EE / CprE / SE 491 – sdmay21-09

Instruction Level Reverse Engineering through EM Side Channel

Week 6 Report

3/29/2021 – 4/12/2021
Client: Akhilesh Tyagi
Faculty Advisor: Akhilesh Tyagi

Team Members:

Noah Berthusen — *Data Analysis Engineer*

Matthew Campbell — *Test Engineer*

Cristian George — *Meeting Scribe*

Jesse Knight — *Signals Processing Engineer*

Evan McKinney — *Integration Engineer*

Jacob Vaughn — *Report Manager*

Weekly Summary

We completed our data collection of a new dataset which used Python code to generate assembly code using random permutations of 2 instructions over 12 clock cycles in order to create a set of training data. For the first set of data we used 10,000 repetitions of 2000 samples of add and asr instructions running in random order for 12 clock cycles. For the second set of data we used 10,000 repetitions of 2000 samples of add and mul instructions. The Machine-Learning task is to label a sample as either containing add and asr or containing add and mul. We struggled to find a way to move from single instruction classification to multiple instruction classification, therefore we hoped that if we could diversify the wave to have many versions of what these instructions look like at different parts of the pipeline, our classifier would be able to pick up on those patterns. The problem with this approach is that it is not very scalable since we can't train every combination of every instruction. We experimented with using precision to split the wave into different clock cycles but struggled with this due to some issues with the assembler optimizing parts of the GPIO triggers. Machine learning still requires progress, which is coming along but a slow process because the large files take a long time to train against.

Past Week Accomplishments

- More data collection

- Data training attempts
    - Multilabel classification on standard and Fourier transformed data
    - Binary relevance, classifier chains, powerset classification.
    - Results not too great; could be due to poor data (as discussed later) or too complex of a problem for a simple Scikit-Learn model.

Pending Issues

- Time series data is being shifted a few clock cycles while being recorded, resulting in data that is not consistent in our data sheets and for machine learning.
    - High level GPIO interface may be the cause
    - Possible fix is using GPIO via in-line assembly

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Noah Berthusen	Research and experimentation with multilabel classification	6	29
Matthew Campbell	New data set & matlab Code	2	20
Cristian George	New data set case statement generation	2	23
Evan McKinney	Machine Learning on multilabel dataset	3	26
Jacob Vaughn	Case statement generation	2	20
Jesse Knight	Took new data set Finalized Matlab Code	2	32

Plans for Coming Week

- Cristian, Jake: Embedded changes
    - Modify embedded code to trigger GPIO using ASM rather than HAL C functions
        - New code will hopefully fix time series discrepancies in data
- Jesse, Matt: Data
    - Collect new dataset based on meeting with Dr. Tyagi
- Machine Learning Team
    - Train a new model using fixed time series data based on suggestions from Dr. Tyagi.

Summary of weekly advisor meeting (If applicable/optional)

We met with Varghese and Tyagi on Monday 4/12 to discuss the current state of the project. Discussed how current data might be too inconsistent for accurate machine learning and tried to determine the cause of the inconsistency. Proposed the idea that the GPIO trigger could be shifting time-series data a few clock cycles, leading to differently looking datasets. Dr. Tyagi suggested we set the GPIO using in-line assembly rather than higher-level C functions to avoid optimization. Additionally he suggested that with the new data

that will be collected trained to create time-based cascading classification models.