization of FPGA bitstreams as data series and the extraction of signature patterns that differentiate benign bitstreams from malicious bitstreams;
- A CNN-based classification framework that detects malicious bitstreams based on features extracted from RO patterns;

The remainder of the paper is organized as follows. Section II discusses the threat model. Section III describes the generation of our CNN-based feature extraction framework for malicious bitstream classification. Experimental results and observations are presented in Section IV.

II. Threat Model

ROs generate oscillations with a frequency that depends on the gate delays of the inverters. ROs have been shown to maliciously affect the power consumption of the FPGA [9]. This can result in DoS and ultimately cause the FPGA to crash within seconds. Alternatively, the voltage fluctuations can also be used to leak encryption keys from an AES module [3].

With advances in technology, FPGAs are increasingly being shared by multiple users, either at the same time or at different times. In such a scenario, both the attacker and victim can configure the FPGA with their own modules. We assume that an attacker has access to the target FPGA and is capable of configuring the FPGA with a malicious bitstream.

III. CNN-based Feature Extraction Framework

A. Dataset Generation

We generate a large dataset consisting of benign and malicious bitstreams that are used to configure an FPGA. We generate 95 benign bitstreams and 80 malicious bitstreams. The bitstreams used for our experiments are as follows:

Benign Bitstreams: We generated bitstreams that implement USBs, keyboard controllers, AES cores, MIPS cores, IPs that support trigonometric, quadratic, and other arithmetic operations, and VGA OpenCores. We also generated bitstreams that implement ISCAS '85, ITC '99, and EPFL benchmarks. We ensured that these bitstreams are representative of data used in real-life benign applications.

Malicious Bitstreams: We define a malicious bitstream as a bitstream implementing a circuit that is capable of overheating the FPGA or launching power analysis and voltage-based fluctuation attacks on the victim FPGA fabric. We focus on simple ROs as well as variants of ROs. Most CAD tools can detect a combinational loop and raise an error during DRC. However a non-combinational oscillator escapes DRC and also supports successful bitstream generation. Therefore, we take into consideration such loop-free oscillators and treat them as malicious circuits in this paper.

*This work was supported by the National Science Foundation under grant no. CNS-2011561.

978-1-6654-6706-3/22/$31.00 ©2022 IEEE

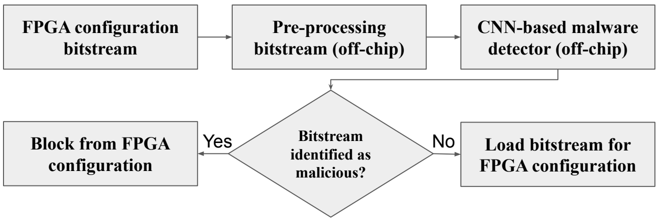

FPGA configuration bitstream
Pre-processing bitstream (off-chip)
CNN-based malware detector (off-chip)
Bitstream identified as malicious?
Yes
No
Block from FPGA configuration
Load bitstream for FPGA configuration

Fig. 1: CNN pipeline for detecting malicious bitstreams.

B. Mapping Bitstreams to Data-Series Representation

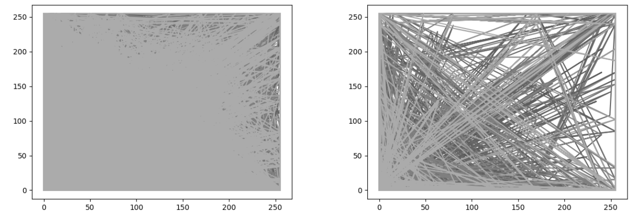
Next, we convert benign and malicious bitstreams to their corresponding data series. This data is then represented as image files for each bitstream. Plotting bitstreams as data series and storing them as image files enables us to 1) identify specific patterns in the image files that represent malicious behaviour, and 2) utilize a CNN-based image classification framework to detect these malicious patterns in images. Note that the procedure of generating a bitstream takes $\sim$ 15 minutes and plotting the bitstream as 2D data series takes upto 10 minutes. Therefore, generating a large dataset of benign and malicious bitstreams and then converting them to image files is time-consuming. Image augmentation, a well-known type of data augmentation technique, is used in such scenarios. Image augmentation increases the size of the training dataset by creating transformed versions of training set images.

C. Malicious Bitstream Detection

We adopt a CNN-based approach to detect malicious FPGA bitstreams. The end-user inputs a bitstream to the pre-trained CNN model for authentication before loading it to the FPGA. If the CNN classifies the bitstream as malicious, it is blocked from FPGA configuration. Fig. 1 illustrates the proposed CNN pipeline. The neural network used in our classification framework has four convolutional layers, four max pooling layers, and four linear layers. We choose this architecture for the following reasons:

1) The image received at the input layer may have noise included in it. Therefore, we increase the number of filters by adding more convolutional layers and extract useful features as the network gets deeper.
2) We use gray-scale images in our training dataset for our CNN. However, extracting meaningful features from a gray-scale image is much more complex than extracting features from a colored image. In order to improve the performance of the model in such a scenario, it is desirable to have more linear layers using the non-linear activation function. We choose the Rectified Linear Unit (ReLU) as our activation function because it trains the CNN model faster and more effectively, without causing a significant drop in classification accuracy [10].

The image files corresponding to a benign bitstream and a malicious bitstream are shown in Fig. 2(a) and Fig. 2(b), respectively. We draw the following qualitative observations from the patterns in the benign and malicious image files. For malicious bitstreams, the intensity of patterns across the image is not uniform; it follows a non-uniform distribution. For benign bitstreams, we observe a higher intensity of a particular pattern in the image. However, the same region appears with less intensity in the case of a malicious bitstream. These observations guide us to use these specific attributes as features to train our proposed neural network-based malicious bitstream detector.



(a) (b)

Fig. 2: Image files corresponding to: (a) benign bitstream; (b) malicious bitstream.

IV. Experimental Results

A. Evaluation of the proposed CNN-based classification framework

Recall that our dataset comprises of 95 image files generated from benign bitstreams and 80 image files generated from malicious bitstreams. The bitstreams are generated using the *write_bitstream* command available in Xilinx Vivado 2018.2. These bitstreams target the Xilinx Virtex Ultrascale FPGA. We choose the $rotation$ and $flip$ as our image augmentation method; this method shows the highest classification accuracy compared to the other commonly used image augmentation techniques, such as the $equalize$ and the $translate$ operations [11]. After image augmentation, the size of our dataset increases to 314 image files. We use k-fold cross validation to evaluate multiple versions of train-test split. In our experiments, we select $k = 5$ in accordance with common practice [12]. We obtain an average training accuracy of 99.2% and an average test accuracy of 96.4% after 300 epochs, over the five folds. Also, we obtain TPR_{mal} = 97.08% and FPR_{mal} = 4.29%.

References

[1] Amazon, "Amazon EC2 F1 Instance," https://go.aws/3ENtUj9, 2021.

[2] K. Eguro and R. Venkatesan, "FPGAs for trusted cloud computing," in *Int. Conf. on Field-Programmable Logic and Applications*, 2012.

[3] M. Zhao and G. Suh, "FPGA-based remote power side-channel attacks," *Proc. IEEE S&P*, 2018.

[4] F. Schellenberg *et al.*, "An inside job: Remote power analysis attacks on FPGAs," in *DATE*, 2018.

[5] T. M. La *et al.*, "FPGADefender: Malicious self-oscillator scanning for Xilinx UltraScale + FPGAs," *ACM Transactions on Reconfigurable Technology and Systems*, 2020.

[6] S. Mahmood *et al.*, "IP core identification in FPGA configuration files using machine learning techniques," in *Proc. IEEE ICCE-Berlin*, 2019.

[7] N. Vashistha *et al.*, "Trojan scanner: Detecting hardware trojans with rapid sem imaging combined with image processing and machine learning," in *ISTFA*, 2018.

[8] K. Hasegawa *et al.*, "Hardware Trojans classification for gate-level netlists based on machine learning," in *IOLTS*, 2016.

[9] J. Krautter *et al.*, "FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018.

[10] X. Glorot *et al.*, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, 2013.

[11] B. Zoph *et al.*, "Learning data augmentation strategies for object detection," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1906.11172.

[12] SKlearn, "Grid search with cross validation," https://bit.ly/3hEHNnQ , [Online; accessed 13-August-2020].