High-Fidelity Model Extraction Attacks via Remote Power Monitors

Anuj Dubey, Emre Karabulut, Amro Awad, Aydin Aysu
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC, USA
{aanujdu,ekarabu,ajawad,aaysu}@ncsu.edu

2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS) | 978-1-6654-0996-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/AICAS54282.2022.9869973

***Abstract*—This paper shows the first side-channel attack on neural network (NN) IPs through a remote power monitor. We demonstrate that a remote monitor implemented with time-to-digital converters can be exploited to steal the weights from a hardware implementation of NN inference. Such an attack alleviates the need to have physical access to the target device and thus expands the attack vector to multi-tenant cloud FPGA platforms. Our results quantify the effectiveness of the attack on an FPGA implementation of NN inference and compare it to an attack with physical access. We demonstrate that it is indeed possible to extract the weights using DPA with 25000 traces if the SNR is sufficient. The paper, therefore, motivates secure virtualization—to protect the confidentiality of high-valued NN model IPs in multi-tenant execution environments, platform developers need to employ strong countermeasures against physical side-channel attacks.**

***Index Terms*—Neural networks, model stealing, time-to-digital converters, secure virtualization**

I. Introduction

Stealing machine learning (ML) models from embedded devices using physical side-channel attacks has become a new booming threat. Several recently published attacks show the possibility to steal the model's internals from a variety of target platforms like microcontrollers [1], GPUs [2], FPGAs [3], and domain-specific accelerators such as Intel's Neural Compute Stick [4]. Meanwhile, the cloud providers like Amazon have started to host FPGAs as services on which a user can run a custom application on a pay-per-use business model. Thus, apart from the edge FPGA platforms, users have also started running ML applications on cloud-based FPGAs because of the cost-effectiveness: the user does not have to permanently buy the FPGA and rent it without the cost of its maintenance.

A single user might not require the resources of an entire FPGA. Thus, it may be economically more feasible to have the cloud provider support multi-tenancy on the FPGA fabric in which the users utilize distinct regions of the same FPGA fabric. Multi-tenancy works well for both the cloud provider that can rent out the same FPGA to multiple users, and the user that can request and pay for only the resources needed for the target application. The users access their allocated regions via authenticated shells to instantiate digital designs and run software on them—for example, a neural network accelerator design and inference. However, sharing the same resources by multiple users can create unintentional and potentially dangerous security vulnerabilities [5].

Prior works have demonstrated the presence of side-channels between designs co-located on the same FPGA fabric due to the shared power distribution network. Schellenberg *et al.* first showed how to extract the secret AES key from an FPGA remotely using a time-to-digital converter (TDC) [6]. Zhao *et al.* demonstrated a successful remote SPA attack to recover the secret key from an RSA module using a ring-oscillator-based attack circuit [7]. We refer interested readers to a recent survey that categorizes works on this topic [5].

However, early works on remote side-channels were only focused on cryptographic ciphers: the victim is an encryption module co-located with a malicious attack circuit on the FPGA. The research on exploring such attacks in the context of machine learning applications is limited. We argue that it is critical to explore such remote attacks for ML applications because of two reasons: 1) the number of published side-channel attacks on ML accelerators has significantly increased over the past few years, and 2) a large fraction of ML applications are deployed on the cloud these days [8].

To the best of our knowledge, there is no work that demonstrates the extraction of model *parameters* remotely from an ML accelerator yet. Among similar works, Moini *et al.* show how to remotely extract the MNIST input images fed to a CNN accelerator and how to extract the model structure (a.k.a., *hyperparameters*) using a TDC [9], [10], and Zhang *et al.* showed how to remotely extract model structure using a ring-oscillator-based attack circuit [11].

In this work, we show for the first time how to extract the weights of a trained binarized neural network running on an FPGA through physical side-channel attacks but without any physical access. We target a *high-fidelity* extraction that aims at extracting exact values of parameters [12]. Using a TDC to remotely measure the power variations on the FPGA, we conduct a differential power analysis (DPA) attack to extract the model weights successfully. Our results show that with 25000 TDC measurements, we can successfully extract the weights of the neural network and that remote access reduces the attack effectiveness[1] by 62.5×. Therefore, our paper urges the need to develop secure virtualization solutions for multi-tenant use in cloud FPGA infrastructures.

[1]By attack effectiveness, we mean the number of measurements needed for a successful attack.

II. Adversary Model and Preliminaries

In this section we state our assumptions on the adversary and present background information on neural networks and remote side-channel attacks.

A. Adversary Model

We follow the threat model of the prior works published on remote side-channel attacks [9], [10]. A summary is given as follows. The malicious attack circuit is co-located with the victim circuit on the FPGA fabric. We assume that the adversary knows the inputs fed to the model and the model hyperparameters. If they are unknown, the adversary can use the techniques proposed in the prior literature to extract the inputs as well as the hyperparameters [11].

The goal of the adversary is to extract the model weights that are highly lucrative because of the expensive training process required to generate them. The adversary by no means can physically access the FPGA. However, the adversary does have the capability to instantiate a DUT on the same FPGA (remotely) and execute computations with it.

B. Neural Networks

Neural networks have become a popular choice for ML applications, especially for common use-cases such as image classification, speech recognition, and cybersecurity, among others. It consists of smaller units called neurons that are arranged in groups called layers. The neurons of one layer may feed the neurons of the next layer through a weighted connection. The neuron then applies a non-linear function on the weighted summation of all the incoming connections. The weights of these connections are determined during training through a process called backpropagation [13]. These weights play a crucial role in determining the accuracy of the network.

The typical floating-point weights make inference quite expensive in terms of memory and computation. Thus, the past few years have seen a significant research in building the so-called quantized neural networks–the weights and activations are represented using lower number of bits in fixed point that reduces the memory footprint and enables efficient bit-level operations such as xnor-popcount. Binarized neural networks (BNN) are the extreme case of quantization where the weights and activations are binary. Prior works have proposed BNN architectures with accuracy as good as the conventional neural networks [14]. Due to the huge memory, computation, and energy savings, BNNs have become a popular choice in both academia and industry [15], [16].

C. Remote Side-Channel Attacks

The key idea behind *remote* power side-channel attacks is that the designs instantiated on the same chip share the power-distribution network which makes the power variations mutually observable. This become a security issue if some design **A** is processing confidential data because another design **B** can use a power sensor to observe **A's** power consumption and to potentially get information about the confidential data.

Prior works show how to launch remote power analysis attacks using circuits such as a ring-oscillator and TDC [6], [7]. A ring oscillator circuit can act as a power sensor because its frequency of oscillations varies with the power supplied to it. A TDC can also act as a power sensor because the propagation delay of the buffers varies with the power supplied. Specifically, a voltage drop anywhere in the FPGA will increase the buffer delay and reduce the number of buffers that the pulse crosses, thus reducing the TDC counts.

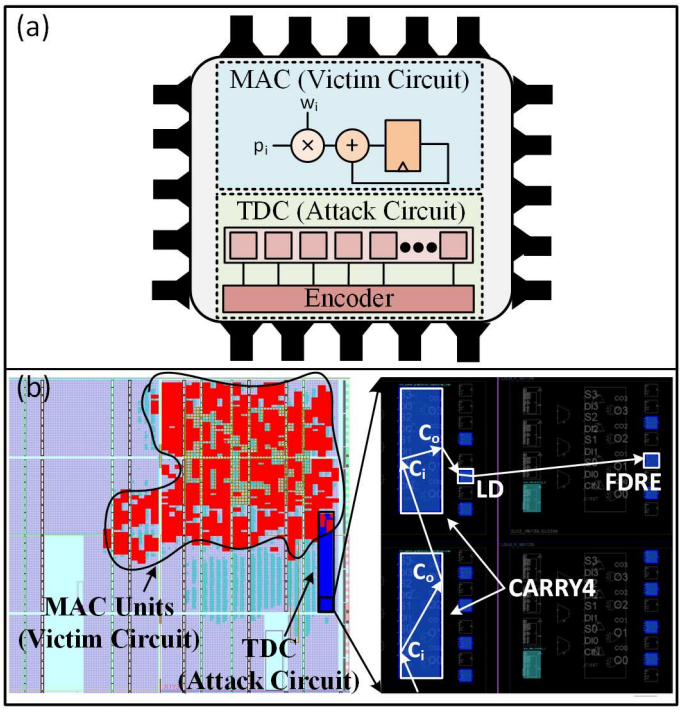

Fig. 1. (a) shows the conceptual remote attack setup that we create for our experiments and (b) shows the actual floorplan of the MAC units and TDC circuit on the FPGA highlighted in red and blue, respectively. The figure on the right zooms in to a portion of the TDC that depicts the `CARRY4` primitives serially connected via the carry-in and carry-out ports to eventually feed an `LD` (latch) and `FDRE` (flip flop) cell.

III. Neural Network and Remote Monitor Design

A. Hardware Design Details

The victim circuit in our attack model is a neural network accelerator, which primarily consists of multiply-accumulate (MAC) units along with some other blocks for non-linear operations. We focus on the MAC units since they are a good first point of attack in DPA—the known input data combines with unknown weights for the first time. We design a hardware that first loads all the inputs in an on-chip memory and then processes them sequentially every cycle. The weights of the neural network are already stored before the computations begin in another on-chip memory. The hardware loads each input every cycle and then either keeps it unchanged or computes its two's complement based on whether its weight is a one or zero, respectively. Next, it feeds this result to the accumulator, which keeps adding all the partial summations to eventually generate the complete summation for one node of a layer [17], [18].

B. Remote Monitor Details

We adopt the design of Schellenberg *et al.* in our work [6]. Fig. 1 shows the design details. The remote power monitor primarily consists of a chain of buffers (also called the delay chain), sequential elements to capture buffer outputs, and an encoder to compress the final output. The input to the delay chain is a clock, which gets slightly delayed everytime it crosses a buffer on the chain. The outputs from the buffers are tapped using latches which also run at the same clock as the delay chain's input clock. The initial part of the chain implemented using LUTs, and it is not observed since the variations only show up towards the end. The observable portion of the chain uses the built-in `CARRY4` primitive of the FPGA because of its low propagation delay that helps to improve the TDC's resolution.

The buffer outputs are tapped using the Xilinx's latch primitives named `LD` that consume the same clock as the delay chain. Thus, the latches capture how many buffers did the high portion of the clock (i.e., half clock cycle) cross before going low during its active region. The latch outputs are finally registered by the TDC output register implemented using the Xilinx's register primitives called `FDRE`. A priority encoder translates the number of ones seen on the delay path, thereby compressing the size of outputs from N to $log_2 N$.

A controller ensures that the TDC measurements are made when the target operation–the MAC operations in this case– executes in the victim circuit. In our experiments, we initiate the victim operations and the TDC measurements simultaneously, which keeps the traces synchronized in time. We agree that triggering the side-channel measurements at the right point in time is indeed a challenge in a remote attack since the attacker does not have a direct visibility over the target computations. This challenge is not unique to our scenario. A typical way to address this is to actually start with randomly choosing the windows and then align them later based on the patterns seen in the traces [6], [11]. For example, in Fig. 2 it is evident that each summation causes a sharp drop in the count and that can be used to identify the point where the first summation is computed. The controller writes the TDC outputs to a register file of parameterized depth. The depth and the frequency of the controller governs the width of the capture window. The attacker later exports the stored TDC counts to an external host PC and launches the DPA.

IV. Results

A. Experimental Setup

We use the Sakura-X FPGA board as our testing platform. The board provides a Kintex-7 FPGA on which we run the MAC operations alongwith the TDC-based attack circuit. We use Verilog to develop the RTL and Xilinx Vivado 2021.1 for synthesis, implementation, and bitstream generation. A C#-based software code on the host PC communicates with the FPGA for register reads and writes. Essentially, it is the software that sends the input data for MAC units, receives the output from the MAC units, verifies the outputs and also receives the stored TDC measurements.

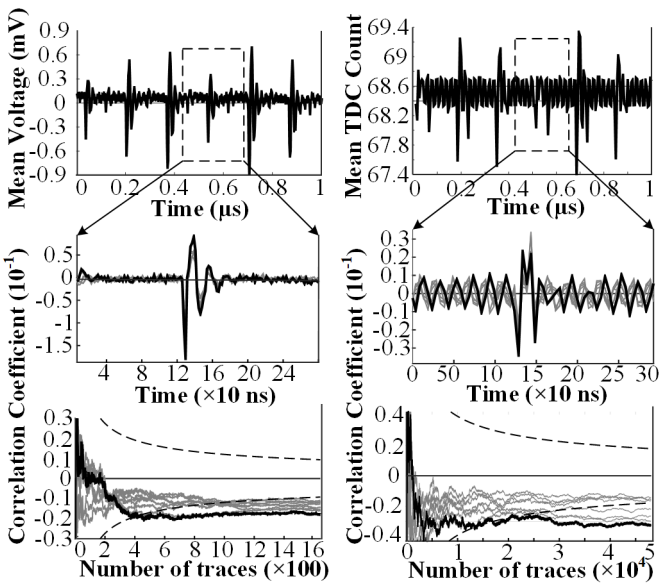

Fig. 2. The left three figures show the DPA results using real power traces from an oscilloscope and the right three figures show the results using TDC counts; the dark and light colored lines denote the correct and incorrect weight guesses. The Pearson correlation coefficient crosses the 99.99% confidence interval (dotted lines) at 400 traces with real power measurements and at 25k traces with the TDC measurements.

B. Attack Results and Comparison

We run a DPA attack on the accumulation register of the MAC units. As Section III-A describes, the hardware adds the product of each pixel with its corresponding weight to the accumulated sum in each cycle. Thus, we can hypothesize on the weights using this knowledge and create the power model for the accumulator register.

We next the derive the power model of the attack in detail. We denote the inputs and the corresponding weights p_i and w_i, respectively, where i is the index of the pixel. Thus, in each cycle i, the value of the accumulator register updates from the sum of first $(i-1)$ partial products to the sum of first i partial products. Since the power activity of an FPGA depends on the number of toggles in the registers, we assume the power model for cycle i to be the hamming distance (HD) of the $(i-1)^{th}$ and i^{th} summations, given as follows:

$$HD(\sum_{k=0}^{i-1} p_k \times w_k \, , \sum_{k=0}^{n} p_k \times w_k)$$

.

Since we target a binarized neural network, the weights can only be ± 1. We propose a sequential attack: extract the first n weights $w_0 - w_{n-1}$ by hypothesizing on the n^{th} partial sum given as $\sum_{k=0}^{n-1} p_k \times w_k$. The 2^n possibilities for $w_0 - w_{n-1}$ yields a hypothesis table of size 2^n. Then, starting from the n^{th} summation, we hypothesize on the next n^{th} partial sum to extract the next n weights and so on.

The proposed attack platform faces challenges that we describe next. The signal-to-noise ratio on Sakura-X for a single MAC unit was too low. We believe two factors attribute to this effect: 1) a single MAC unit is a relatively small hardware when compared to a larger design such as AES, and

2) the power consumption of the cells on the Kintex-7 is low due to the smaller technology node, when compared to other common targets such as Sakura-G with a Spartan-6 FPGA. Even for AES, we observe a voltage drop of only $\pm$1mV, when measured using an oscilloscope. Nevertheless, there is still enough signal to run a DPA attack because the 10 peaks are clearly visible. Thus, we first increase the SNR of our design to create a noticeable power difference between idle and active regions, like we see for AES. We instantiate multiple copies of the same MAC unit in the design which creates noticeable power variations at 256 instantiations. Note that the proposed attack works regardless but instantiating multiple copies will result in using fewer number of traces.

Fig. 2 shows the DPA results when we attack the first three weights (n=3) using both real and TDC measurements. The correlations cross the 99.99% confidence interval when the number of measurements reach 400 and 25000 with the real and TDC traces, respectively. The number of traces needed for a successful attack is low in this experiment because of sufficient SNR due to 256 instantiations of the target block and the low noise platform of Sakura-X[2]. From prior works [19], we know that the number of traces required for a successful attack inversely varies with the SNR. Thus, we believe that the attack will still be successful with lesser number of MAC instantiations but would require more measurements.

There is no prior work on remote DPA of ML accelerators to extract the parameters but some works have shown vulnerabilities using profiling to extract the image and hyperparameters [9]–[11]. Those works required 10 and 50 traces to construct the power profile of each possible image and hyperparameters, respectively. It is difficult to make a fair comparison between our attack results and prior works given the differences in the goals, post-processing techniques, and most importantly the attack class.

V. Conclusion

We expose the feasibility of remotely extracting the weights of an ML model running on an FPGA using a TDC-based remote power monitor. The current large-scale deployment of machine-learning applications on cloud FPGAs makes this work very relevant and warrants immediate research to explore countermeasures against such attacks. More research is needed on the attacks front too, to explore how well the attacks scale with other neural network topologies.

VI. Acknowledgments

This work is supported in part by NSF award #1943245, SRC GRC Task 2908.001, and Office of Naval Research (ONR). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for public release. Distribution is unlimited.

[2]Sakura-X board provides the the isolated power line as a direct output.

References

[1] L. Batina, S. Bhasin, D. Jap, and S. Picek, "CSI NN: reverse engineering of neural network architectures through electromagnetic side channel," in *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*. USENIX Association, 2019.

[2] L. Chmielewski and L. Weissbart, "On reverse engineering neural network implementation on GPU," in *Applied Cryptography and Network Security Workshops - ACNS 2021 Satellite Workshops, AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, and SiMLA, Kamakura, Japan, June 21-24, 2021, Proceedings*, ser. Lecture Notes in Computer Science, vol. 12809. Springer, 2021, pp. 96–113.

[3] A. Dubey, R. Cammarota, and A. Aysu, "MaskedNet: The first hardware inference engine aiming power side-channel protection," in *2020 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2020, San Jose, CA, USA, December 7-11, 2020*. IEEE, 2020.

[4] Y. Won, S. Chatterjee, D. Jap, A. Basu, and S. Bhasin, "WaC: First results on practical side-channel attacks on commercial machine learning accelerator," in *ASHES@CCS 2021: Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security, Virtual Event, Republic of Korea, 19 November 2021*, C. Chang, U. Rührmair, S. Katzenbeisser, and D. Mukhopadhyay, Eds. ACM, 2021, pp. 111–114.

[5] G. Dessouky, A.-R. Sadeghi, and S. Zeitouni, "SoK: Secure FPGA multi-tenancy in the cloud: Challenges and opportunities," in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2021, pp. 487–506.

[6] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori, "An inside job: Remote power analysis attacks on FPGAs," *IEEE Des. Test*, vol. 38, no. 3, pp. 58–66, 2021.

[7] M. Zhao and G. E. Suh, "FPGA-based remote power side-channel attacks," in *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 2018, pp. 229–244.

[8] A. Mishra, *Machine Learning in the AWS Cloud: Add Intelligence to Applications with Amazon SageMaker and Amazon Rekognition*. John Wiley & Sons, 2019.

[9] S. Moini, S. Tian, D. E. Holcomb, J. Szefer, and R. Tessier, "Remote power side-channel attacks on BNN accelerators in FPGAs," in *Design, Automation & Test in Europe Conference & Exhibition, DATE 2021, Grenoble, France, February 1-5, 2021*. IEEE, 2021, pp. 1639–1644.

[10] S. Tian, S. Moini, A. Wolnikowski, D. E. Holcomb, R. Tessier, and J. Szefer, "Remote power attacks on the versatile tensor accelerator in multi-tenant FPGAs," in *29th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, FCCM 2021, Orlando, FL, USA, May 9-12, 2021*. IEEE, 2021, pp. 242–246.

[11] Y. Zhang, R. Yasaei, H. Chen, Z. Li, and M. A. A. Faruque, "Stealing neural network structure through remote FPGA side-channel analysis," in *FPGA '21: The 2021 ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Virtual Event, USA, February 28 - March 2, 2021*. ACM, 2021, p. 225.

[12] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020.

[13] S. Linnainmaa, "Taylor expansion of the accumulated rounding error," *BIT Numerical Mathematics*, vol. 16, no. 2, pp. 146–160, 1976.

[14] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[15] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "FiNN: A framework for fast, scalable binarized neural network inference," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017.

[16] Techcrunch. (2020) Apple buys edge-based ai startup xnor.ai for a reported $200m. https://techcrunch.com/2020/01/15/apple-buys-edge-based-ai-startup-xnor-ai-for-a-reported-200m/.

[17] A. Dubey, R. Cammarota, and A. Aysu, "BoMaNet: Boolean masking of an entire neural network," in *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2020, pp. 1–9.

[18] A. Dubey, A. Ahmad, M. A. Pasha, R. Cammarota, and A. Aysu, "ModuloNET: Neural networks meet modular arithmetic for efficient hardware masking," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 506–556, 2022.

[19] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks: Revealing the secrets of smart cards*. Springer Science & Business Media, 2008, vol. 31.