Instruction Level Reverse Engineering (Disassembly) through EM Side Channel

Design Document

Team Number: 09
Client: Prof. Akhilesh Tyagi
Advisers: Varghese Vaidyan

Team Members:
Matthew Campbell
Noah Berthusen
Cristian George
Jesse Knight
Jacob Vaughn
Evan McKinney

sdmay21-09@iastate.edu
https://sdmay21-09.sd.ecec.iastate.edu

Revised: 10/02/2020

Executive Summary

Development Standards & Practices Used

- Agile Scrum model used
- IEEE UART protocols
- Python PEP 8 Style
- Arduino Style Guidelines
- IEEE Code of Ethics

Summary of Requirements

- Program collects EM data and converts it to a usable format
- Model will predict opcodes with 90%+ accuracy and operands with 80%+ accuracy.
- Written in Python
- Well-documented code
- Predictions are formatted in a user-friendly format
- Large amount of data used to train the model

Applicable Courses from Iowa State University Curriculum

- COM S 311 (Introduction to the Design and Analysis of Algorithms)
- COM S 474 (Introduction to Machine Learning)
- CPRE 288 (Embedded System I)
- CPRE 381 (Computer Organization and Assembly Level Programming)
- CPRE 482x (HW Design for Machine Learning)
- EE 224 (Signals and Systems I)
- EE 321 (Communication Systems I)
- EE 201 (Electrical Circuits)
- EE 230 (Electronic Circuits and Systems)

New Skills/Knowledge acquired that was not taught in courses

- Convolutional neural networks (CNNs), Markov chains
- Side-channel observation of processors

Table of Contents

1 Introduction

1.1 Acknowledgement

We would like to thank Vhargese Vaidyan for sharing his knowledge about working with the EM side channel for reverse engineering, and for sharing his equipment and experience with us in order to jump start the project.

1.2 Problem and Project Statement

General problem statement:

We want to be able to determine the assembly level code that is currently running on a processor by only reading the electromagnetic radiation that comes off of the processor. This kind of research has cyber security implications in that you could bypass a lot of security if you could find out what code is running just by measuring the physical electromagnetic (EM) radiation that the processor gives off.

General solution statement:

Our solution is to capture data using an electromagnetic probe and send that data to a machine learning algorithm. The machine learning algorithm will be able to look at the data and the surrounding data points to determine with a degree of certainty what opcode and operand is being ex

1.3 Operational Environment

The resulting end product from this project will be used in a laboratory environment with minimal electromagnetic interference. At the moment, the operational environment will be in 301 Durham. 301 Durham has an oscilloscope and EM antenna powerful enough to capture the radiation emitted by our microcontroller.

1.4 Requirements

Requirements:

- Program collects EM data and converts it to a usable format
- Model will predict opcodes with 90%+ accuracy and operands with 80%+ accuracy.

Non-functional requirements:

- Written in Python
- Well-documented code
- Predictions are formatted in a user-friendly format
- Large amount of data used to train the model

1.5 Intended Users and Uses

Our single intended use is to measure EM radiation from a Cortex Arm M4 processor and output the corresponding opcodes and operands. The intended user for our project is our client Akhilesh Tyagi and other EM side-channel researchers.

1.6 Assumptions and Limitations

Assumptions:

- End users will have access to the necessary hardware and software
- Design is running on a Cortex Arm M4 processor with a 4 stage pipeline running at 200 MHz
- EM is measured using same tools used to train the machine learning model

Limitations:

- Program will need to run on a high-end GPU
- Input data to model must be in a specific format
- Processor must have at least a 4 stage pipeline

1.7 Expected End Product and Deliverables

1. A machine learning algorithm capable of 90% opcode, and 80% operand detection:

   The machine learning algorithm will be delivered at the end of the project, approximately May 2021. The algorithm will be created using python and will include the datasets used to test, train, and validate the algorithm. Additionally, the algorithm will include documentation describing how to redeploy the algorithm to a separate system.

2. Automated data extraction tool to pull data from EMR probe.

   The extraction tool will take data from either the oscilloscope or digitizer and convert it into a format usable by the machine learning algorithm. This tool will allow the client to easily extract additional data to run against the trained algorithm. We expect to have this data extraction tool to be completed by November 2020, but will be delivered to the client with the machine learning algorithm in May 2021.

2 Project Plan

2.1 Task Decomposition

Hardware:

We will need to set up some interface to probe the EM radiation from the processor which will need to be precise and consistent in order to get as clean data as possible. Also we should set up some way to control, monitor and save the processor's executing instructions.

subtasks: modify existing arduino code and matlab file to be compatible with Nucleo board ISA, for automating data collection

Design a structure for the probe and processing board to be mounted on for consistent measuring.

Collect many instances of single opcodes being executed on our processor.

subtask: data with varying operands

Collect many instances of multiple opcodes being executed on our processor.

subtask: data with multiple varying operands

Create an interface between processor chip EM radiation measurements and trained ML model for live instruction recognition.

Software:

We need to perform machine-learning experimentation such that we understand what algorithms work best for our data. Including techniques that include some form of recall to understand signals mutating and persisting as they travel through pipeline stages.

We will need a working classification model for opcode prediction when data only includes a single instruction at a time in our processor pipeline.

subtask: create file that structures classification model, can input data for training and output predictions, but not ready to train effectively yet

subtask: polish file such that utilizes the more effective processing and training algorithms

Then we want to use advanced data processing techniques and automate data exportation with multiple instructions in the pipeline.

subtask: be able to save trained model and load in separate runtime environment to calculate performance analysis

2.2 Risks And Risk Management/Mitigation

For each of the tasks in our project each one comes with an associated risk. Things may not go as well as planned or tools may fail to function. Listed below are some of the risk and risk factors of each of the tasks.

Signal Capturing: Risk factor 0.55
Instruments may not be sensitive enough to detect 90% of the opcode and operands. Because of the high risk factor multiple EM antena need to be available for purchase.

Data Conversion: Risk factor 0.2
Risks include conflicts with interfacing sensor output to python.

Filtering Through Machine Learning: Risk factor 0.5
Inability of chosen machine learning technique to properly filter between valid and invalid data. This risk can be mitigated by having an understanding of multiple ML techniques and algorithms.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Observation of Operand and opcode with EM radiation probe with stable and consistent output.

- Successful separation of waveforms for multiple operations in pipeline

Interface between EM radiation probe and software.

- Completely modified arduino and matlab code for Nucleo board compatibility

Collection and identification of many instances of multiple opcodes being executed on the processor.

- Waveforms exported into basic format (Excel) with opcode instruction along with accompanying EM characteristics

Interface between processor chip EM radiation measurements and trained ML model

- Collected data converted into format for ML
- Process of data collection -> ML format can be completed automatically

Working classification model for opcode prediction

- 90%+ accuracy in opcode detection

Working classification model for opcode AND operand prediction

- 90%+ accuracy in opcode detection
- 80%+ accuracy in operand detection

2.4 Project Timeline/Schedule

2.5 Project Tracking Procedures

To track progress the team will be using GitLab's issue tracking feature. This feature creates a kanban-like board that allows for the assignment of tasks and progress tracking for groups and individuals. Additionally, the team has also chosen to use Slack for IM communications and WebEx teams for voice/video communications.

2.6 Personnel Effort Requirements

Task	Total-Person Hours	Explanation
Data Collection Interface	50	Creation of EM capture and waveform separation method.. Includes validation of signal data after using technique.
Automation Code	10	Automation of data collection technique created, testing of correct functionality.
Mount	5	3D Modelling and Print Time

Single Ops Data	30	Identification of Ops
Varying Operands Single Ops	10	Runthrough and identification of all relevant Ops
Multiple Ops Data	40	Identification of operations in a pipeline, differentiation between consecutive Ops
Varying Operands Multiple Ops	10	Generate data with randomized instruction operands for data variance
EM to ML Interface	40	Set up automation code to connect processed probe readings to trained model for live classification
ML Experimentation	80	Study advanced ML techniques and practice applications on other datasets
Classification Model for Ops	30	Train Models to classify Ops
Classification Ops Framework	30	Design code framework that sets up basic interface to ML packages
Classification Ops Improvement	30	Work on implementing advanced techniques to improve prediction
Data Export Automation	30	Automate data processing on instructions with multiple instructions and varied operands
Trained Model Export	5	ML Package save to file and performance statistical analysis
Total	400	42: answer to life, the universe and everything

2.7 Other Resource Requirements

- STM32H7 Nucleo-144 (MB1364) board
- Power supply, capable of powering above-mentioned microcontroller
- TBPS01 Electromagnetic Probe

- Digital Oscilloscope/Digitizer with a bandwidth of at least 400MHz

2.8 Financial Requirements

- Up to $70 to purchase another microcontroller

3 Design

3.1 Previous Work And Literature

Include relevant background/literature review for the project

– If similar products exist in the market, describe what has already been done

– If you are following previous work, cite that and discuss the **advantages/shortcomings**

– Note that while you are not expected to "compete" with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

3.2 Design Thinking

Detail any design thinking driven design "define" aspects that shape your design. Enumerate some of the other design choices that came up in your design thinking "ideate" phase.

3.3 Proposed Design

Include any/all possible methods of approach to solving the problem:

- Discuss what you have done so far – what have you tried/implemented/tested?
- Some discussion of how this design satisfies the **functional and non-functional requirements** of the project.
- If any **standards** are relevant to your project (e.g. IEEE standards, NIST standards) discuss the applicability of those standards here
- This design description should be in **sufficient detail** that another team of engineers can look through it and implement it.

3.4 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

3.5 Design Analysis

- Did your proposed design from 3.3 work? Why or why not?
- What are your observations, thoughts, and ideas to modify or iterate over the design?

3.6 Development Process

Discuss what development process you are following with a rationale for it – Waterfall, TDD, Agile. Note that this is not necessarily only for software projects. Development processes are applicable for all design projects.

3.7 Design Plan

Describe a design plan with respect to use-cases within the context of requirements, modules in your design (dependency/concurrency of modules through a module diagram, interfaces, architectural overview), module constraints tied to requirements.

4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or software.

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study or acceptance testing for functional and non-functional requirements).
2. Define/identify the individual items/units and interfaces to be tested.
3. Define, design, and develop the actual test cases.
4. Determine the anticipated test results for each test case

5. Perform the actual tests.
6. Evaluate the actual test results.
7. Make the necessary changes to the product being tested
8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you have determined.

4.1 Unit Testing

– Discuss any hardware/software units being tested in isolation

4.2 Interface Testing

– Discuss how the composition of two or more units (interfaces) are to be tested. Enumerate all the relevant interfaces in your design.

4.3 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

4.4 Results

– List and explain any and all results obtained so far during the testing phase

- Include failures and successes
- Explain what you learned and how you are planning to change the design iteratively as you progress with your project
- If you are including figures, please include captions and cite it in the text

5 Implementation

Describe any (preliminary) implementation plan for the next semester for your proposed design in 3.3.

6 Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc,. PCB testing issues etc., Software bugs etc.