Hands-on Labs for Secure Programming on Modern Trusted Platforms

Yuzhe Tang
ytang100@syr.edu

Wenliang Du
wedu@syr.edu

Syracuse University
900 South Crouse Ave, Syracuse, NY 13244

Abstract - With the increasing awareness of cyber-security issues, cyber-security workforce becomes an urgent societal need. Of particular importance is the development skills of building secure applications. To meet the educational demand, we propose to develop hands-on labs and education tools for "secure development on modern trusted platforms". This work focuses on two emerging trusted platforms, that is, trusted execution environments (e.g., recently released Intel SGX CPU) for secure application hosting on a third-party cloud (e.g., Amazon), and the Blockchain technology that underlies the Bitcoin and other cryptocurrencies by trustworthy data recording. We develop two sets of lab modules, respectively for Intel SGX and Blockchain. The SGX labs address the necessary skills and techniques on software partitioning, SGX memory protection, side-channel security and software attestation. For Blockchain, we build an education tool enabling the integration of Blockchain in students' course-taking experience. We also develop two Blockchain labs on transaction programming and logging applications. Through evaluation, it is shown that the project helps improve students' interest in SGX and Blockchain, and helps them develop secure applications on these platforms.

Keywords

Secure development, Blockchain, TEE, SGX

1. INTRODUCTION

As our daily lives increasingly rely on computer technology, it is no news that cyber-security issues, manifesting in forms such as cyber attacks and hacks, raise societal awareness and concerns at an unprecedented level. Cyber-security workforce capable of mitigating and defending the attacks is in dire needs and the shortage of cyber-security workforce becomes real — it is estimated that at least one million cyber-security positions are left unfilled in the United States [12]. To bridge the gap, the key lies in the education of cyber-security skills of building secure systems and defending real-world attacks.

Two trusted security platforms emerge recently, that is, 1) trusted execution environments (TEE) and 2) Blockchain. Concretely, 1) commercial TEEs become real recently and are offered by different vendors, ranging from Intel's Software Guard eXtension (SGX [10]), ARM's TrustZone [2], to AMD's encrypted memory [1]. SGX, as available in the 6th generation Intel Skylake CPU, allows one to set up a trusted execution environment (called "enclave") in an otherwise untrusted host. The canonical application is to secure the public clouds such as Amazon EC2 and Google Cloud Platform (note that SGX is offered on GCP [8]), and to enable these cloud platforms to securely process sensitive data for clients. 2) Blockchain is the technology that underlies Bitcoin [4], Ethereum [7] and other crypto-currencies. The success of crypto-currencies has shown Blockchain's potential to assume the role of trusted third-party and to enable many applications beyond crypto- currencies. Overall, both platforms have the potential to solve the long lasting lack-of-trust problem in our society and to disrupt many relevant industries, such as finance technology, public cloud computing, banking industry, supply-chain, etc. To realize this potential, it is critical to educate the skills of building security applications on these platforms.

In this work, we develop hands-on lab modules and education tools on Intel SGX and Blockchain, with the goal of teaching the application development on these platforms.

For the design of SGX labs, we focus on the unique techniques required in developing SGX applications. Concretely, the architecture of SGX features two execution environments, the trusted world (or so-called enclave) and untrusted host. Developing any applications on SGX requires partitioning the software to two pieces, respectively executed in the two worlds. The first development skill required on SGX is software partitioning and placement. SGX applications normally run in a third-party cloud-like platform. To set up the enclaves securely, it requires running software attestation procedure [13]. The second development skill is the use of software attestation. Given that the SGX architecture is known to be vulnerable to advanced side-channel attacks [22, 14], guarding the program execution against the attacks is important, especially for running critical applications in enclave. The third development skill is techniques for defending side-channel attacks on SGX.

The learning objective of our SGX labs focuses on the skills on software partitioning, software attestation and side-channel security. We propose three SGX labs accordingly: The first lab (§2.1) requires students to partition enclave computation and to observe the performance difference under different placement. This also allows students to observe and understand the performance implication of SGX memory protection. The second lab (§2.2) requires students to invoke software attestation procedures properly when calling remote enclaves. The third lab (§2.3) teaches students to understand and defend side-channel attacks on SGX, where they can observe the information leakage of different sorting algorithms under the side-channel attacks and are asked to choose a side-channel secure algorithm.

For the Blockchain part, we propose to first build a Blockchain based education platform, and then to develop Blockchain hands-on labs. The two tasks are interrelated, as the first task helps build interests among students that facilitates the second task. The two tasks are built on a local Blockchain platform that runs in a campus environment (§3). By this means, we can control the size of network and adjust the difficulty level of Blockchain mining for students to mine coins. The Blockchain education tools (§3.1) address the use of Blockchain for standard education workflow, with the goal of enabling

novel applications. In this paper, we report one application for allowing students to earn coins and to buy late homework submission (§3.1). On the local Blockchain platform, we develop two Blockchain labs, respectively for Blockchain transaction programming (§3.2.1), and for Blockchain applications (§3.2.2).

We have used the developed labs and tools in residential courses and educational workshops. The project has resulted in the hands-on labs and relevant materials packaged in the forms such as Virtual Machine Images, and has been disseminated through the SEED education platform [18].

2. SGX LABS

Hardware enclave is new security-oriented computer architecture and is adopted in the recent commercial processors – a notable example is the Intel Software Guard eXtension or SGX [10]. The hardware-enclave architecture allows to set up a trusted execution environment (TEE) on an otherwise untrusted host (e.g., the public cloud instance). Hardware enclaves have been deployed in the cloud service provider. For instance, Google Cloud Platform provides machines with Intel SGX [8]. With the availability of hardware platform, it is in dire need to develop software systems leveraging the platform and to support various security applications in healthcare, finance, smart-home, etc.

To meet this need, the key is the understanding of the hardware enclave architecture from security perspective. In particular, cryptography is the core technique to enable security in hardware enclave: At the hardware level, cryptography is used to ensure 1) the security of setting up the trusted "enclave" on an untrusted remote machine, a process called remote software attestation, and 2) the security of running the software program in the enclave, which is provided by enclave-memory protection. At the software level, advanced cryptographic protocols (more specifically, oblivious RAM [19]) are needed to provide 3) strong program-execution security under the known side-channel attacks [22].

This teaching topic is for students to understand the intricate security of hardware enclaves in these three aspects mentioned above. We propose three hands-on labs respectively for the memory protection (§2.1), software attestation (§2.2) and side-channel security (§2.3). In designing the hardware-related labs, we adopt a software-simulation approach that has advantages in easy dissemination and exposing low-level events to developers.

2.1 Lab 1: Understanding Memory Protection in SGX.

The lab provides to students a programming environment in a VM image pre-installed with Intel SGX SDK. The lab pre-requires the programming skills on Intel SGX, including the concept of ECalls/OCalls [10] that allows a program to call outside/inside the enclave a function inside/outside the enclave. Of particular importance is the argument passing in ECalls/OCalls where there are options to pass arguments by reference or by data.

In Intel SGX, there are three computing modes with respect to the placement of data and code: 1) program outside enclave accessing data outside enclave, 2) program inside enclave accessing data outside enclave, and 3) program inside enclave accessing data inside enclave. This lab requires students to run the program in all three modes and observe the difference in execution time.

Experiment results: Table 1 shows the performance result reported by one of the students who did the lab in the last author's course. The performance difference of the same computation under different execution modes is clearly shown in the table.

Table 1
Lab 1 performance results

Mode	**inside**	**outside**	**outside(data)**
Execution time (us)	115	45	76

2.2 Lab 2: Software Attestation in SGX.

Software attestation is an enclave authentication scheme that convinces a client that the server-side enclave is loaded with the authentic client-provided image and is hosted by a trusted processor in the server. Software attestation is a fundamental service in establishing a trusted execution environment (TEE) in a distributed system. It can be combined with TLS style authenticated key exchange to set up shared secrets and establish a secure channel. At its core, the software attestation produces a verifiable proof about the binding between software image and trusted processor. Internally, the proof is produced by the trusted processor signing the measurement of the software image.

The learning objective of this lab is the use of SGX software attestation for remote applications. Students will be able to write programs that call proper functions for enclave attestation in SGX machines. The lab focuses on the interfaces and functions for calling software attestation services in Intel SGX SDK. The lab treats the internal of software attestation as a black box and hides from students the various internal mechanisms (e.g., quoting enclaves for signing and measurement).

The lab requires students to write programs to set up a secure channel between two enclaves and to issue remote procedure calls securely. This lab includes two exercises: 1) to implement an argument-encoding scheme when establishing shared secrets and secure channel, 2) to implement new applications functions in the attested enclave.

2.3 Lab 3: Side-Channel Attacks and Defense in Enclave.

Side-channel vulnerability is common in the trusted execution environment and particularly in the hardware enclaves there are known and emerging attacks exploiting the memory access pattern to disclose secret memory data [22, 14]. The defense of such attacks is mainly left to the job of software developers [9]. This raises an education need of side-channel attacks and defense which is critical to the secure development of software systems in SGX.

The learning objective is an in-depth understanding of the root cause of side-channel vulnerability, and the skills to develop practical side-channel attacks and effective defenses. In SGX, the side-channel security of an in-enclave computation refers to whether an adversary outside the enclave will be able to infer the enclave secret data from the side-channel observation, such as memory access pattern. To defend the attacks, the side-channel security of a computation depends on whether the computation accesses memory data in an "oblivious" fashion. Informally, data obliviousness means the data-access of a computation should be the same no matter what input values it takes, thus a data-oblivious computation decouple the secret data from the data-access trace.

This lab requires students to observe data obliviousness of classic sorting algorithms including bubble sort, merge sort, etc. Based on the understanding of obliviousness, the lab further requires students to design concrete attacks and defense strategies against enclave sorting. Following this design, the lab includes five exercises: The first two ("Exercise 1/2: Data Obliviousness of Merge/Bubble sort") are about understanding the "data obliviousness" of two textbook sorting algorithms; In particular, in the merge sort [15], the merge phase (merging multiple sorted arrays) accesses data non-obliviously and is vulnerable under side-channel attacks. The exercise asks students to observe the memory-access pattern of the merge under different input arrays. The next two exercises ("Exercise 3: Attacking Merge sort" and "Exercise 4: Protecting bubble sort") are about designing and developing functional attack strategy and defense mechanisms for sorting algorithms. The exercise is based on the understanding (in)security of different sorting algorithms and asks students to choose the "secure" sort algorithm under side-channel attacks. The last exercise is about applying the learned analysis skills to computation beyond sorting.

3. BLOCKCHAIN TOOLS AND LABS

The Blockchain technology has been receiving an increasing amount of attention and has been revolutionizing the society in the financial industry and beyond. At its core, the Blockchain provides a trustworthy third-party to store transactions in a way that cannot be controlled by individual entities including

government. Cryptography plays a key role in enabling the third-party trustworthiness — The storage of transactions, or Blockchain as a ledger, leverages cryptographic hash (a one-way function) for the transaction commitment, and writing of a transaction to Blockchain leverages the hardness in solving hash puzzles (i.e., the inverse of calculating a hash, called "mining") to ensure security in a world without identity. This teaching topic aims at teaching the application of cryptography in Blockchain. It includes an education tool based on Blockchain and two hands-on labs.

The required concept of Blockchain is covered in the online lecture and online textbook [21, 3].

On-Campus Blockchain Education Platform: To start with, our goal is to build a Blockchain-based education platform that enables 1) curriculum and hands-on lab development on Blockchain, and 2) novel education applications using Blockchain.

To enable students' interaction with Blockchain, we deploy an Ethereum network on campus, leveraging idling machines in our research laboratories. We have tried using the public Ethereum network in our previous courses; students interested in mining using their personal laptops were eventually frustrated by the difficulty of winning it. Opting in the on-campus Ethereum network, it is much easier to mine a coin from the Blockchain.

3.1 Tool: ChainGrader and Late Homework Submission

The first application on our on-campus Blockchain platform is ChainGrader that grades student's homework submission and enforces late-submission policy.

Motivation: One of our motivating scenarios is to allow students to budget their 'points' and enable late homework submission. To be specific, if a student spends time on mining and obtains sufficient coins, she can spend the coins to buy homework late submission without penalty. This will motivate students who are late in homework preparation to still learn through the process.

Developed Tool: We have built ChainGrader by running smart-contract programs the on-campus Blockchain. Briefly, the ChainGrader smart contracts support function calls from the instructor (including TA) and students. The instructor can set the standard solution and students can submit their homework solution to the ChainGrader. The ChainGrader has a data-feed contract to be aware of the current time, and categorizes the students' submission as standard and late. In the case of late submission, the ChainGrader would request the student to issue a transaction to pay for the late submission. Students can do the mining using their laptops to earn themselves coins.

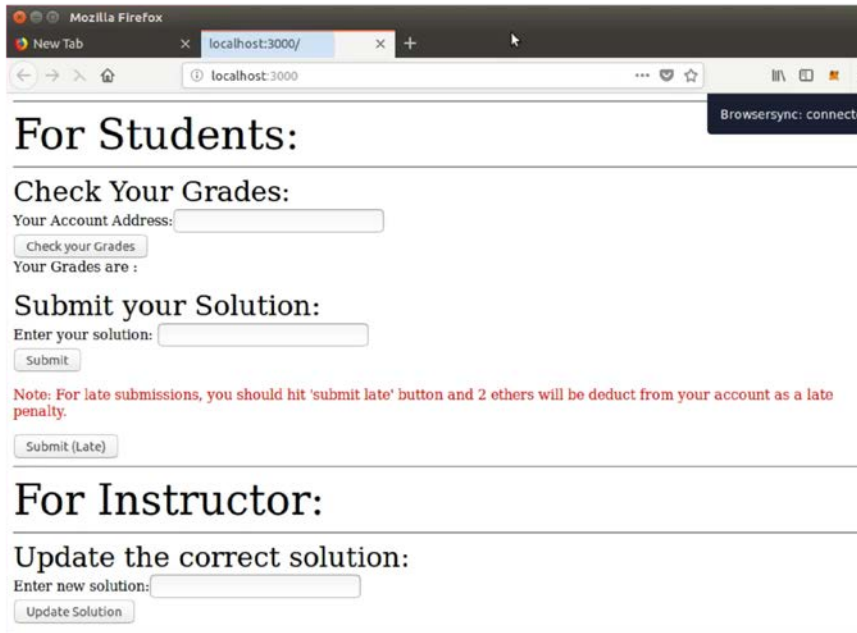

Figure 1: ChainGrader GUI on the web.

3.2 Blockchain Labs

The first application on our on-campus Blockchain platform is ChainGrader that grades student's homework submission and enforces late-submission policy.

3.2.1 Lab 4: Blockchain Interfaces for Transaction Storage

The Blockchain provides the storage of transaction history in cryptocurrency and exposes interfaces for reading/writing transactions. In this lab, the learning objective is for students to have a hands-on experience and in-depth understanding of Blockchain's transaction-storage interfaces.

This lab consists of four exercises: The first three exercises explore the transaction interface exposed by a Blockchain. The last exercise requires designing off-chain programs to visualize the Blockchain transaction history. In these exercises, students are given the primitive functions such as read/write a specific transaction and block, create an account, etc. They are asked to implement an off-chain program composed of these primitive functions to realize the requested functionalities. These functionalities include checking coin balance after mining, listing the selected transactions on Blockchain, sending transactions and observing the transaction life cycle.

3.2.2 Lab 5: Blockchain Application for Logging

The Blockchain technology, while whose main application is storing transaction history in cryptocurrency, can be used in many other application domains. One common paradigm is to repurpose Blockchain for logging target applications and to increase the transparency. In this lab, the setting is to use Blockchain to log the operation trace of a remote file service. The learning objective for the lab is for students to understand, design and implement the non-cryptocurrency application of Blockchain for logging distributed services.

The lab provides students programs that simulate a remote file storage service where client users access/write files stored on the remote server. The access path includes classic information-security (InfoSec) protocols for permission-based access control and password authentication.

The lab consists of four exercises: 1) Password authentication and user login, 2) Permission-based access control, 3) Implementing Blockchain logging services, and 4) Instrumenting target systems to trace user login and access requests in Blockchain.

3.3 Project Outcomes

The project outcome for each module includes the followings: 1) Lab description: For each lab module, we will provide detailed descriptions of lab tasks, as well as guidelines. This is the main document used by students. 2) Web page: For each lab module, we create a web page, which contains the lab description, and files needed, along with some additional readings. Instructors can link this web page to their course page if they want to assign the lab to students. Our web page is invited to be posted to the website of a nationwide education platform, called SEED labs [17, 18]. 3) Instructor manual: For each module, we will produce an instructor manual, which provides step-by-step instructions, including sample code. Instructors can follow the instructions to set up the lab sessions when preparing for their classes. Instructor manuals will only be provided to instructors upon requests. 4) Virtual-Machine (VM) preparation: All the needed software and configuration for the lab modules will be pre-installed on the SEED VM, so instructors and students only need to download our SEED VM, without the hassle of installation.

4. RELATED WORK

While there are many Blockchain courses offered from different institutes, only two courses include hands-on labs for Blockchain. The labs from UMD [16] are mostly designed for smart-contract programming. While they are good for program security, they lack the necessary context in finance, supply chain, information security, etc. The labs from Princeton [20] teach the internal mechanism of Blockchain (e.g., proof-of-work mining) and feature a series of small programming tasks to simulate the mechanisms. Our labs are designed to emphasize the real-world Blockchain systems by focusing on the interfaces.

The SGX training labs from Intel [11] cover various aspects of SGX enclave development including design and build enclaves, interact with enclaves, maintain state across power events, and debugging enclaves. Our SGX labs complement the Intel labs in addressing the side-channel security, performance understanding, and software attestation.

5. EVALUATION

5.1 ChainGrader

Evaluation: We have used ChainGrader in a course taught in Spring 2018 ("CIS644/488: Internet Security"). The course is targeted for graduate students in the computer-science program. The use of ChainGrader is given to students as an option, but not required. There are 17 students who opt-in (out of 56 total students in the class) and are active in mining. Among them, 7 students actually use the coins they mine for late submission. There are 10 homework assignments in the course and the following is the distribution of the length of late submission among the 10 homework assignments.

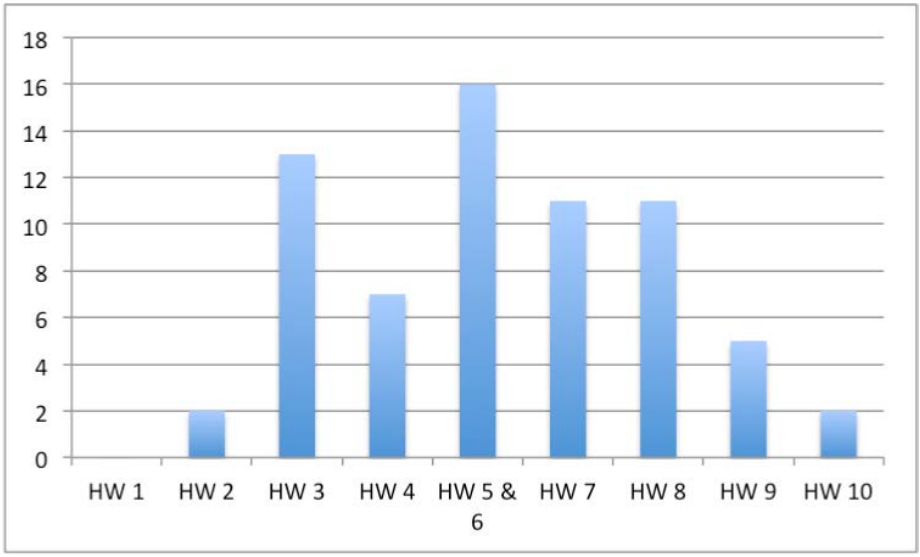

Figure 2: ChainGrader evaluation in adoption.

As reported by students, the ChainGrader allows the students "with many courses and hectic schedules to be able to catch up the course schedule with less pressure because of the extension purchases."

5.2 Lab Evaluation

We have used and evaluated our labs in two settings: 1) a graduate course "CIS700 Information Security and Privacy" [5] which is intended for PhD and master students in the computer science program, and 2) a lab tutorial [6] invited to present in the SEED workshop 2018 which is intended for cyber-security educators in colleges. The CIS700 is a small course consisting of six students. In the SEED workshop 2018, the audience consists of 90 people, majority of whom are college teachers in computer science, business, etc. The central theme of the workshop is to disseminate reusable lab modules in cyber-security education for the adoption nationwide.

In both settings, we send exit surveys to the audience after they finish the lab exercises. We collect data and report them in Figures 4, 5 and 3.

Figure 3 shows the evaluation results of Blockchain lab (Lab 4) in the SEED workshop. This is a 90-min session where the last author presents an introductory lecture before the lab, since most audience may not have enough the background on Blockchain. The result shows that most attendee reports the lab/lecture is effective and worth their time. Lesson: However, it shows the two ratings related to adoption are low; we suspect the reason is that the Blockchain lab requires extra efforts in setting up environments to run Ethereum locally on campus, which may intimidate peer teachers to adopt.

Figures 4 and 5 show the evaluation results of Intel SGX lab (Lab 1) and Blockchain lab (Lab 4). In this course, we present lectures about Intel SGX and Blockchain in details prior to the two labs. Students are instructed to do the lab in classroom, with the help of instructor. They are consulted by an exit survey

after they finish and submit the lab. The survey results are shown in Figure 3, 5 and 4.

For Lab 1 results, students are content with the lab clarity and effectiveness. However, the lesson we learned is that the lab prerequisite on Bash scripting and C/C++ programming is lacking in some of students, which makes them think the lab is difficult.

For Lab 4 results, students exhibit similar concerns on the lab pre-requisite on programming and difficulty levels. In addition, the lack of interests is more severe in this lab. We will add needed programming tutorials to the lab to make students more comfortable. In addition, we will add more real-world scenarios to relate students to the lab and spark their interest.

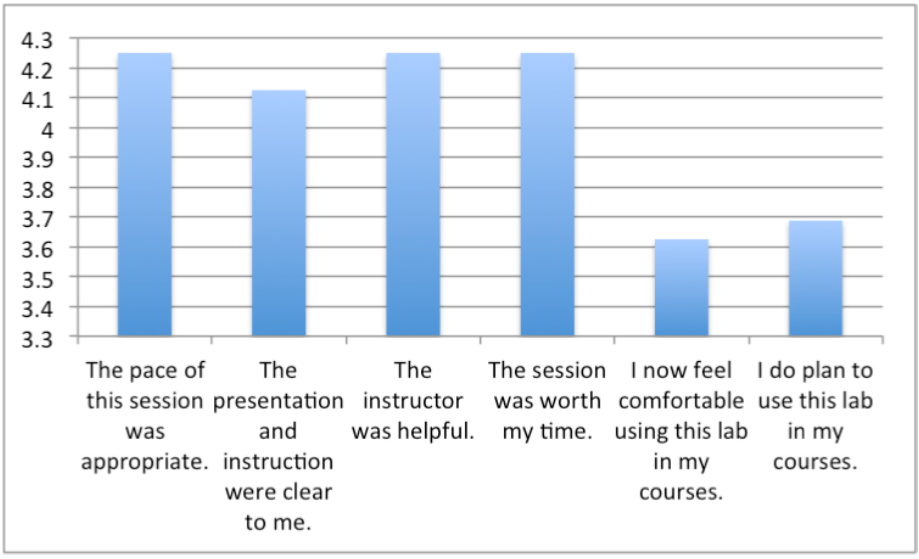

Figure 3: Lab 4 evaluation (SEED Workshop).

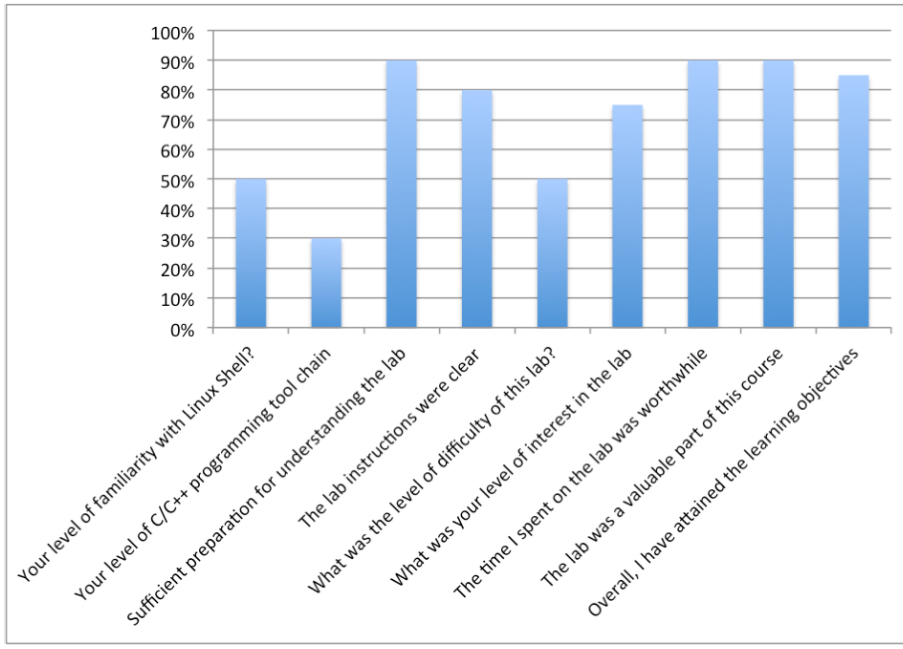

Figure 4: Lab 1 evaluation (CIS700).

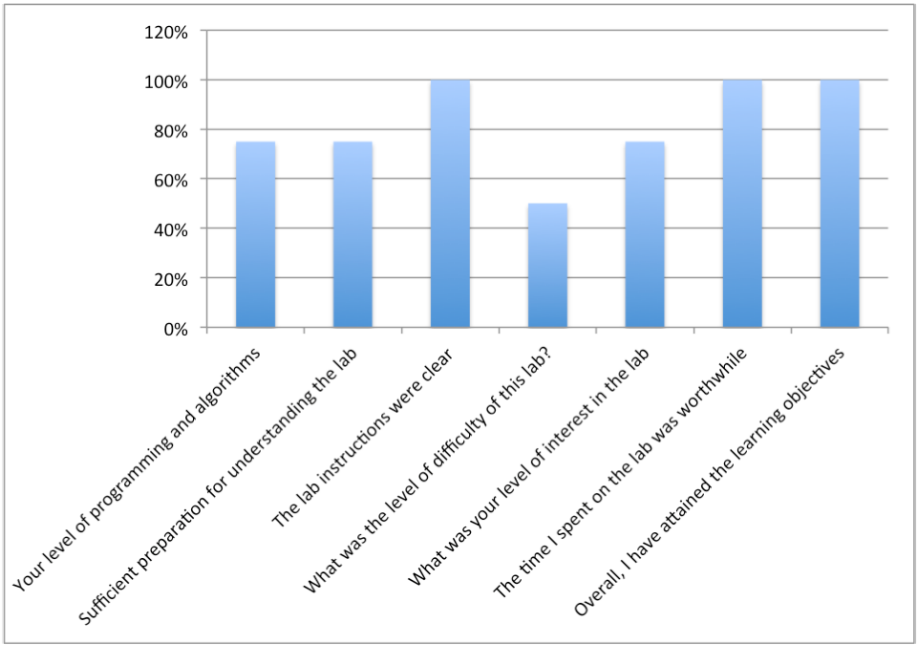

Figure 5: Lab 4 evaluation (CIS700).

6. CONCLUSION

This paper presents a suit of hands-on labs and education tools built on trusted platforms, including Intel SGX and Blockchain. The education materials aim at the secure application development skills in high demand in the job markets. The target education goals include partitioning software in enclave, software attestation and side-channel secure development (on Intel SGX), as well as transaction programming and trusted logging applications (on Blockchain). We also propose an education tool based on Blockchain, enabling new education applications and exposing students to Blockchain. The lab

modules and education tools are used in residential course teaching and a nationwide education workshop targeted for cyber-security educators. The results show that the education tool can improve students' learning experiences, effectively reducing pressures with busy course taking schedule. The lab modules spark students' interests in learning new techniques and modern platforms.

REFERENCES

[1] Amd memory encryption, https://developer.amd.com/amd-secure-memory-encryption- sme-amd-secure-encrypted-virtualization-sev/.

[2] ARM TrustZone, https://www.arm.com/products/security-on-arm/trustzone.

[3] Bitcoin and cryptocurrency technologies: http://bitcoinbook.cs.princeton.edu/.

[4] Bitcoin: https://bitcoin.org/en/.

[5] Cis700 information security and privacy: https://goo.gl/JV66GN.

[6] Dr. tang's blockchain lab session in seed workshop 2018: http://www.cis.syr.edu/ wedu/seed/work-shop_agenda2018.html.

[7] Ethereum project: https://www.ethereum.org/.

[8] Google cloud platform supports intel sgx.

[9] Intel sgx and side-channels.

[10] Intel sgx programming reference, 2014 no. 329298-002, https://goo.gl/n9DgHX.

[11] The intel sgx web-based training lab bundle, https://software.intel.com/en-us/download/the-intel-sgx-web-based-training-lab-bundle.

[12] One million cybersecurity job openings in 2016: https://www.forbes.com/sites/stevemorgan/2016/01/02/one-million-cybersecurity-job-openings-in-2016/#25c65db627ea.

[13] R. Bahmani, M. Barbosa, F. Brasser, B. Portela, A. Sadeghi, G. Scerri, and B. Warinschi. Secure multiparty computation from SGX. IACR Cryptology ePrint Archive, 2016:1057, 2016.

[14] J. V. Bulck, N. Weichbrodt, R. Kapitza, F. Piessens, and R. Strackx. Telling your secrets without page faults: Stealthy page table-based attacks on enclaved execution. In E. Kirda and T. Ristenpart, editors, 26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18 2017., pages 1041-1056. USENIX Association, 2017.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Magazine.

Introduction to Algorithm 3rd Edition. MIT Press, 2009.

[16] K. Delmolino, M. Arnett, A. E. Kosba, A. Miller, and E. Shi. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In J. Clark, S. Meiklejohn, P. Y. A. Ryan, D. S.Wallach, M. Brenner, and K. Rohloff, editors, Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers, volume 9604 of Lecture Notes in Computer Science, pages 79–94. Springer, 2016.

[17] W. Du. The SEED web page: http://www.cis.syr.edu/~wedu/seed/.

[18] W. Du. The SEED project: Providing hands-on lab exercises for computer security education. IEEE Security and Privacy Magazine, September/October 2011.

[19] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious rams. J. ACM, 43(3):431–473, 1996.

[20] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016.

[21] A. Narayanan, J. Bonneau, E. W. Felten, A. Miller, and S. Goldfeder. Bitcoin and Cryptocurrency Technologies - A Comprehensive Introduction. Princeton University Press, 2016.

[22] Y. Xu, W. Cui, and M. Peinado. Controlled-channel attacks: Deterministic side channels for untrusted operating systems. In 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015, pages 640–656. IEEE Computer Society, 2015.