# Enhancing Information Security Courses With a Remotely Accessible Side-Channel Analysis Setup

Abubakr Abdulgadir
George Mason University
Fairfax, VA, USA
aabdulga@gmu.edu

Jens-Peter Kaps
George Mason University
Fairfax, VA, USA
jkaps@gmu.edu

Ahmad Salman
James Madison University
Harrisonburg, VA, USA
salmanaa@jmu.edu

## ABSTRACT

The ever-increasing security threats to our digital infrastructure impose the training of a sufficient number of engineers on real-world equipment and attacks. A significant investment in equipment is often needed to teach hardware security. Additionally, the global COVID-19 pandemic demonstrated that online-accessible educational systems are crucial to the continuity of the teaching process. In this work, we describe our experiment with teaching hardware security using a centralized shared setup that can be accessed remotely by students. Our setup reduces the cost and makes teaching such advanced topics more accessible while keeping the benefits of using real hardware to gain practical experience.

## CCS CONCEPTS

• **Applied computing** → **Computer-assisted instruction**; • **Hardware** → **Reconfigurable logic applications**; • **Security and privacy** → **Side-channel analysis and countermeasures**.

## KEYWORDS

Education, Hardware Security, FPGA, Side-Channel Analysis Cryptography

**ACM Reference Format:**
Abubakr Abdulgadir, Jens-Peter Kaps, and Ahmad Salman. 2022. Enhancing Information Security Courses With a Remotely Accessible Side-Channel Analysis Setup. In *Proceedings of the Great Lakes Symposium on VLSI 2022 (GLSVLSI '22), June 6–8, 2022, Irvine, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3526241.3530347

## 1 INTRODUCTION

With the growing security threats to our digital infrastructure, training engineers who can design and verify secure systems is critical for the prosperity of our computing-based economy.

Among the significant threats to the security of systems are Side-Channel Attacks (SCAs), where an adversary uses unintended system outputs such as variations in power consumption to reveal keys. SCA experimental setups help augment teaching the concepts of side-channel analysis and other implementation attacks. Such setups allow students to run experiments and analyze data. In addition to clarifying concepts, practical experiments allow students to appreciate the feasibility and practicality of these attacks, encouraging them to consider such threats in their future designs seriously. Although attackers can perform such attacks at a low cost, providing SCA setups in a classroom setting can be costly if each student or small group is provided with their setup. This issue was already noted at GLSVLSI 2019, where Aysu reported the cost of SCA boards as a significant challenge for teaching hardware security lab experiments and stated that there is a need for cheap and accessible side-channel analysis boards [2].

Moreover, the COVID-19 pandemic and its associated lockdowns clearly showed that the continuity of the educational process depends heavily on the availability of online-accessible systems that can be used remotely.

This paper discusses our experiment in supporting information security classes using an SCA experimental setup to provide hands-on experiments on hardware security. The setup is online-accessible, so students can log in remotely, run experiments on real hardware, and analyze results. This experiment started in Spring 2020 when the semester was abruptly changed to online teaching. We added online reachability and time-sharing multi-user features to an SCA setup. The system has since supported three class offerings in two universities.

Our system is designed with fault tolerance in mind to recover from error conditions. This is implemented through a watchdog mechanism that monitors the system status and resets failing components. As a result, we claim the following contribution:

(1) An SCA setup that can be shared by multiple users remotely which reduces cost and allows social distancing.
(2) Educational materials consisting of labs that guide students through experiments to learn the fundamentals of SCA and assess their progress.

## 2 BACKGROUND AND PREVIOUS WORK

### 2.1 Side-channel Analysis

Side-channel attacks (SCA)[9] have been recognized for two decades to be a severe threat to the practical deployment of cryptography. In these attacks, an adversary utilizes side channels such as power consumption and electromagnetic radiation to reveal secret keys. Differential power analysis (DPA) is one of the most potent SCA attacks since its ability to extract keys increases as more side-channel traces are available.

In a typical DPA attack, the cryptographic function inside the design-under-test (DUT) is invoked several times with different inputs. The power consumption is measured using an oscilloscope or an analog-to-digital converter and recorded for analysis. A small part of the key (sub-key) is guessed, and the power consumption of



*GLSVLSI '22, June 6–8, 2022, Irvine, CA, USA*

ACM ISBN 978-1-4503-9322-5/22/06.
https://doi.org/10.1145/3526241.3530347

the DUT is estimated using a power model for each guess. Statistical analysis is performed to determine the most likely key guess. This operation is repeated for each sub-key until the full key is revealed.

Several SCA platforms to perform SCA are readily available. The DPA Workstation from Rambus [12] and Inspector from Riscure [13] are examples of commercial systems. SAKURA boards [6] are widely used in academia and support FPGAs and smart cards; however, the project concentrates more on the hardware than analysis tools. NewAE Chipwhisperer SCA platform offers several target options and allows target and sampling clocks to be synchronized for precise measurements [11].

The Flexible Opensource workBench fOr Side-channel analysis (FOBOS) [1] is an SCA platform that uses commercially available low-cost FPGA boards (e.g., Digilent Nexys-A7) whenever possible. The FOBOS control board is compatible with Chipwhisperer FPGA targets and can also use a synchronized sampling clock, and includes analysis scripts that can be used for attacks and leakage assessment.

## 2.2 Test Vector Leakage Assessment

The Test Vector Leakage Assessment (TVLA) methodology [5] is used to test leakage from cryptographic implementations. This test is widely accepted as a first check for security against SCA attacks.

The idea of the test is as follows; if a device is DPA-resistant, its power consumption must be independent of the algorithm's intermediate values. Consequently, power traces collected when processing fixed data and traces collected when processing random data should be statistically indistinguishable. This variation is called the fixed-vs-random test. We label the two trace sets $Q_f$ and $Q_r$, respectively. The $t$ value is calculated as follows:

$$t = \frac{\mu_f - \mu_r}{\sqrt{\frac{s_f^2}{n_f} + \frac{s_r^2}{n_r}}}$$

where $\mu_f$ and $\mu_r$ are the means, $s_f$ and $s_r$ are the standard deviations and $n_f$ and $n_r$ are the number of samples in the sets. The null hypothesis is that the means of the two trace sets $Q_f$ and $Q_r$ are equal (i.e., the two trace sets are indistinguishable). We use the calculated $t$ value as an indicator to accept/reject the null hypothesis at a certain confidence level. If the $t$ value is greater than a predefined threshold, the device fails the test.

## 2.3 Hardware Security Courses and Equipment

Several researchers have explored teaching hardware security to 4-year and 2-year college students. In [15], Wiesen et al. propose educational guidelines for hardware reverse-engineering courses with a hands-on component. Additionally, they designed a course that included five hands-on projects to evaluate their methodology. Chandy et al. [4] developed a set of hardware-security oriented courses with hands-on components to train students in this area. Based on student feedback, the authors concluded that theoretical and practical components are essential in such courses. In [2], Aysu argued that even though hardware-security courses are being taught in many US universities, the number of offered courses does not meet the demand. He also presented a hardware security course focusing on SCA. Schaumont described an experiment to teach an online digital signal processing to conform to social distancing imposed by the COVID-19 pandemic [14]. In this course, a low-cost setup was used to replace expensive lab equipment used in on-campus offerings of the same course.

A common issue with many of these courses is that the lab setting they require is not affordable to most students, limiting their hands-on experience to time in the lab. Recently, NewAE introduced relatively affordable hardware, ChipWhisperer-Nano [7], with a built-in target Arm Cortex-M0 processor, which has 16KB of FLASH and 4KB of SRAM. Even though their capabilities are limited compared to a ChipWhisperer-Lite, DPA Workstation, or Inspector, they are sufficient for an introductory course to SCA. At a price between $40 each in the classroom pack (20 Nanos) and $50 in single quantities, they are ideal for an in-person lab setup.

Due to COVID-19 restrictions, an in-person lab was not possible. The Nano was not an option either, as we did not want to impose an additional cost on students, did not have the funds to purchase 30 or more boards for students to borrow, and did not want the headache of managing the lending of boards with subsequent full functional verification and eventual updates. Even in the absence of a pandemic, there are other circumstances where an in-person lab is not possible for example when the department does not have lab space available to set up an in-person lab or the course does not have a dedicated lab section associated with it. For these reasons, we decided to develop an online accessible SCA evaluation setup based on the FOBOS platform and created a series of lab exercises.

# 3 METHODOLOGY

Our goal was to augment information security courses with an experimental setup that can be shared by multiple students and provide educational materials targeting students from different backgrounds, including non electrical and computer engineering students. Below we detail the courses, lab, and experimental setup used.

## 3.1 Course description and Student Background

Our online SCA setup and the associated labs were introduced in Spring 2020 in ECE/CYSE 476 shortly after COVID19 forced a shutdown of in-person classes at George Mason University (GMU) and used again in Spring 2021 (still no in-person classes) and currently in Spring 2022 with in-person classes. IT 435 at James Madison University (JMU) started using the labs in Fall 2021, and classes were in-person.

*3.1.1 CYSE/ECE 476 Cryptography Fundamentals.* is a senior-level course and required for students in the Cybersecurity Engineering program (CYSE) and a technical elective for students in the Electrical Engineering and the Computer Engineering programs (ECE). This course introduces the students to historical ciphers, modern secret-key stream and block ciphers, modes of operation, and public-key cryptosystems such as RSA, elliptic curve, and post-quantum cryptography. It covers the mathematical background required for understanding the algorithms and security estimates. The course lets the students explore historical ciphers, the limits of key management for public-key ciphers, and attack implementations of cryptographic algorithms in hardware using side-channel analysis through hands-on projects. The prerequisite for this course

are introductory courses in Python programming and digital electronics. One 1 hour and 15 minute lecture is devoted to the basics of side-channel analysis and explains simple power analysis and correlation power analysis in particular. It includes the mathematical background on computing the power model for the AES implementation that they will be attacking in the lab. The lab consists of three parts, and the students have one week to complete each. At the end of each lab, the student has to write a report in which they answer all questions posed in the lab instructions. Students were working on these labs in groups of two. A second lecture on side-channel analysis covers timing and cache attacks and is scheduled later in the semester after completing the SCA labs.

*3.1.2 IT 435 Cryptography for Information Technology.* is a senior-level course in the Computer Science department. The course is intended for students enrolled in the Information Technology (IT) program and its topics include elementary combinatorics and number theory, classical ciphers and accompanying attacks, and modern encryption schemes, including public-key cryptography, secret-key ciphers, hash functions, side-channel attacks, and post-quantum cryptography. The prerequisite for this course is an introductory course in information security which introduces general topics in network and computer security. Side-channel analysis is introduced as a concept in two 1 hour and 15 minutes long lectures that include different types of side-channel analysis, timing attacks, simple-power analysis, and correlation power analysis. The students are then presented with a three parts hands-on lab identical to the three parts project from CYSE/ECE 476. The only difference is that in IT 435, students perform the hand-on labs during class time, with the instructor available to answer questions that may arise while performing the lab and explain concepts. Each student was working individually on each of the three parts of the lab. The students also get the chance to see the FOBOS hardware setup in-person in that format. Each lab part is scheduled for one week (2 lecture times), with the lab report due one week after it is introduced.

## 3.2 Side-Channel Analysis Setup

Our experimental setup is based on the FOBOS open-source project, which provides a comprehensive yet affordable side-channel platform. The version available at the start of this project, FOBOS 2, has a few drawbacks. First, it requires an external oscilloscope, increasing the setup cost. Also, the interface to the PC is limited in speed since it uses USB/UART communication. Furthermore, FOBOS 2 does not have the capability to be time-shared by multiple users.

To address these shortcomings, we performed a significant upgrade to the FOBOS 2 resulting in a platform that we will refer to as FOBOS 3 in the rest of this paper. The FOBOS 3 Side-Channel analysis platform is composed of the following components:

(1) The *SCA workstation* runs all capture and analysis scripts. To collect traces, the users execute a script that connects to the control board, sends test vectors (e.g., plaintext and key) one at a time, and receives results (e.g., ciphertext) and traces back from the control board. Results and traces are stored in files in the SCA workstation for further analysis.
(2) The *control board* receives test vectors from the SCA workstation and forwards them to the DUT. Once the DUT starts processing a test vector (running the cryptographic algorithm), the control board collects the instantaneous power consumption changes of the DUT using a measurement circuit and the ADC on the FOBOS Shield. The control board also generates the clock signal for the DUT and controls the trigger signal (used to trigger ADC / oscilloscope to collect a trace).
(3) The *target board* (DUT board) is where the cryptographic core that will be attacked or tested is instantiated. The board allows the measurement of changes in the power consumption of the core voltage rail. So far, we tested using a modified Digilent Nexys 3 board, which contains a Xilinx Spartan-6 FPGA as a target board, and a Tektronics CT-1 current probe was used for measurement. Our setup is also compatible with the NewAE ChipWhisperer CW305 DUT, which contains a Xilinx Artix-7 FPGA.

The FOBOS 3 setups used at GMU and JMU are shown in Fig 1 and Fig 2 respectively. The control board consists of a PYNQ-Z1 board and the FOBOS 3 Shield. The FOBOS 3 Shield is a custom board that includes an ADC, a pre-amplifier, power regulators, glitch control, and the DUT communication interface. The PYNQ-Z1 features a Xilinx Zynq 7020 System on Chip, which consists of a dual-core ARM processor running Linux and an FPGA. FOBOS 3 uses the FPGA to generate a clock signal for the DUT, communicate with the DUT, and collect measurements from the ADC on the FOBOS 3 shield. The control board is connected to the control PC using a Gigabit Ethernet interface to transfer commands, data, and power traces. The ARM processing system runs the control software, which is based on the PYNQ [16] library, a Python-based abstraction layer that facilitates writing applications for Zynq.

The high-level architecture of the data acquisition system is shown in Fig. 3. The DUT Communication module interfaces the control board and the target board. DUT Control performs reset, triggering, and timeout functions. The ADC module is optionally used to measure power traces instead of an oscilloscope. It has a maximum sampling rate of 105 MS/s and is based on the OpenADC project [10] with drivers adapted from [3]. The control board also includes modules for power measurement and glitch control.

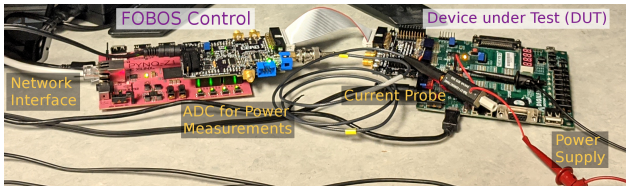


**Figure 1: FOBOS 3 setup at GMU. The target here is a modified Digilent Nexys-3 board**

The architecture of our online-accessible SCA setup is shown in Fig. 4. The user can access a server hosting the control scripts and connect to the control board via a communication interface. The server uses a protocol to send commands and receive status. All data (results and trace) are stored in the host server and can be analyzed or downloaded locally. We utilize Jupyter Lab, an interactive web-based development environment that provides access to many tools, including Jupyter notebooks, to run Python code [8].

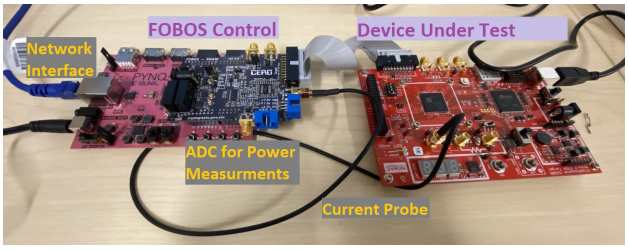


**Figure 2: FOBOS 3 setup at JMU. The target here is a NewAE ChipWhisperer CW305 board**

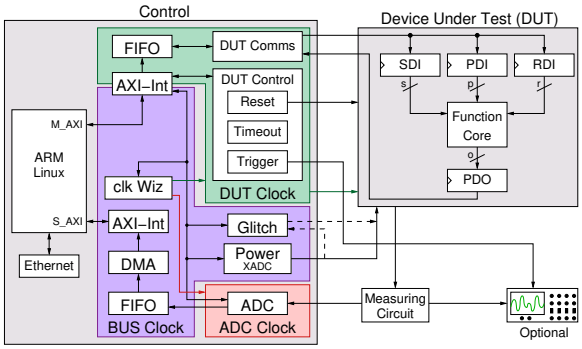


**Figure 3: FOBOS 3 architecture**

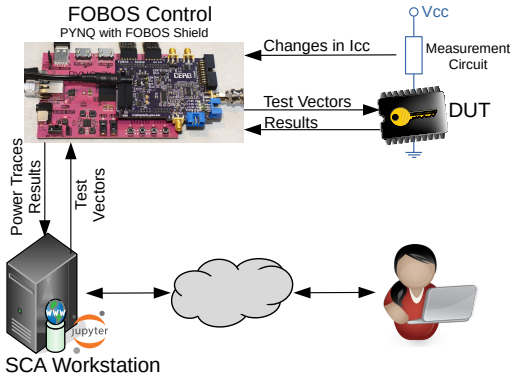


**Figure 4: Remote access to the FOBOS platform**

## 3.3 Laboratory Exercises

The lab consists of three parts, data acquisition, correlation power analysis and side-channel leakage assessment. For each part, the students get a copy of a Jupyter worksheet which contains detailed instructions, Python code, and several questions. The lab report that the students will have to submit at the end of each part consists of the answers to the questions. We included several questions in these notebooks asking for student feedback. The feedback is analyzed in Sect. 4.1.

*3.3.1 Data Acquisition:* In this lab the students get familiar with FOBOS, the SCA setup, and the JupyterLab environment. In particular, the students learn the relationship between samples and traces as well as sampling frequency and DUT frequency. After configuring the FOBOS acquisition settings, such as samples per trace, number of traces, ADC and DUT frequencies, and ADC gain, FOBOS will generate test vectors (plain text inputs), run the data acquisition and plot the waveform of the changes in current consumption of the DUT. Using their knowledge of AES, they have to determine which samples correspond to the *interesting* clock cycle, i.e., the point of attack, and re-run the acquisition step for those samples with 10,000 or more traces. Students will experience that the power traces look slightly different from each other due to noise and that each run will yield again slightly different results due to different test vectors used. An example is shown in Fig. 5.

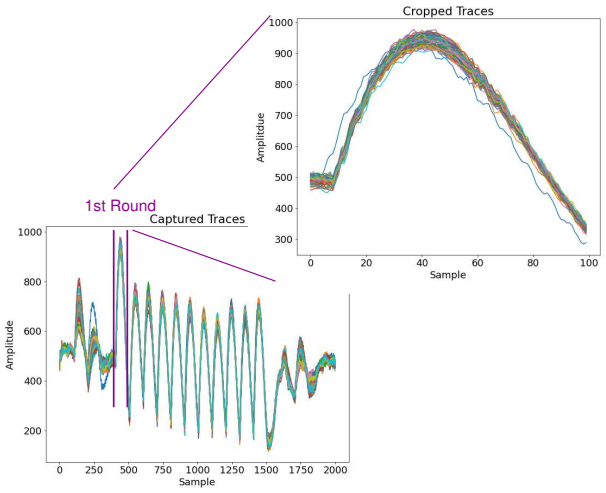


**Figure 5: Power traces of AES-128 with crop to the "interesting" clock cycle**

*3.3.2 Data Analysis - Correlation Power Analysis (CPA):.* Students use the power traces collected in the first part of this lab for CPA. First, they have to learn how to generate a hypothetical power model for the point of attack based on hamming distance and perform this calculation manually for one key-guess, one sub-key, and one plaintext input. After the students crop the power traces more closely to the point of attack (Fig 5), FOBOS computes the power model and uses Pearson's product moment correlation coefficient to generate *Measurements to Disclosure (MTD)* graphs as well as *Correlation* graphs. Examples are shown in Fig 6 and Fig 7 respectively. MTD graphs show on the x-axis the number of traces analyzed, and on the y-axis in blue, the highest and the lowest correlation value found for any key guess except the presumed correct one for that many analyzed traces. Additionally, they show in red the correlation for the assumed correct key guess. The number of traces where the red curve (presumed correct key) leaves the space between the blue lines (best other key guesses) is the minimum number of traces required to obtain this subkey byte. The Correlation graph shows on the x-axis the sample number, based on the start of the cropped sample window, and on the y-axis the correlation value. The black line is the correlation for the presumed correct key guess.

The graph shows when in time (sample number), the correlation value is the highest. Students will experience that the sample number for the highest correlation differs for different key bytes due to propagation delay differences in the AES implementation and that MTD differs for different key bytes due to noise.

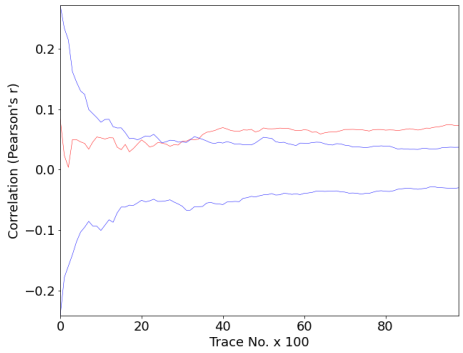


**Figure 6: Measurements to Disclosure**

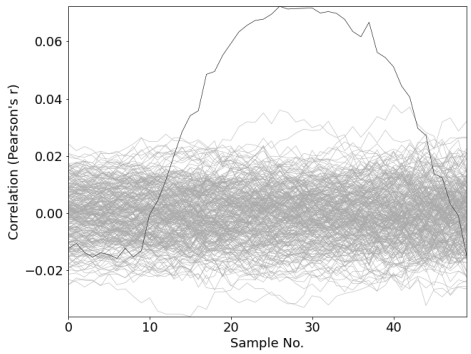


**Figure 7: Correlation Graph**

*3.3.3 Side-Channel Leakage Assessment:* Students get familiar with the Test Vector Leakage Assessment (TVLA) methodology based on Welch's T-Test. Using their knowledge from part 1, they configure FOBOS's data acquisition. FOBOS will produce fixed and random test vectors interleaved at random and perform the data acquisition. Then it performs the T-Test on the recorded traces and presents the results in graphical form. Students run these steps multiple times to examine the effects of the number of traces used and the speed with which the samples are being taken (number of samples per DUT clock cycle) on the result of the T-Test. Students are also asked to investigate any correlation between the T-Test results and the operations of AES to determine during which clock cycle information is leaking.

# 4 ANALYSIS AND DISCUSSION

## 4.1 Student Feedback

Feedback from students was requested for the data acquisition and the data analysis part in all offerings of these labs. The leakage assessment part only requested feedback in the Spring 2021 and Fall 2021 offerings. For each lab part, we asked the students what concepts were the hardest to understand and which step of the lab took the most time. The feedback is summarized below.

*4.1.1 Data Acquisition:* The relationship between the number of samples and clock cycles of AES (samples per trace) was the hardest to understand concept. This directly leads to understanding how to determine at which sample in the power traces graph the AES computation starts and at which it ends. This is needed to find the *interesting* clock cycle. The students observed that the power traces vary and found it hard to determine the reason. Students spent most of the time understanding the complex concepts and re-doing the measurements until all settings were correct, including the ADC gain, in order to obtain a good graph.

*4.1.2 Data Analysis:* The hardest to understand concepts in this part of the lab were Pearson's Correlation, the hypothetical power model calculations, and reading the MTD graphs. Determining why the number of samples and the sample number are different for different subkey bytes was also difficult to understand. The most time was spent on computing the power model (HD) for one subkey, one key guess, and one plaintext manually (bonus question), determining the minimum number of traces (MTD) for each subkey byte, and re-running the data acquisition part to get more measurements.

*4.1.3 Leakage Analysis:* Not surprisingly, the most challenging concept to understand was Welch's T-Test. However, also hard to understand was how to match the peaks of the T-Test to the clock cycles of AES and how sampling frequency affects the T-values. Most time spent in this part was on running the T-Test multiple times for different frequencies and comparing the results.

*4.1.4 Overall:* These lab part-specific questions were followed up with some questions on their experience with the laboratory exercises overall. Table 1 summarizes the results to the question of "Do you think it was worth it accessing actual hardware and performing real physical measurements or would a simulation be sufficient for you to understand the details and difficulties of correlation power analysis?" A majority of students thought it was worth accessing actual hardware, even though it was remotely. Five student groups thought that the lab was a simulation. For IT 435, the answer to this question was yes unanimously. The students in this course were in the lab and could see the hardware setup.

**Table 1: Was it worth accessing actual hardware**

| Semester | Yes | No | Undecided | Simulation |
|---|---|---|---|---|
| Spring 2020 (CYSE/ECE 476) | 7 | 3 | 2 | |
| Spring 2021 (CYSE/ECE 476) | 12 | 4 | 2 | 5 |
| Fall 2021 (IT 435) | 7 | | | |

The biggest hurdles the students had to overcome were for CYSE/ECE 476: Scheduling time with a partner and that only one person at a time can access the Jupyter netbooks, re-reading instructions and lecture notes / slides to understand the concepts, and verifying that all parameters are correct. Students suggested that the labs could be improved by providing more detailed explanations

on core topics such as Pearson's Correlation, MTD, CPA, and TVLA. Some questions need to be formulated more clearly. Some students suggested that a method should be found so that they can confirm the outcomes or at least have solutions available for earlier parts before starting the next part to avoid repeating earlier mistakes. Also, for CYSE/ECE 476, a demonstration in class before starting the labs was suggested.

## 4.2 Course Outcome

Table 2 shows the time students reported to have needed for completion of the three lab parts computed across all three course offerings. While the average times are between 3 to 5 hours per lab and the medians show an even more encouraging 2 to 3.5 hours per lab, the standard deviation of up to 3.72 shows a large spread on how long students took. On the other hand, self-reported times have to be taken with a grain of salt, as the unrealistic minimum and maximum times show.

**Table 2: Time Spent in Hours**

| Lab Part | Avg. | Median | Std. | Min | Max |
|---|---|---|---|---|---|
| Data Acquisition | 3.51 | 2.00 | 3.35 | 0.50 | 16.00 |
| Data Analysis | 4.64 | 3.50 | 3.72 | 1.00 | 20.00 |
| Leakage Assessment | 2.62 | 2.00 | 1.57 | 0.80 | 6.50 |

Tables 3 shows the points the students have achieved on each lab part computed across all three course offerings. All labs were graded out of 100 points. The *data analysis* lab has a 10 point extra credit question. Here the averages and medians are close to full points, and the standard deviation is low, indicating that most students completed the labs successfully.

**Table 3: Points Achieved**

| Lab Part | Avg. | Median | Std. | Min | Max |
|---|---|---|---|---|---|
| Data Acquisition | 96.8 | 98.1 | 6.7 | 35.2 | 100.0 |
| Data Analysis | 102.7 | 106.1 | 8.8 | 69.0 | 110.0 |
| Leakage Assessment | 98.3 | 100.0 | 3.5 | 81.6 | 100.0 |

## 4.3 Lessons Learned

Several student groups pointed out that accessing actual hardware and not just running a simulation has several benefits. Hardware shows that the theory learned is actually working. While many aspects could have been simulated, real hardware leads to less consistent results, and understanding these variations is essential. The ability to change many parameters and observe the outcomes was also appreciated, leading to a deeper understanding.

Some students pointed out that they enjoyed working with real hardware and that the current CYSE degree program does not offer enough work on physical hardware. The reason for this sentiment, as reported by students, is that performing real physical measurements rather than a simple simulation gave them more confidence that the concepts learned in class will work in real life and that correlation mathematics does account for noise and other variations. This confidence translated for several students into higher motivation and more effective learning.

# 5 CONCLUSIONS

We presented our experiment in augmenting information security courses with labs based on an online accessible shared SCA setup. This lowers the cost per student, allows the convenience of remote access, and allows social distancing when necessary. The setup has been used successfully to support three course offerings in two universities. Further, we developed teaching materials and analyzed student feedback on the advantages, challenges, and proposed improvements to this offering. FOBOS 3, including source code, hardware schematics, and educational material is available at https://cryptography.gmu.edu/fobos.

With reasonable effort, such a setup can support massive-online courses with a large number of students. A pool of instances can be dynamically allocated to students to achieve this. We believe this to be interesting future work.

# ACKNOWLEDGMENTS

This work was partially funded by the Commonwealth Cyber Initiative (CCI), an investment in the advancement of cyber R&D, innovation and workforce development (Grant no. E2237761).

# REFERENCES

[1] Abubakr Abdulgadir, William Diehl, and Jens-Peter Kaps. 2019. An Open-Source Platform for Evaluation of Hardware Implementations of Lightweight Authenticated Ciphers. In *2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. IEEE, Cancun, Mexico, 1–5. https://doi.org/10.1109/ReConFig48160.2019.8994788

[2] Aydin Aysu. 2019. Teaching the Next Generation of Cryptographic Hardware Design to the Next Generation of Engineers. In *Proceedings of the 2019 on Great Lakes Symposium on VLSI - GLSVLSI '19*. ACM Press, Tysons Corner, VA, USA, 237–242. https://doi.org/10.1145/3299874.3317994

[3] Matthew Carter. 2018. *Enabling a Control System Approach to Side-Channel and Fault Attacks*. Master's thesis. George Mason University, Fairfax, VA.

[4] John A. Chandy, Zhijie Shi, Mark Tehranipoor, Megan Welsh, Chujiao Ma, Ujjwal Guin, and Qihang Shi. 2015. Hardware Hacking: An Approach to Trustable Computing Systems Security Education. *Journal of The Colloquium for Information Systems Security Education* 3, 1 (June 2015), 1–19.

[5] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. 2011. A Testing Methodology for Side-Channel Resistance Validation. In *NIST Non-invasive Attack Testing Workshop*. Nara, Japan.

[6] Hendra Guntur, Jun Ishii, and Akashi Satoh. 2014. Side-Channel AttacK User Reference Architecture Board SAKURA-G. In *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*. IEEE, Tokyo, Japan, 271–274.

[7] NewAE Technology Inc. 2019. *CW1101: ChipWhisperer-Nano*. https://media.newae.com/datasheets/NAE-CW1101_datasheet.pdf.

[8] Project Jupyter. [n. d.]. Jupyter. https://jupyter.org/.

[9] Paul C. Kocher. 1996. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology — CRYPTO '96*, Neal Koblitz (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 104–113.

[10] NewAE. 2014. OpenADC Product Datasheet. http://newae.com/files/openadc-datasheet.pdf.

[11] Colin O'Flynn. 2017. *A Framework for Embedded Hardware Security Analysis*. Ph. D. Dissertation. Dalhousie University, Halifax, Nova Scotia.

[12] Rambus. 2019. DPA Workstation Analysis Platform - Rambus. https://www.rambus.com/security/dpa-countermeasures/dpa-workstation-platform/.

[13] Riscure. 2019. Side Channel Analysis Security Tools. https://www.riscure.com/security-tools/inspector-sca/.

[14] Patrick Schaumont. 2021. Socially-Distant Hands-On Labs for a Real-time Digital Signal Processing Course. In *Proceedings of the 2021 on Great Lakes Symposium on VLSI*. ACM, Virtual Event USA, 425–430. https://doi.org/10.1145/3453688.3461490

[15] Carina Wiesen, Steffen Becker, Marc Fyrbiak, Nils Albartus, Malte Elson, Nikol Rummel, and Christof Paar. 2018. Teaching Hardware Reverse Engineering: Educational Guidelines and Practical Insights. In *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. 438–445. https://doi.org/10.1109/TALE.2018.8615270

[16] Xilinx. 2021. http://www.pynq.io/.