HASP 2023

The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

DINAR: Enabling Distribution Agnostic Noise Injection in Machine Learning Hardware

Karthik Ganesan, Victor Kariofillis, Julianne Attai, Ahmed Hamoda, Natalie Enright Jerger

October 29, 2023

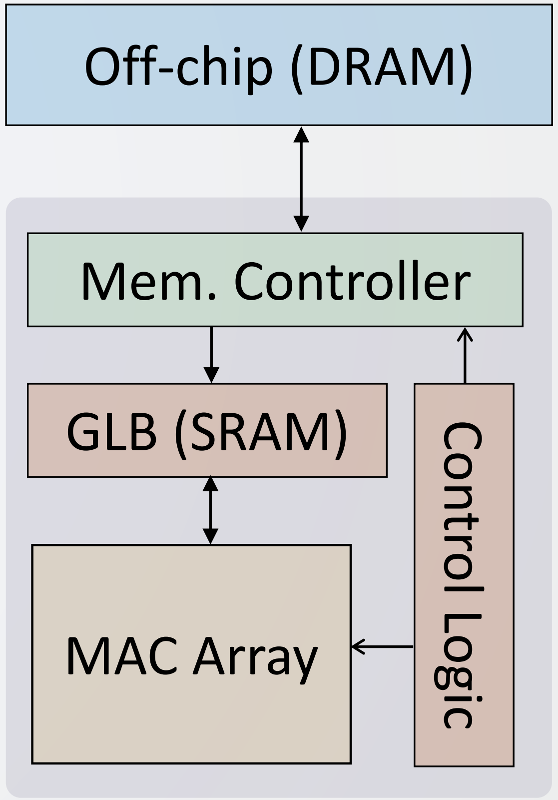
Executive Summary

- Security-centric ML algorithms require random noise, which current edge ML accelerators cannot provide.
- Existing hardware techniques for generating noise add significant overhead and leaks side-channel information.
- DINAR: light-weight hardware modifications to support noise addition.
- DINAR enables important ML algorithms while adding <0.5% area, energy and latency overheads.

Edge ML accelerators

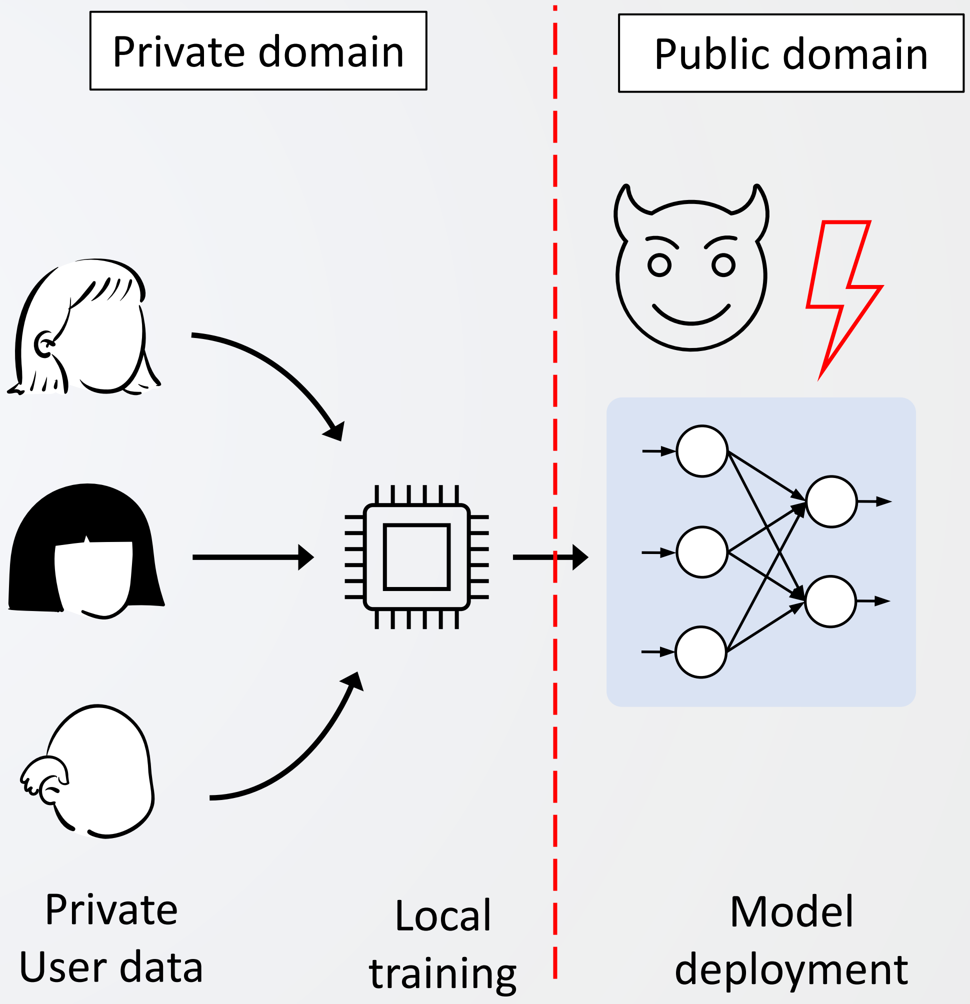
- ML is being deployed in devices ranging from cloud to edge.
- We focus on edge ML accelerators, as prior works show attacks against them.[1,2]
- Security-centric ML algorithms require **random noise.**
  - Current chips lack CPUs to provide noise.[3-5]
- Important algorithm that requires noise:

  Differentially private ML (DP-ML)



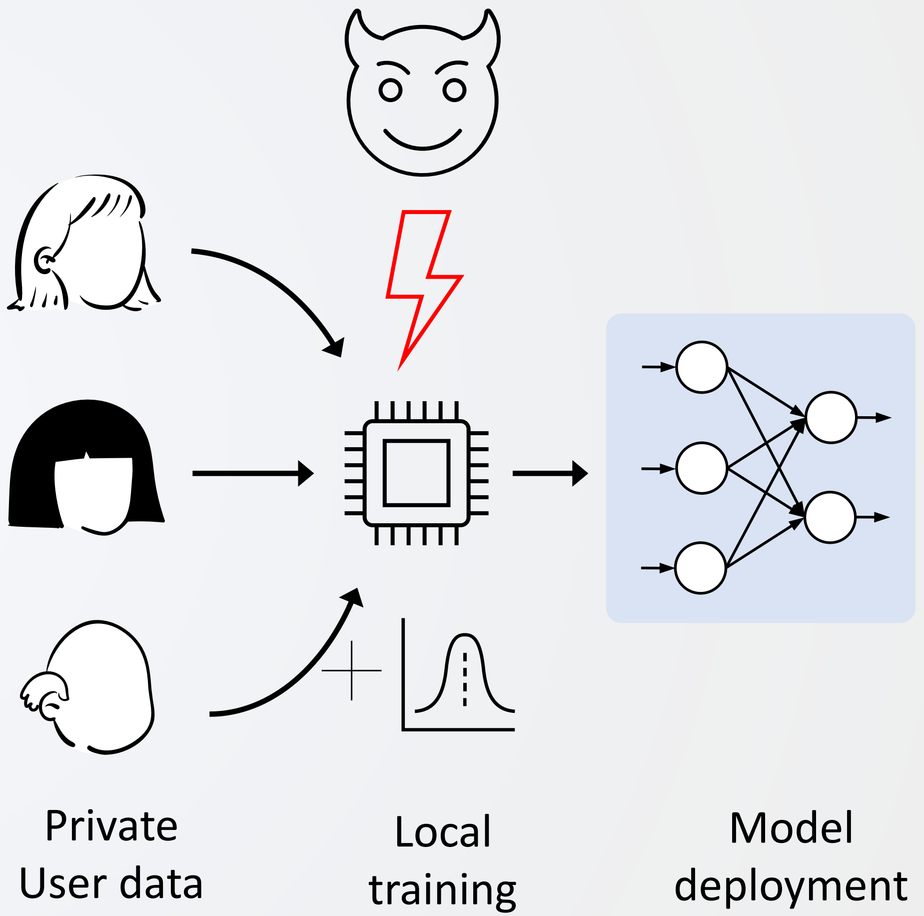
Regular edge training

- ❑ Avoids sending private user data to a central server and instead trains locally.
- ❑ The trained model is then released, while keeping user data private.
- ❑ However, even the trained model can leak private user data.[9]



Differentially private training

- DP-ML avoids this by adding a small amount of noise to each user's data.
- Most common is adding Laplace or Gaussian sampled noise.[10]
- However, learning the added noise can undermine security.[11,12]
- We focus on securely producing noise on-chip.

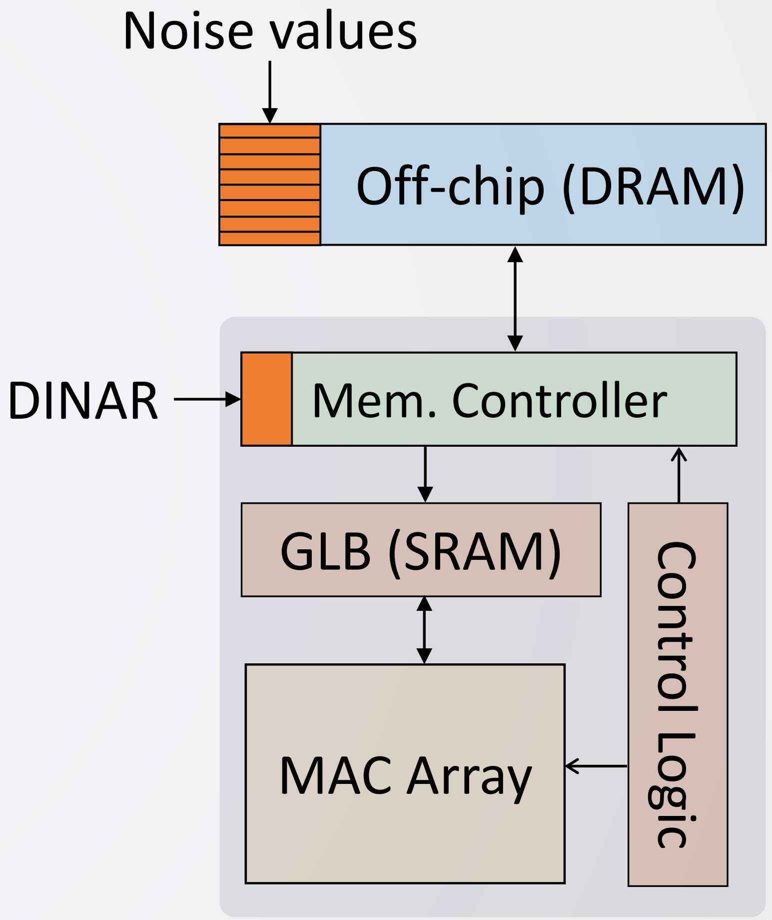


Hardware noise generation

- Many prior works have proposed techniques to generate Laplace and Gaussian noise in hardware.[13-19]
- However, these approaches suffer from several drawbacks:
  1. Use complex functions (e.g., Cos/Sin, ln, sqrt) which require lookup tables.
  2. Only produce fixed-point values and must be converted to floating point.
  3. Suffer from timing-side channels.[11,12]

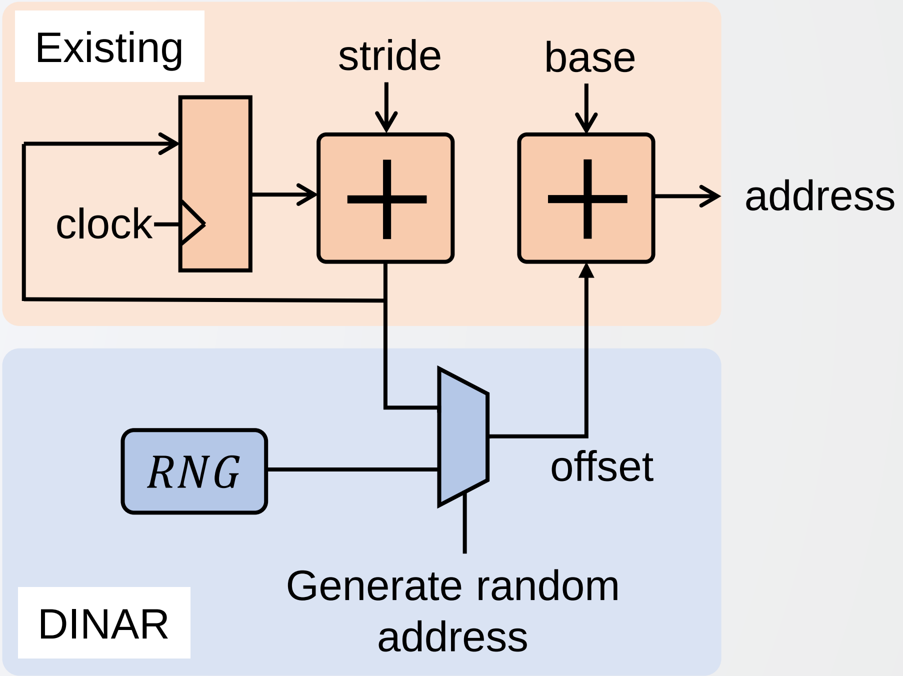
DINAR: Overview

- Pre-compute and store the noise points ahead of time in plentiful off-chip DRAM.
- Noise values are then loaded along with model weights from DRAM.
- Only requires changes to the on-chip DRAM memory controller.



DINAR: Hardware

- DINAR modifies the DRAM address generation logic
- DRAM address is typically calculated as a *base* + an *offset*
- For sequential reads, the *offset* is incremented by a *stride* each cycle
- For DINAR, we modify this hardware to support random reads



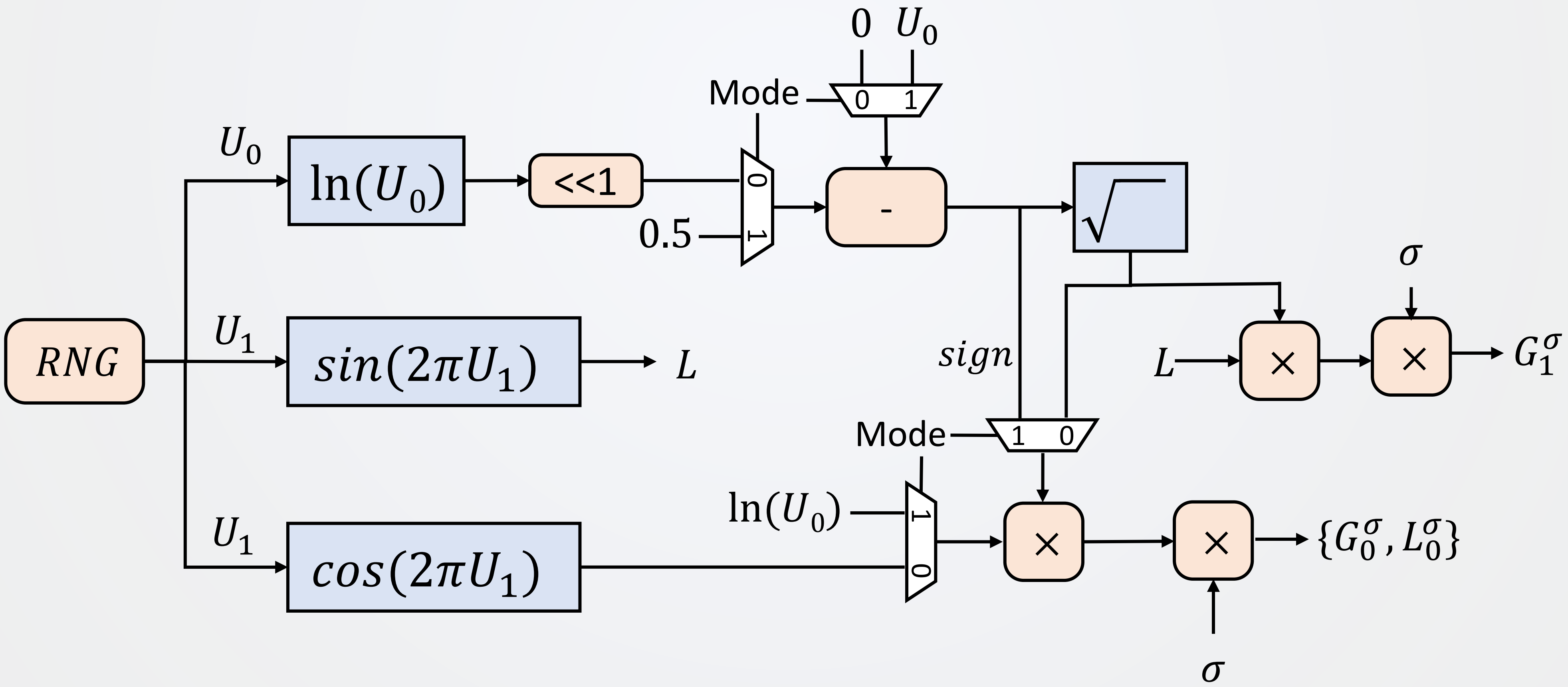
DINAR: Hardware Scrambling

- The datatype width is typically less than the bus width.
  - We read a fixed set of values each time.
  - Could lead to a loss of security.
- To increase randomness, we also include hardware to randomly scramble the values read.

Values in DRAM	A	B	C	D
Read 0	C	A	D	B
⋮				
Read N	D	C	A	B

Baseline implementation: NoiseGen

- Our baseline implementation of prior work to produce Gaussian and Laplace random numbers.



Methodology: Test setup

- ❑ We use DiVa [20] as our baseline accelerator.
  - ❑ DiVa has no support for adding noise.
- ❑ We evaluate two designs:

  DiVa-NoiseGen and **DiVa-DINAR**

Model	CIFAR-10	CIFAR-100
PreActResNet-18	✓	✓
WideResNet-32	✓	✓
VGG-16	✓	

- ❑ Model accelerators using Accelergy [21].
- ❑ Evaluate 3 models using two datasets, using the Opacus [22] library for PyTorch.

Evaluation

- Latency:
  - Both designs are optimized for performance so add <0.5% latency overhead.
- Area and Energy:

Metric	DiVA-DINAR	DiVA-NoiseGen	
Area overhead	0.4%	9.38%	23x ↑
Energy overhead	0.2%	8.15%	40x ↑

Evaluation: DRAM overhead

- ❑ DINAR requires storing noise points in DRAM.

No. of noise points	Storage (KB)	Footprint over smallest model
65536	128	0.56%
131072	256	1.11%
262144	512	2.23%
524288	1024	4.47%

- ❑ Even with 2^{19} points, we only add 5% overhead compared to our smallest model.

Conclusion

- Existing edge ML accelerators cannot run security-critical ML algorithms, as they lack CPUs.
- We present DINAR: light-weight hardware for using pre-computed noise points.
  - Demonstrate DINAR using differentially-private ML.
  - Also show DINAR for adversarial robustness in the paper.
- DINAR enables key algorithms while adding <0.5% area, energy and latency overheads.