High Performance Library for Big Data Analysis (NotScared)

Preston Petersen, Shawn Robinson, Jadon Procter

Team

Preston Petersen, Shawn Robinson, Jadon Procter

- Computer Science Undergraduates at EECS eCampus
- Senior Capstone Project Team

Vincent Immler (Project Partner)

- Assistant Professor at Oregon State with a focus in hardware security
- Provided a lot of guidance for the project and taught us fundamental hardware security ideas from the ground up.

What is our project?

- The library is a collection of tools useful for hardware security research.
- Side Channel Analysis (SCA)
  - Encryption is in theory perfect, but…
    - Extra information leakage (Power usage and magnetic fields)
    - Leakage correlates with the data (the data + the key) being processed
- An oscilloscope is used to measure either power draw or magnetic field strength across the duration of encryption (creating a “trace”)
- Since we know the encryption algorithm in advance (AES-128), we can calculate how strongly the trace correlates with each key value

Key Features

- Fast
- Open-source
- Expandable
  - Easily add tasks, models
- Multiple Leakage Models
- NxM Tile compatibility
  - Heat Map
  - Multiprocessing
- Signal to Noise Ratio
- Correlation Power Analysis

File Format

- Custom specification
- HDF5 files
  - Fast
  - Simple
  - Open-source
  - Works well with Numpy
  - Transparent compression
  - Metadata embedding

traces/

/plaintext/tile_x/tile_y/[ptxts]

/key/tile_x/tile_y/[keys]

/samples/tile_x/tile_y/[samples]

/ciphertext/tile_x/tile_y/[ctxts]

plaintext is a 2D array of integers, each element a byte of the plaintext.

key is a 2D array of integers, each element a byte of the key.

samples is a 2D array of integers, each element a reading from the sample.

ciphertext is a 2D array of integers, each element a byte of the ciphertext.

```
AES_SBOX = np.array([
    0x63, 0x7C, 0x77, 0x7B, 0xF2, 0x6B, 0x6F, 0xC5, 0x30, 0x01, 0x67, 0x2B, 0xFE, 0xD7, 0xAB, 0x76,
    0xCA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0, 0xAD, 0xD4, 0xA2, 0xAF, 0x9C, 0xA4, 0x72, 0xC0,
    0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC, 0x34, 0xA5, 0xE5, 0xF1, 0x71, 0xD8, 0x31, 0x15,
    0x04, 0xC7, 0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A, 0x07, 0x12, 0x80, 0xE2, 0xEB, 0x27, 0xB2, 0x75,
    0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0, 0x52, 0x3B, 0xD6, 0xB3, 0x29, 0xE3, 0x2F, 0x84,
    0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A, 0xCB, 0xBE, 0x39, 0x4A, 0x4C, 0x58, 0xCF,
    0xD0, 0xEF, 0xAA, 0xFB, 0x43, 0x4D, 0x33, 0x85, 0x45, 0xF9, 0x02, 0x7F, 0x50, 0x3C, 0x9F, 0xA8,
    0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5, 0xBC, 0xB6, 0xDA, 0x21, 0x10, 0xFF, 0xF3, 0xD2,
    0xCD, 0x0C, 0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17, 0xC4, 0xA7, 0x7E, 0x3D, 0x64, 0x5D, 0x19, 0x73,
    0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A, 0x90, 0x88, 0x46, 0xEE, 0xB8, 0x14, 0xDE, 0x5E, 0x0B, 0xDB,
    0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C, 0xC2, 0xD3, 0xAC, 0x62, 0x91, 0x95, 0xE4, 0x79,
    0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x4E, 0xA9, 0x6C, 0x56, 0xF4, 0xEA, 0x65, 0x7A, 0xAE, 0x08,
    0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6, 0xE8, 0xDD, 0x74, 0x1F, 0x4B, 0xBD, 0x8B, 0x8A,
    0x70, 0x3E, 0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E, 0x61, 0x35, 0x57, 0xB9, 0x86, 0xC1, 0x1D, 0x9E,
    0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94, 0x9B, 0x1E, 0x87, 0xE9, 0xCE, 0x55, 0x28, 0xDF,
    0x8C, 0xA1, 0x89, 0x0D, 0xBF, 0xE6, 0x42, 0x68, 0x41, 0x99, 0x2D, 0x0F, 0xB0, 0x54, 0xBB, 0x16
])
```

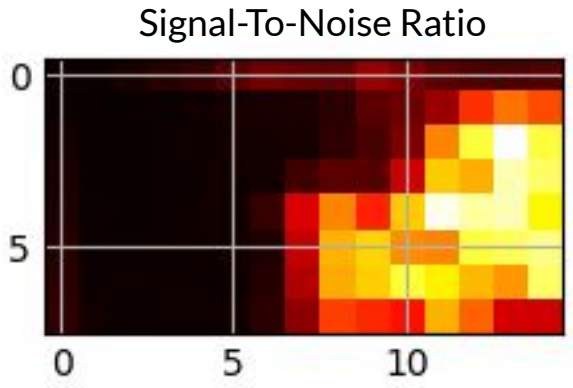
Leakage Models

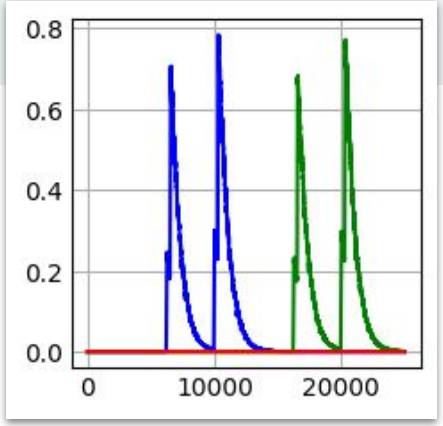
- Different Hardware Systems may have different leakage patterns based on how they flip bits during encryption.
- Current models
  - Hamming weight - The number of bits set to 1 in the output
  - Hamming distance - The number of bits that differ between the output and input

```
# Hamming weight lookup table
HW_LUT = np.array([
    0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4,
    1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5,
    1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5,
    2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6,
    1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5,
    2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6,
    2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6,
    3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7,
    1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5,
    2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6,
    2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6,
    3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7,
    2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6,
    3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7,
    3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7,
    4, 5, 5, 6, 5, 6, 6, 7, 5, 6, 6, 7, 6, 7, 7, 8
])
```

Gridded Data Compatibility

- Different chip architectures have the highest magnetic field leakage in different spots
- NxM tile processing is not supported by the current industry standard
- Heat Map
- Multithreading across tiles





Signal to Noise Ratio

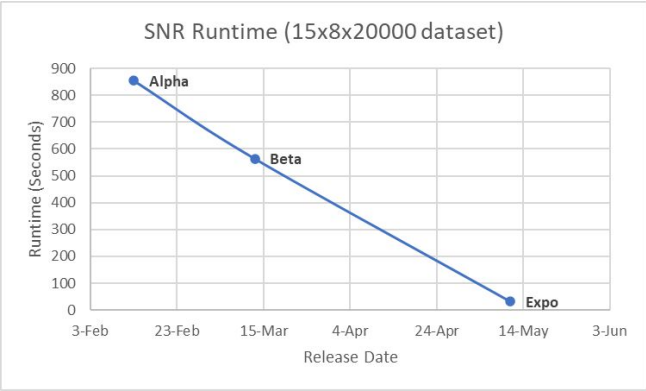
Signal to Noise Ratio is a measurement of desired signal against the background noise. This can be used to reveal points of interest within the traces.

Our Benchmarks

Alpha: 14 minutes, 14 seconds

Beta: 9 minutes 22.6 seconds

Current: 31.6 seconds



```
np.set_printoptions(precision=1)
key_candidates = ns.results[0][0] # Access the results for tile (0,0)
print("KEY CANDIDATES:\n", key_candidates[0])
print("CORRELATION:\n", key_candidates[1])
```

```
[11]
...
KEY CANDIDATES:
 [129,  29,  25,  73,  204,  136,  214,  88,  157,  132,  183,  130,  195,  44,  60,  51]
CORRELATION:
 [0.561, 0.546, 0.078, 0.075, 0.078, 0.070, 0.070, 0.080, 0.069, 0.076, 0.074, 0.069, 0.066, 0.076, 0.071, 0.071]
```

Correlation Power Analysis

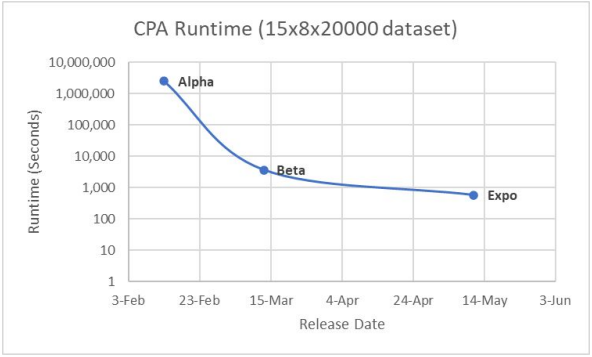
Correlation Power Analysis is a statistical equation for calculating the relationship between two variables. In SCA this is the correlation between the oscilloscope readings and theoretical leakage model.

Our Benchmarks (Log Scale)

Alpha: 28.5 days

Beta: 70 minutes, 37 seconds

Current: 9 minutes, 32 seconds



Designing - Fall

- Architecture Plan
  - Explanatory documents on the scope of the project.
  - Design documents
  - Technical requirements
- Research
  - Algorithms
  - File formats
  - Numerical stability
  - Memory usage

Sprint #1 - Winter

Planned

- Implement MIA
- Implement CPA
- Implement SNR
- File Conversion (sqlite to hdf5)
- Histogram Method for Variance and Mean
- Welfords Online Algorithm

Completed

- Barebones CPA
- Barebones SNR
- Single Tile File Conversion
- Histogram
- Welfords

Not completed

- CLI (Postponed)
- MIA (Cancelled)

Sprint #2 - Winter

Planned

- Command Line Interface
- Conversion Filtering Profiled Data
- Cursor based read
- CPA results post-processing
- Bug Bash
- Alpha Release

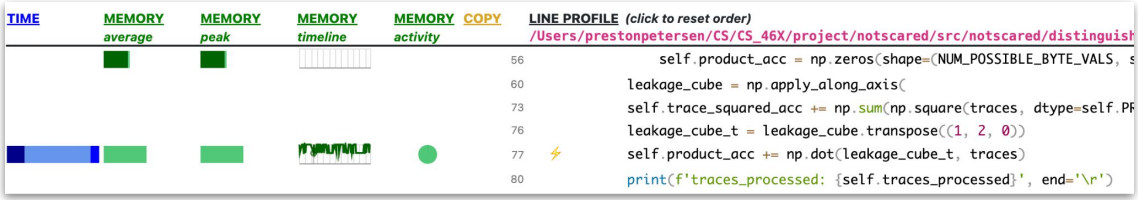
Completed

- Conversion Script Filtering
- Cursor Based Read
- CPA results post-processing
  - Exposed bugs
- Bug Bash
- Alpha Release

Not Completed

- Command Line Interface (Cancelled)

Sprint #3 - Winter



Planned

- Optimize CPA
  - Vectorized (Switch to numpy broadcasting instead of loops)
  - Calculate (Method)
  - Push batch (Method)
  - Product accumulator (Data structure)
- File handling
  - Tile handling
  - Fancy indexing
- Demo jupyter notebook
- SNR
  - Rewrite
  - Plotter
- Compressed HDF5 benchmarks
- Beta Release

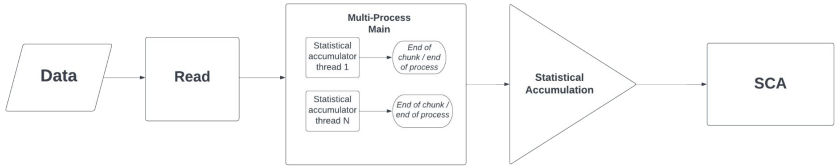
Completed

- Optimize CPA
- File handling
- Demo Jupyter Notebook
- SNR Rewrite
- Beta Release

Not completed

- Compressed HDF5 Benchmarks (Postponed)

Sprint #4 - Spring



Planned

- Re-architecture
  - Modularization for usability and expandability
  - Containerization
  - Tasks Class
  - NotScared Class
- Byte-based threading
- Compressed HDF5 Benchmarks

Completed

- Re-architecture
- Compressed HDF5 Benchmarks

Not Completed

- Byte-based threading (Extended)

Sprint #5 - Spring - Ongoing

Planned

- Dynamic Memory Management
- Usability Improvements
- Heat Map
- Cache hit optimization
- Byte-based threading
  - Continuation and Benchmarking
- Thorough Unittesting
  - Data Mocking
- Expo Materials

Completed

- Heat Map
- Tile-based threading
- Thorough Unittesting
- Expo Materials

Not Completed

- Cache hit optimization (Cancelled)
  - Uninspiring performance benefits
- Byte-based threading
  - Continuation and Benchmarking

Future of the Project

- A full time research assistant will be taking over the project in the Summer
- The combination of materials from our team as well as the other team will be used to continue research.

Huge thank you to Vincent Immler for creating opportunities for Oregon State Students to participate in industry relevant work. He has been an incredible resource for learning and progression towards career readiness.