Project 1: Robust Deep-learning-based Side-Channel Attacks

Channing Smith, College of Charleston
Joel Ward, Cedarville University

Chenggang Wang, University of Cincinnati

Mentors: Dr. Boyang Wang, Dr. Marty Emmert

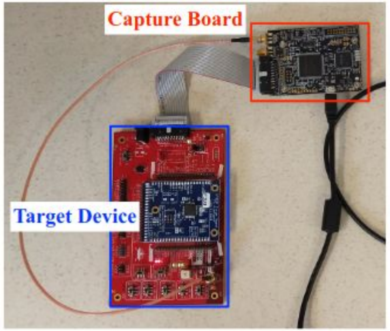
University of CINCINNATI

RHEST

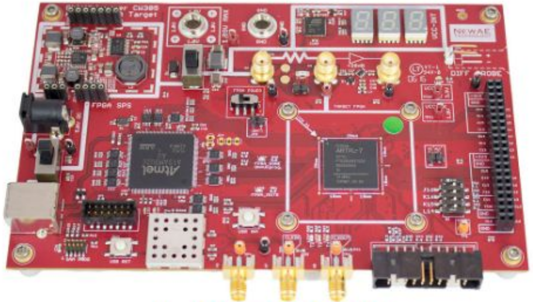
Side-Channel Attacks (SCA)

- An attacker analyzes power or electromagnetic (EM) signals of a target (microcontroller or FGPA) when it runs encryption algorithm (e.g., AES) and recover encryption keys

- Why? power consumption is correlated with the value processed by target
  - 0x00 requires less power than 0xFF

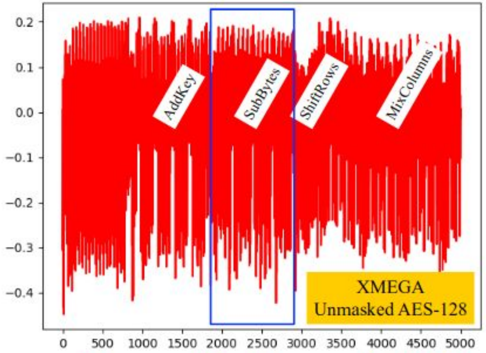


Arm STM32F3



Artix-7 FPGA

Attack Window [1800, 2800]

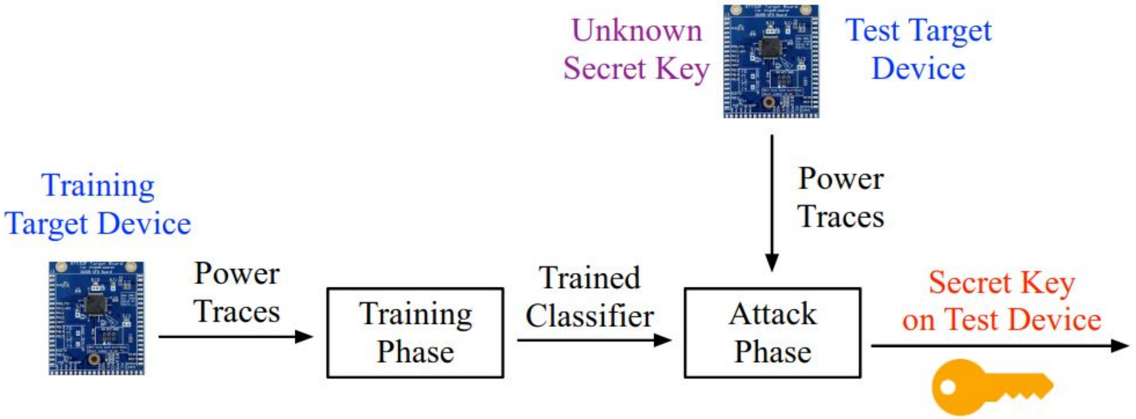


Power Pattern of AES

Deep-Learning SCA

- Advantages compared to traditional SCA attacks
  - No need to pre-process traces
  - Can defeat existing countermeasures (masking & random delays)

- High accuracy (>90%) in the **same-device setting**
  - Train with device A, test with device A



RHEST

Large number of traces

Device A

Unknown distribution shifts (hardware imperfection, setup, etc.)

Challenge:
Limited number of traces

Device B

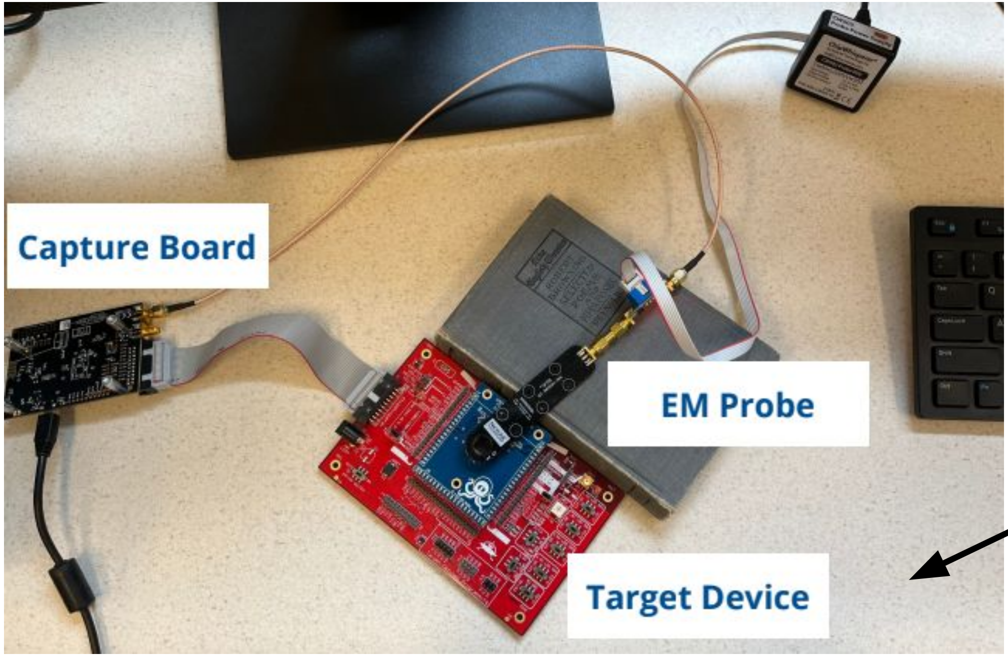
- Poor performance (<10% accuracy or fail to recover keys) in **cross-device setting** (a real-world attacker)
  - Train with device A, test with device B

- Challenges: (1) Limited traces from Device B; (2) unknown key from Device B; (3) complex discrepancies caused by hardware and software

Objectives

- Task 1: Collect EM traces on microcontrollers and test results with our existing ML code
- Task 2: Study instruction rewriting in assembly on AVR XMEGA and ARM STM32 as well as examine the impact of instructions rewriting in deep learning side channel attacks
- Task 3: Collect EM traces of AES encryption compiled with different optimizations and study the optimizations’ effects

EM Data Collection Setup



Either XMEGA or STM32

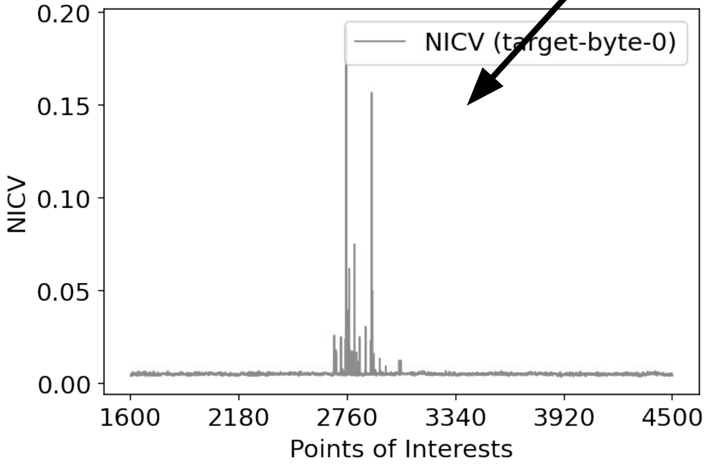
EM Data Collection

XMEGA	STM32
- 50k unmasked AES, PC1 - 50k unmasked AES, PC2 - 50k masked AES, PC1 - 50k masked AES, PC2	- 50k unmasked AES, PC1 - 50k unmasked AES, PC2

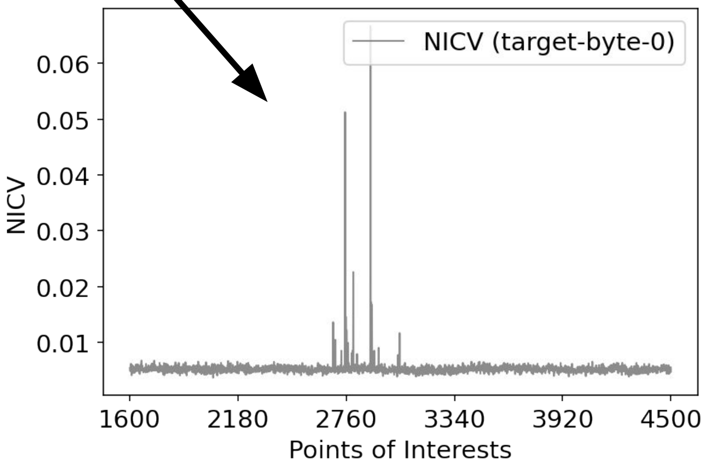
EM Data Analysis

- Performed Normal Inter-Class Variance (NICV)

Shows leakage within defined attack window of [1600, 4500]



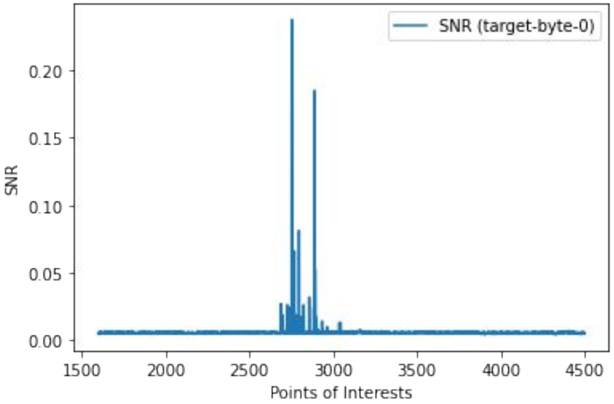
NICV results from XMEGA masked, PC1



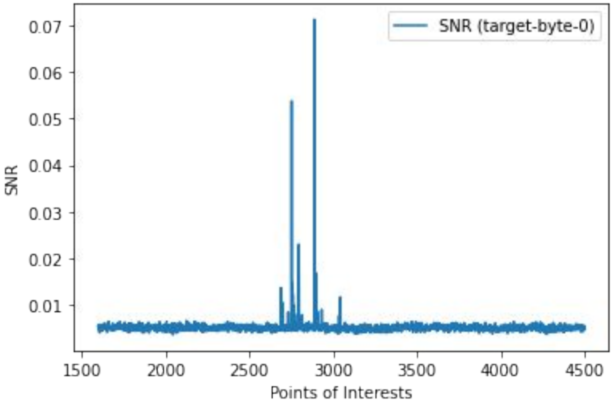
NICV results from XMEGA masked, PC2

EM Data Analysis

- Performed Signal to Noise Ratio (SNR)



SNR results from XMEGA masked, PC1



SNR results from XMEGA masked, PC2

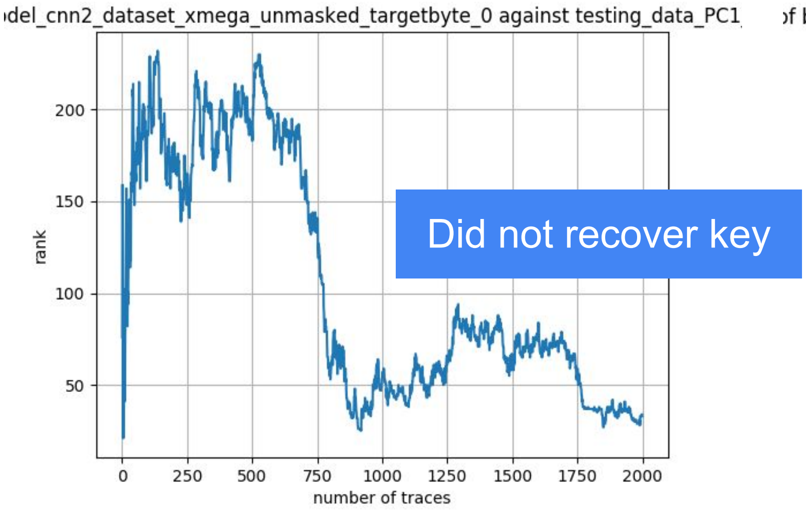
- Ran CPA attack

```
Key guess:   0xc6
Correlation:   0.25582897783658465
Correct Key:   0xc6
```

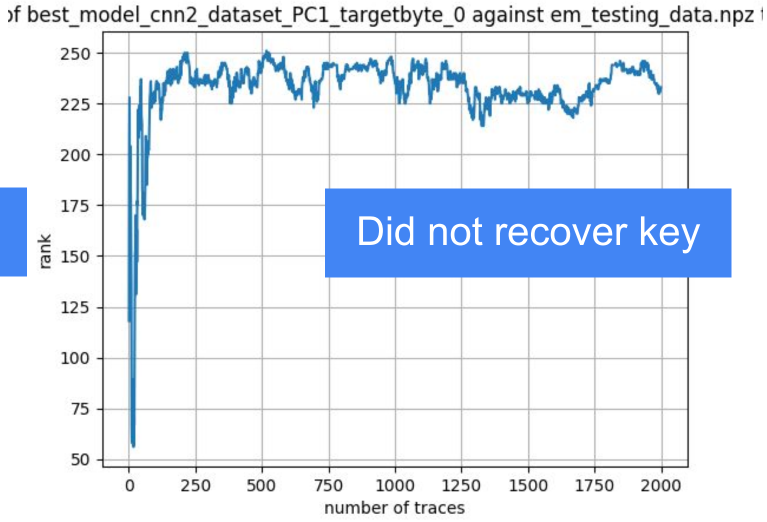
```
Key guess:   0x70
Correlation:   0.11420753433103399
Correct Key:   0x70
```

EM Data Convolutional Neural Network (CNN) Results

- Train and test data using Convolutional Neural Network (CNN)
  - For cross-device scenario, use 40k for training (PC1) and 10k for testing (PC2)



Same-device results from XMEGA unmasked



Cross-device results from XMEGA masked

Working with EM Data

- Collecting EM data is much more difficult than power traces. (And results produced by CNN are not always promising even with same-device)
- Improved data collection process would benefit the data, as it is easy for the EM probe to move positions during the collection.
- Although CNN did not always show us the results we were hoping for, we did get a lot of promising pics from NICV and from the CPA attack which was able to recover most keys.

Instruction Rewriting

- Causes software discrepancy
- Train with masked AES, test with rewritten AES
- Rewrote lines of assembly code with 1-3 comparable lines
- Focused on SubBytes and addRoundKey routines
- 24 lines rewritten

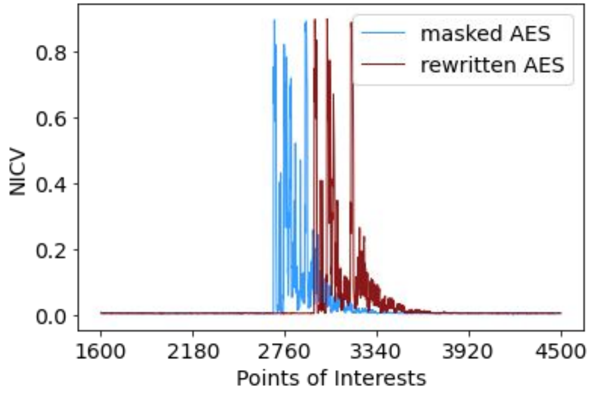
CLR r19 → EOR r19, r19

MOVW r26, r22 → EOR r26, r26
ADD r26, r22

Power Trace Data Collection

- Collect 50k masked AES power traces
- Collect 50k rewritten masked AES power traces
  - 40k traces for training
  - 10k traces for testing
- Run NICV and CPA

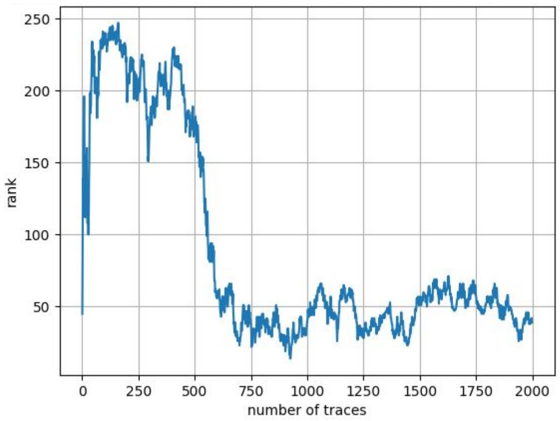
NICV results




```
Key guess:   0x2b
Correlation:   0.9446525225032248
Correct Key:   0x2b
```

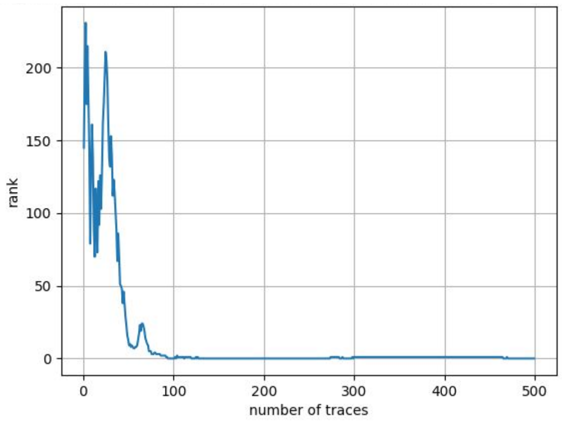
CPA results for masked AES dataset

Did not recover key



attack window: [1600,4500]

Recovered key



attack window: [1900,4800]

EM Data Collection with Optimization

- Optimization → software discrepancy
- Compiled with either o1, o2, or o3 optimization in gcc

- Gcc command before optimization:

  make PLATFORM=CWLITEXMEGA CRYPTO_TARGET= TINYAES128C

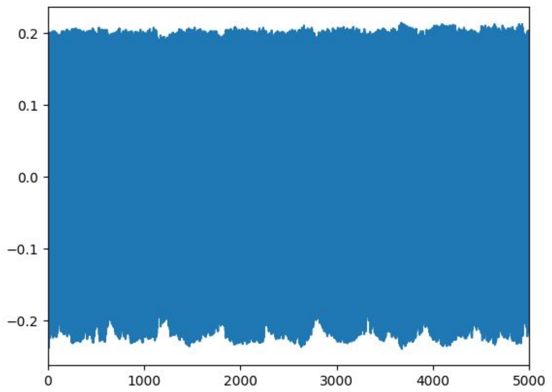
- Gcc command after optimization (o1):

  make PLATFORM=CWLITEXMEGA CRYPTO_TARGET= TINYAES128C **OPT=1**

RHEST

EM Data Collection with Optimization

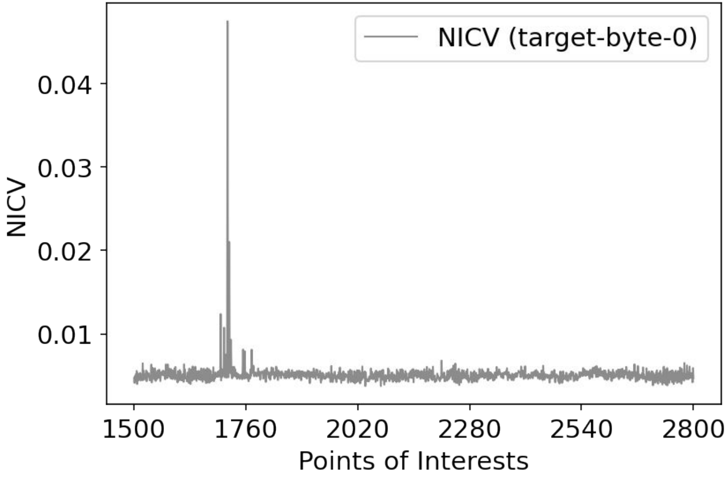
- Collected 4 50k EM datasets:
  - XMEGA masked, (modified with instruction rewriting), PC2
  - XMEGA unmasked, (compiled with o1 optimization in gcc), PC2
  - XMEGA unmasked, (compiled with o2 optimization in gcc), PC2
  - XMEGA unmasked, (compiled with o3 optimization in gcc), PC2

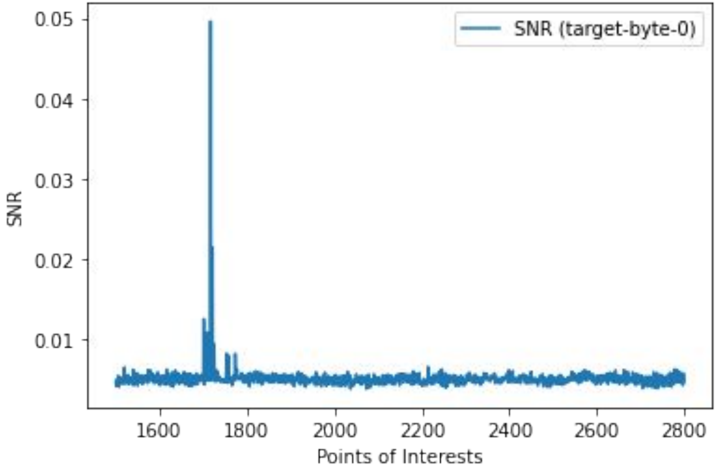


First 5,000 traces from dataset compiled with o1 optimization

EM Data with Optimization Analysis

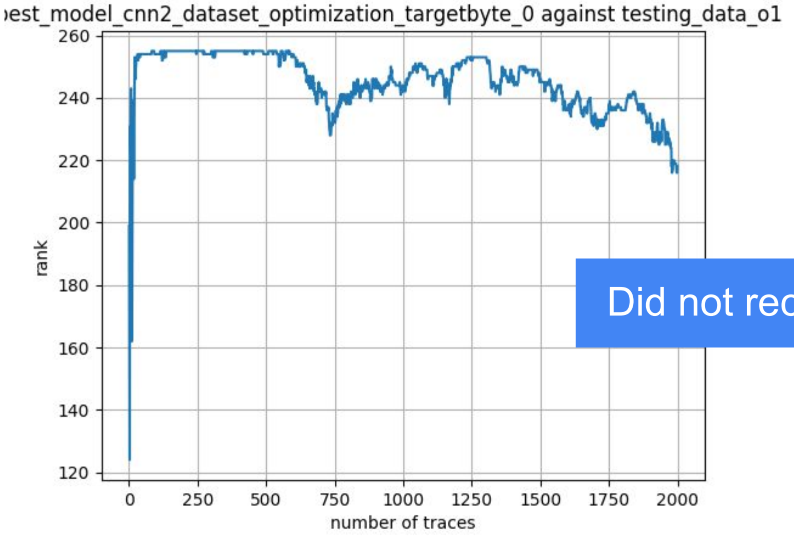
- Performed Normal Inter-Class Variance (NICV), Signal to Noise Ratio (SNR), and CPA attack on XMEGA unmasked EM dataset compiled with o1, o2, and o3 optimization in gcc.





EM Data with Optimization Results (CNN)

- Trained (40k) and tested (10k) EM dataset compiled with o1 optimization in gcc on Convolutional Neural Network (CNN) on same-device scenario.



Did not recover key

Limitations & Challenges

- Pre-Data Collection (being able to run required scripts).
- ChipWhisperer has a limited number of integrated AES implementations.
- EM datasets are noisy, which oftentimes doesn’t show promising results.

RHEST

Future Direction

- Instruction rewriting STM32
- Analyzing Trojans on FPGA’s
- Transfer learning with datasets
- Improving data collection process of EM data
- EM data collection and analysis of STM32 masked

Thank you!

- Collected 10 EM datasets and 2 power datasets used for instruction rewriting
    - 700k power and EM traces
    - 52 gb of data
- GitHub link: https://github.com/UCdasec/CrossSide

This work is supported by National Science Foundation (NSF), CNS-2150086, RHEST: NSF REU Site In Hardware and Embedded Systems Security and Trust