Machine Learning and Hardware Security: A Winning Combo!

CoLA: Convolutional Neural Network Model for Secure Low Overhead Logic Locking Assignment

Yeganeh Aghamohammadi[1] and Amin Rezaei[2]

[1] University of California, Santa Barbara
[2] California State University, Long Beach

Great Lakes Symposium on VLSI (GLSVLSI) - 2023

Outline

1. Introduction
    1.1. Is logic locking a cat & mouse game?
    1.2. Where does machine learning come into play?
2. Objectives
    2.1. Research Gaps
    2.2. Contributions
3. CoLA
    3.1. Data Gathering & Labeling
    3.2. Data Augmentation
    3.3. CoLA Architecture
    3.4. Quantized CoLA

4. Experiments

5. Conclusion

1. Introduction

- Fabless manufacturing ⊂ Zero-trust environment
  - Challegne: Hardware IP piracy and overproduction
  - Solution (Thesis): Logic locking and obfuscation → The notion of “locking” via a “key”
  - Anti-solution (Antithesis): Attacks to retrieve the correct key or extract the original circuit

New solution (Synthesis)

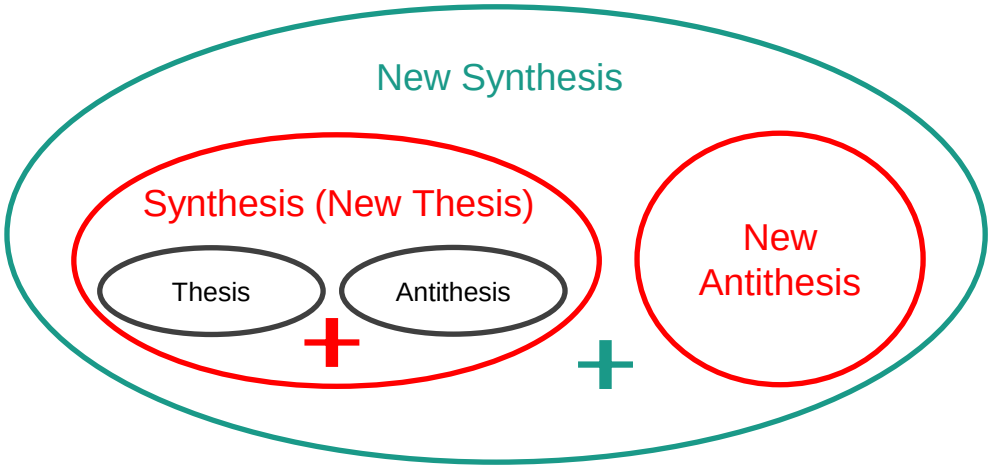
- Overhead vs. Security → There ain't no such thing as a free lunch!

1.1. Is logic locking a cat & mouse game?

It depends…

- ❑ Are there incremental works? → Yes
  - ➢ A problem in academic research
  - ➢ Must avoid incremental works
    How? By bringing in formalism and cryptographic foundations
- ❑ Are there sequences of attacks and defenses? → Yes
  - ➢ It is not necessarily a bad thing!

Moving from “particular” to “universal” based on the Hegelian dialectic.

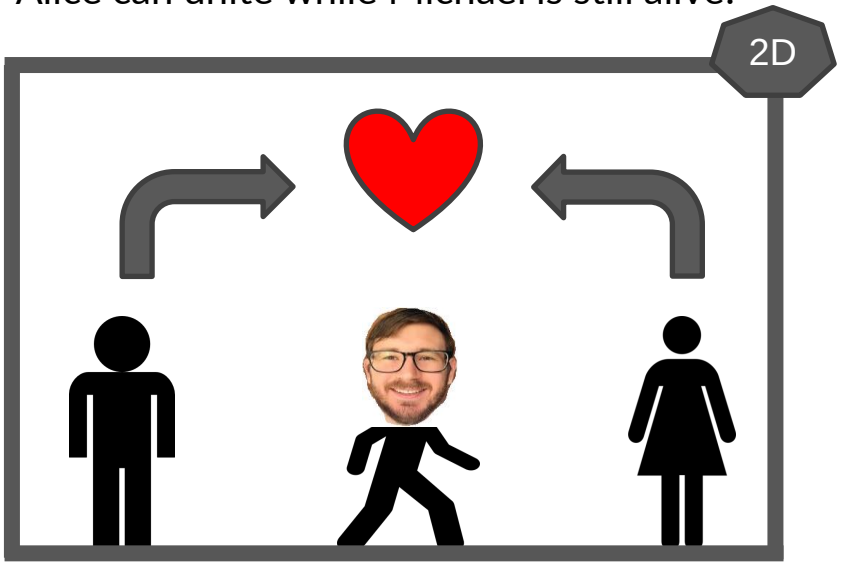


1.1. Is logic locking a cat & mouse game? (cont'd)

Alice & Bob Thesis: Love is above anything. But Michael is an obstacle between us. Let's kill and then eat him and unite.

Michael's Antithesis: Thou shalt not murder! Not murdering is above anything, and there is no way for Alice & Bob to unite.

Kaveh's Synthesis: Open your eyes! You are living in a 2D world, not a 1D one. Bob & Alice can unite while Michael is still alive!



1.3. Where does machine learning come into play?

- Training (Inductive Learning) → General (universal) rules are learned from specific (particular) historical data.
  - The realization of the Hegelian dialectic towards a universal thesis.
- Prediction (Deductive Inference) → Asking the general (universal) model to determine specific (particular) outcomes.
  - The opportunity to test (validate) the universal thesis.

2. Objectives

To answer the following questions:

- How secure is a logic-locked circuit?
  - An attacker can fail 99 defenses -and succeed in one- and win.
  - A defender fails if he or she stops 99 attacks but not the last one.
- How much overhead is imposed by locking?
  - It is counter-intuitive if the locking overhead is beyond a small amount.

2.1. Research Gaps

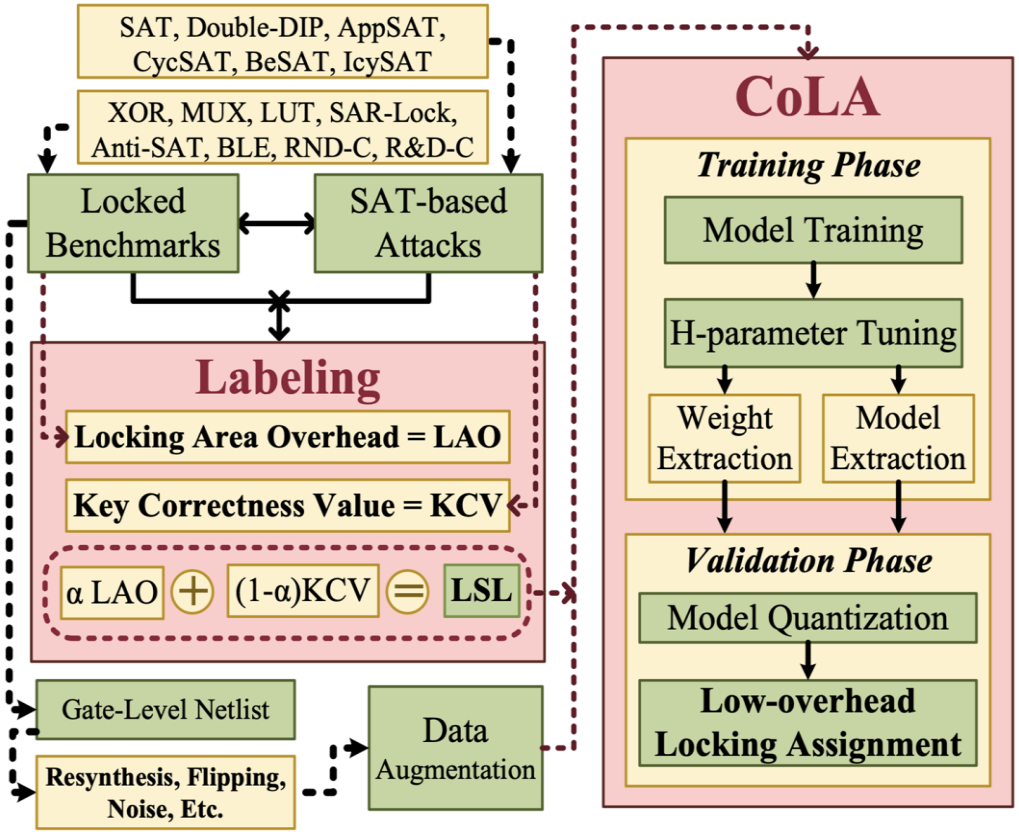
- Lack of multi-objective goals in logic locking research field. How much are you willing to pay for security?
- Most initiatives in ML + logic locking focus on the attacker side (antithesis). What about the opportunities that ML can provide on the defense side (thesis)?
- Lack of large enough datasets in logic locking to achieve the “Law of Large Number” required for ML models to become “universal”.

2.2. Contributions

- **The sublime goal:** To find a logic locking method that provides security against various attacks while minimizing overhead.
- Need for a comprehensive understanding of circuit structure to achieve the sublime goal.
- We created a measurement model for the sublime goal.
  - Developed a multi-label security and overhead degree dataset consisting of more than 10,000 benchmarks locked with 8 distinct logic locking methods;
  - Built and trained an accurate CNN-based ML model under 6 different attacks with hyperparameter tuning;
  - Tested the created CNN model on seen and unseen logic-locked benchmarks and evaluated the model's security and overhead prediction accuracy.

3. CoLA

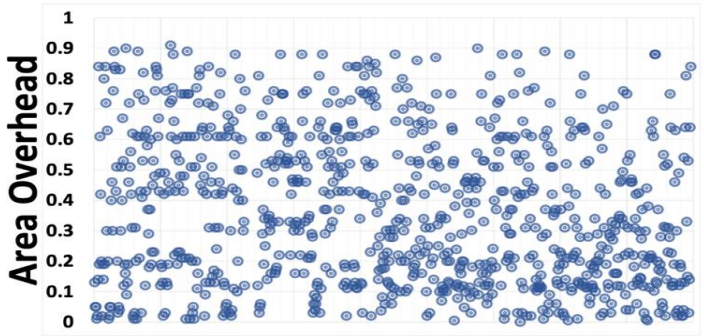
- Convolutional Neural Network (CNN) for Logic Locking Assignment
  - Data Gathering, Labeling, and Augmentation
  - Model Building and Training
  - Validation



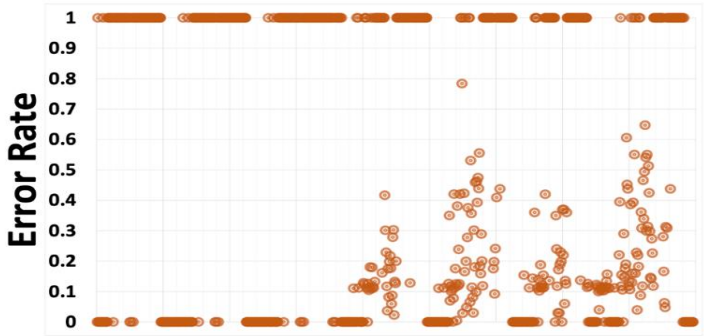
3.1. Data Gathering & Labeling

- One label to distinguish each locking method.
- One label to consider both security degree (KCV) and area overhead (LAO).

$$LSL = \alpha LAO + (1 - \alpha)\, KCV$$

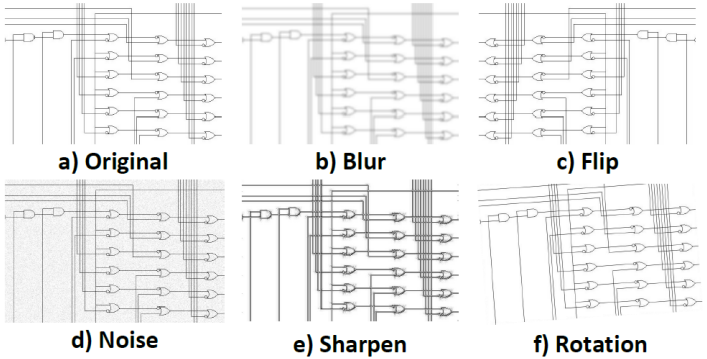


a) Area Overhead Distribution



b) Error Rate Distribution

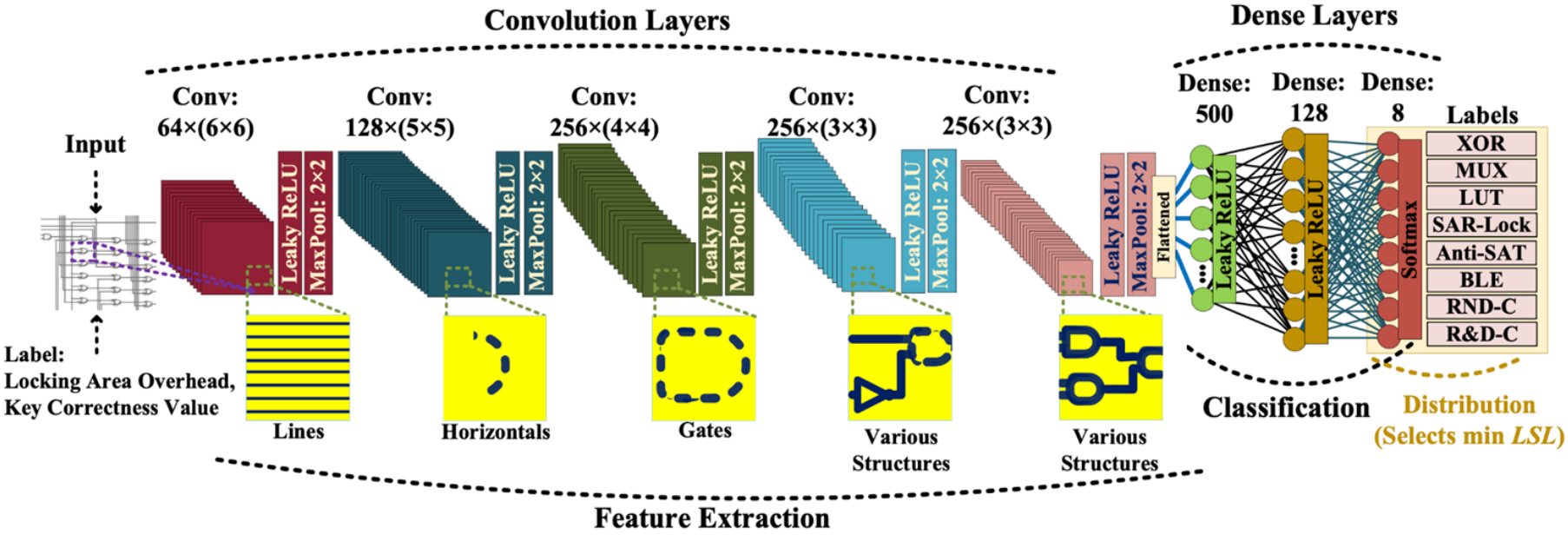
3.2. Data Augmentation



- Primary dataset: 240 locked benchmarks
- Keras[1]-only augmentation: 4560 data elements
  - Noise injection, random brightness, random flip, rotation, etc.
- All augmentation techniques: 10560 data elements
  - Keras + resynthesized layouts + different positions of the elements

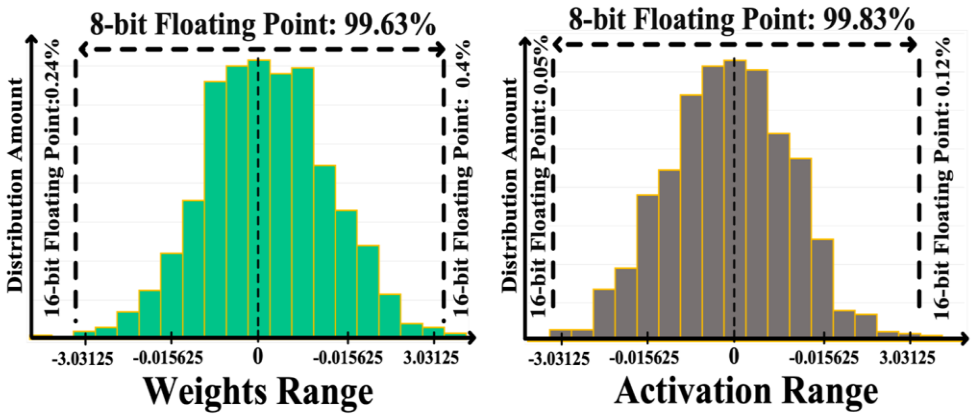
[1] https://keras.io/

3.3. CoLA Architecture



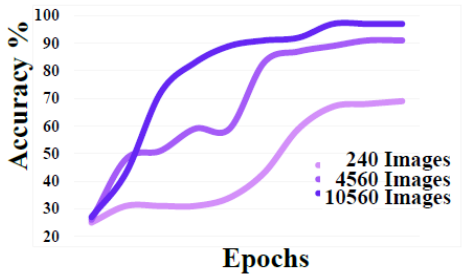
3.4. Quantized CoLA

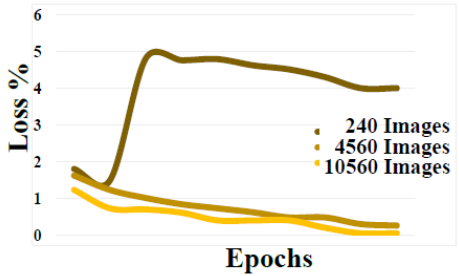
- Original CoLA for training → Offline training allows large model usage without constraints.
- Quantized CoLA for validation → Online validation requires memory usage reduction and computation speed-up.
- 8-bit quantized model can still achieve very high accuracy.



4. Experiments

- 1056 items of validation data
  - Quantized CoLA → 95.61% Accuracy
  - Original CoLA → 97.3% Accuracy





Bench	Overhead	Q time (ms)	R time (ms)	Pred. LSL	Same Label?
ex1010	5%	360	1179	Anti-SAT	Yes
ex1010	10%	173	612	Anti-SAT	Yes
c3540	25%	271	843	Anti-SAT	Yes
c7552	5%	149	577	Anti-SAT	No
c7552	5%	159	593	Anti-SAT	Yes
c1355	5%	124	541	SAR-Lock	No
c1355	10%	169	627	SAR-Lock	Yes
c3450	5%	173	663	R&D-C	Yes
c3540	10%	233	760	R&D-C	Yes
c7552	10%	207	827	R&D-C	Yes
ex1010	25%	268	873	BLE	Yes
c2670	5%	145	659	BLE	Yes
c6288	5%	142	736	BLE	Yes
c7552	25%	186	619	BLE	Yes

5. Conclusion

- A defensive perspective on ML + logic locking
- Dataset: Over 10,000 image circuits
- CoLA: Comparative model for logic locking security and overhead analysis

Thank You For Your Attention!