IEEE TRANSACTIONS ON
INFORMATION FORENSICS
AND SECURITY
TIFS Volume 17 | 2022

Stealing Neural Network Structure through Remote FPGA Side-channel Analysis

Yicheng Zhang, Rozhin Yasaei, Hao Chen, Zhou Li,
Mohammad Abdullah Al Faruque

University of California, Irvine

INTERNATIONAL SYMPOSIUM ON FIELD-PROGRAMMABLE GATE ARRAYS
FPGA 2021 CFP Program Tutorials Guidelines Artifacts Diversity Committees Archive

UCI Samueli School of Engineering
University of California, Irvine

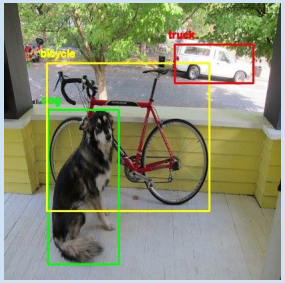
Deep-learning models: Highly Valuable IP

- Deep-learning models are everywhere.
- Large market size.

Computer Vision ~$2.37 Billion

Facial Recognition

Object Segmentation



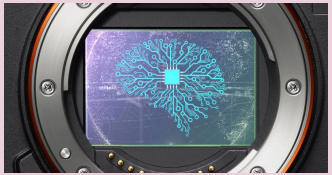
Speech Recognition ~$21.5 Billion

Voice Assistants

Auomatic Machine Translation

Embedded Devices ~$6.6 Billion

Consumer Electronics



Self-driving Cars

Data Centers ~$20 Billion

Video Recommendation

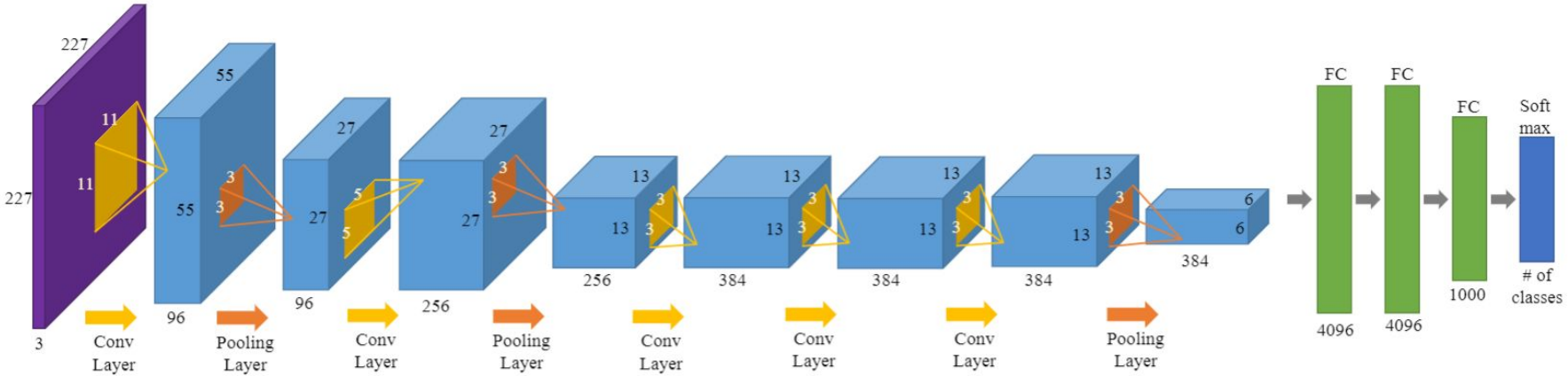
YouTube

Advertisement Prediction

Google

Deep-learning models: Highly Valuable IP

- Designing a deep-learning model with good performance requires great time and effort.
  - Deep-learning model structure has a fundamental impact on its performance.
  - Some deep-learning models have complex structures.
  - Large seach space.
    - The search space for VGG16 is 5.4×10^12 [1].

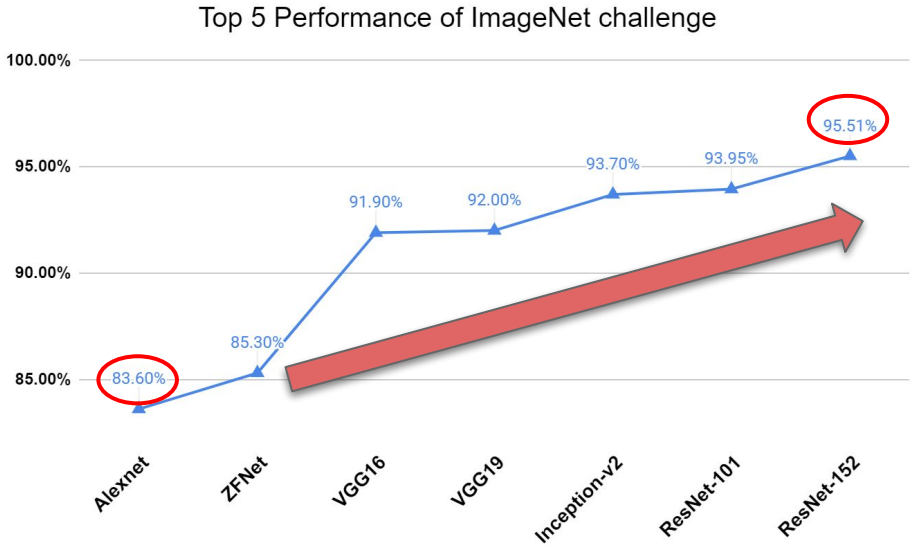


AlexNet

[1]. Mengjia Yan, Christopher W Fletcher, and Josep Torrellas. Cache telepathy: Leveaging shared resource attacks to learn{DNN}architectures. In29th{USENIX}Security Symposium ({USENIX}Security 20), pages 2003–2020, 2020

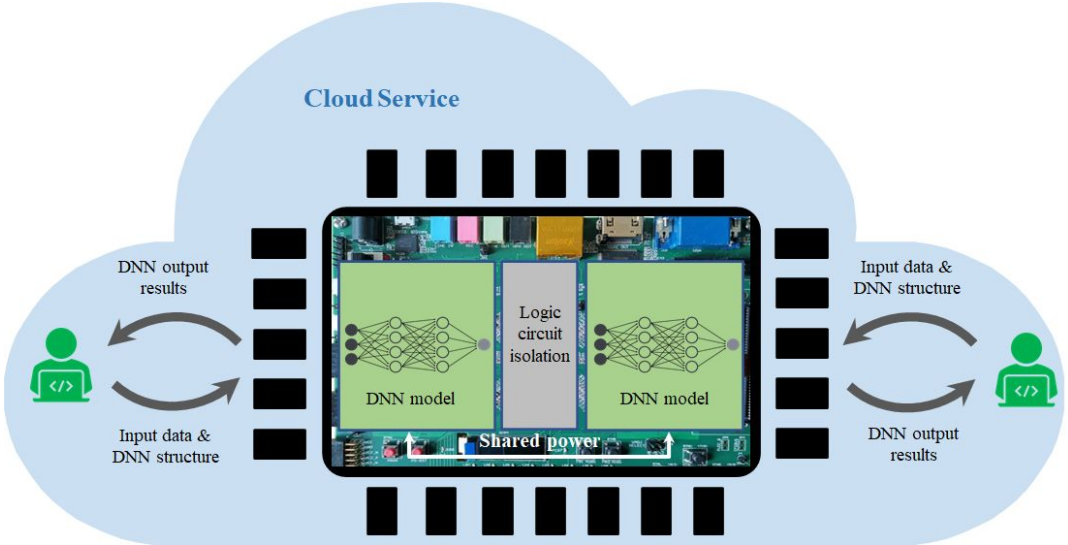
Deep-learning models: Highly Valuable IP

- Designing a deep-learning model with good performance requires great time and effort.
  - Deep-learning model structure has a fundamental impact on its performance.



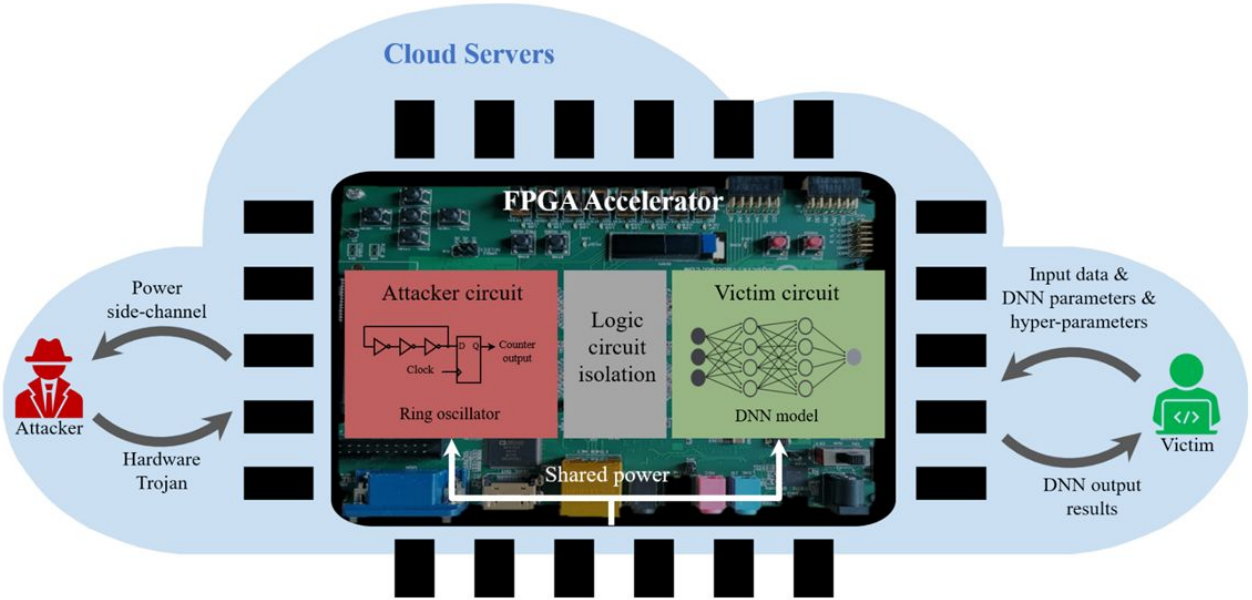
Deep-learning model training on the cloud

- FPGA has become the dominant hardware to train and run deep-learning models.
- FPGA multi-tenancy.
  - Multiple users may share the same physical FPGA [2-11].
  - Share with same power supply unit or power distribution network.
- Isolation.
  - User does not have direct access to other users.



Deep-learning model training on the cloud

- Spy and victim share the same FPGA.
- Power side-channel leakage.
- Can an adversary infer deep-learning models by exploiting power side-channel?



Prior work

- Stealing deep-learning model secrets on the cloud through side-channel.
  - Most of them are CPU-based cache side channel.
- The research closest to ours was done by Hua et al. [12], and Yuet al[13].
  - Complete control of and physical access to the FPGA board.

2018 IEEE Symposium on Security and Privacy

FPGA-Based Remote Power Side-Channel Attacks

Mark Zhao and G. Edward Suh
Computer Systems Laboratory
Cornell University
Ithaca, New York
yz424@cornell.edu, suh@ece.cornell.edu

***Abstract*—The rapid adoption of heterogeneous computing has driven the integration of Field Programmable Gate Arrays (FPGAs) into cloud datacenters and flexible System-on-Chips (SoCs). This paper shows that the integrated FPGA introduces a new security vulnerability by enabling software-based power side-channel attacks without physical proximity to a target system. We first demonstrate that an on-chip power monitor can be built on a modern FPGA using ring oscillators (ROs), and characterize its ability to observe the power consumption of other modules on the FPGA or the SoC. Then, we show that the RO-based FPGA power monitor can be used for a successful power analysis attack on an RSA cryptomodule on the same FPGA. Additionally, we show that the FPGA-based power monitor can observe the power consumption of a CPU on the same SoC, and demonstrate that the FPGA-to-CPU power side-channel attack can break timing-channel protection for an RSA program running on a CPU. This work introduces and demonstrates remote power side-channel attacks using an FPGA, showing that the common assumption that power side-channel attacks require specialized equipment and physical access to the victim hardware is not true for systems with an integrated FPGA.**

I. Introduction

As we increasingly rely on hardware acceleration to improve the performance and energy efficiency of computing systems, Field Programmable Gate Arrays (FPGAs) have recently been widely adopted in large-scale datacenters. For example, Amazon offers FPGA instances in its EC2 service, allowing customers to rent FPGAs in its cloud computing environment [1]. Microsoft heavily utilizes FPGAs in its datacenters for various tasks ranging from web searches to network crypto and machine learning [2]. Similarly, Baidu also accelerates deep neural networks in its datacenters with FPGAs [3]. Furthermore, hardware vendors such as Intel and Xilinx have introduced heterogeneous System-on-Chip (SoC) designs, which integrate both processing cores and FPGA fabric in one silicon die. These FPGA-based SoCs will likely to be utilized in mobile and embedded applications.

In this paper, we show that these integrated FPGAs introduce a new security vulnerability that can be exploited to perform power side-channel attacks in software, without requiring physical access or proximity to the target system. Power side-channel attacks infer confidential information based on the data-dependent variations in a target system's power consumption [4]. In order to obtain power consumption traces, traditional power analysis attacks require physical access to the system; attackers insert a low-impedance resistor in series with the power supply and use an oscilloscope to measure the power consumption as the voltage drop across the resistor. In this paper, we demonstrate that an on-chip power monitor can be constructed using the programmable logic of an FPGA, allowing us to measure dynamic power consumption with sufficient resolution to enable power analysis attacks. In essence, the integrated FPGA opens the door for remote power analysis attacks.

This FPGA-based power side channel may be exploited in a variety of system architectures that allows an untrusted user to program a part of an FPGA. In cloud computing infrastructures, many studies from both academia and industry have proposed mechanisms to virtualize and share FPGAs among multiple users so that multiple accelerators co-reside on one physical FPGA [5]–[10]. Even in cloud platforms where each FPGA is allocated to a single user, untrusted user logic is co-resident with privileged control logic called the 'shell' [11]. Similarly, in personal computing platforms, an FPGA fabric can be shared among multiple programs, including a potentially malicious application. In such a shared FPGA platform, we show that an FPGA-to-FPGA attack, where an attack circuit in one part of the FPGA steals a secret used by another circuit on the same FPGA, is viable.

As a concrete example, we demonstrate a simple power analysis (SPA) attack on an RSA accelerator on an FPGA. In this example, we implement an RSA decryption engine and a power monitor on one FPGA, but isolate both logically and physically; there is no connection between the two modules and they are implemented at physically different locations on the FPGA. This is analogous to either two users or one user and privileged control logic co-resident on one FPGA. Our experiments show that the FPGA-based SPA can reliably recover RSA private keys only with a small number of power traces and without manual efforts; the power analysis can be fully automated.

In addition to the attacks within an FPGA, we also study attacks on an FPGA-CPU SoC. Surprisingly, we found that an FPGA power monitor can not only detect the power consumption of an FPGA but also the power consumption of other components on an SoC, in particular a CPU. This implies that various FPGA-to-CPU attacks are possible where the malicious logic on an FPGA can monitor the power consumption of software programs running on a CPU core. Our experiments show that the FPGA power monitor can indeed detect software operations on an ARM processing core. The CPU power trace can also be used to enable timing

DOI 10.1109/SP.2018.00049

IEEE computer society

DeepEM: Deep Neural Networks Model Recovery through EM Side-Channel Information Leakage

Honggang Yu*, Haocheng Ma†, Kaichen Yang*, Yiqiang Zhao† and Yier Jin*
*Department of Electrical and Computer Engineering, University of Florida
†School of Microelectronics, Tianjin University
honggang.yu@ufl.edu, hc_ma@tju.edu.cn, bojanykc@ufl.edu, yq_zhao@tju.edu.cn, yier.jin@ece.ufl.edu

***Abstract*—Neural Network (NN) accelerators are currently widely deployed in various security-crucial scenarios, including image recognition, natural language processing and autonomous vehicles. Due to economic and privacy concerns, the hardware implementations of structures and designs inside NN accelerators are usually inaccessible to the public. However, these accelerators still tend to leak crucial information through Electromagnetic (EM) side channels in addition to timing and power information. In this paper, we propose an effective and efficient model stealing attack against current popular large-scale NN accelerators deployed on hardware platforms through side-channel information. Specifically, the proposed attack approach contains two stages: 1) Inferring the underlying network architecture through EM side-channel information; 2) Estimating the parameters, especially the weights, through a margin-based, adversarial active learning method. The experimental results show that the proposed attack approach can accurately recover the large-scale NN through EM side-channel information leakages. Overall, our attack highlights the importance of masking EM traces for large-scale NN accelerators in real-world applications.**

I. Introduction

Neural Networks (NNs) have recently shown tremendous progress in various real-world applications, ranging across object recognition [1]–[3], natural language processing [4] and autonomous vehicles [5], [6]. Additionally, there has been an increasing effort to deploy large-scale NN models on dedicated hardware platforms such as GPU, FPGAs, or customized ASICs in order to improve the performance and efficiency of data processing systems. Hardware vendors including Xilinx and Intel spend great efforts collecting a data set, training these NNs models on it, and developing the NNs accelerators, and thus want to keep the trained models private and secret.

However, recent studies have demonstrated that severe vulnerabilities exist in hardware implementations of these NN accelerators. An adversary, who has no knowledge of the details of structures and designs inside these accelerators (i.e., black-box), can effectively reverse engineer the neural networks by also present that NNs are extremely susceptible to timing side-channel attacks. In their attack scheme, adversaries recover the layer's depth by applying timing side-channel information and exploit a reinforcement learning technique to search for the best substitute model with functionality similar to the victim networks. It is important to note that IP vendors do not always allow users to access these architectural side-channel information, such as memory and cache due to security and privacy concerns. Therefore, these attacks can not be conducted while targeting NNs protected in this way. To solve this problem, Batina *et al.* [10] propose a new model theft attack that exploits EM side-channel analysis to effectively reverse engineer the network characteristics of small-scale multilayer perception (MLP) and convolutional neural networks (CNNs). Specifically, the authors perform correlation electromagnetic analysis (CEMA) using the Hamming weight model to recover the networks weights. However, uniform weight setting makes current leakage models, i.e., Hamming weight and Hamming distance, slightly deviate actual EM leakages [11]. Considering the enormous parameters (e.g., weights) those large-scale neural network accelerators maintain, this deviation will significantly degrade the effectiveness of EM based model theft attacks.

To address these challenges, we present a new black-box attack that exploits EM side-channel information to effectively reverse engineer Binarized Neural Networks (BNNs), which are commonly used NNs for IoT/edge devices that apply binary values for activations and weights. Different from the previous attacks, in this study we assume the adversary has no access to the exact training data, network architecture, parameters, etc, but can only collect EM side-channel information under inference operations and observe the networks outputs (e.g, labels or confidence scores).

The key idea of our attack method is that we exploit EM side-channel information to reconstruct the network archi-

Reverse Engineering Convolutional Neural Networks Through Side-channel Information Leaks

Weizhe Hua, Zhiru Zhang, and G. Edward Suh
School of Electrical and Computer Engineering, Cornell University, Ithaca, NY
{wh399, zhiruz, gs272}@cornell.edu

ABSTRACT

A convolutional neural network (CNN) model represents a crucial piece of intellectual property in many applications. Revealing its structure or weights would leak confidential information. In this paper we present novel reverse-engineering attacks on CNNs running on a hardware accelerator, where an adversary can feed inputs to the accelerator and observe the resulting off-chip memory accesses. Our study shows that even with data encryption, the adversary can infer the underlying network structure by exploiting the memory and timing side-channels. We further identify the information leakage on the values of weights when a CNN accelerator performs dynamic zero pruning for off-chip memory accesses. Overall, this work reveals the importance of hiding off-chip memory access pattern to truly protect confidential CNN models.

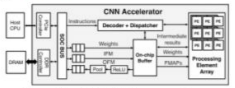


Figure 1: A typical CNN inference accelerator.

1 INTRODUCTION

Convolutional neural networks (CNNs) are quickly becoming an essential tool in a wide range of machine learning applications. In many application scenarios, CNN models — both its network structure and learnable parameters (i.e., weights) — need to be protected as confidential information: (1) for companies that rely on a CNN to provide a core or value-added service, the underlying neural network model represents an important piece of intellectual property; (2) in personalized applications such as digital assistants, CNN models are trained using private data, and the weights need to be kept confidential for privacy [13]; (3) furthermore, recent studies on the adversarial network show that an attacker can intentionally affect the outcome of CNN-based classification and object detection by perturbing input images when the network model is known [5].

This paper investigates reverse-engineering attacks on CNN models exploiting information leaks through memory and timing side-channels. Specifically, we study attacks on a hardware accelerator that is protected by secure processor techniques similar to the scheme used in Intel SGX [2]. In this setting, an adversary can feed inputs to a protected computation and observe off-chip accesses, but cannot observe or change the computation and the internal state. Surprisingly, we show that an adversary can effectively reverse engineer both the structure and the weights of an encrypted CNN model running on a hardware accelerator that performs the inference (i.e., forward propagation). Because the CNN states (feature maps) and parameters (weights) are often quite large, it is impractical to hold all feature maps, weights, and intermediate results in the on-chip memory of an accelerator. As a result, CNN accelerators typically store feature maps and weights in off-chip memory and access them as needed. Even if data values are encrypted, memory access patterns reveal which memory locations are accessed and whether each access is a read or a write. In this study, we show that the memory access patterns expose key parameters of the network structure such as the number of layers, input/output sizes of each layer, the size of filters, data dependencies among layers, etc. Given this information, an attacker can infer a small set of possible network structures by further considering the execution time of a CNN accelerator, which indicates the amount of computation. In our experiments, we demonstrate the proposed attack by reversing engineering the structures of two popular CNN models in AlexNet [9] and SqueezeNet [8].

In addition to revealing the network structure, this study shows that the memory access patterns also leak information on weight values when dynamic zero pruning is used for off-chip memory accesses. The optimization is based on the observation that the feature maps from the intermediate layers of a CNN model contain a large number of zeros. Recent studies in [1, 11, 12] have shown that these feature maps can be compressed in DRAM by only storing non-zero values and the associated indices to significantly reduce the memory bandwidth usage. Unfortunately, this optimization leaks the number of zero-valued pixels pruned by the activation function, which can be leveraged to infer the ratio between each weight and the bias value. To the best of our knowledge, this paper represents the first study on reverse engineering of convolutional neural network models on hardware accelerators, especially in the context of exploiting the side channel through memory access patterns.

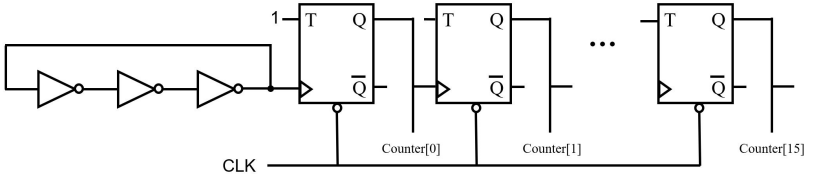
The rest of the paper is organized as follows: Section 2 defines the assumed threat model; Sections 3 and 4 present two reverse-engineering attacks on the structure and the weights of a CNN model and evaluate the effectiveness of the proposed attacks; Section 5 discusses the related work, and Section 6 concludes the paper.

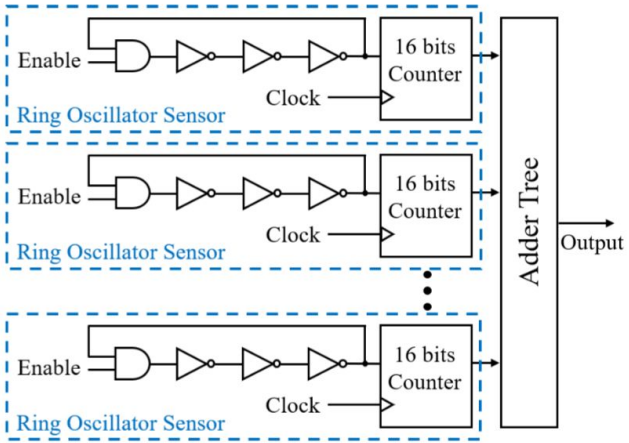
Our work

- First to investigate the remote FPGA side-channel attack on stealing DNN models.
- Cloud scenario.
  - Passive attacker.
    - No control of input.
  - Covertly.
    - No physical access to FPGA instance.

Design of FPGA Power Sensor

- Ring Oscillator (RO) Power Sensor.
  - Voltage fluctuations have a prominent impact on the frequency of RO.
  - Previous work [14] shows that the frequency of RO can be treated as power side-channel leak to recover RSA key.
  - The output of last inverter is connected to 16-bit T flip-flop counter.
  - The sum of 20 RO power sensors reading.

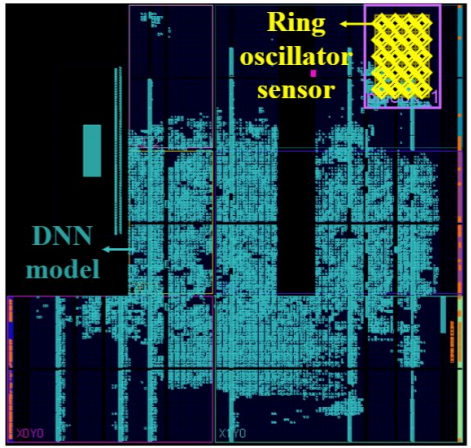




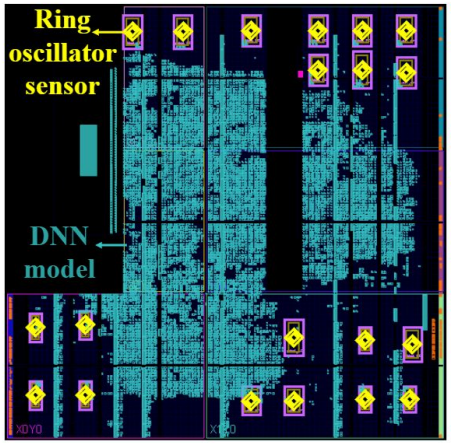
[14]. Mark Zhao and G Edward Suh. FPGA-based remote power side-channel attacks.In2018 IEEE Symposium on Security and Privacy (SP), pages 229–244. IEEE, 2018

Design of FPGA Power Sensor

- Placement of RO power sensors.
  - RO power sensors are contrained in a virtual FPGA (slot).
  - RO sensors are equally distributed on the board. (More Agressive)
- Reading from the RO power sensors.
  - Sent to a workstation for further analysis through xillybus [15].



a. RO sensors are restricted in a slot.



b. RO sensors are equally distributed on the board.

[15]. Xillybus. Xillybus product brief. http://xillybus.com/downloads/xillybus_product_brief.pdf .

DNN Layers and Computational Workload

- Different layers introduce different types of operators and different workloads.
  - Optimized MAC (multiply and accumulation) for FC layer and Conv layers.
  - MAX operations by pooling layers.
- Computational workload of three popular layers(The number of MAC operations).
  - Fully-connected (FC) layer.
  - Convolutional (Conv) layer.
  - Pooling layer(MAX operations).

Layer Type	Hyper-Parameter	Definition
FC layer	N_i	Number of neurons of the $layer_i$
Conv layer	W_i	Width of the output feature map of the $layer_i$
	F	Size of the filter
	D_i	Depth of output feature map (Number of filters)
	S	Stride
Pooling layer	W_i	Width of the output feature map of the $layer_i$
	F	Size of the filter
	S	Stride

$$FC_{mac} = N_{i-1} \times N_i \tag{1}$$

$$CONV_{mac} = W_i^2 \times F^2 \times D_{i-1} \times D_i \tag{2}$$

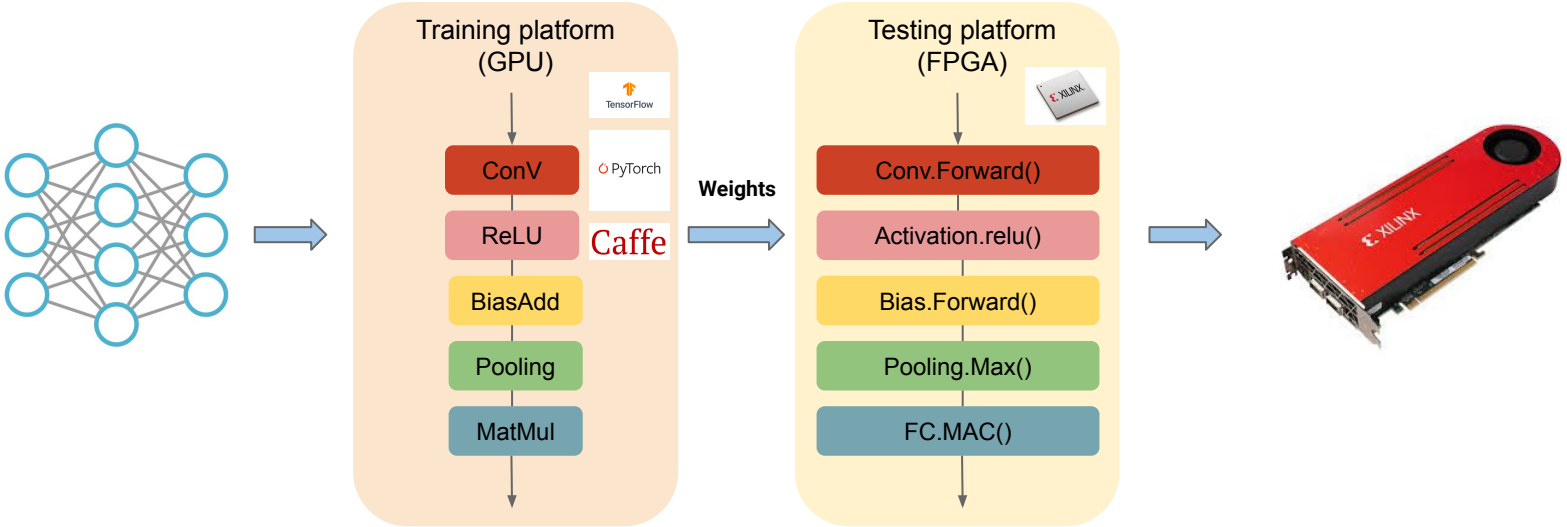
$$W_i = \frac{W_{i-1} - F + 2 \times P}{S} + 1 \tag{3}$$

$$W_{pooling} = \frac{W_{i-1} - F}{S} + 1 \tag{4}$$

$$S \leq F \leq \frac{W_i}{2} \tag{5}$$

Experiment setting

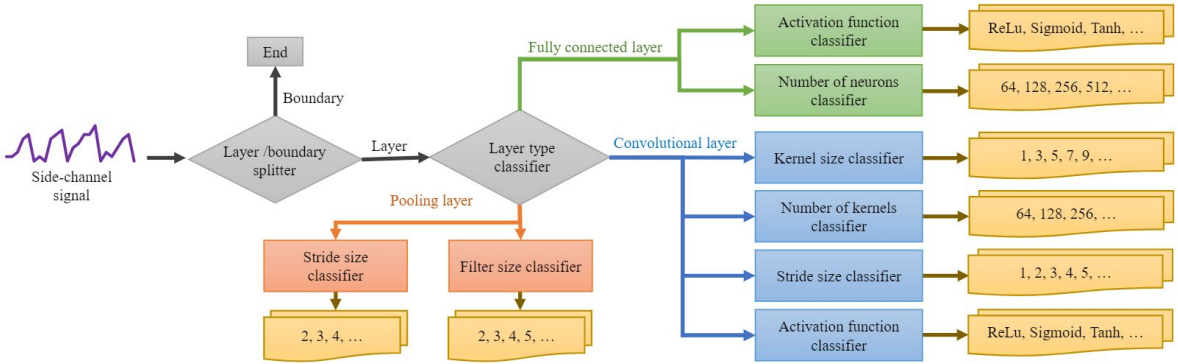
- Experiment platform.
  - Xilinx ZedBoard [16].
  - Xilinx Vivado.
- Deep-learning model inference on FPGA accelerator.
  - DNN models are trained on GPU and weights of model are fixed.
  - Victim DNN runs in the inference stage.



[16] Xilinx. Zedboard. https://www.xilinx.com/products/boards-and-kits/1-8dyf -11.html.

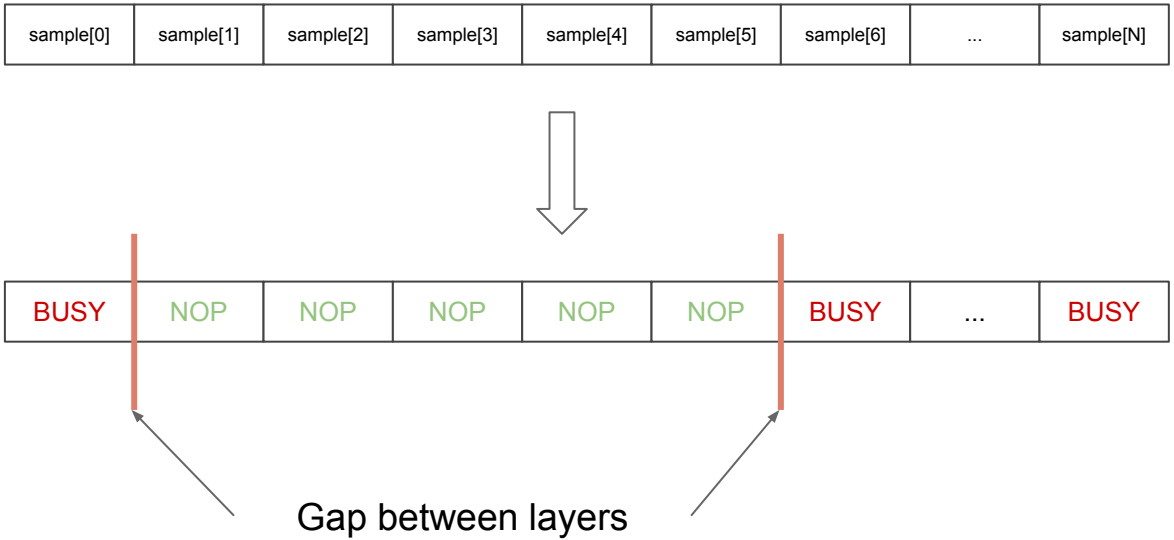
Our attack framework

- Overview of attack.
  - Profiling.
    - Before the actual attack, the adversary profiles a set of models to train the inference models.
  - Extraction.
    - Use trained inference models to extract model structure of victim deep-learning models.



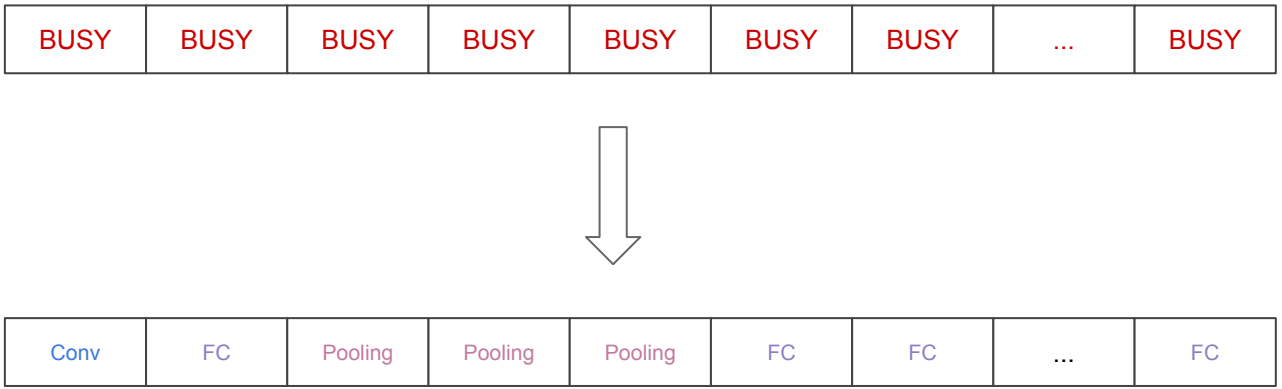
Our attack framework MoSRePS

- Splitting layers.
  - Classify samples into ‘NOP’ or ‘BUSY’ by Xgboost machine.
  - Split iterations if the number of consecutive ‘NOP’ is above threshold.



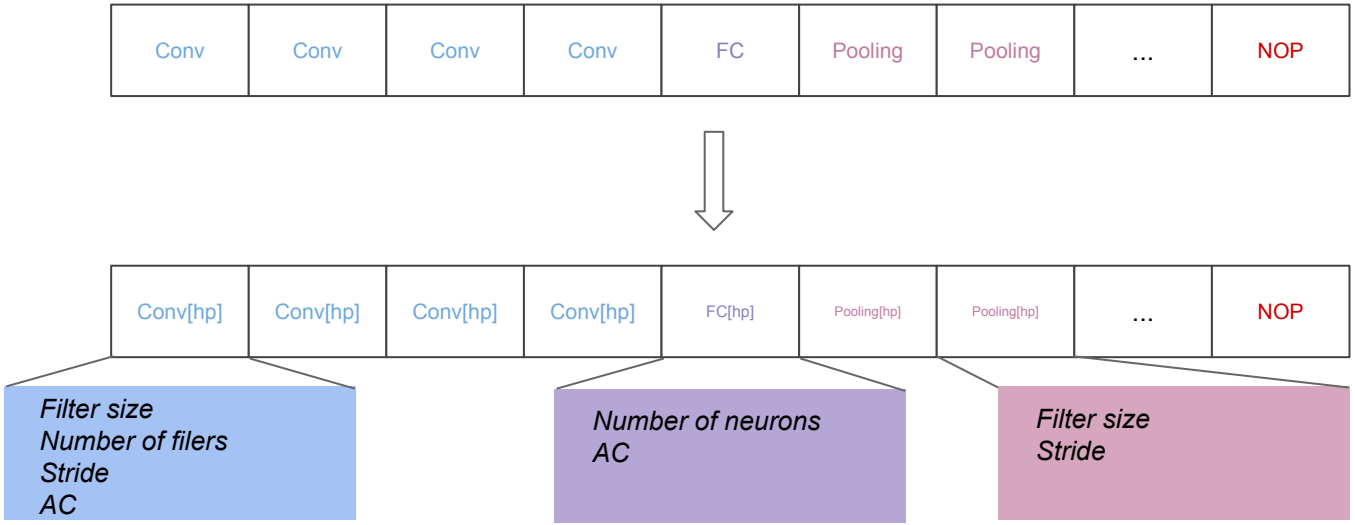
Our attack framework MoSRePS

- Recognize layer type.
  - Convolutional layers, fully-connected layers and Pooling layers.
- Classify each ‘BUSY’ into ‘Conv’, ‘FC’ or ‘Pooling’.
  - ‘Conv’, ‘FC’ or ‘Pooling’ are short for ‘Convolution’, ‘Fully-connected’ and ‘Pooling layers’.



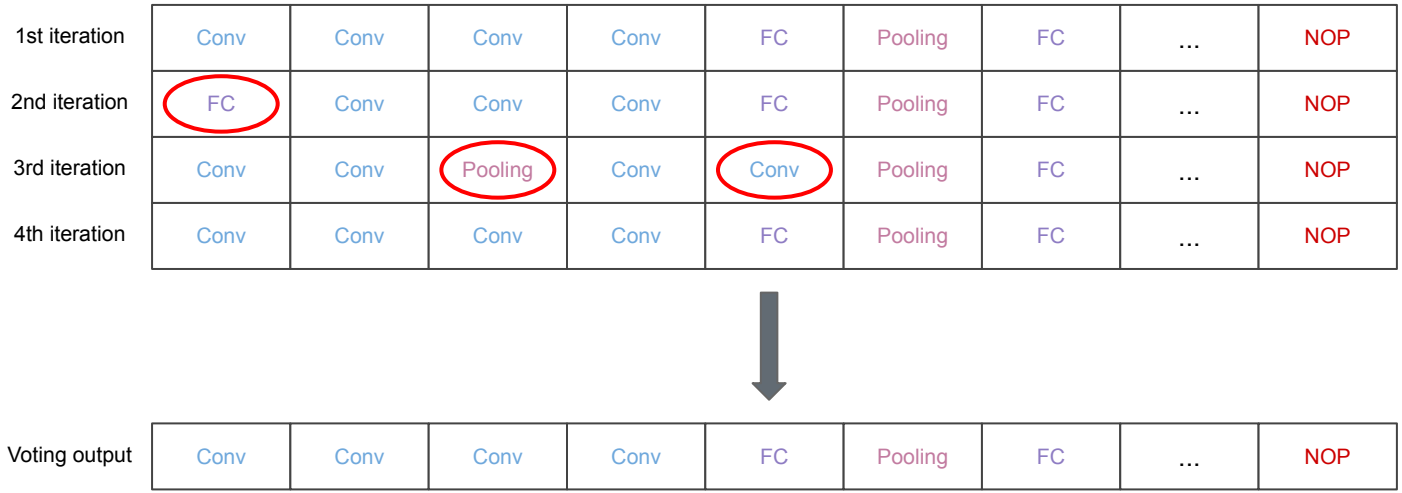
Our attack framework MoSRePS

- Infer hyper-parameters.
  - Infer the hyper-parameters for ‘Conv’, ‘FC’ and ‘Pooling’.
  - ‘hp’ is short for hyper-parameter.



Our attack framework MoSRePS

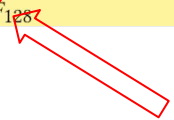
- Voting.
  - Combine multiple predicted DNN layer sequences to correct the wrong predictions.



Experimental evaluation

- Splitting iterations.
  - Accuracy is over 96%.
- Layer inference.
  - Accuracy reaches 100%.
- Hyper-parameter inference.
  - Accuracy is over 94.28%.
- Case study on VGG16.
  - For layer type and sequence.
    - 100% (**16**/**16**).
  - For hyper-parameters.
    - 94.11% (**64**/**68**).

Ground-truth	$C_{3,64,1} - C_{3,64,1} - P_{2,2} - C_{3,128,1} - C_{3,128,1} - P_{2,2} - C_{3,256,1} - C_{3,256,1} - C_{3,256,1} - P_{2,2} - C_{3,512,1} - C_{3,512,1} - C_{3,512,1} - P_{2,2} - C_{3,512,1} - C_{3,512,1} - C_{3,512,1} - P_{2,2} - F_{512} - F_{256} - F_{128}$
Predicted structure	$C_{3,64,1} - C_{3,64,1} - P_{2,2} - C_{3,128,1} - C_{3,128,1} - P_{2,2} - C_{3,512,1} - C_{3,512,1} - C_{3,256,1} - P_{2,2} - C_{3,512,2} - C_{3,512,2} - C_{3,512,1} - P_{2,2} - C_{3,512,1} - C_{3,512,1} - C_{3,512,1} - P_{2,2} - F_{512} - F_{256} - F_{128}$



Misprediction for one Conv layer

Our contribution

- First to exploit the FPGA remote power side channel to steal DNN models.
- We use 8 different classifier models as inference models to infer model secrets.
- Our attack extracts the whole structure of deep-learning models.
  - Layer.
    - Layer type.
    - Layer sequence.
  - Layer hyper-parameters.
    - Neuron number.
    - Filter size.
    - Filter number.
    - Stride.
  - Activation function(ReLu, Sigmoid and Tanh).
- Achieve high inference accuracy among a wide range of deep-learning models.
  - 5-layer MLP.
  - ZFNet.
  - VGG16.

Future work

- FPGA single-tenancy Scenario.
  - Giechaskiel et al. [17] proved the power supply unit (PSU) can be exploited to construct FPGA-to-FPGA, CPU-to-FPGA, and GPU-to-FPGA covert channels between different boards.
    - Cross-FPGA model-stealing attack can be one direction in future.
- Model weight inference.
  - Hua et al. [12] showed weights can be inferred, but they assume the adversary can feed input to the CNN inference accelerator.
    - Assuming input data is public and known to the adversary.
- Defense.
  - Hiding the power consumption patterns unique to the DNN layers/hyper-parameters.
    - Add supplementary hardware to mask power consumption.
    - Embed active fences around each tenant.
  - Rejecting the deployment request of suspicious FPGA logic.
    - Verify each tenant's RTL design or netlist file.

[12] Weizhe Hua, Zhiru Zhang, and G Edward Suh. Reverse engineering convolu-tional neural networks through side-channel information leaks. In2018 55thACM/ESDA/IEEE Design Automation Conference (DAC), pages 1–6. IEEE, 2018.
[17] K Rasmussen, I Giechaskiel, and Jakub Szefer. Capsule: Cross-fpga covert-channe lattacks through power supply unit leakage. In IEEE Symposium on Security and Privacy, volume 1. IEEE, 2020.

Stealing Neural Network Structure through Remote FPGA Side-channel Analysis

Yicheng Zhang, Rozhin Yasaei, Hao Chen, Zhou Li,
Mohammad Abdullah Al Faruque

Thanks!

Q&A

University of California, Irvine

UCI Samueli School of Engineering
University of California, Irvine

Reference

1. Mengjia Yan, Christopher W Fletcher, and Josep Torrellas. Cache telepathy: Lever-aging shared resource attacks to learn{DNN}architectures. In29th{USENIX}Security Symposium ({USENIX}Security 20), pages 2003–2020, 2020
2. Anuj Vaishnav, Khoa Dang Pham, and Dirk Koch. A survey on fpga virtualization.In2018 28th International Conference on Field Programmable Logic and Applications(FPL), pages 131–1317. IEEE, 2018
3. Amran A Al-Aghbari and Muhammad ES Elrabaa. Cloud-based fpga customcomputing machines for streaming applications.Ieee Access, 7:38009–38019, 2019.
4. Guohao Dai, Yi Shan, Fei Chen, Yu Wang, Kun Wang, and Huazhong Yang. Onlinescheduling for fpga computation in the cloud. In2014 International Conferenceon Field-Programmable Technology (FPT), pages 330–333. IEEE, 2014.
5. Naif Tarafdar, Thomas Lin, Daniel Ly-Ma, Daniel Rozhko, Alberto Leon-Garcia,and Paul Chow. Building the infrastructure for deploying fpgas in the cloud. InHardware Accelerators in Data Centers, pages 9–33. Springer, 2019.
6. Ke Zhang, Yisong Chang, Mingyu Chen, Yungang Bao, and Zhiwei Xu. Computerorganization and design course with fpga cloud. InProceedings of the 50th ACMTechnical Symposium on Computer Science Education, pages 927–933, 2019.
7. Stuart Byma, J Gregory Steffan, Hadi Bannazadeh, Alberto Leon Garcia, and PaulChow. Fpgas in the cloud: Booting virtualized hardware accelerators with open-stack. In2014 IEEE 22nd Annual International Symposium on Field-ProgrammableCustom Computing Machines, pages 109–116. IEEE, 2014
8. Fei Chen, Yi Shan, Yu Zhang, Yu Wang, Hubertus Franke, Xiaotao Chang, andKun Wang. Enabling fpgas in the cloud. InProceedings of the 11th ACM Conferenceon Computing Frontiers, pages 1–10, 2014.
9. Joel Mandebi Mbongue, Festus Hategekimana, Danielle Tchuinkou Kwadjo, andChristophe Bobda. Fpga virtualization in cloud-based infrastructures over virtio.In2018 IEEE 36th International Conference on Computer Design (ICCD), pages242–245. IEEE, 2018.
10. Joel Mandebi Mbongue, Alex Shuping, Pankaj Bhowmik, and Christophe Bobda.Architecture support for fpga multi-tenancy in the cloud. In2020 IEEE 31st Inter-national Conference on Application-specific Systems, Architectures and Processors(ASAP), pages 125–132. IEEE, 2020.
11. Jagath Weerasinghe, Francois Abel, Christoph Hagleitner, and Andreas Herk-ersdorf. Enabling fpgas in hyperscale data centers. In2015 IEEE 12th Intl Confon Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Auto-nomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computingand Communications and Its Associated Workshops (UIC-ATC-ScalCom), pages1078–1086. IEEE, 2015.
12. Weizhe Hua, Zhiru Zhang, and G Edward Suh. Reverse engineering convolu-tional neural networks through side-channel information leaks. In2018 55thACM/ESDA/IEEE Design Automation Conference (DAC), pages 1–6. IEEE, 2018.
13. Honggang Yu, Haocheng Ma, Kaichen Yang, Yiqiang Zhao, and Yier Jin. Deepem:Deep neural networks model recovery through em side-channel informationleakage.
14. Mark Zhao and G Edward Suh. Fpga-based remote power side-channel attacks.In2018 IEEE Symposium on Security and Privacy (SP), pages 229–244. IEEE, 2018
15. Xillybus. Xillybus product brief. http://xillybus.com/downloads/xillybus_product_brief.pdf .
16. Xilinx. Zedboard. https://www.xilinx.com/products/boards-and-kits/1-8dyf -11.html.
17. K Rasmussen, I Giechaskiel, and Jakub Szefer. Capsule: Cross-fpga covert-channelattacks through power supply unit leakage. InIEEE Symposium on Security andPrivacy, volume 1. IEEE, 2020.