

Adding practical experience to computer security course

Zarko Stanisavljevic  | Pavle Vuletic

Department of Computer Engineering and Information Theory, School of Electrical Engineering, University of Belgrade, Belgrade, Serbia

Correspondence

Zarko Stanisavljevic, Department of Computer Engineering and Information Theory, School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11120 Belgrade, Serbia.
Email: zarko.stanisavljevic@etf.bg.ac.rs

Abstract

Computer security has become one of the core areas according to IEEE Computer Society and ACM Joint Task Force curriculum guidelines for undergraduate degree programs in computer engineering and software engineering. At the University of Belgrade, School of Electrical Engineering there is a course called Computer Security that is taught to software engineering and computer engineering students for the past 9 years. During this time a significant change was made to the course regarding the level of practical students' work. Several factors initiated the change: the recommendations in the curriculum guidelines, the analysis of the students' success at the exams, and the students' opinion. The impact of gradual transition from no practical activities at all to laboratory exercises and practical project at the course are described in this paper. Results show that the introduction of laboratory exercises at the course helped students to better understand some of the core areas and that the introduction of practical project increased their overall knowledge in the computer security area.

KEYWORDS

computer security education, cryptographic algorithms, practical engineering education, teaching/learning strategies, X.509 certificates

1 | INTRODUCTION

Computer security has become one of the most important areas in computer engineering and software engineering. More and more resources are invested in computer security each year, and still more and more security flaws are constantly reported. Therefore, it is obvious that computer engineering and software engineering students need to learn about this area during their studies. Although some computer security topics can be taught through courses like computer networks, internet programming, databases, and e-business infrastructure, the significance of the area indicates that a separate course is needed. IEEE Computer Society and ACM Joint Task Force recent efforts to create curriculum guidelines for undergraduate degree programs in computer engineering [9] and software engineering [10] can be used to support

previous claim. Both guidelines put security among core areas.

At the University of Belgrade, School of Electrical Engineering (UB-SEE) there is a Computer Security (CS) course that is mandatory for the fourth year students at the undergraduate study program in software engineering (SEU) and for the third year students at the undergraduate study program in computer engineering (CEU). The course covers topics from cryptographic algorithms, network security and applications, and system security.

For most of the courses in computer engineering and software engineering curriculum one of the important aspects is hands-on experience for students (as suggested in [9,10]). This kind of experience is usually offered to students through laboratory exercises [2–4,18,20,21,23,24], and practical projects. For complex and constantly evolving areas like

computer security it is a challenging task to select topics that should be covered through practical work and even more challenging to find or design educational systems that can be used for this.

At the CS course practical work is organized through both laboratory exercises and practical project. Laboratory exercises cover cryptography algorithms, one of the topics that is less attractive and cognitively challenging to students and yet very important for understanding other topics. They are conducted using a software system for algorithm visualization called COALA (CryptOgraphic ALgorithm visuAl representation) [22] which improved students' understanding of the most challenging topics in computer security. Several years after the introduction of COALA, practical project was introduced to the course in order to motivate students further and to stimulate their thinking about the real world scenarios where the concepts they learned about were implemented. Unlike the laboratory exercises where a change of topic would require a significant effort in course preparation, because in most cases it would require the use of a new educational system, the topic for the practical project can easily be changed every year. The topic that was initially selected was understanding the structure and methods for generation of X.509 certificates. The rest of the paper is organized as follows. Section 2 gives course description and presents previous and related work, section 3 explains how practical work at the CS course at UB-SEE is organized, section 4 summarizes and discusses impact of practical work introduction and section 5 concludes the paper.

2 | COURSE DESCRIPTION, PREVIOUS, AND RELATED WORK

The CS course is worth 6 ECTS credits and consists of two lecture classes, two problems classes and one laboratory exercises class per week during one 14-week long semester. All classes last for 45 min. The lectures cover theoretical aspects, problems classes show students how some theoretical concepts are deployed in practice, while laboratory classes allow students to implement some of the concepts on their own. Lectures and problems classes are given every week and laboratory exercises are organized three times during the semester, upon explaining sufficient number of theoretical concepts. Final exam is conducted in pencil-and-paper written form. Final grades are ranged from 5 (did not pass) to 10 (excellent). To pass the exam, student should collect more than 50 out of 100 points. To achieve highest final grade 10, student should collect more than 90 points.

The CS course covers topics ranging from cryptographic algorithms to network security, applications and system security. The first part of the course covers the basics of cryptography theory, symmetrical cryptographic algorithms,

and public key cryptography principles. The second part of the course introduces the notion of network security and covers authentication applications, e-mail security protocols, traffic security mechanisms, and internet security principles. The third part of the course covers topics from the system security area with the goal to demystify security threats and how to protect from them.

The rationale behind introducing laboratory exercises to the course and the previous work on this topic was explained in detail in [22]. Before introducing laboratory exercises cryptographic algorithms main concepts (classical, symmetric, and asymmetric cryptography) were explained at the total of six lecture classes (out of 28), and the specific algorithms (e.g., substitution, transposition, production, DES, AES, RSA, DH, etc.) were explained in detail with problems done for these algorithms at the total of fourteen classes (out of 28). By analyzing students results it was possible to isolate cryptographic algorithms as an area that students had trouble to understand, and this fact influenced the overall lower results at the exam. During laboratory classes students use the COALA simulator to follow the execution of cryptographic algorithms and have a test about the simulation using an LMS system [5]. The time devoted to cover this topic at the lecture classes and the problem classes remained the same after the introduction of the laboratory exercises. In [22] the authors showed that introducing laboratory exercises helped students to better understand this area, thus build a better basis for the rest of the material and improve the overall exam results. Since the publication of [22] no similar system or laboratory exercises could be found in the open literature, as far as the authors are aware.

Adding laboratory exercises to the CS course improved students' understanding of cryptographic algorithms, which is one of the core topics in computer security. On the other hand, students still needed practical work that would be close to the real world applications. The authors noticed this problem when they assigned students their bachelor thesis on topics from computer security area. As part of their research, students had to tackle real world problems. After completing their work, the students' feedback was that solving real world problems helped them a lot to better understand the concepts they learned about. That is why a practical project assignment was added to the CS course.

The effect of the practical project in which students are supposed to solve specific problems, using concepts taught at the CS course, was expected to be twofold: to motivate students to find and fill the gaps in their knowledge and to show them an example of the real world scenario they might encounter after they finish their studies.

Kapur [13] showed that groups of students who passed a preparatory problem-oriented phase without instructional facilitation significantly outperformed other groups of students who had traditional lecture and practice or those who passed

instructor facilitated problem-oriented phase. Similarly, Gloger-Frey et al. [7] showed that student teachers who were given paper-based course material spent more time on the follow-up coursework than teachers who were given a well-developed computer-based learning environment. Researchers agree that challenge which is intrinsic to the problem-oriented practical project activity enhances student's motivation to learn [15]. Also a recent study [19] showed that situational interest and curiosity are triggered in those situations where knowledge incompleteness exists. Practical projects efficiently reveal such gaps. However, there is still no agreement on the advantages of discovery learning over direct learning techniques, especially for those students who have low knowledge prior to the problem phase [14].

Computer security education is one of the computer science areas with the highest potential in the future, considering its central point in the technological development. That is why similar courses to the UB-SEE CS course exist all around the world. Most of the courses that are described in the open literature use laboratory exercises to provide hands on experience and most of them are based on some sort of a network laboratory [8,16,17,25]. For example Hill et al. in [8] created a course where they use an isolated network laboratory to teach advanced network security concepts. They divide their students into two teams where one team has a goal to attack the other teams computers, while the other team has a goal to defend their computers. In this way students learn through hands on experience. There are also approaches where the laboratory is developed on top of an operating system like Linux, which allows system security concepts to be taught. In [6] the author developed such a laboratory system and a set of laboratory exercises covering different topics of computer security. The laboratory exercises are organized in three groups: vulnerability and attacks labs, exploration labs, and design labs. In most cases students are presented with a security situation that they need to resolve.

Laboratories described in [8,16,25], and [17] are all designed very well and the papers describing them report excellent results from the students using them, but they cover topics in the field of computer systems, network security, and computer networks which are not the main target of the UB-SEE CS course. Laboratory described in [6] includes some topics that are covered by laboratory exercises in the UB-SEE CS course, but the focus is not on the details of the algorithms. In the open literature the authors did not find any practical project similar to the one assigned at the UB-SEE CS course.

3 | PRACTICAL EXPERIENCE AT THE CS COURSE

Practical work was added gradually to the course. The course was taught for the first time in the school year 2008/2009. In

the school year 2011/2012 we introduced laboratory exercises to the course. In the school year 2015/2016 we introduced a practical project to the course.

3.1 | Laboratory exercises

Some topics described in section 2 are more attractive to students and therefore easier to understand. On the other hand some topics that are crucial to the course are complex and rely on mathematics and are therefore very challenging to students. Based on the analysis of exam results at the course the authors noticed that students had trouble understanding cryptographic algorithms. That is why a set of laboratory exercises covering some of the most important and yet hard to understand algorithms was designed. The selected algorithms were Data Encryption Standard (DES), Advanced Encryption Standard (AES), Diffie-Hellman (DH), and Rivest-Shamir-Adelman (RSA).

In order to make laboratory exercises possible a software system for algorithm visualization was needed. The existing systems were analyzed and none of them met the CS course requirements. That is why a new software system for CryptOgraphic ALgorithm visuAl representation (COALA) was developed. The COALA system and experiences from using it in the CS course are presented in [22].

3.2 | Practical project

As mentioned in section 2, adding laboratory exercises resolved only a part of the issues noticed at the CS course. The other part considering the need of the practical work that would be close to the real world applications still remained unresolved for a couple of years. The general idea behind introducing a practical project was to create a real world scenario around the concepts that are taught at the course, but for which not all the details are provided. The project was assigned after two thirds of the semester, after students completed direct instructional lectures and laboratory exercises about the cryptographic algorithms, PKI and X.509 certificates, which are the key topics of the practical project, described in more details in the remainder of this chapter. All students have necessary skills to develop the projects they were assigned due to the programming courses they completed prior to the start of the CS course. Besides that, the project was assigned with minimum information that was necessary for solving it, but due to the lectures that are completed before the start of the project, this should not be considered as a minimal-guidance approach to learning. In this manner the authors hoped to inspire students to show their engineering abilities and their creativity. The teacher assigns a project

and guides students throughout the project by answering their questions using mail list or at the classes. There are also few problem classes devoted to the project requirements detailed explanations.

The practical project assignment was added to the CS course in the school year 2015/2016. The goal of the practical project was that students acquire deeper knowledge about the structure of the public-key infrastructure and X.509 certificates and the way of generating and using these certificates. The main reasons for choosing these topics were the importance and the widespread use of digital certificates today (e.g., nowadays more than a half of the internet traffic is being protected using HTTPS where digital certificates are a mandatory part of server identity verification and data protection scheme) and the requirement to use various security related algorithms in digital certificate implementation. The positive side of this choice is that it has low hardware and software demands (any PC with standard configuration and standard software can be used for the project). The assignment was to design and implement application with GUI written in Java programming language with following functionalities:

1. generation of a new keypair for X.509 certificate,
2. import/export of an existing keypair for X.509 certificate,
3. view of details for existing keypairs for X.509 certificates,
4. signing the X.509 certificate,
5. export the X.509 certificate.

In order to simplify the project, so that students could finish it within a reasonable time, certain restraints were defined. For item (1) only a subset of possibilities should be supported:

1. only RSA in combination with the SHA-1,
2. only version 3 of the X.509 certificate,
3. only basic constraints, issuer alternative name, and key usage for extensions.

For item (2) only PKCS#12 file format should be supported. For item (4) first a certificate signing request (CSR) should be generated in a PKCS#10 file format and then signed with a CA certificate. For item (5) export should be available in base-64 encoded file format (CER).

Students had 2 months to complete the project (time from publication of project assignment until the deadline for submission) and were paired in teams of maximum two persons. They are allowed to use any open-source cryptographic libraries. The project carries 20% of the points for the exam that cannot be won in any other way, but it is not obligatory in order to pass the exam. Students submit their finished project until the deadline. After that, project is examined by the teachers where functional testing is carried out and a number of points is assigned based on the testing results. Finally, an oral examination is organized where students are asked questions about the code they wrote and are given a chance to improve the faults that were found during functional testing. Figure 1 shows an example of a successfully finished project.

3.3 | Practical project experience

By implementing the application described in section 3.2 students were able to understand how CAs work, because with their application it is possible to simulate the entire process of obtaining a valid X.509 certificate. Students did not

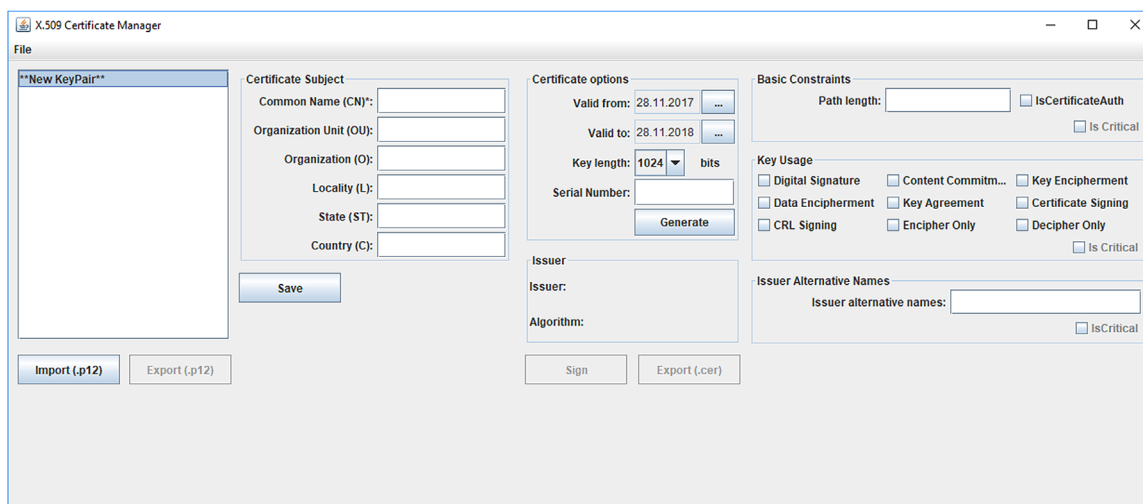


FIGURE 1 An example of a successfully finished project

TABLE 1 The number of students who took the course in each year (N1)

	2014	2015	2016
SEU	49	50	64
CEU	NA	NA	176

have any limitations in choosing the method of the project implementation, except that only open-source and commercial components for which UB-SEE has a license for were allowed. Students had to complement their knowledge by browsing online resources, which is the methodology they are going to use often in their workplaces once they finish the studies.

One of the drawbacks of assigning the project to the teams of two students was the sometimes observed practice among students to divide the work among team members. In some of the teams one member did the GUI programming and the other did the security mechanism programming. Therefore, only one student was focused on the target part of the project, while the other was working on the parts not directly related to the CS course.

As mentioned in 3.2, students were allowed to use any open-source cryptographic and other libraries. One of the main problems students had to cope with was finding the appropriate libraries and including them in their project in the right way. This was something that students did not have a chance to learn in a formal way. It appears that it was a bigger problem for the CEU group of students, since this was one of the first projects of this type that they had to work on during their studies. For the GUI most of the students used default frameworks that come with the IDE they used, while some students used advanced frameworks, and there were also some that even did not use a framework at all, but instead implemented a GUI from scratch. For the security mechanisms development students used two distinct approaches: the first was the use of Java security [12] and Java crypto [11] packages, which had to be extended because they do not give enough support for everything that is required by the project, and the second was the use of the bouncycastle open-source API [1]. The second approach was the most popular one among the students, as it provides all the necessary data

TABLE 2 The percentage of students who passed the exam during one school year (P1) and the average grade on the exams during 1 school year (G1)

	2014	2015	2016
SEU (P1/G1)	77.59/7	86.54/7.67	84.62/8
CEU (P1/G1)	NA	NA	73.99/7.23

TABLE 3 The percentage of scores above 90% of the points on the exam during 1 school year for the SEU group of students

	2011	2012	2013	2014	2015	2016
SEU	3.33	8.7	14.55	17.24	19.23	32.69

structures and cryptographic algorithm implementations, so that students can focus on key elements of the project task. The first approach is also in line with the practical project goals, but it requires a bit more programming than necessary (not all data structures are supported, so students have to implement that part as well).

4 | RESULTS

In the previous work the authors analyzed how the COALA system influenced students' grades and satisfaction [22]. In this paper the same analysis is conducted for the next 2 years, which were not included in the previous work. The analysis on the effects and students' opinion about practical project is also presented. Table 1 shows the number of students who took the course in each year (N1).

In the previous accreditation process (2013) at the CE curriculum the course was moved from the list of electives at the fourth year of studies to the list of mandatory at the third year of studies, which significantly increased the total number of students at the course in the last school year that was analyzed.

4.1 | Student grades

In [22] the exam results and many other numerical indicators were compared in order to show that the introduction of the COALA system made positive impact. Here only the most significant parameters from that analysis will be presented for the next 3 years. Table 2 shows the percentage of students who passed the exam during 1 school year (P1) and the average grade on the exams during 1 school year (G1).

If the P1 and G1 parameters are observed for the SEU group of students in the next 3 school years it can be seen that P1 never dropped under 76 percent and that G1 never dropped under 7 (out of 10). In the last school year that was analyzed the practical project was added (as explained in section 3.2). With this improvement the P1 parameter remained high (85 percent) while the G1 parameter reached its highest value 8 (out of 10). By analyzing these two parameters it can be concluded that practical experience described in section 3 had positive influence on students' results, and thus on the level of their knowledge.

TABLE 4 The results from the laboratory exercises survey conducted in 2016 separated for SEU and CEU students (29 and 80 answers, respectively)

Question	SEU	CEU
The COALA system helped me to better understand the course material.	4.48	4.40
The COALA system helped me to prepare for the exam.	4.03	3.88
The user interface of the COALA system is intuitive and user-friendly and the abstractions used to present the concepts are easy to understand.	4.17	4.35
The laboratory exercises helped me to understand the details of the algorithms.	4.5	4.28
The way of the algorithm visualization helped me to follow the details of the algorithms.	4.69	4.68
My experience of using the COALA system was excellent.	4.59	4.55

In the last school year that was analyzed another representative sample appeared: CEU group of students. Their P1 and G1 parameters are at the high level, but not as high as the SEU group. The number of students in this group is significantly higher than in the SEU group (176 as opposed to 64), so this is one factor that influenced the statistics. Another significant factor is the fact that the students from the CEU group have the course 1 year before the students from the SEU group during their studies, which appears to be relevant in this case, especially when it comes to completing the practical project (as will be explained in section 4.3). All things considered this is still a good start, but it shows that adaptation of the practical project needs to be done for this group of students.

The analysis of the percentage of students who scored above 90% of the points on the exam during 1 school year from [22] was repeated here, but with new data. It is shown for the SEU group of students in the consecutive 6 school years in Table 3.

Table 3 shows that the percentage of students who score the top grade on the exam (over 90% of the points) shows constant increase and reaches the peak in the last school year that was analyzed when the project was introduced—an increase of 69.9% over the previous year. For the CEU group of students this percentage was 21.39, which is higher than

any previous for the SEU group of students in the observed time period.

4.2 | Laboratory exercises survey

In the school year 2012/2013 we conducted an anonymous survey among the students who took the course, as explained in [22]. The same survey was conducted in the following years and the scores were mainly higher than before in absolute values, but the relative relationship among the questions remained similar.

Table 4 shows the averages for questions from the laboratory exercises survey conducted in 2016, if we use the scores 1–5 to represent the choices strongly disagree to strongly agree, separated for SEU 4th year and CEU 3rd year students. It is interesting that when their answers from the survey were compared there were only negligible differences. This result leads to a conclusion that the year of study difference did not influence the usefulness of the laboratory exercises.

4.3 | Practical project survey

As was described in section 3.2, in the school year 2015/2016 a practical project was assigned to students. In addition to the analysis based on the measurable results and the laboratory exercises survey, a separate survey was conducted among the students who completed the practical project in 2016. The survey was anonymous and 95 students (22 SEU students and 73 CEU students) took the survey. The survey consisted of the multiple choice, Likert-scale and free response questions.

The multiple choice question was used to determine the amount of time needed to complete the project. Students were offered to choose one of the following options: 1–8, 9–16, 17–24, 25–32, and 33+ all expressed in work hours. Table 5 shows the results separated for the SEU and the CEU groups of students.

Here it can be seen that the year of studies and students' experience had an impact on the apprehended difficulty of the practical project (even though the project was exactly the same for both groups). As shown in Table 5 majority of the SEU 4th year students (around 76%) completed the project in 4 work days or less. On the other hand almost 50% of the CEU 3rd year students needed at least 5 work days to complete the project. This was expected since the 4th year SEU students during their studies already had several courses with practical projects of a similar scale, while for the 3rd year CEU students this was the first of many such projects. That is why project assignment was scaled so that 3rd year CEU students could complete it in a reasonable amount of time. As a consequence the SEU students needed a little less time than it would be optimal.

TABLE 5 The time needed to complete the project

	1-8 hr	9-16 hr	17-24 hr	25-32 hr	33+ hr
SEU	0%	19%	33%	24%	24%
CEU	1%	6%	22%	23%	48%

TABLE 6 The results from the practical project survey conducted in 2016 separated for the SEU and the CEU students

Question	SEU	CEU
The project helped me to better understand the material.	3.77	3.60
The project should be obligatory to pass the exam in the future.	2.36	2.34
My experience while working on the project was excellent.	4.14	3.96

The Likert-scale questions were:

QP1. The project helped me to better understand the material.

QP2. The project should be obligatory to pass the exam in the future.

QP3. My experience while working on the project was excellent.

The scales were the same as the ones described in section 4.2 for the laboratory exercises survey. Table 6 shows the results of the Likert-scale questions calculated in the same way as was for the laboratory exercises survey in section 4.2 (maximum is 5).

By analyzing Table 6, it can be seen that the most of the students considered the practical project as useful to better understand the material. The authors believe that the scores would be even higher if the project would be relaxed from the GUI design and programming and narrowly focused on security issues. Over 50% of both groups of students agree that the project should not become mandatory to pass the exam. This is consistent with the authors' original reasoning that students should be encouraged to work on a project but not forced to do it. Finally, the analysis shows that students were generally satisfied with their experience while working on the project.

In the free response questions, students commented on pros and cons regarding the way of examination, adding laboratory exercise that would help in completing the project and in general about the project. Some of the students from the CEU group suggested that at the examination a modification should be requested from student, others suggested that the examination should be public. The SEU group of students thought that the way of examination was good and did not suggest alterations. Regarding laboratory exercise majority in the SEU group did not think it is needed, while the students from the CEU group had divided opinion on this matter. Regarding the general cons about the project both groups of students agreed that the assignment was obscure. However, this was the intention because the authors' goal was to trigger situational interest and curiosity through such knowledge

incompleteness and to make students analyze the problem, read the literature and make their own assumptions in order to create situation as close as possible to the real world problem. The CEU group of students additionally complained that the project carried too little points for the work it required. Regarding the general pros about the project both groups of students agreed that the work on practical problem helped them to understand the material and that they enjoyed the teamwork. The students from the CEU group additionally mentioned that the project helped them to improve their programming and research skills.

5 | CONCLUSION

In this paper the authors have described evolution of CS course from no practical work at all to laboratory exercises that cover cryptographic algorithms and practical project that covers real world problems.

Based on the conducted research three major conclusions can be drawn. The first is that the laboratory exercises are at the point where no significant further improvement to the learning process can be obtained from them.

The second is that the practical project shows a significant positive impact to the overall knowledge shown at the exams, but there are still places for improvement. The main problem that was observed, as was discussed in section 4, is that the GUI programming is a big overhead for students, especially the CEU group. This is a relevant issue since the idea of the project is to improve information security knowledge and not the overall programming skills. Another consequence that comes from this fact is that the students often divided their work so that one group member would do GUI programming and the other would do security programming, which led to the situation in which only one student is focused on the significant part of the project. In order to overcome these issues, a new software system that will be used as a template for the next years' practical project is already prepared. It solves the last years' problem, but with many different options included and with the GUI and the logic parts clearly separated. This will allow teachers to make many different variations of the problem and it will relieve students of handling a GUI.

Third conclusion comes from the fact that the older group which had a wider set of prior courses in different areas of computer science and especially programming (SEU) required less time to complete the project and gave slightly better overall opinion about the usefulness of the practical project. This signals that practical projects should be introduced with care to students that are in the earlier stages of their education and with less knowledge acquired in the prior education, in order to avoid the excessive working memory and cognitive load for these students, which are

created in cases of minimal guidance and incomplete long-term knowledge.

ORCID

Zarko Stanisavljevic  <http://orcid.org/0000-0003-0272-5139>

REFERENCES

1. Bouncycastle API Documentation, available at: <https://www.bouncycastle.org/docs/docs1.5on/index.html> (Accessed October 2017)
2. E.T.H. Chu and C.W. Fang, *CALEE: A computer-assisted learning system for embedded OS laboratory exercises*, *Comput. Educa.*, **84** (2015) 36–48.
3. L. Cui et al., *Wefilab: A web-based wifi laboratory platform for wireless networking education*, *IEEE Trans. Learn. Technol.*, **5** (2012), 291–303.
4. J. Djordjevic, B. Nikolic, and A. Milenkovic, *Flexible web-based educational system for teaching computer architecture and organization*, *IEEE Trans. Educ.*, **48** (2005), 264–273.
5. D. Draskovic, M. Mistic, and Z. Stanisavljevic, *Transition from traditional to LMS supported examining: A case study in computer engineering*, *Comput. Appl. Eng. Educ.*, **24** (2016), 775–786.
6. W. Du, *SEED: Hands-on lab exercises for computer security education*, *IEEE Security & Privacy*, **9** (2011), 70–73.
7. I. Glogger-Frey et al., *Inventing motivates and prepares student teachers for computer-based learning*, *J. Comput. Assist. Learn.*, **31** (2015), 546–561.
8. J. Hill et al., *Using an isolated network laboratory to teach advanced networks and security*, *ACM SIGCSE Bulletin*, **33** (2001), 36–40.
9. IEEE Computer Society and ACM, Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, Available from: <http://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf> (Accessed January 2017)
10. IEEE Computer Society and ACM, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, Available from: <http://www.acm.org/binaries/content/assets/education/se2014.pdf> (Accessed January 2017)
11. Java Crypto Documentation, available at: <https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html> (Accessed October 2017)
12. Java Security Documentation, available at: <https://docs.oracle.com/javase/7/docs/api/java/security/package-summary.html> (Accessed October 2017)
13. M. Kapur, *A further study of productive failure in mathematical problem solving: Unpacking the design components*, *Instr. Sci.*, **39** (2011), 561–579.
14. P.A. Kirschner, J. Sweller, and R.E. Clark, *Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching*, *Educ. Psychol.*, **41** (2006), 75–86.
15. M.R. Lepper, *Motivational considerations in the study of instruction*, *Cogn. Instr.*, **5** (1988), 289–309.
16. I. Marsa-Maestre et al., *Design and evaluation of a learning environment to effectively provide network security skills*, *Computers & Education*, **69** (2013), 225–236.
17. M. O'Leary, *A laboratory based capstone course in computer security for undergraduates*, *ACM SIGCSE Bull.*, **38** (2006), 2–6.
18. S. Ristov, D. Spasov, and M. Gusev, *Successful integration of practical Cisco CCNA in the Computer Networks Design course*, *IEEE Global Engineering Education Conference (EDUCON)*, 2015, pp. 694–703.
19. J.I. Rotgans and H.G. Schmidt, *Situational interest and learning: Thirst for knowledge*, *Learn. Instr.*, **32** (2014), 37–50.
20. A. Ruiz-Martinez et al., *Teaching advanced concepts in computer networks: Vnuml-um virtualization tool*, *IEEE Trans. Learn. Technol.*, **6** (2013), 85–96.
21. Z. Stanisavljevic et al., *SDLDS—System for digital logic design and simulation*, *IEEE Trans. Educ.*, **56** (2013), 235–245.
22. Z. Stanisavljevic et al., *COALA—System for visual representation of cryptography algorithms*, *IEEE Trans. Learn. Technol.*, **7** (2014), 178–190.
23. J. Tan, X. Guo, and W. Zheng, *Case-based teaching using the Laboratory Animal System for learning C/C++ programming*, *Comput. Educ.*, **77** (2014), 39–49.
24. M. Wannous and H. Nakano, *NVLab, a networking virtual web-based laboratory that implements virtualization and virtual network computing technologies*, *IEEE Trans. Learn. Technol.*, **3** (2010), 129–138.
25. L. Xu, D. Huang, and W.T. Tsai, *Cloud-based virtual laboratory for network security education*, *IEEE Trans. Educ.*, **57** (2014), 145–150.



Z. STANISAVLJEVIC received his BSc (2007), MSc (2008), and PhD (2015) degrees in Computer Engineering from University of Belgrade, School of Electrical Engineering. He is currently an assistant professor at the University of Belgrade, School of Electrical Engineering at the Department of Computer Engineering and Information Theory, teaching several courses on computer architecture and organization and computer security. His research interests include eLearning tools, computer architecture and organization, network security, cryptography, and internet programming.



P. VULETIC obtained his BSc, MSc, and PhD in Computer Systems and Network Architecture from University of Belgrade, School of Electrical Engineering. He used to work at the University of Belgrade, Computer Centre as a senior network engineer while Computer Centre was responsible for the development of Serbian Research and

Education Network (AMRES), and later as a deputy director of AMRES. He is currently assistant professor at the University of Belgrade, School of Electrical Engineering at the Department of Computer Engineering and Information Theory, teaching Advanced Computer Networks and Software Defined Networks courses. His research interests span from network security to network performance, monitoring and modern network management principles. Dr. Vuletic was involved in several EC projects in the area of research and education networking. He was a leader of research tasks exploring multi-domain control and management principles and network service performance in four Geant projects.

How to cite this article: Stanisavljevic Z, Vuletic P. Adding practical experience to computer security course. *Comput Appl Eng Educ*. 2018;26:384–392. <https://doi.org/10.1002/cae.21891>