Lab 04: Side Channel Analysis Attacks (Part 1) Statistical Analysis of Information Leakage

Jasmin Kaur and Alec Braynen
University of South Florida
Tampa, FL 33620

I. Introduction

Electronic devices often leak information about their internal functionality in the form of supply curent, electromagnetic (EM) radiation, timing, or power [1]. Observing and analyzing differences in patterns of these side channel information leakages could be used by a knowlegeable adversary to gain sensitive information [1]. Exploiting such physical information leakage to gain sensitive information is known as side-channel analysis (SCA) attacks which are often easy and inexpensive to mount on target devices [1].

The main idea behind SCA is to try and identify how power consumption or timing of a device varies with respect to some internal process occurring such as data processing. For example, the attackers could observe differences in how long it takes to complete a complex task for a timing based SCA, or one may observe the power consumption of different chip components based on data processing for a power analysis [1]. Statistical analysis helps us to determine what difference in side-channel leakage patterns can be considered significant to extract information from given data. A statistical hypothesis test (SHT) called a Welch's t-test is used to analyse whether two different sets of recorded data have identical mean when same inputs are used. A *trace* is the recording of a side channel leakage as it varies over time for a single process [1]. A t-test is determining whether two independent data sets have the same mean with respect to a null hypothesis (there is no difference between two data sets that is statistically significant). In terms of SCA, this means observing any slight differences in side-channel leakages if two different keys are used to encrypt the same plaintext or if two different plaintexts are encrypted using the same key [1].

In this work, the authors perform statistical analysis using Test Vector leakage Assessment (TVLA) as a baseline approach to determine information leakage by changing an internal parameter and observing any differences [1]. ChipWhisperer (CW) Nano board is utilized to gather power traces from two different data lists for three different operations - multiplication, XOR operation, and modular multiplication performed on all the elements of the given lists. The gathered traces are then used to perform statistical analysis where the differences in these traces are observed and analyzed through the graphed plots. The P-values of the gathered traces are used to determine whether the null hypothesis is rejected (significant differences exists if ($P-value \leq 0.05$)) or accepted ($P-value > 0.05$) [1].

II. Reading Check

Question 1: What are some examples of physical parameters that an attacker can leverage in an SCA attack?

Answer: Attackers can utilize timing side channels and power side channels to leverage SCA attacks. Additionally, physical parameters such as supply current, power consumption, EM radiation, noise, or timing can be leveraged by an attacker to perform an SCA on an electronic device. [1].

Question 2: Why does an unprotected electronic device (one without any countermeasures in place) leak information through side channels?

Answer: This happens because as the device performs different computations, or computations on the same data, different transistors switch on and off, different values flow along the circuit busses, etc. [1] For example, devices may leak side-channel information when performing complex mathematical operations such as multiplication which could differ in processing time and power consumption when compared to an XOR operation [1]. Or XOR operation consumes slightly more power when there are more 1's than 0's present in the output [1].

Question 3: What does the p-value represent in a statistical hypothesis test, such as the t-test?

Answer: The probability value (P-value) is the deterministic value to accept or reject the null hypothesis which states that there is no statistically significant difference between two data sets (population). Generally, significance is evaluated at 5% (0.05) probability which means that there is a 5% probability of rejecting a true null hypothesis and 95% probability of not rejecting it. This significance level is called the P-value (between 1 and 0). If ($P-value \leq 0.05$)) Then the statistical significance exists otherwise if ($P-value > 0.05$) then it does not exist. [1].

III. Methods

For this experiment we gather power traces of computation on two data lists using CW nano board and Python. The PHS-VM includes ChipWhisperer v5.5 and Python v3.8-3.10 to interface with CW. The CW is configured for communication using the methods CW setup scripts contained in the directory ”cw-base-setup” and a firmware script. The ”cw-base-setup/simpleserial-base” is used as a template firmware to program the nano board.

A. *Software Setup*

The software setup for the experiment consists of a virtual machine running Ubuntu OS on Virtualbox. The Python and C code is implemented via Jupyter notebooks on this virtual machine. ChipWhisperer v5.5 and Python v3.8-3.10 to interface with CW and CW setup library ”simpleserial” is used to send and recieve data from CW. The random class of numPy library is used to generate random data lists for testing the functionality of CW and well as to gather power traces for the experiment.

The directory structure of the files in the Jupyter notebook is structured as follows:

```
PHS-Lab-04
│
├─ Firmware
│  ├─ makefile
│  ├─ tvla-test-CWNANO.hex
│  └─ tvla-test.c
│
├─ Python
│  └─ Lab-04-SCA-Statistical-Analysis.ipynb
│
└─ Figures
   ├─ tvla_a.pdf
   ├─ tvla_a2.pdf
   ├─ tvla_a_diff.pdf
   ├─ tvla_a_t.pdf
   ├─ tvla_b.pdf
   ├─ tvla_b2.pdf
   ├─ tvla_b_diff.pdf
   ├─ tvla_b_t.pdf
   ├─ tvla_c.pdf
   ├─ tvla_c2.pdf
   ├─ tvla_c_diff.pdf
   └─ tvla_c_t.pdf
```

1) The ”Python” directory hold the Jupyter notebook files where the python code is executed for compiling firmware, generating random data lists of 16 elements, reading CW responses as well as data analysis and plotting graphs.
2) The ”Firmware” directory holds the C file containing the functions that are performed by CW on the recieved data - multiplication of all the list elements, XORing all the list elements, and performing modular multiplication of all the list elements. It also contains a makefile used to compile the C code and a ”.hex” file that is used to configure CW.

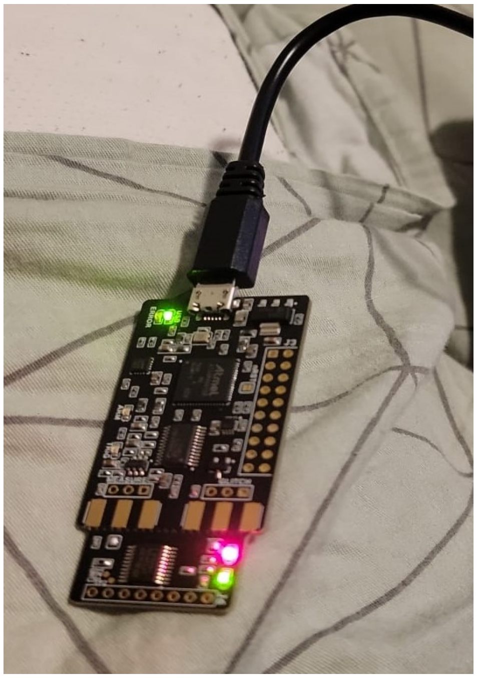


Fig. 1: CW nano board Connected to PC

3) The ”Figures” directory holds the generated figures of the analyzed data.

B. *Hardware Setup*

The hardware setup consists of the CW nanoboard connected via a microUSB cable to a desktop machine running Windows (Fig. 1).

C. *Experimental Procedure*

Part A: Firmware script

The C code for programming CW nano board is as follows:

- The code files ”simpleserial-base.c” and the ”makefile” from ”cw-base-setup/simpleserial-base” are used as a template for the firmware script for CW.
- The ”simpleserial-base.c” file is re-named to ”tvla-test.c” and contains functions ”tvla-test-A”, ”tvla-test-B”, and ”tvla-test-C”.
- The ”tvla-test-A” multiplies all the 16 elements of the input array.
- The ”tvla-test-B” performs an XOR of all the 16 elements of the input array.
- The ”tvla-test-C” performs a modular multiplication of all the 16 elements of the input array with modulo ”0xFB” (251).
- These functions are called in the main using simpleserial parameters ’a’, ’b’, and ’c’.
- The makefile is modified to the map to the correct path of the tvla-test.c file

Part B: Obtain Data

Obtaining the data is done by:

- Python file ””Lab-04-SCA-Statistical-Analysis.ipynb” imports chipwhisperer, matplotlib.pyplot, numpy, and scipy.stats.
- The C code is compiled in the python file and the CW is then programmed with the C code after verifying the correctness of C code.
- A reset target function is defined to reset the target on the board.
- A function is defined to interface with the board to send and receive data
- Code is written in Python that performs the same operations as the tvla-tests defined in the tvla-test.c file. These are used to verify the board is giving a correct response on each test.
- A function is defined to capture the power trace of the board during its processing
- Then for each operation of computing the: product, xor and modulus, the function above is used, along with two data lists, to read two power trace responses from the board.
- These two traces are plotted, and also differentiates against each other (and this differentiated trace is plotted as well.)

Part C: Analyze Data

The analysis of the responses is done by:

- These two traces, for each operation, are plotted, and also differentiates against each other (and this differentiated trace is plotted as well.)
- The Welch t-test is performed on every trace and their t and p values are obtained.
- The t values are plotted and the mean and standard deviation of the t values are calculated.
- The percentage of statistical significance (count of p values less than 0.05) is calculated
- The board is disconnected.

IV. Results

The results of our work is shown in Figures 2-4 respectively. The graphs show the 32 power traces obtained using CW for two sample data sets containing 16 elements each.
From the power trace results for implemented product, XOR, and modular multiplication, we can gain information about which of the each function is being performed as multiplication uses more power than XOR operation. The power consumption for modular multiplication (4.a and 4.b) is the highest since it is an expensive operation to perform for large numbers. From the power trace graphs we can also infer when a 1 or a 0 is being read or sent. The differential power traces of the two data sets for the above mentioned functions help us realize if the two datasets show any statistical correlation. The statistical correlation is further analyzed using the Welch's t-test where the t and p values are plotted in figures 2.d, 3.d, and 4.d

TABLE I: Mean and standard deviation for the captured power traces

Function	Dataset	Mean	SD
Multiplication (Trace A)	A	0.0386	0.1398
Multiplication (Trace A)	B	0.0389	0.1397
Multiplication (Trace A)	Diff.	-0.0002	0.0065
XOR (Trace B)	A	0.0397	0.1354
XOR (Trace B)	B	0.0397	0.1351
XOR (Trace B)	Diff.	0.0000	0.0021
Modular Mult. (Trace C)	A	0.0394	0.1217
Modular Mult. (Trace C)	B	0.0392	0.1255
Modular Mult. (Trace C)	Diff.	0.0002	0.1309

TABLE II: Percentage of p-values below 0.05

Trace	Percentage p
Trace A	52.8%
Trace B	5.4%
Trace C	71.8%

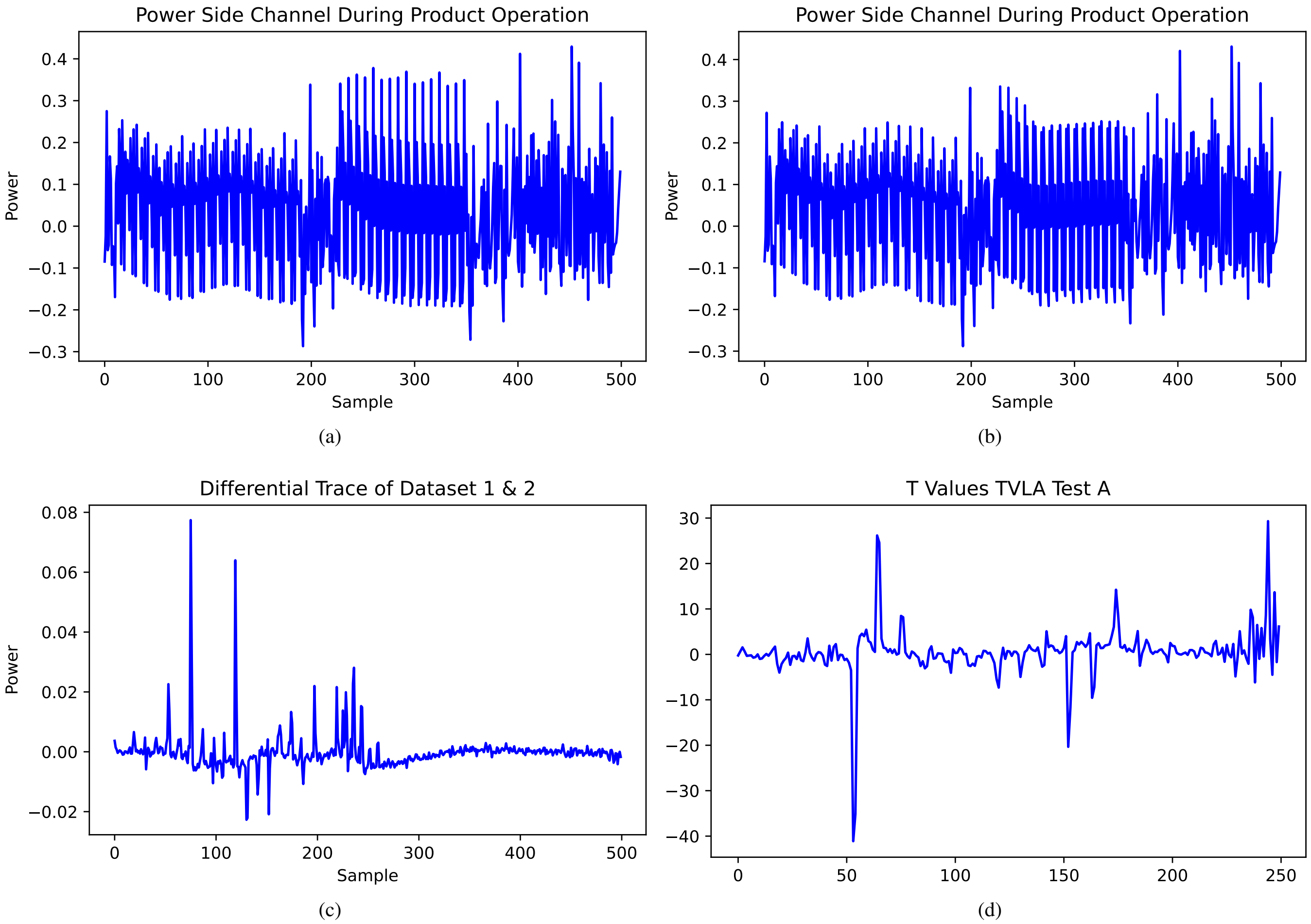
V. Discussion

These results in Section IV demonstrate the leakage that occurs during various operation on the CW Nano board. Each operation, multiplication, XOR, and MOD has a distinct power trace and the t-values and plots of each operation mostly show this. Additionally, with the modulus operation, the differential analysis shows a strong correlations between the two data lists.

From the graphs and tables presented in Section IV, it is evident using side channel attacks can be easily mounted on various devices as shown by this experiment using CW nano board. With a basic side channel analysis, one can determine which operations the board is performing. Additionally, there is also leakage with a differential analysis, with all plots of these traces showing some statistical information, and the MOD operation showing significant data shape leakage.

During this experiment, we had issues adapting to programming the CW nano board in the programming language of C. Additionally, we had issues properly setting up the makefile. These issues mainly stemmed from Jupyter notebooks not providing an autosave function by default for these files.

In future work, we would like to investigate how to reduce the side channel information leakage on the CW nano board. Additionally, we would like to investigate how this leaked data can be used to further attack the device.



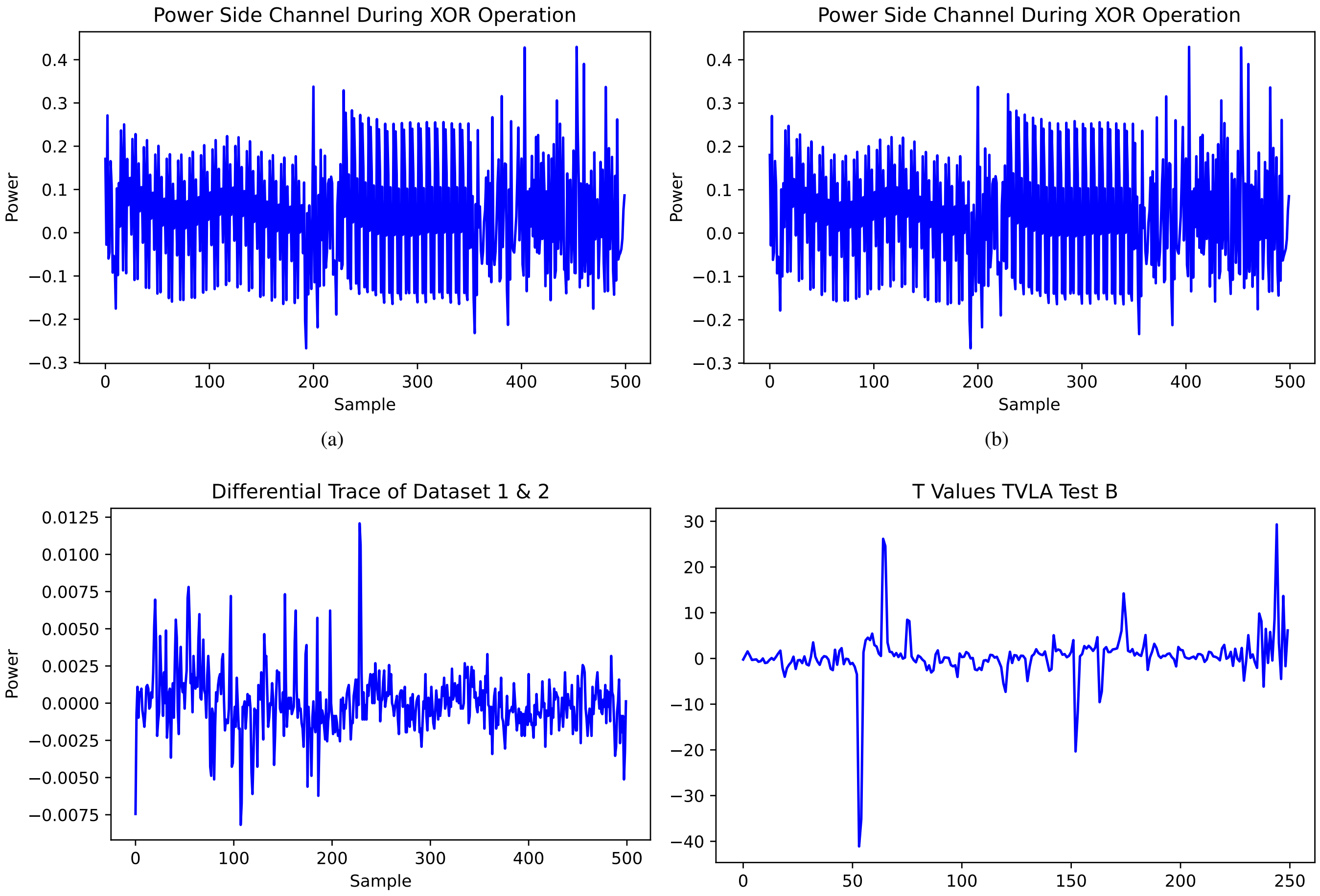
(a)

(b)

(c)

(d)

Fig. 2: Data analysis of 32 power traces for 2 data lists of 16 elements each using ”tvla_test_A”



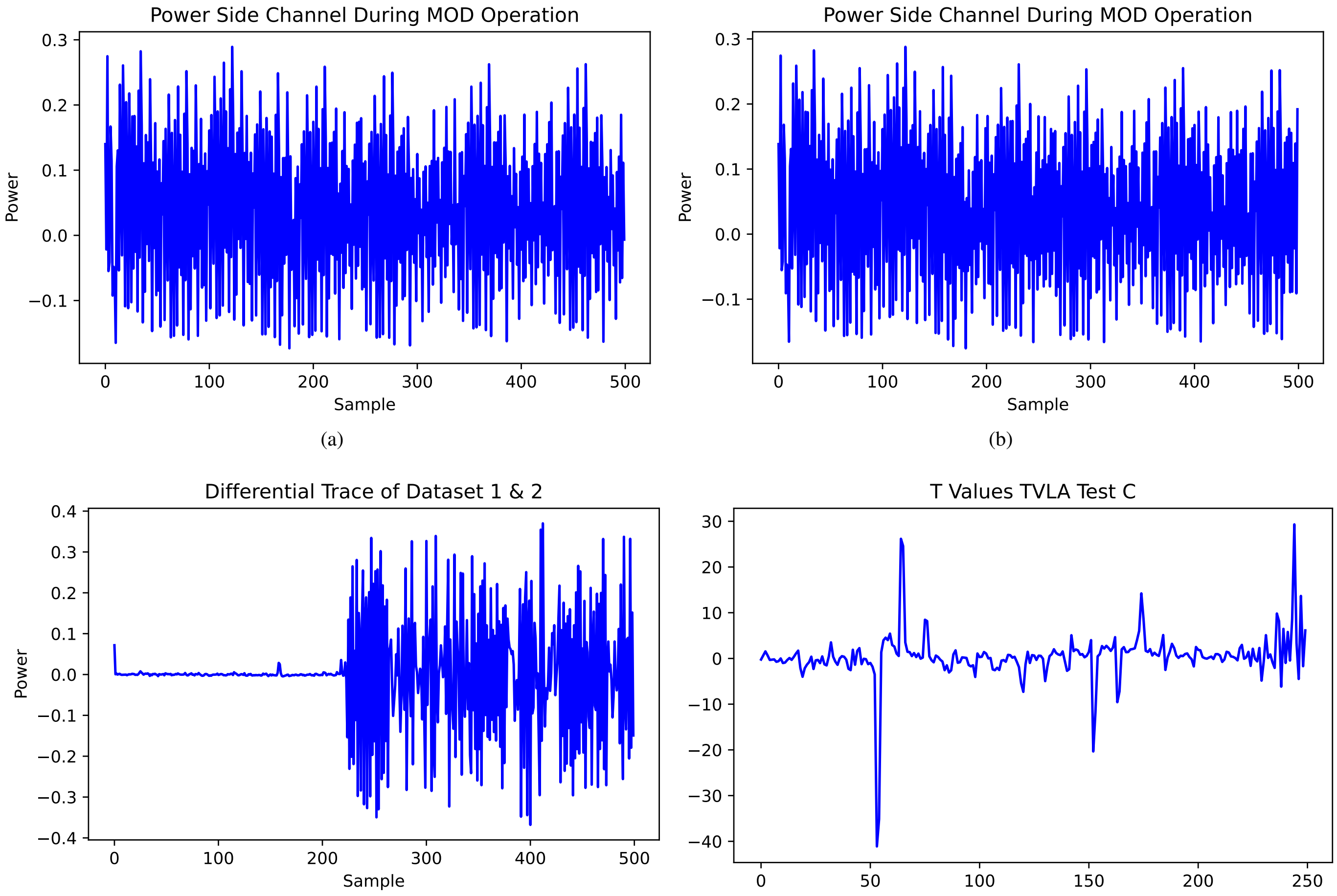
(a)

(b)

(c)

(d)

Fig. 3: Data analysis of 32 power traces for 2 data lists of 16 elements each using ”tvla_test_B”



(a)

(b)

(c)

(d)

Fig. 4: Data analysis of 32 power traces for 2 data lists of 16 elements each using ”tvla_test_C”

VI. Conclusion

In this lab, we captured power traces for two data sets using CW nano board programmed to perform multiplication, XOR and modular multiplication operations on the given data. The captured trace values were plotted onto the graphs which were analyzed to determine power consumption based on the data processing and function performed at a given time. The correlation between the two data sets was observed by performing differential power analysis on the average power traces of each data set. Welch's t-test using the Scipy library was also performed for each data set to determine the statistically significance (depending upon the P-value) between the two data sets for each function performed. This experiment taught us the extremely useful skills of side channel analysis, that can be used in many modern devices to gather sensitive information.

References

[1] R. Karam, S. Katkoori, and M. Mozaffari-Kermani, "Experiment 4: Side Channel Analysis Attacks (Part 1)," in *Practical Hardware Security Course Manual*. University of South Florida, Aug 2022.