Visualizing and Interacting with Physically Unclonable Function Experiment Data

Jason Pereira
jperei5@uic.edu
Department of Computer Science
University of Illinois at Chicago

October 4, 2022

Client Details

Interviewer: Jason Pereira (Graduate Student at the Department of Computer Science at UIC)
Client name: Professor Wenjing Rao (Associate Professor at the Department of Electrical and Computer Engineering) and Yeqi Wei (PhD researcher at the Department of Electrical and Computer Engineering)
Location: Zoom
Duration: 1 hour
Date October 3, 2022

Abstract

This paper proposes a tool to visualize and analyze experiment data of physically unclonable functions. This tool can help its users gain new insight and catch patterns that haven't been seen before from the data.

Introduction

With the rise of mobile devices and the Internet of Things, the number of small, moblie devices in the world is increasing. Most of these devices in the field communicate with a server to both send and recieve data. For security purposes, the server requires that any device connecting to it be autheticated prior to establishing the connection. Currently, public key cryptography is the state of the art standard for secure communication between 2 devices. Public key cryptography however, requires that each communicating party have a private key, which is known only to the party itself. In the context of mobile devices, this is implemented by embedding the private key of each device in the device's ROM. This approach is expensive both in terms of design area and power consumption. Additionally, ROM can be cloned, in which case the entire system is comprimised.[1]

Physical unclonable functions (PUFs) are a promising new idea that are used for authentication and secret key storage without the requirement of secure EEPROMs and other expensive hardware.[1] PUFs derive a secret from the physical characteristics of the integrated circuit. Because of intrisic defects and variablity in the manufacturing process of ICs, no two dervied secrets will be the same. It is also currently impossible to clone hardware at the level required to mimic these defects, and thereby the derived secret.

This paper proposes a tool to visualize experiment data of PUFs. The chief entities are:

- Response matrix R
- Challenge matrix C
- Response Bias $B(C)$ over C

- Uniqueness for a pair of PUFs x and y which is the hamming distance between x and y
- Uniqueness for a batch M of PUFs

Requirement Analysis

Humans

- PhD students and Prof. Wenjing Rao and her research group.
- PhD students from the ECE department
- Anyone from outside UIC may use and play with the tool if we decide to post it online.

Tasks

- Given a response matrix does any row have more than 50% similarity?
- Given a response matrix, interchange any 2 rows and view the result.
- Given a response matrix, interchange any 2 columns and view the result.
- Select a submatrix S of a PUF matrix and input a function F, what is the value of $F(S)$?
- How different is PUF x from PUF y?
- Visualize a 100×1000 (or more) binary matrix on screen.
- Visualize a 100×1000 (or more) real-valued matrix on screen.
- Visualize the function $\Delta_n(c)$ as a histogram.
- How different is each PUF from every other PUF?
- Visualize/compute Reliability of PUFs
- Visualize/compute Bad PUF vs. good PUFs

Data

The users use random sampling APIs of Python to generate data to work with. Results of experiment data are stored in Excel sheets. Currently, the users have a myriad number of Excel sheets of different formats for data of each experiment. Therefore, the tool must be atleast capable of reading data from CSV files.

There also exists a use case in which the user may want to:

1. Generate a sample of random data directly in the tool itself
2. Integrate the tool with a Python runtime so that the user can use their existing data generation code to input data directly into the tool, instead of exporting data from a Jupyter notebook and importing it into the tool every time.

Flow

The user frequently performs the challange-response workflow, which she would like to optimize using our viz. tool. The workflow has the following steps:

1. Generate PUF data
2. Generate a challange matrix
3. Generate a response matrix
4. Visualize and interact with these matrices
5. Select a submatrix and and calculate $F(submatrix)$

Non Functional Requirements

Security: The user does not work with any confidential or private data. They generate data from random sampling APIs and feed that data into their existing tools. They do not any electronic sensors or physical methods of gathering data. Therefore, security is not a large concern.

Scalablity: The system needs to potentially be able PUF matrices of size $10^4 \times 10^6$. The users do occasionally work with matrices of this size.

Probes

Are you currently using any tools for data viz?
Excel: Used to store, manage and visualize data. However, the level of customization and interactivity provided by Excel visualizations is not enough for the primary user's needs.

Is visualization appropriate for this project?
Yes, because the client requires a tool that provides a degree of customization and interactivity that any of their current tools do not provide.

Visualization Design Challenges

Interaction with the visualization: The client requires features like drag and drop, and being able to edit values in the visualization after it has rendered. Most existing tools do not provide this level of interaction and implementing this will be a challange

Data density: The client regularly works with matrices of size 10000×10^6. Showing such a large number of records all at once on screen is an unsolved challage in data viz.

Work Plan

Week 1-2: Gain more insight into requirements, bounce off ideas with the client
Week 2-3: Build mockup of the layout of the primiry screens and get approval from client
Week 3-4: Start Implementation of a prototype
Week 4-5: Demo 1 and discussion of changes, discuss new ideas and additional features to be prioritized
Week 5-6: Implement changes required after demo 1
Week 6-7: Demo 2, discussion of minor changes
Week 7-8: Final presentation, documentation, hosting on server

C.V

Jason Pereira
Graduate Student, Department of Computer Science, UIC
Skills: Python, Java, d3, three.js, Javascript, Scala, concurrency, CSS, React, ExpressJS

Wenjing Rao
Associate Professor, Department of Electrical and Computer Engineering, UIC
Research Interests: VLSI test, Fault-tolerance, Reliability, Novel computation paradigms in emerging nanoelectronic systems, VLSI Hardware Security, Computer Aided Design (CAD)

Yeqi Wei
Ph.D. Student, Department of Electrical and Computer Engineering, UIC
Research Interests: Fourier analysis, asynchronous circuits, binary codes, encoding, error correction codes, fault diagnosis, field programmable gate arrays

References

1. Herder C., Yu M.D., Koushanfar F., Devadas S. Physical unclonable functions and applications: A tutorial

2. Yeqi Wei, Tim Fox, Vincent Dumoulin, Wenjing Rao, Natasha Devroye; APUF faults: impact, testing, and diagnosis -