caslab.csl.yale.edu

# Fingerprinting Cloud FPGA Infrastructures

Shanquan Tian†, Wenjie Xiong†, Ilias Giechaskiel‡, Kasper Rasmussen‡, Jakub Szefer†

*† Yale University, ‡University Oxford*

{shanquan.tian, wenjie.xiong, jakub.szefer}@yale.edu, {ilias.giechaskiel, kasper.rasmussen}@cs.ox.ac.uk

## 1. Project Overview

Cloud FPGAs have been recently deployed in data centers. Several research papers [1,2,3] have reported Cloud FPGA side or covert channel attacks. The requirement for such attacks is that the adversary knows the identification information of the Cloud FPGA instances. However, existing Cloud FPGA providers, such as AWS, secure their infrastructures through a number of countermeasures. Users do not have access to physical pins, but only a strict RTL "shell" interface that prevents access to Xilinx eFUSE and Device DNA primitives, for example.

This work shows that it is still possible to fingerprint Cloud FPGAs through Physical Unclonable Functions (PUFs) based on the decay of Dynamic Random Access Memories (DRAMs) attached to the Cloud FPGA boards, and, by extension, the FPGAs themselves.

## 2. System Diagram

The workflow of AWS F1, shown below, requires users to upload Design Checkpoint (DCP) files generated by Xilinx Vivado design tools, and then performs Design Rule Checks (DRCs) before generating bitstream files based on them. The generated bitstream files are managed by AWS and registered as Amazon FPGA Image (AFI), users can only reconfigure their FPGAs referencing AFI numbers.

A set of FPGA boards is connected to a server over PCIe. The boards contain FPGA chips, and are placed in fixed slots in the server. Each FPGA has access to four dedicated DRAM modules.

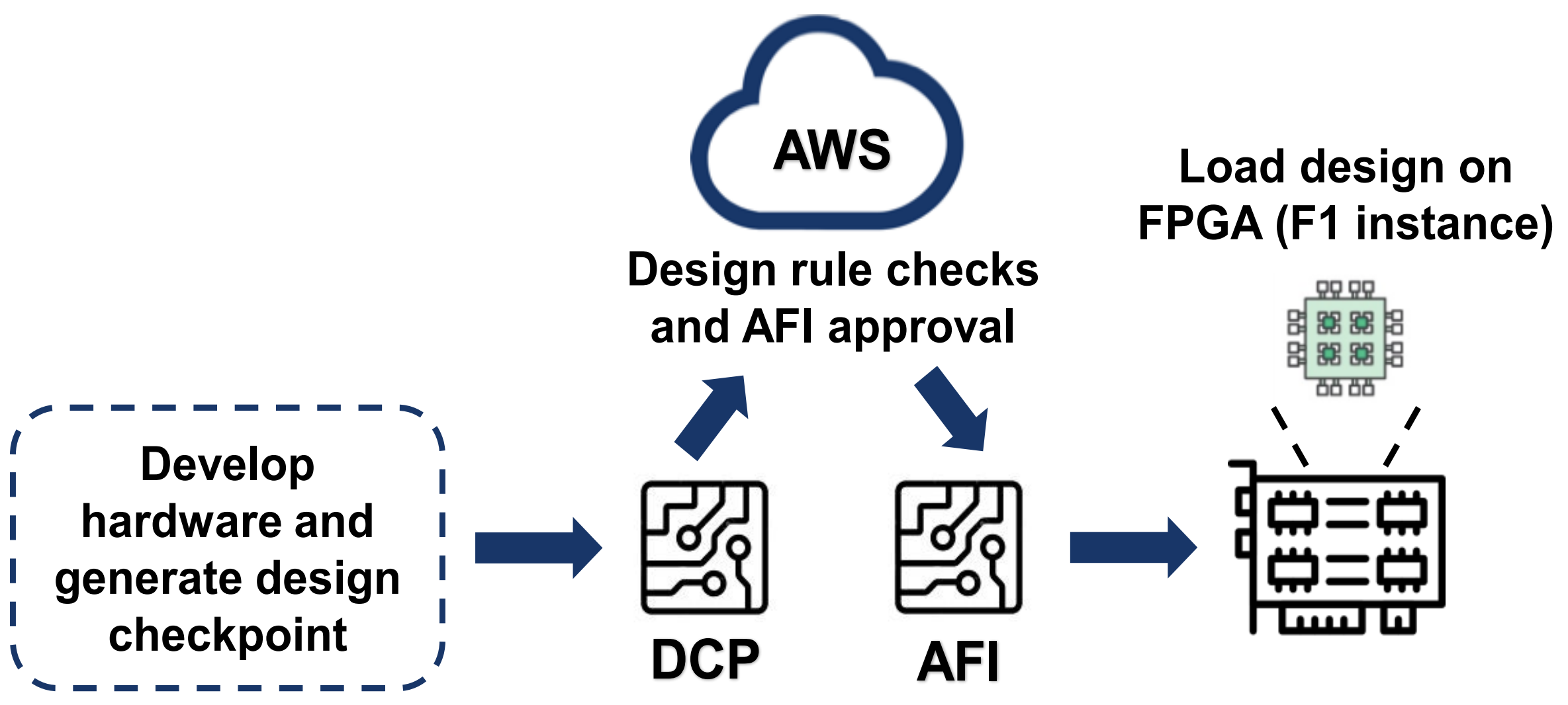

AWS
Design rule checks and AFI approval
Load design on FPGA (F1 instance)
Develop hardware and generate design checkpoint
DCP
AFI

**Figure 1. FPGA workflow on AWS: developers compile their custom logic and send encrypted Design Checkpoints (DCPs) to Amazon. DCPs which pass Design Rule Checks (DRCs) generate bitstream files that can be loaded onto F1 instances in the form of Amazon FPGA Images (AFIs)**

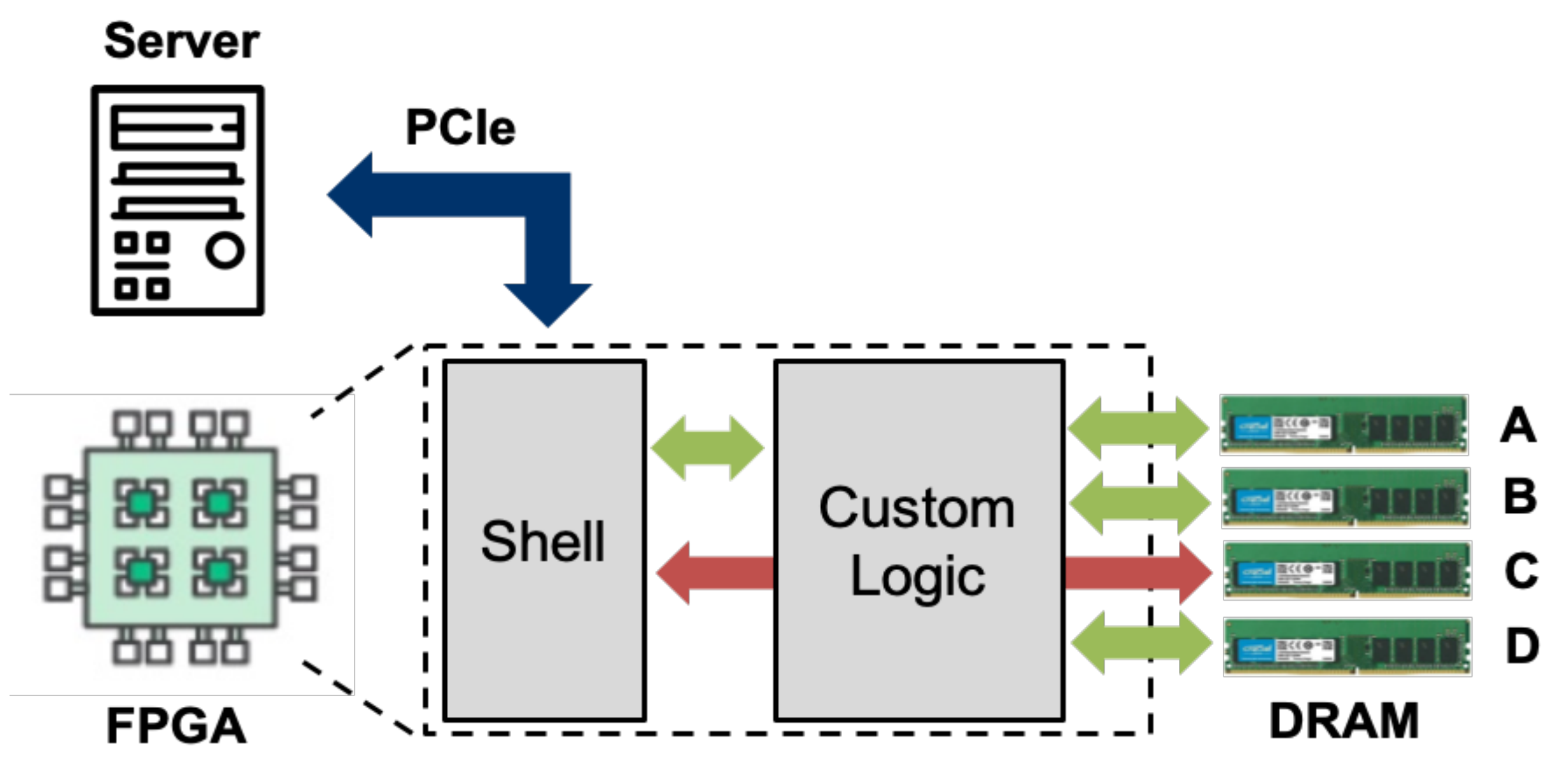


**Figure 2. System diagram: a virtual machine communicates with one or more FPGAs over PCIe**

## 3. DRAM PUFs on AWS F1

Each DRAM chip consists of DRAM banks, which are arrays of DRAM cells. The capacitor charge in each cell leaks over time through different leakage paths. If DRAM self- refresh and Error-Correcting Code (ECC) are disabled, bit flips (errors) will occur at different location after a certain decay period, and the variation can be used in DRAM PUFs.

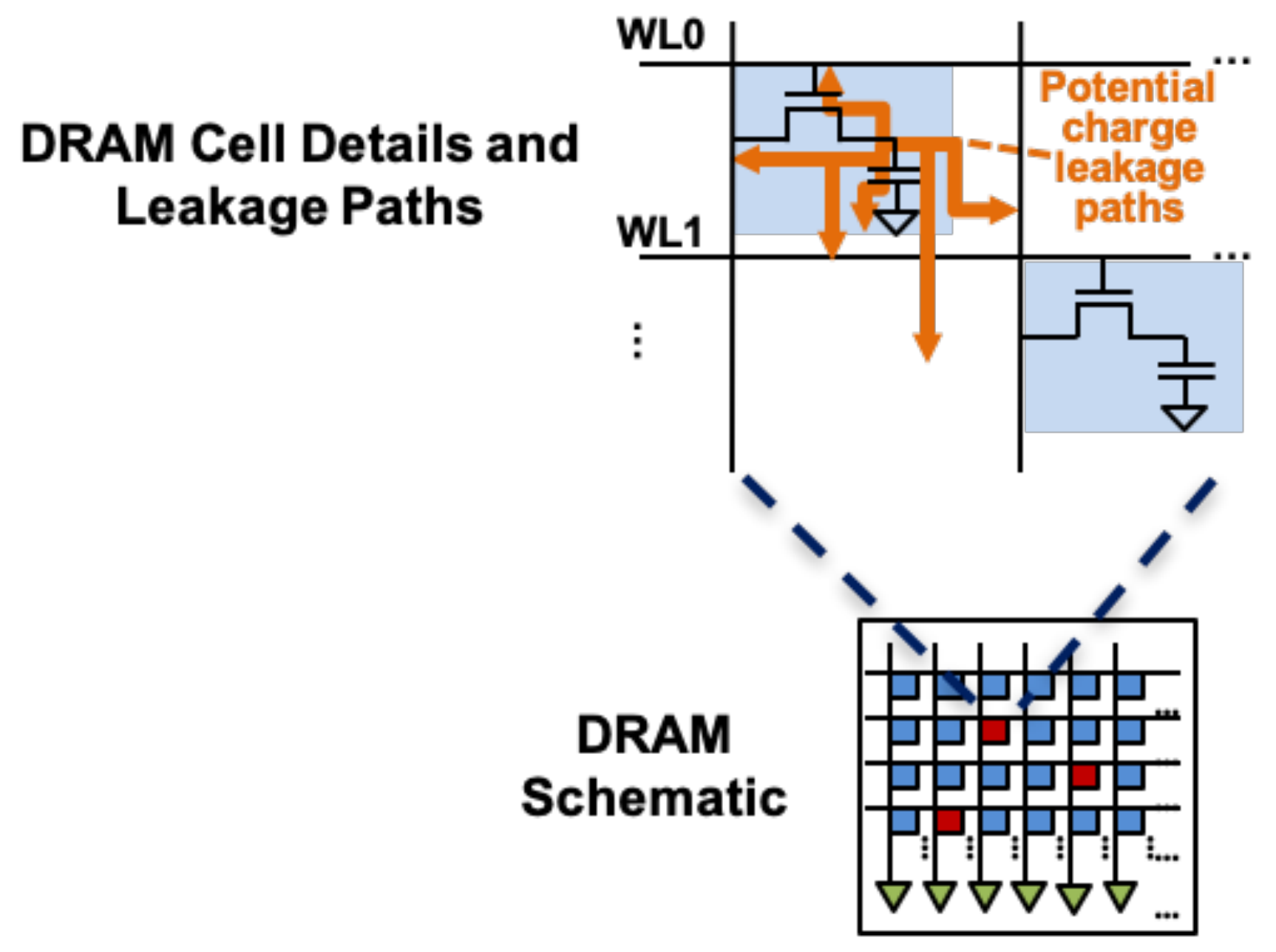

DRAM Cell Details and Leakage Paths
WL0
WL1
Potential charge leakage paths
DRAM Schematic

**Figure 3. PUFs that exploit DRAM charge leakage on DRAMs can uniquely identify the underlying hardware**

As shown in Figure 4, AFI-0 is first loaded to write all 1s to a certain area of a DRAM module. Then AFI-1 is loaded to stop memory self-refresh. Finally, after a fixed amount of time, AFI-0 is re-loaded to measure bit flips in the written addresses.

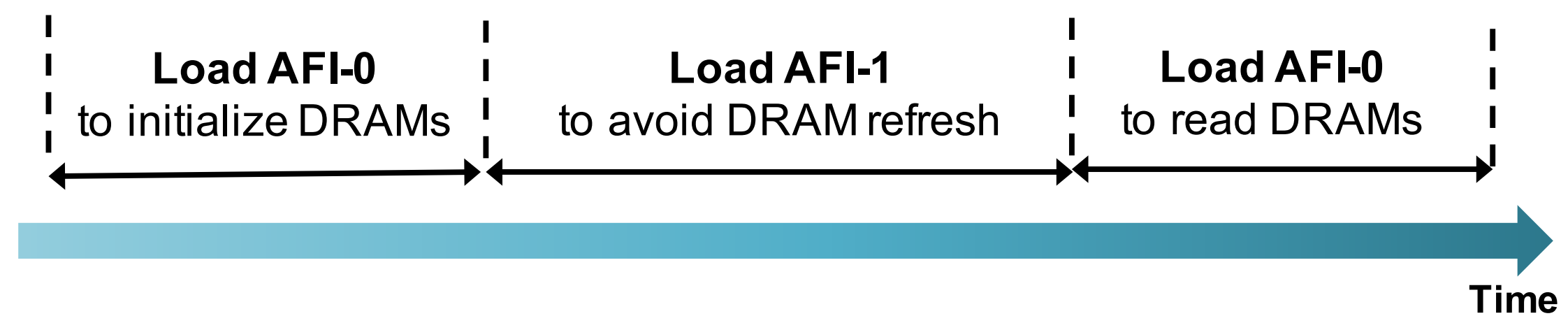


**Figure 4. Three Steps to measure DRAM PUFs with two AFIs**

## 4. Fingerprinting Metric

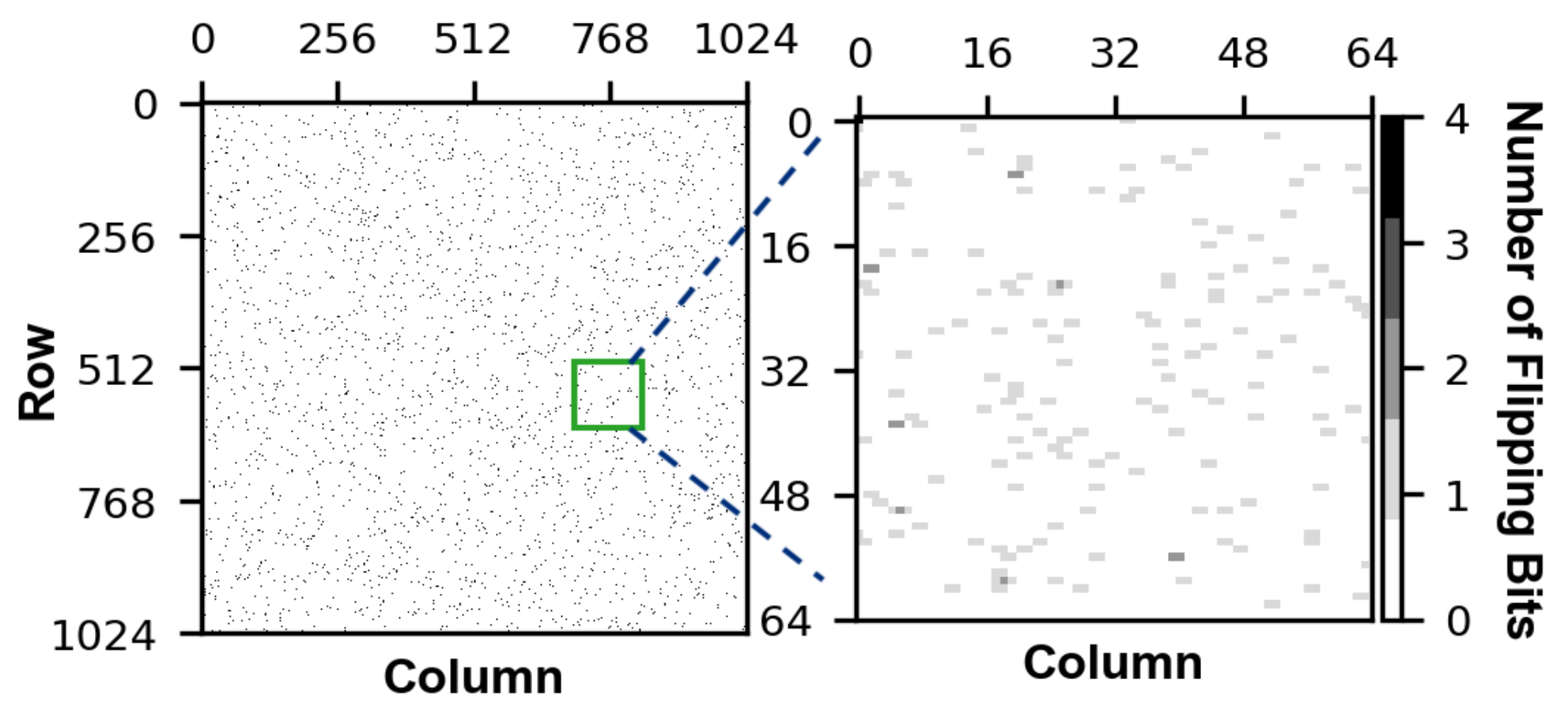


**Figure 5. Bitmap of an example DRAM PUF response**

To quantify how similar or different DRAM PUF responses are, we use the Jaccard index [4, 5]. Let F1 and F2 denote the set of bit flips in two DRAM PUF responses. The Jaccard Index (J) is defined as:

$$J(F_1, F_2) = \frac{|F_1 \cap F_2|}{|F_1 \cup F_2|}$$

Intra-device: from the same DRAM, $J(F_1, F_2) \to 1$
Inter-device: from different DRAMs, $J(F_1, F_2) \to 0$

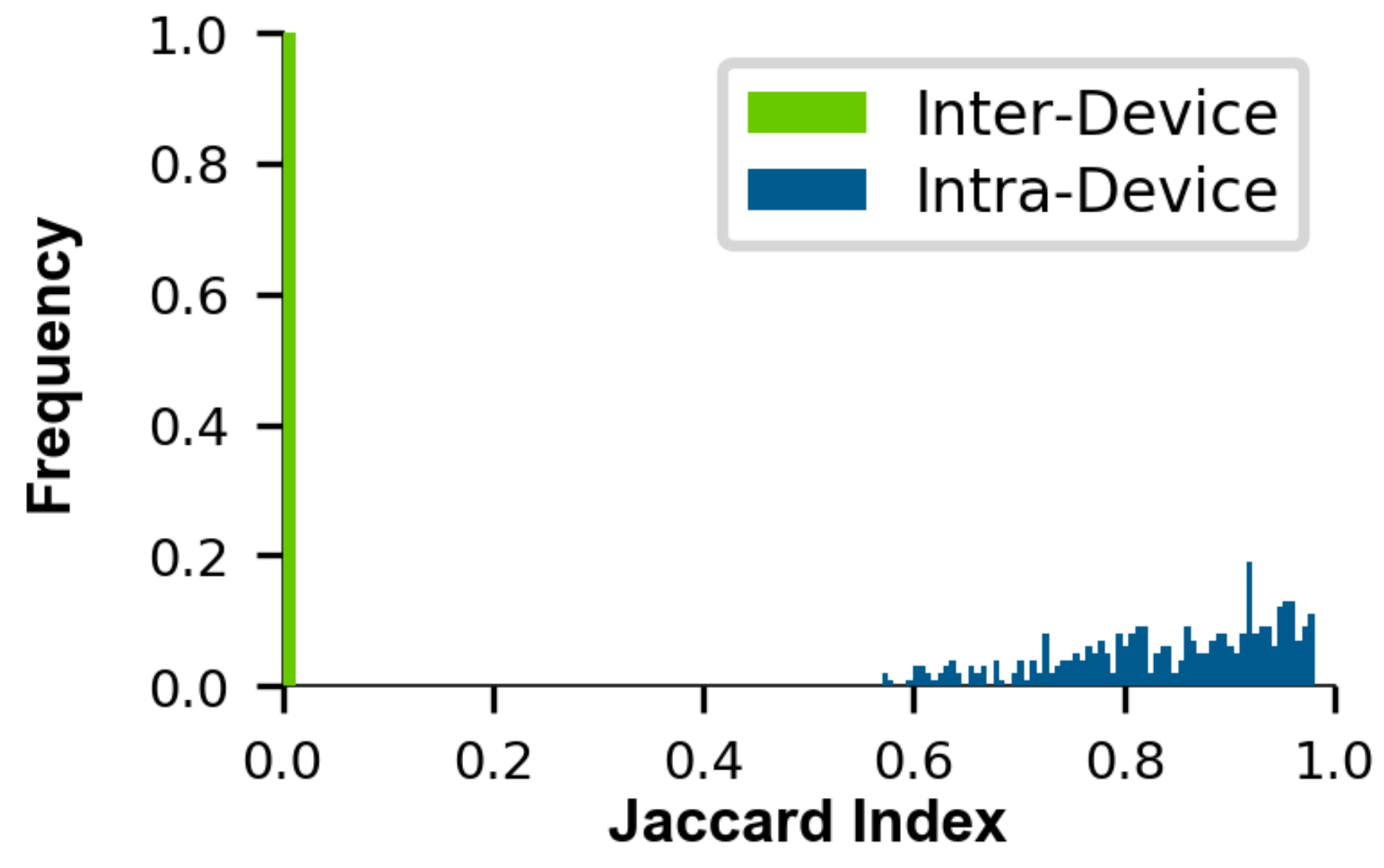


**Figure 6. Distribution of Jaccard indices for each pair of DRAM PUF responses on f1.2xlarge instances**

## 5. Evaluation

Our evaluation is the first to calculate the probability of renting the same FPGA as a function of time, and demonstrate that there is no overlap between FPGAs of different instance types.

**Table 1: Number and type of FPGA instances rented, along with the number of unique sets of FPGAs found and the approximate experimental cost using spot instances**

| F1 Type | Number of Slots | Number of FPGAs | Unique FPGAs | Approx. Cost ($) |
|---|---|---|---|---|
| 2xlarge | 1 | 60×1 | 10×1 | 3.47 |
| 4xlarge | 2 | 60×2 | 6×2 | 8.91 |
| 16xlarge | 8 | 60×8 | 8×8 | 83.16 |

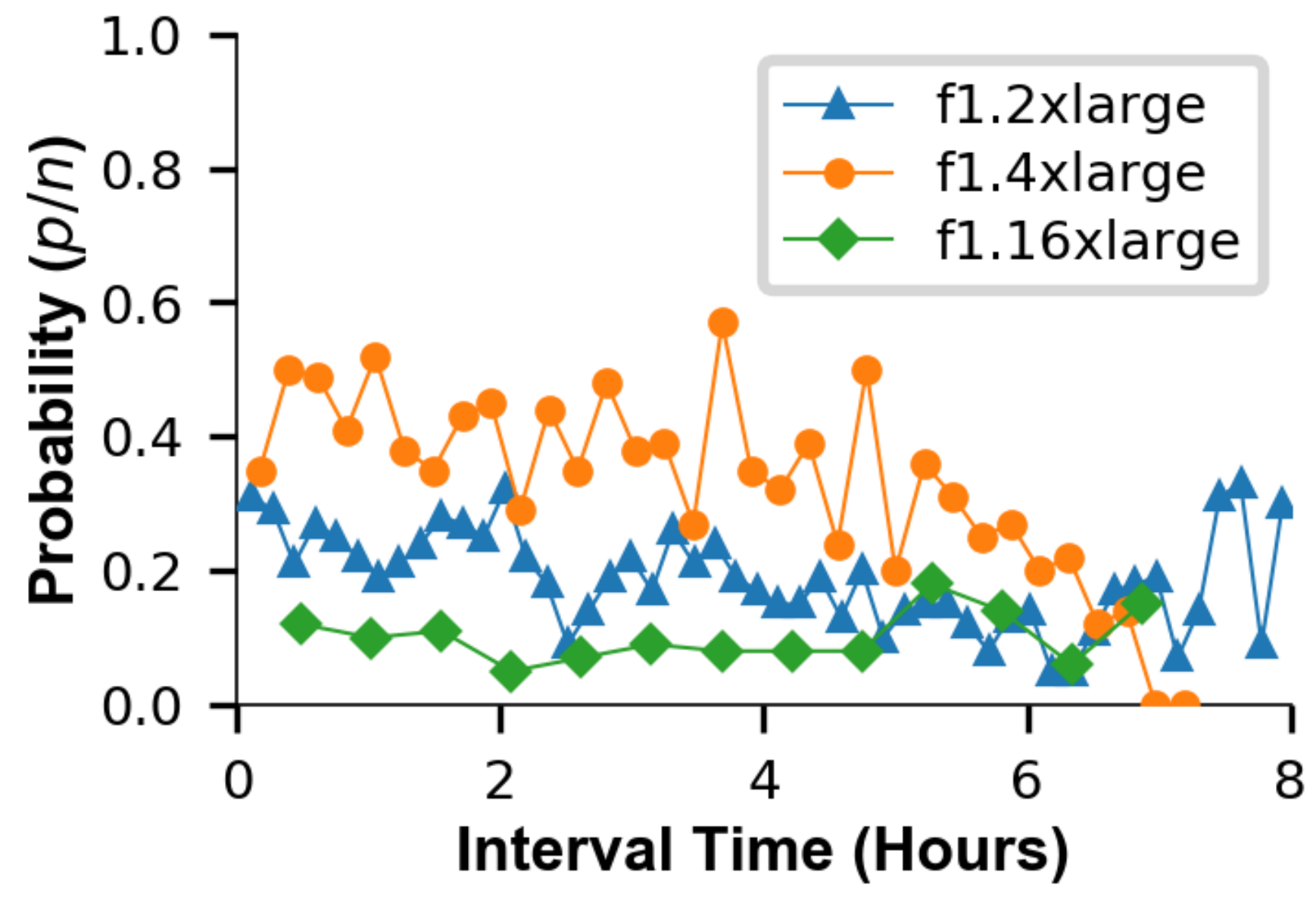
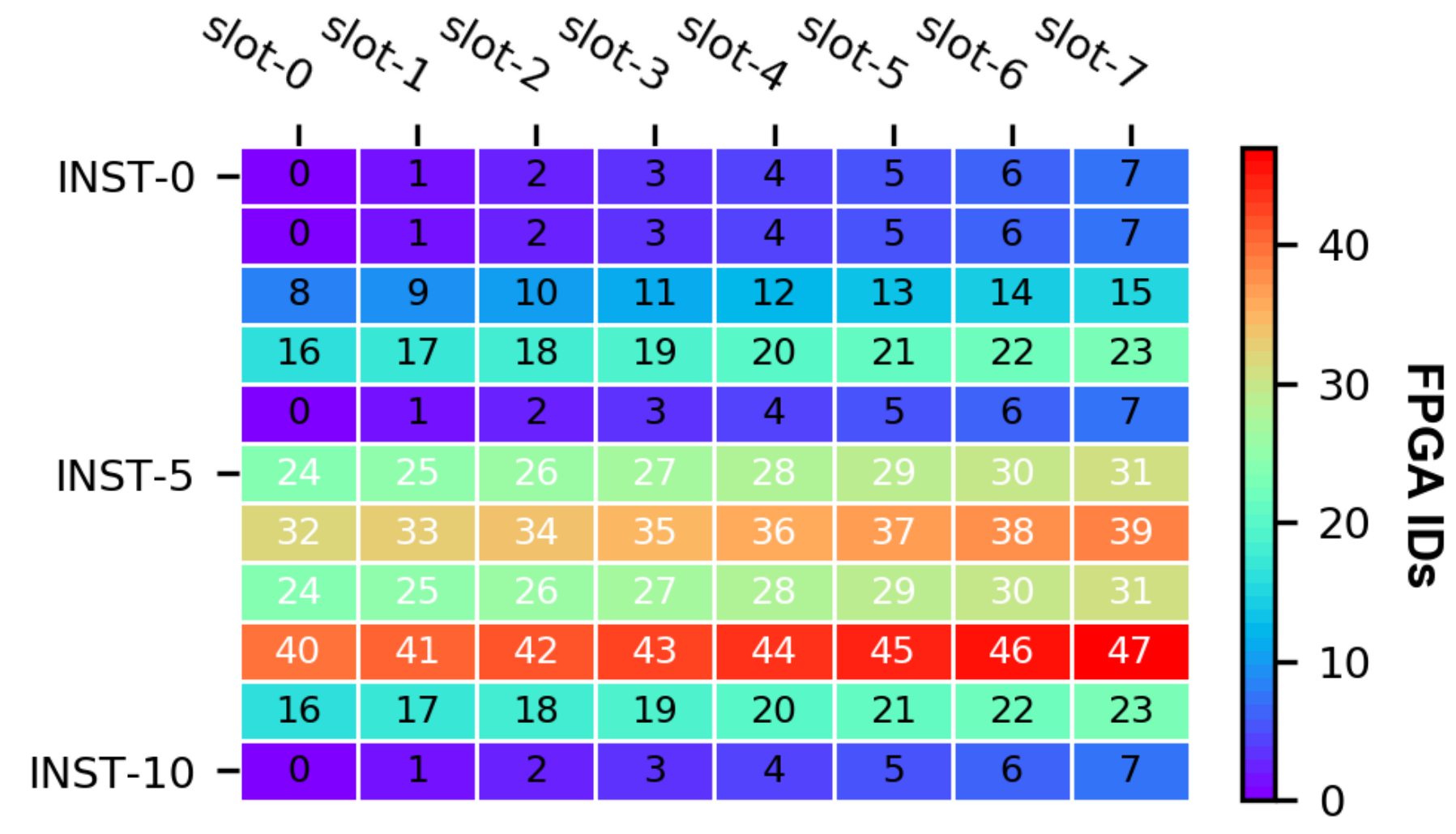


**Figure 7. Probability of renting re-allocated FPGA boards for different waiting periods. $p$ denotes the number of pairs (out of $n$) for which Jaccard indices are bigger than 0.5. Although the figure only shows slot 0, the probability for all slots is identical, as FPGA ordering does not change within instances**



| | slot-0 | slot-1 | slot-2 | slot-3 | slot-4 | slot-5 | slot-6 | slot-7 |
|---|---|---|---|---|---|---|---|---|
| INST-0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| INST-5 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| INST-10 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

**Figure 8. Fingerprinting FPGAs on f1.16xlarge instances with 8 FPGA slots: out of 11 spot instances, only 6 different sets of FPGAs are allocated. In the remaining instances, only 2 additional sets were identified (Table 1)**

## 6. Conclusion

The contributions of this paper are as follows:

- We introduce a novel experimental setup which uses DRAM PUFs to fingerprint AWS cloud FPGAs.
- We conduct the first fingerprinting experiments on cloud FPGAs, extracting unique and stable fingerprints of several Amazon f1.2xlarge, f1.4xlarge and f1.16xlarge instances.
- We propose a set of countermeasures against cloud FPGA fingerprinting.

Our code as well as pre-compiled AFI will be made available at https://caslab.csl.yale.edu/code/cloud-fpga-fingerprinting.

1. I. Giechaskiel, K. B. Rasmussen, and J. Szefer, "Measuring long wire leakage with ring oscillators in cloud FPGAs," in International Confer- ence on Field Programmable Logic and Applications (FPL), 2019.
2. I. Giechaskiel, K. B. Rasmussen, and J. Szefer, "Reading between the dies: Cross-SLR covert channels on multi-tenant cloud FPGAs," in IEEE International Conference on Computer Design (ICCD), 2019.
3. S. Tian and J. Szefer, "Temporal thermal covert channels in cloud FP- GAs," in ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA), 2019.
4. PaulJaccard.1901.ÉtudeComparativedelaDistributionFloraledansunePortion des Alpes et du Jura. Bulletin del la Société Vaudoise des Sciences Naturelles 37(1901), 547–579.
5. Wenjie Xiong, André Schaller, Nikolaos A. Anagnostopoulos, Muhammad U. Saleem, Sebastian Gabmeyer, Stefan Katzenbeisser, and Jakub Szefer. 2016. Run- time Accessible DRAM PUFs in Commodity Devices. In International Conference on Cryptographic Hardware and Embedded Systems (CHES).