COALA-System for Visual Representation of Cryptography Algorithms

Zarko Stanisavljevic, Jelena Stanisavljevic, Pavle Vuletic, and Zoran Jovanovic

Abstract—Educational software systems have an increasingly significant presence in engineering sciences. They aim to improve students' attitudes and knowledge acquisition typically through visual representation and simulation of complex algorithms and mechanisms or hardware systems that are often not available to the educational institutions. This paper presents a novel software system for CryptOgraphic ALgorithm visuAl representation (COALA), which was developed to support a Data Security course at the School of Electrical Engineering, University of Belgrade. The system allows users to follow the execution of several complex algorithms (DES, AES, RSA, and Diffie-Hellman) on real world examples in a step by step detailed view with the possibility of forward and backward navigation. Benefits of the COALA system for students are observed through the increase of the percentage of students who passed the exam and the average grade on the exams during one school year.

Index Terms—AES, algorithm visualization, cryptographic algorithms, data security, DES, Diffie-Hellman, RSA, security education

1 Introduction

In the modern e-world era security and reliability of e-services are of the utmost importance. Nowadays people live their entire life, both private and professional, online. This would not be possible without complete trust in e-services that are in use. Confidentiality and authentication are the two services that became indispensable in the everyday use of Internet.

Therefore many schools that teach computer engineering worldwide have supplemented their curriculums with courses in this area. A Data Security course was offered to the undergraduate final-year students at the study program in software engineering (SEU) and to the one year master degree students at the study program in computer engineering (CEM) at the School of Electrical Engineering, University of Belgrade (SEE-UB). The course covers topics from cryptographic algorithms, network security and applications and system security. Cryptographic algorithms are one of the crucial parts of the course, since the confidentiality and authentication services are achieved by using various combinations of different cryptographic algorithms. Data Security is a one-semester mandatory course, which consists of two lecture classes, two problems classes, and one laboratory class per week for the SEU group of students and an elective course, which consists of two lecture classes and two problems classes for the CEM group of students. The lecture classes cover theoretical concepts of security mechanisms, the problems classes deal with concrete security mechanisms that are based on the concepts covered in lectures, and the laboratory classes provide practical experience in dealing with complex algorithms.

In previous years we have analyzed exam results from the Data Security course and noticed that students had trouble understanding cryptographic algorithms, which resulted in lower grades on the part of the exam covering this area and consequently the final grades for the exam. The reason for this was the difficulty for students to closely follow the execution of algorithms, because they are too complex to be calculated manually on paper.

In order to help students to better understand the material many teachers in different areas use educational software systems. These systems have a significant presence in engineering sciences, where it is important for students to have practical work in addition to classes. Systems differ significantly depending on the area in which they are used. It is common in hardware related courses to use software simulators [1] in order to avoid the high price of the necessary equipment for the laboratory exercises. In software related areas that teach algorithms, software systems for visual representation of the algorithms [2] are often used for both teaching and laboratory exercises.

Therefore we have decided to introduce a software system to the course, which would help students to better understand these topics. There are many available algorithm visualization (AV) tools, which can be used in education, with various features. We have analyzed the possibility of using some of the existing tools in the Data security course. These tools are presented and compared in the next section. The analysis showed that although all of them have been used successfully elsewhere, each was developed to meet very specific requirements (e.g., specific algorithms, specific level of execution, etc.), and therefore none of them meets the Data Security course requirements. That is why the decision was made to develop a new software system.

- *Z. Stanisavljevic, P. Vuletic, and Z. Jovanovic are with the Department of Computer Engineering, School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia. E-mail: {zarko.stanisavljevic, pavle.vuletic, zoran.jovanovic}@etf.bg.ac.rs.*
- *J. Stanisavljevic is with the Asseco SEE, Bulevar Milutina Milankovica 19g, 11070 Belgrade, Serbia. E-mail: jelena.stanisavljevic@asseco-see.rs.*

Manuscript received 9 July 2013; revised 17 Mar. 2014; accepted 31 Mar. 2014. Date of publication 7 Apr. 2014; date of current version 2 July 2014.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TLT.2014.2315992

The software system is designed to support laboratory exercises, which cover complex cryptographic algorithms that caused most of the problems for students. Supported algorithms are: Data Encryption Standard (DES), Advanced Encryption Standard (AES), RSA, and Diffie-Hellman. We have developed a system for CryptOgraphic ALgorithm visuAl representation (COALA) and designed the related laboratory exercises that make use of the COALA system. The system aims to enable students to study the various complex algorithms covered by the course in a user-friendly, visual, and interactive environment. For every algorithm that is supported, the system is able to visually represent complete execution with the possibility of navigating forward and backward through the execution, obtaining intermediate results for every operation of the algorithm, and presenting details of every operation in the algorithm. The laboratory exercises intend to enable students to execute complex algorithms quickly on real examples, where manual execution on paper would be time consuming, and to help them learn details and notice important attributes of these algorithms. The COALA system and experiences from using it in the Data Security course are presented in this paper.

The paper is organized as follows. Section 2 outlines the related work and shows what motivated us to develop the COALA tool, Section 3 gives short descriptions of each algorithm and analyzes several key design issues that had to be resolved during the implementation of the COALA system, Section 4 presents functional description and explains the main usage patterns of the COALA system, Section 5 presents and discusses the impact of the COALA system introduction to the teaching process and Section 6 concludes the paper.

2 Motivation and Related Work

Algorithm visualization tools have been used in the education process for a long time and a lot of different tools have been created [3]. Some of these are described in a survey paper focusing on program visualization and algorithm animation systems [4]. In order to classify these tools based on their learning outcomes, the survey uses the finding that the interaction between students and a visualization system is more important than the visualization contents [5]. The engagement level taxonomy [6] that includes six engagement levels: no viewing, viewing, responding, changing, constructing, and presenting, which are organized in a hierarchical structure where higher level of engagement leads to higher educational benefit, was used for the classification. Based on the analysis of this classification, the survey suggests that moving from the no viewing level to the responding level can improve attitude and knowledge acquisition.

The introduction of active learning in the educational process through algorithm visualization tools is explained and design criteria are proposed in [7]. Five desirable design characteristics for interactive classroom visualizations that were considered are: high interactivity, easily understood abstraction, simple is better, relevancy, and robust user interface. The way these design features are implemented in the COALA system is given in the next section.

Hundhausen et al. in [5] explained algorithm visualization effectiveness. Four underlying theories were defined: Epistemic Fidelity, Dual-coding, Individual Differences, and Cognitive Constructivism. It was stated that the Cognitive Constructivism theory has had the most significant impact on AV effectiveness out of these four. According to this theory students will not benefit from an AV system by passively viewing the algorithm simulation, but instead they must become actively engaged with the technology in order to benefit most from it. The second conclusion that is important is the way in which the effects of an AV system can be measured. There are two important factors that impact measurement of an AV system effectiveness: the type of knowledge that is measured (conceptual knowledge and/or procedural knowledge) and the way knowledge acquisition is measured (post-test results measurement and/or pre-test to post-test improvement measurement).

It was emphasised that using active learning techniques increases students' motivation and holds their attention longer than lecturing alone [8]. Some topics covered in security courses can be very interesting to students (viruses, hackers, etc.), in contrast to the topics concepts and theory on which they are based on (e.g., algorithms) which are typically less attractive and more challenging to students. Schweitzer et al. in [8] suggested that these less attractive topics should be covered by active learning techniques in order to keep students' interest.

Although there are a lot of visualization tools that are used in the education process, there is a relatively low number of these tools that are used in security education and especially for cryptographic algorithms. We analysed the possibility to use aforementioned available tools and their suitability for the Data Security course.

A visualization tool for wireless network attacks is presented in [9]. Some of the most popular wireless network attacks (eavesdropping, evil twin, man in the middle, ARP poisoning, and ARP request replay) have been demonstrated with this tool. Demonstrations are in the form of animations and options that are available to users during an animation are: play, trace step by step, rewind to see the previous step, forward to see the next step, pause, and stop. The tool provides a quiz that users can take with questions on the animation they just finished.

In [10] a set of interactive visualization applets for teaching cryptography algorithms is presented. Algorithms covered by this tool are: Shift Cipher, Simple Substitution Cipher, Affine Cipher, Vigenere Cipher, RC4 Stream Cipher, RSA Cipher, and DES Cipher. The interface allows users to input the text to be encrypted and the key and to interactively control analysis tools such as frequency graphs, key length analysis, and diagram maps. The applet for DES Cipher shows two rounds of a Feistel system as a diagram with textual description of operations in a round, and it interactively moves bits through the diagram to illustrate data flow in the algorithm.

Grasp [11] is a visualization tool for teaching security protocols. It can be configured by a user to visualize any security protocol (e.g., Diffie-Hellman), since it provides protocol specification language that allows an arbitrary number of actors and message passing with appropriate commands needed for security protocols. Users can edit

protocol commands during the visualization allowing them to observe "what-if" scenarios and they can control the protocol execution by moving a step forward or backward, resetting or finishing the execution.

ECVisual [12] is a tool for visualization of elliptic curve based ciphers. This tool allows users to visualize elliptic curves over the real field and over a finite field of prime order, perform arithmetic operations, do encryption and decryption, and convert plaintext to a point on an elliptic curve.

In [13] a digital Lego system is designed, where atomic security units such as cryptographic operations, authentication, etc. have been represented as pieces of a puzzle and security protocols are presented as structures built from these pieces. Users can construct security protocols in two ways as a plain text input or through the interactive visualization interface. There are few preloaded protocols and few preloaded attacks available with the system.

DESVisual [14] is a visualization tool for the DES algorithm. It visually presents the execution of initial permutation and the first round of the DES algorithm, using 8 or 16 bits input and 6 or 10 bits key, as a diagram. There are two modes of operation: trace mode and guided encryption (or decryption) mode. In trace mode the tool executes all operations and users can follow a specific bit across operations by clicking on it. In guided encryption (decryption) mode the tool steps through each operation and asks users to calculate the result of a current operation.

Tools described in [9], [12], and [13] cover different topics from the ones selected to be covered by laboratory exercises in the Data Security course. DESVisual [14] and visualization applets [10] can both visually present only part of the DES algorithm execution with limited size of the input. Grasp [11] is able to visually represent Diffie-Hellman protocol only at the conceptual level. Visualization applets [10] enable execution of the RSA algorithm without showing details of the algorithm. None of the tools visualizes the AES algorithm.

Previously mentioned tools were developed to meet very specific requirements with a narrow scope of algorithms covered and parts of algorithms that are visualized. These tools only partially fulfill the need to improve the critical topics of the Data Security course (e.g., algorithms explained in Section 3 and Section 4). Therefore with the COALA tool we aimed to close this gap between the available tools and the needs of the Data Security course.

3 Key COALA Design Issues

General COALA design guidelines are based on several recommendations from the literature. COALA system is designed to be at the responding level of engagement [4], which means that students are answering questions concerning the visualization presented by the system. Previously our students were at the no viewing engagement level, since no visualization tool was used. In order to have high interactivity [7], the COALA system was designed to enable students to control the execution of algorithms forward and backward, it allows them to configure the algorithm parameters before starting an execution, and it makes it possible for them to follow the result of every operation in any time. Abstractions used to present concepts are quite intuitive and thus easy to understand. The user interface in the COALA system is designed to show only relevant information at a time (tabs are used to separate operations in algorithms) and it includes only the minimum set of necessary options. Algorithms presented in the COALA system are the ones that were found difficult for students to cope with and at the same time the ones that represent concepts on which they are based on. The COALA system was designed in a way that there is minimum room for user errors, especially for users that are familiar with algorithms that are executed. Discussion made in [8] about what topics should be covered by active learning techniques assisted us to decide to visually represent four algorithms covered in the Data Security course (DES, AES, RSA, and Diffie-Hellman) in the COALA system.

TABLE 1
Differences between the DES and the AES Algorithm

	DES	AES
Feistel structure	Yes	No
Plaintext size	64 bits	128 bits
Key size	56 bits	128/192/256 bits
Round key size	48 bits	128 bits
Number of iterations	16	10/12/14

The COALA system was implemented in the Java programming language, using the NetBeans IDE as a development environment. The Swing API was used for GUI development. New custom components were developed and used in addition to the standard components of the Swing API.

In the remainder of this section we will describe some interesting details of the implementation of the COALA system. These details are results from solving key problems that came up while designing the COALA system.

3.1 Details of DES and AES Implementation

DES and AES are both symmetric block algorithms that operate with data in several iterations before producing ciphertext. There are several differences between these algorithms, which are presented in Table 1.

Despite the differences mentioned above it was still possible to create a uniform visual representation for both algorithms, which is important for visualization effectiveness, as explained in Section 2. Algorithm execution software implementation, on the other hand, had to be separate for each of the algorithms. A short description of the algorithms, the selection of the options available to students during the algorithms' execution, algorithm execution itself, the way in which iteration is graphically presented, and the graphical representation of the binary data are explained next.

3.1.1 A Short Description of the Algorithms

The DES algorithm consists of the initial permutation, 16 identical iterations, 32-bit swap and inverse of the initial permutation. Each iteration in the DES algorithm starts by

Fig. 1. Control panel for the DES and the AES algorithm execution.

the expansion permutation which permutes and extends the right 32 bits to 48 bits in order to match the size of a round key. This expanded value is XORed with the round key for that iteration and the result is reduced to 32 bits using the substitution. After the substitution, the value is permuted once again, this time without the changes in the size, and XORed with the left 32 bits in order to make the right 32 bits for the next iteration. The left 32 bits for the next iteration are the right 32 bits from the current iteration. The iteration key creation includes an initial permutation of the original key and a left circular shift and a permutation in each iteration to create a 48-bit iteration key.

The AES algorithm consists of the initial XOR with the original key, nine identical iterations and the incomplete final iteration. Each iteration in the AES algorithm starts with the substitute bytes operation, which substitutes each byte of the state with a byte determined by a specially designed substitution matrix. The iteration continues with the shift rows operation, which performs a byte circular left shift on each row of the state matrix. The mix columns operation is next for all iterations except the last one, which does not have this operation. The mix columns operation makes that each byte in a column is calculated based on all four bytes in that column. Each iteration finishes with a simple XOR of the state matrix with the iteration key. Iteration keys are created by using the expansion function on the original key.

The detailed explanations of the algorithms can be found in [15].

3.1.2 The Options Available to Students during Execution of the Algorithms

It is necessary to make it possible for students to follow each step of the execution in order to improve algorithm understanding. The algorithms are too complex to be graphically presented as a whole. However, since both algorithms are iterative, it is possible to graphically represent an iteration of the algorithm at a time. We have decided to graphically represent an iteration of the algorithm at a time with results of each step in the iteration available in its original size. Students have the option of moving one iteration forward or backward, as shown in Fig. 1.

The tool [10], which supports the DES algorithm, graphically presents only the first two rounds of the DES algorithm as a static diagram that shows flow of bits through those rounds without the possibility to view the actual values of steps in the algorithm and without the possibility to move through iterations. In the tool described in [14], which also supports the DES algorithm, only one iteration is graphically presented in a form of a diagram with concrete values, but the size of the plaintext is limited to 8 or 16 bits, and the size of the key is 6 bits which is far from realistic values. Also there is no possibility to move through the iterations. As far as we know, there are no tools that support the AES algorithm execution found in open literature.

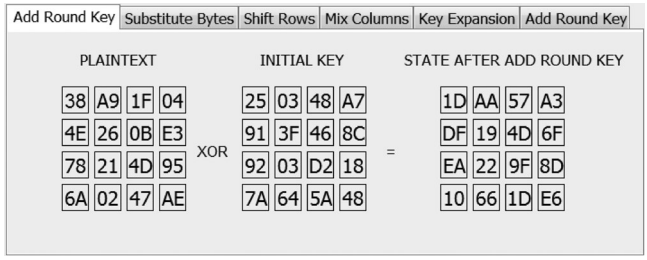


Fig. 2. Graphical representation of an iteration of the AES algorithm.

3.1.3 The Execution of Algorithms

One of the challenges was the implementation of the full DES and AES algorithm execution in a manner that allows the desired visual representation of each algorithm step. There are numerous implementations of these algorithms available online and in the literature, and all of them are capable of giving correct plaintext/ciphertext pairs. However, all these implementations are black box type solutions, which do not show intermediate steps, but only execute the whole algorithm. Therefore we designed new implementations of the DES and the AES algorithm from scratch, in a way that is compatible with the visual representation where each step of each round of the algorithm can be presented to a user.

3.1.4 Graphical Representation of an Iteration

One way to graphically represent an iteration of these algorithms is in a form of a diagram, where the results of operations would be presented as parts of a diagram and details of operations could be obtained when the button is pressed. Due to relatively large plaintext and key sizes and therefore large results of operations of iteration, this approach would produce graphical representations that are hard to follow. In [10] and [14] this issue in the DES algorithm was resolved by presenting a static diagram without values or a diagram with shortened lengths of plaintext, key and intermediate results. Since we wanted to present iteration with results of all operations with original lengths, this approach was not appropriate for COALA, especially when considering the AES algorithm, which does not follow the Feistel structure and deals with much bigger data. That is why we developed a new approach used in the COALA system. Fig. 2 shows how operations of an iteration are graphically presented using tabs.

Students are able to view the results of all operations of an iteration and the graphical representation is easy to follow.

3.1.5 Graphical Representation of the Binary Data

An important design choice was the way binary data, used in the DES and the AES algorithm, were going to be graphically represented in the COALA system. Since the operations in the AES algorithm are executed on byte level, it was decided that states and iteration keys are represented divided to bytes, as a 4×4 byte matrix, which is consistent with their original definition in the algorithm (Fig. 3a). The hexadecimal values are used for individual byte values instead of binary values, to improve clarity in visual

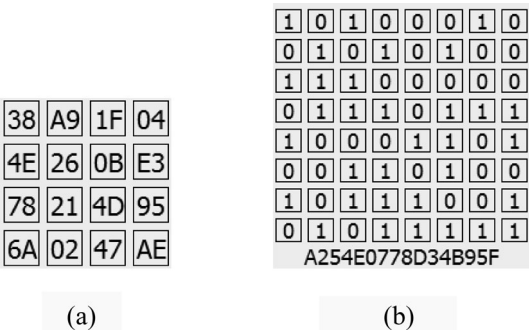


Fig. 3. Graphical representation of: (a) the AES algorithm state matrix and (b) the DES algorithm plaintext.

representation. On the other hand, because the operations that are used in the DES algorithm are executed on a bit level, data and iteration keys should be represented on the bit level, making it possible to fully represent details of operations. It was decided to represent them also as a matrix, in order to keep the same graphical abstraction as in the AES algorithm (Fig. 3b).

For both algorithms a matrix representation of binary data is needed. These matrices are of varying size and that is why a general matrix representation had to be designed. Other tools that support the DES algorithm [10] and [14] avoided this issue and decided not to show real size data in visual representation.

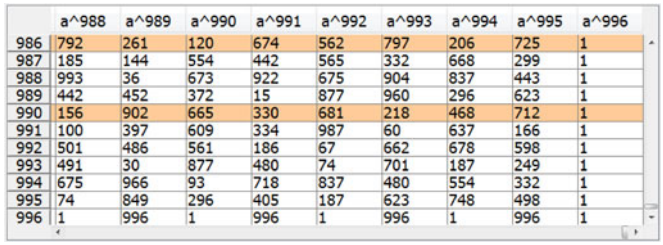
3.2 Details of Diffie-Hellman and RSA Implementation

Diffie-Hellman and RSA are both asymmetric public key cryptography algorithms. The key difference between them is that Diffie-Hellman can be used only for key exchange between two users, while RSA can be used for encryption and decryption of data. Even though these algorithms are not the same it was still possible to create a uniform visual representation for them. Unlike with the DES and the AES algorithm where working with the real world sizes of parameters was important, we believe that there would be no more benefit to students if they observe real world lengths for these two algorithms, especially since that would be extremely difficult to visualize. A short description of the algorithms, the visual representation of a primitive root for prime number calculation, and the visual representation of the algorithm for fast exponentiation are explained next.

3.2.1 A Short Description of Algorithms

The Diffie-Hellman algorithm enables two users to securely exchange a secret value. First, global public elements must be defined (prime number q and its primitive root α). Then each user selects a private value X which is less than q, and calculates a public value Y as $\alpha^X \ mod \ q$, using the previously selected private value. Public values are then exchanged by the users. At the end, each user calculates a common secret value by himself using the formula $K = Y^X \ mod \ q$, where X is his private value and Y is the public value obtained from the other user.

The RSA algorithm starts with the key generation process. At the beginning of this process two large prime numbers (p and q) are selected and multiplied to form n. Next, the Euler's totient function for n ($\Phi(n)$) is calculated. Then integer e is selected and it must fulfill the following conditions: $gcd(\Phi(n), e) = 1$ and $1 < e < \Phi(n)$. After that integer d is calculated based on a formula $d \equiv e^{-1} mod \ \Phi(n)$. Finally, the public key is created as $\{e, n\}$ and the private key is created as $\{d, n\}$. Encryption of a message M, which must be less than n, is done using public key and formula $M^e \ mod \ n$. Decryption of a message C, which must be less than n is done using private key and formula $C^d \ mod \ n$.



	a^988	a^989	a^990	a^991	a^992	a^993	a^994	a^995	a^996
986	792	261	120	674	562	797	206	725	1
987	185	144	554	442	565	332	668	299	1
988	993	36	673	922	675	904	837	443	1
989	442	452	372	15	877	960	296	623	1
990	156	902	665	330	681	218	468	712	1
991	100	397	609	334	987	60	637	166	1
992	501	486	561	186	67	662	678	598	1
993	491	30	877	480	74	701	187	249	1
994	675	966	93	718	837	480	554	332	1
995	74	849	296	405	187	623	748	498	1
996	1	996	1	996	1	996	1	996	1

Fig. 4. Determination of all possible values for primitive root for a prime number visual representation.

3.2.2 Visual Representation for a Primitive Root for a Prime Number Calculation

The tool described in [11] enables users to follow the Diffie-Hellman protocol scenarios with different actors on a conceptual level, which is appropriate to present the attacks like the man-in-the-middle. However, this kind of representation does not allow users to view the details of the algorithm execution with the specific values. That is why the Diffie-Hellman in the COALA system is designed to enable students to observe the execution of the algorithm with the values he/she assigns. Even though this algorithm is not as complex as the DES or the AES algorithm, the key strength of the algorithm is the use of very large prime numbers in the algorithm, which makes it complex for computing. It is possible to demonstrate the algorithm using simple examples, but we decided to make it possible for students to configure these values on their own and observe the algorithm with the values they specified. In that way the students are able to understand how much the calculation in the algorithm becomes more complex as the sizes of the parameters get bigger. During the execution of the algorithm visual representation was used to demonstrate details like representation of all possible values for a primitive root for a prime number (Fig. 4) and the usage of the algorithm for fast exponentiation. Fig. 4 is a table used to determine which values between 1 and 996 could be a primitive root for a prime number 997. There is one row in the table for each potential primitive root and only those who create a sequence without repetition are primitive roots for 997.

3.2.3 The Visual Representation of the Algorithm for Fast Exponentiation

The tool described in [10] has support for the RSA algorithm execution, it uses fixed length (16-bit) blocks to demonstrate the encryption and enables users to select p and q values. We have decided that there should be no limitations when it comes to the size of the data blocks for encryption and

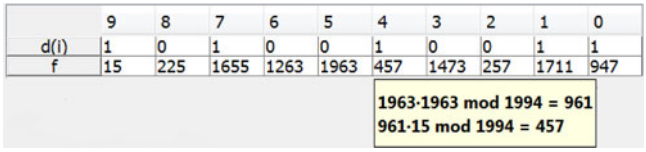


Fig. 5. Visual representation of the algorithm for fast exponentiation (calculation of 15^{659} mod 1994).

that all configurable parameters in the algorithm $(p, q, e,$ and $d)$ could be changed by the students. The fast exponentiation algorithm was used for the calculation for the RSA encryption and decryption in the same way as in the Diffie-Hellman execution (Fig. 5). By positioning the mouse pointer on any intermediate result, a student can obtain a tooltip that shows how the result was calculated (Fig. 5 shows a tooltip for column 4). In Fig. 5 *d(i)* are bits from the binary representation of the exponent (659) and f represents intermediate results for each step of the algorithm for fast exponentiation.

4 The Usage of the COALA System

This section describes the way in which the COALA system is used in the Data Security course.

4.1 Interaction

Three laboratory exercises were designed: (1) DES, (2) AES, and (3) RSA and Diffie-Hellman. Each laboratory exercise lasts for 135 minutes. Students prepare for the exercises by learning the algorithm concepts in lectures. At the laboratory exercises they are assigned with a set of questions (between 25 and 40 questions, depending on the exercise). A part of the questions requires concept knowledge and understanding of the algorithm and the other part is focused on the algorithm execution in the COALA system with specific values. Questions are devised to complete the entire encryption and decryption process for every algorithm, with the attention to every detail of the algorithm. Students use the COALA system to follow execution of an algorithm and answer these questions about the execution on paper. Fig. 6 shows some typical questions for the first laboratory exercise.

4.2 DES

The first laboratory exercise consists of three parts: a) DES encryption, b) DES encryption with 1 bit changed in the plaintext compared to a), and c) DES decryption. The first part aims to show students the encryption process with attention to every operation of the algorithm. The second part requires students to notice the avalanche effect in the DES algorithm. The third part uses ciphertext that was obtained as a result of the encryption in the first part and the same key that was used in the first part in order to show students how Feistel structure uses the same algorithm for both encryption and decryption with the inverse use of round keys.

The window showing the first iteration of the DES encryption is given in Fig. 7a. One iteration of the algorithm execution is presented at one time in the COALA system. Since the DES algorithm iteration consists of several operations, each operation has its own tab in the window. Fig. 7a

1. What is the value of the plaintext?
2. What is the value of the encryption key?
3. Draw a sketch of the DES algorithm for encryption.
4. Based on values in the simulation draw the table, which defines the PC1 permutation?
5. What is the value after the PC1 permutation?
6. What is the value after the Left Shift Key rotation? How many bits were rotated and why?
7. Based on values in the simulation draw the table, which defines the PC2 permutation?
8. What is the value after the PC2 permutation? What does this value represent?
9. Based on values in the simulation draw the table, which defines the initial permutation.
10. What is the value after the initial permutation?

Fig. 6. Typical questions at the first laboratory exercise.

presents the PC1 permutation. The initial key and the result of the PC1 permutation are shown and for each bit of the result the position in the initial key can be viewed. Fig. 7b shows the substitution for the first iteration of the DES algorithm for encryption. The result of the Xor with the round key is shown divided in the eight groups of six bits and the result of the substitution is presented as the eight groups of four bits. Between these two values there are eight buttons representing eight S-box operations and by pressing any of them a student can obtain details about substitution for the pressed S-box (Fig. 7c).

4.3 AES

The second laboratory exercise consists of three parts, similar to the first one: a) AES encryption, b) AES encryption with 1 bit changed in the plaintext compared to a), and c) AES decryption. The first two parts have the same goal as the first two parts in the first laboratory exercise, only this time with the AES algorithm instead of the DES. The third part uses ciphertext that was obtained as a result of the encryption in the first part and the same key that was used in the first part in order to show students how non-Feistel structure uses different algorithms for encryption and decryption. Students should notice how decryption has lower performance than encryption in the AES algorithm.

Fig. 8 represents the first iteration of the AES encryption. Execution of the algorithm is organized in the way that one iteration is presented at one time in the COALA system. Tabs in the window are used to show each operation of the current iteration. Fig. 8a shows the substitute bytes operation of the first iteration of the AES algorithm for encryption. The input state is shown on the left, and the resulting state after this operation is shown on the right. In the middle there is a matrix of buttons 4×4, in which each button can be pressed to show the S-box and how each byte of the input state is substituted in this operation. Fig. 8b gives a portion of the extracted key that is needed for the first iteration of the AES algorithm for encryption. The initial key is shown, and so is the key for the first iteration.

The COALA system provides additional help while executing an algorithm. In each operation parts that are connected are highlighted with the same colour when a mouse

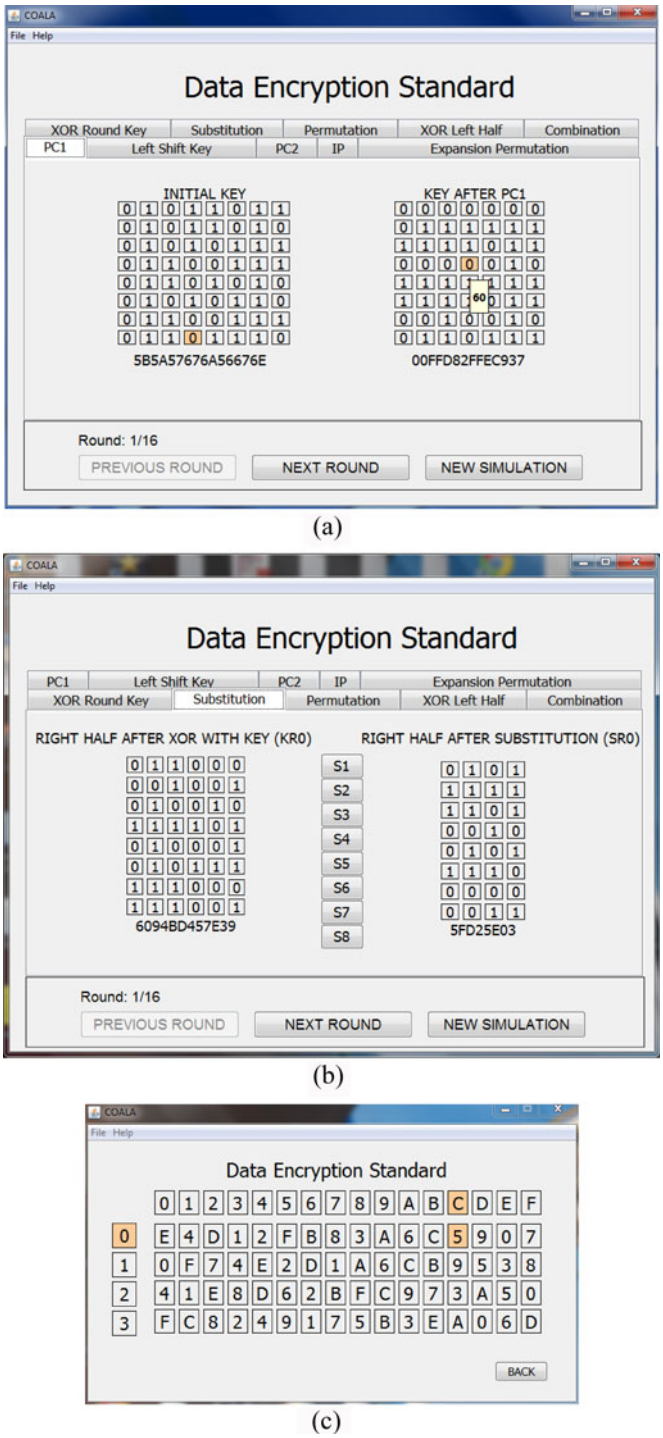


Fig. 7. DES first iteration: (a) PC1 permutation, (b) substitution, (c) S-box example view.

crosses some of these parts. Binary representation of Xor operation can be viewed as a tool tip where it is possible.

Fig. 9a presents an add round key operation at the beginning of the first iteration of the AES algorithm for encryption, where both a tooltip and colour are used as an aid. The initial state and the initial key are used in the Xor function and the result of this operation is shown. Fig. 9b depicts the mix columns operation of the first iteration of the AES algorithm for encryption where colour is used as an aid. Here the multiplication of a state and the

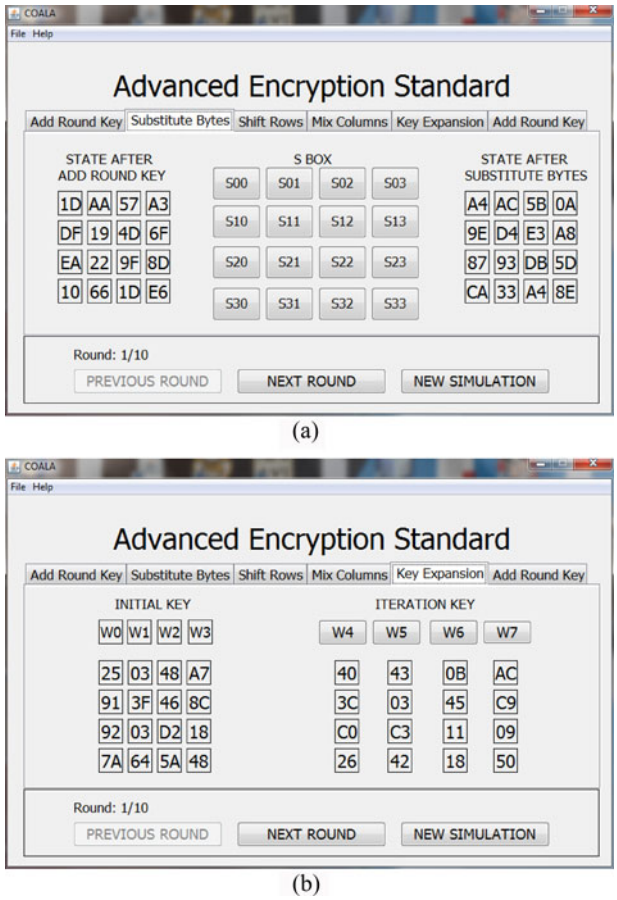


Fig. 8. AES first iteration execution: (a) substitute bytes, (b) part of the expanded key.

multiplication constant matrix is shown, along with the resulting state after the operation.

The COALA system enables users to view details of some complex operations. Fig. 10a shows how the first byte of the state is obtained in the mix columns operation. This window is shown when a user clicks on the first button of the resulting state in the mix columns operation (Fig. 9b).

Fig. 10b shows how the fourth word (W4) of the portion of the expanded key is obtained. This window is shown when a button W4 is pressed in the key expansion operation (Fig. 8b).

4.4 RSA and Diffie-Hellman

The third laboratory exercise consists of two parts: RSA and Diffie-Hellman. The first part requires students to generate a key pair for the RSA algorithm and to use it for encryption and decryption. The fast exponentiation algorithm is demonstrated while executing encryption/decryption. The second part starts by showing students how global public elements (q and α) are chosen and continues by creating a secret key for two users based on the selected global public elements values.

The key generation process for the RSA algorithm (Fig. 11a) starts with a student choosing two private prime numbers p and q from two dropdown lists ($p \neq q$). Then the public n is calculated by multiplying p and q. Also $\Phi(n)$ is calculated as $(p-1)(q-1)$. Next, a student chooses a public

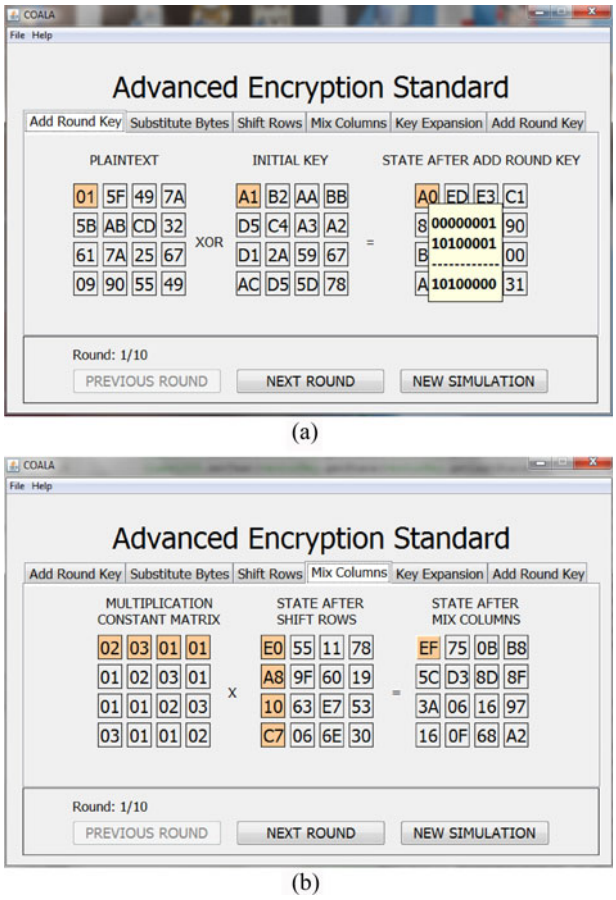


Fig. 9. AES additional visual aid in following the execution: (a) add round key, (b) mix columns.

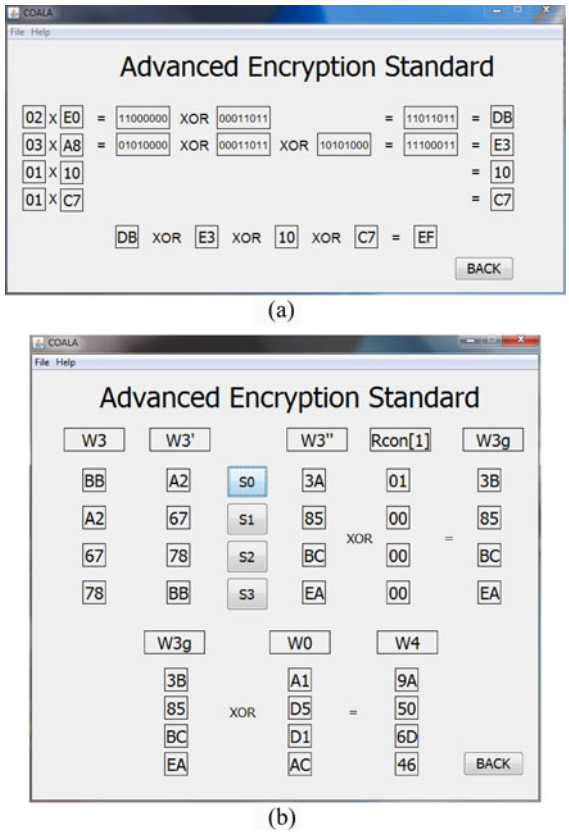


Fig. 10. AES: (a) the mix columns operation first byte generation view, (b) W4 in the first round key generation.

e from a dropdown list. The offered values for e fulfil required conditions ($gcd(\Phi(n), e) = 1$ and $1 < e < \Phi(n)$). Finally, the value for the private d is selected by a student from the dropdown list, which is populated based on the selected value for e, so that it fulfils the required condition ($d \equiv e^{-1} mod\, \Phi(n)$). At the end, the public key (e, n) and the private key (d, n) are formed and the algorithm is ready to be used for encryption and decryption.

The usage of the RSA algorithm for the encryption/decryption (Fig. 11b) starts with a student entering a plaintext value for which the ciphertext is calculated. The algorithm used for the calculation is the fast exponentiation algorithm. The table has as many columns as is the length of the binary representation of the exponent. The way that the value is calculated is shown as a tooltip when mouse is positioned on a column (Fig. 5). As a demonstration for each ciphertext value the calculation of the plaintext value is also represented.

The global public elements selection process for the Diffie-Hellman algorithm (Fig. 12a) starts with a student choosing a prime number q from a dropdown list. After that, the table which makes it possible to calculate primitive roots for q is shown. In each row of the table sequence of different values is highlighted. Only rows that are completely highlighted represent primitive roots for q and these values are offered in a dropdown list for α. Then a student chooses a value for α which completes the global public elements selection process.

The usage of the algorithm for a user A (Fig. 12b) for the Diffie-Hellman algorithm starts with a student choosing a private number X_A from a dropdown list, which offers only values that fulfil the required condition $(1 \leq X_A \leq q - 1)$. After that, the public value Y_A is calculated based on the selected value of X_A. At the end, the secret key for a user A (K_A) is calculated based on the private value X_A and the public value Y_A. User B (Fig. 12c) similarly calculates secret key (K_B) and the secret key value calculated for a user A (K_A) and the secret key value calculated for a user B (K_B) are the same.

5 Results

The COALA system was introduced to the SEU group of students in the school year 2012, and it has been used in the last two school years. Our analysis of the COALA system results consists of three parts. In the first part of the analysis we measure the effect of the COALA system introduction to the course in the school year 2012. In the second part of the analysis we compare the results of the first and the second year of the COALA system usage in the course. In the third part of the analysis we present a survey which shows the assessment of the students' satisfaction with the COALA system and how helpful it was.

5.1 Effects of the COALA System Introduction

At the SEE-UB for most of the courses, including the Data Security course, students are graded independently

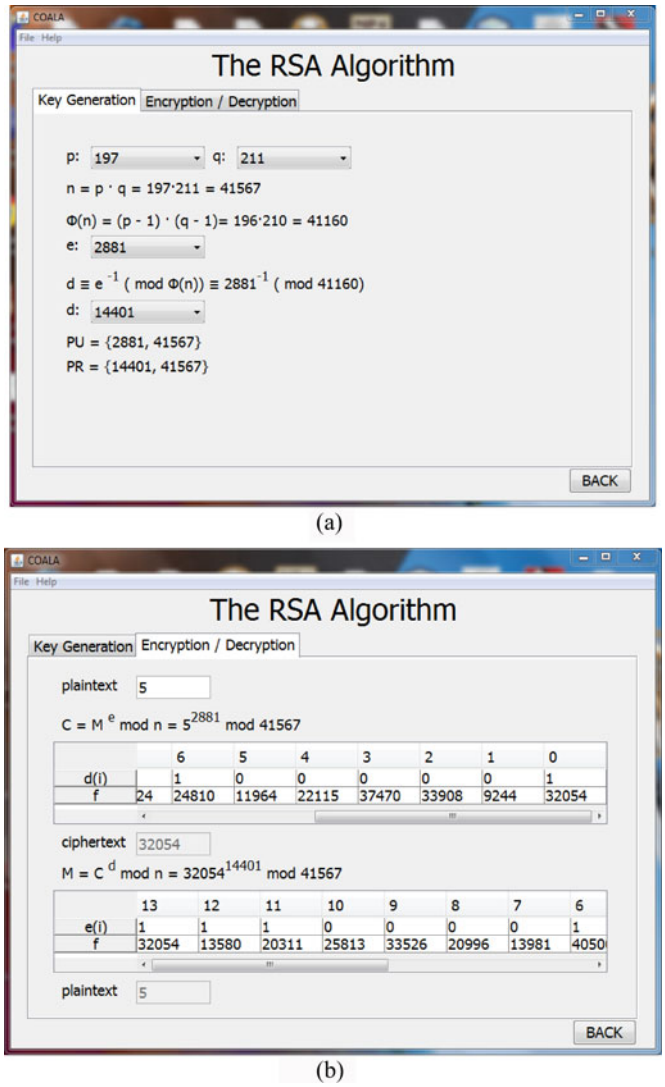


Fig. 11. RSA: (a) key generation and (b) execution.

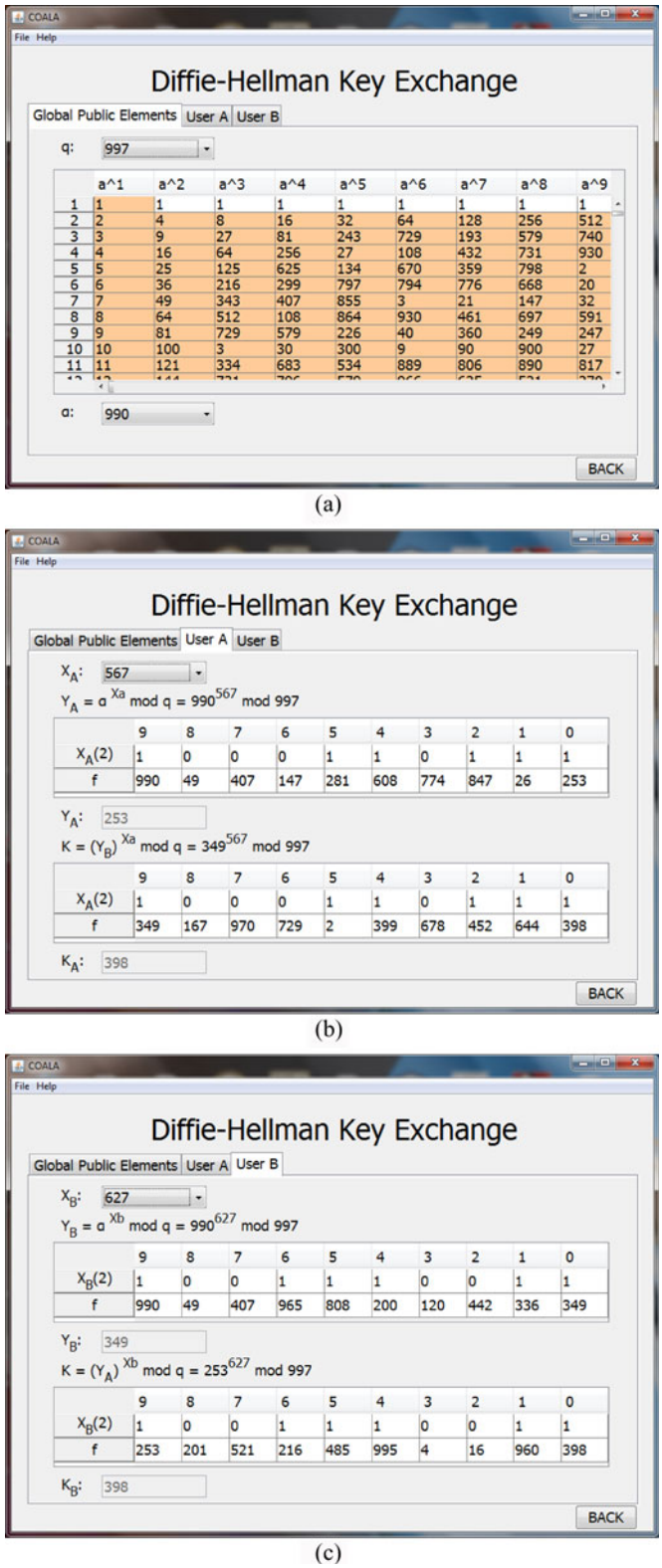


Fig. 12. Diffie-Hellman: (a) global public elements selection, (b) user A secret key calculation, and (c) user B secret key calculation.

according to the percentage of the course material they learned. Grades are formed in the following manner: less than 51 percent grade 5 (student failed the exam), between 51 and 60 percent grade 6, between 61 and 70 percent grade 7, between 71 and 80 percent grade 8, between 81 and 90 percent grade 9, and between 91 and 100 percent grade 10.

In the Data Security course, part of the points that form the final grade is earned through the tests that students take with the laboratory exercises using the COALA system. This motivates students to continuously work during the semester and helps them to prepare for the exam. Since the laboratory exercises impact the final grade, students are motivated to attend them and therefore they are familiarized with the algorithms' execution.

The fact that there are two study groups (SEU and CEM) attending the Data Security course allowed us to measure the effectiveness of the COALA system and compare it to the case when this tool is not used. The students from the CEM group are one year older than the students from the SEU group, but both groups are taught the same material since they are at the same level of prior knowledge from this area (no prior knowledge). The COALA system was introduced to the course for the SEU group of students in the school year 2012 (45 students in this group) and it was not introduced to the CEM group of students (28 students in this group). In the second year in which the COALA system was used (2013) 57

TABLE 2
The Number of Students Who Took the Course in Each Year (N1) and the Number of Students Who Took the Exam at the First Examination Period in a School Year (N2)

	2008.	2009.	2010.	2011.	2012.	2013.
SEU	23 (23)	26 (20)	38 (32)	28 (13)	45 (24)	57 (38)
CEM	-	29 (17)	20 (13)	28 (14)	28 (9)	13 (7)

TABLE 3
The Percentage of Students Who Passed the Exam during One School Year (P1)

	2008.	2009.	2010.	2011.	2012.	2013.
SEU	91.7	85.18	78.72	62.5	76.09	81.82
CEM	-	88.9	76.92	77.42	63.3	69.23

TABLE 4
The Average Grade on the Exams during One School Year (G1)

	2008.	2009.	2010.	2011.	2012.	2013.
SEU	8.58	7.67	7.23	6.59	7.24	7.27
CEM	-	8.67	7.54	7.16	7	7.31

students from the SEU group took the course and 13 students from the CEM group took the course. Since the number of students in the CEM group in 2013. was significantly smaller, these results should be taken with a grain of salt. That is why we focused on the SEU group of students and comparison with the previous year. Based on the discussion on the way in which the effects of an AV system can be measured [5], we decided to measure both procedural and conceptual knowledge in order to assess the effectiveness of the COALA system. The procedural knowledge is tested with the laboratory exercises, while the conceptual knowledge is tested in the exam. The post-test methodology is used for knowledge acquisition measurement.

Table 2 shows how many students there were in each study group in the past five years (N1) and the second value is the number of students who took the exam at the first examination period in a school year (N2).

Several numerical indicators were chosen in order to measure the effect of the COALA system. These indicators are: the percentage of the students who passed the exam during one school year (P1), the average grade on the exams during one school year (G1), the percentage of the students who passed the exam at the first examination period in a school year (P2), and the average grade on the exam in the first examination period in a school year (G2). The first two parameters show students' success in learning the course material during an entire school year. The last two parameters show students' success in the first examination period in a school year.

The P1 indicator (Table 3) shows a constant drop for both groups of students until the year in which the COALA system was introduced. Then for the SEU group of students

TABLE 5
The Percentage of the Students Who Passed the Exam at the First Examination Period in a School Year (P2)

	2008.	2009.	2010.	2011.	2012.	2013.
SEU	91.3	100	81.25	100	91.7	86.84
CEM	-	94.12	69.23	85.71	66.7	85.71

TABLE 6
The Average Grade on the Exam in the First Examination Period in a School Year (G2)

	2008.	2009.	2010.	2011.	2012.	2013.
SEU	8.57	7.95	7.31	7.85	7.75	7.63
CEM	-	9.06	7.31	7.57	7.78	7.86

this indicator shows an increase, but for the CEM group of students the drop continues. The P1 indicator for the SEU group of students continues to rise in the school year 2013.

The G1 indicator (Table 4) shows a constant drop for both groups of students until the year in which the COALA system was introduced. Then for the SEU group of students this indicator starts to recover, but for the CEM group of students the drop continues. The G1 indicator for the SEU group of students shows a negligible increase in the school year 2013.

The P2 indicator (Table 5) for the SEU group of students shows a constant high value, over 80 percent, which continues after the introduction of the COALA system (2012. 91.7 out of 100; 2013. 86.84 out of 100). The percentage of the students who passed the exam at the first examination period in a school year for the CEM group of students shows an erratic trend, which also continues in the year when the COALA system was introduced.

The G2 indicator (Table 6) shows a constant high value for both groups of students at all the time.

The analysis shows that for the SEU group of students who had the opportunity to use the COALA system the first two indicators have shown growth, and for the CEM group of students who did not use the COALA system these indicators show another drop. Determination of the cause for the previous downward trend would require a separate analysis of many different factors and it is out of the scope of this paper. It is important to emphasize that during the observed period there were no other changes in the course, except the introduction of the COALA system for the SEU group of students (same instructors, same material, and same examination criteria every year). This result is better than expected, because reluctance from students to use the system was expected, since the new system demanded an extra effort on their behalf. We believe that this result is going to be even better in the following years.

The percentage of the students who passed the exam at the first examination period in a school year shows that students who took the exam in the first examination period had less problems than the other students in completing the exam successfully and with a high grade. Second, the analysis shows no significant impact to the average grade

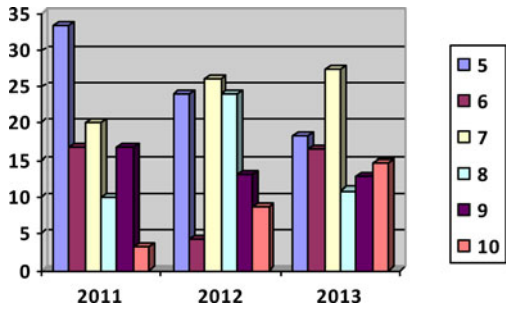


Fig. 13. The distribution of the students' grades on the exam.

in the first examination period for both SEU and CEM groups of students.

Students who take the exam in the first examination period are those who attend classes regularly during the semester and typically don't struggle to finish their responsibilities in time and pass their exams with high grades. This part of the analysis shows that these students had benefit from the COALA system (SEU group). However, the usage of the COALA system did not affect their anyway above average success on the exam significantly, just as the lack of the COALA system (CEM group) did not affect the success on the exam. Analysis results show that the introduction of the COALA system brought the most benefit to those students who postponed the moment when they took the exam to the later examination periods in order to have more time to study. These are typically students who do not attend all of their classes and often skip some of their responsibilities that are not mandatory. These students tend to learn the material on their own. Before the introduction of the COALA system they had trouble understanding cryptographic algorithms, which is one of the key topics of the Data Security course, and this caused them to have lower grades or even to fail on the exam. We can see that the introduction of the COALA system helped this group of students to understand these complex topics and to be more successful on the exam.

5.2 Comparison of the First and the Second Year of Usage

The P1 indicator has increased compared to the year 2012 and it reached the highest value in the last four years (Table 3). The G1 also reached the highest value in the last four years, but the increase compared to the previous year was not significant (Table 4). Therefore we investigated the influence of the system on the students' success on the exam in more detail.

The distribution of the students' grades on the exam for an entire school year for the previous three years is given in Fig. 13.

TABLE 7
The Percentage of Scores Above 90 Percent of the Points on the Exam during One School Year

	2008.	2009.	2010.	2011.	2012.	2013.
SEU	42	23.33	8.7	3.33	8.7	14.55

TABLE 8
The Average Percentage of the Points Earned on the Laboratory Exercises (L1) and the Percentage of Students Who Scored over 90 percent of the Points on the Laboratory Exercises (S1) during a School Year

	2012.	2013.
L1	62.2	75
S1	24.57	53.63

The percentage of students who scored above 90 percent of the points on the exam during one school year is given in Table 7 for the previous six years.

Fig. 13 shows that the percentage of students who fail the exam (grade 5) is constantly dropping since the introduction of the COALA system. The percentage of students that are graded with the best scores on the exam (grades 9 and 10) is increasing since the introduction of the COALA system. Table 7 shows that the percentage of students who score the top grade on the exam (over 90 percent of the points) showed a significant drop from year to year until the year 2012 when the COALA system was introduced, when it increased compared to the previous year and in the year 2013 the increase continues.

Although in the first part of our analysis we concluded that the system brought the most benefit to the students who previously had trouble passing the exam, this conclusion can be extended with the observation that the introduction of the COALA system also brought benefit to the best students to improve their grades.

Finally, in the second year that the COALA system was used, we were able to compare the results at the laboratory exercises in two consecutive school years. Table 8 shows the average percentage of the points earned on the laboratory exercises in a school year (L1) and the percentage of students who scored over 90 percent of the points on the laboratory exercises in a school year (S1).

We can see that students had better scores at the laboratory exercises in the second year of the COALA system usage. We attribute lower scores in the first year to the problems that usually appear in the transitioning phase of the use of a new system and a necessary habituation period for the teachers.

1. The COALA system helped me to better understand the course material.
2. The COALA system helped me to prepare for the exam.
3. 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. The laboratory exercises helped me to understand the details of the algorithms.
5. The way of the algorithm visualization helped me to follow the details of the algorithms.
6. My experience of using the COALA system was excellent.

Fig. 14. The Likert-scale questions used in a survey.

TABLE 9
The Results from the Survey Conducted in 2013

	SD	D	NAND	A	SA
1	1	8	12	19	13
2	5	6	16	16	10
3	0	7	11	14	21
4	2	8	12	12	19
5	2	4	10	12	25
6	1	6	13	21	12

5.3 Survey

In addition to our analysis based on the measurable results, we conducted a survey among the students who took the course in 2013. The survey was anonymous and 54 students took the survey. The survey consisted of the Likert-scale and free response questions. The Likert-scale questions were as shown in Fig 14.

Table 9 shows the results of the Likert-scale questions (the scales are strongly disagree (SD), disagree (D), neither agree nor disagree (NAND), agree (A), strongly agree (SD).

If we use the scores 1 to 5 to represent the choices strongly disagree to strongly agree, the averages for our questions are 3.66, 3.38, 3.92, 3.72, 4.02, 3.70, respectively. We can see that students gave very high scores for questions 3 and 5, which means that they were satisfied how the COALA system was designed. The question 2 had the lowest grade and we believe that the reason for this was that the algorithms covered by the laboratory exercises are only a part of the exam and thus it cannot help students to prepare for the entire exam. Scores for questions 1 and 4 indicate that the COALA system did help students to better understand the course material and especially the details of the algorithms covered by it. The sixth question shows us general evaluation of the COALA system by the students who used it and this score shows that most of the students were satisfied.

In the free response questions, students commented on problems and possible improvements. Some students complained that the font in the COALA system is too small. A couple of students suggested that more explanation about the algorithms in the COALA system would help them to better understand the algorithms. Many students complained that some of the questions on the laboratory exercises were exhausting because too much text had to be written down on the paper. These students suggested that the number of that type of question should be lower.

6 Conclusion

In this paper we have described the novel COALA system for visual representation of the cryptographic algorithms, and experiences from using it at the Data Security course taught at the SEE-UB. The software system is designed to support the laboratory exercises which cover complex cryptography algorithms like: DES, AES, RSA and Diffie-Hellman that are taught in the course. To the best of our knowledge this is the first implementation of the AES algorithm in a learning supporting tool. Also, unlike other existing tools which describe other mentioned algorithms, COALA has the possibility to walk students through the whole algorithm execution step-by-step using real world examples. The functional description of the system and several key design and implementation details are presented in the paper.

The main goal of the introduction of the COALA system in the Data Security course was to help students to better understand the algorithms taught in the course and to help them prepare for the exam. Numerical indicators show that the percentage of the students who passed the exam and the average grade on the exams during one school year increased for the students who used the COALA system. Results of measurements showed that the introduction of the COALA system brought benefit to all students, especially to those students who previously could not pass the exam in the first examination period of a school year and to some of the best students to improve their grades.

References

[1] Z. Stanisavljevic, V. Pavlovic, B. Nikolic, and J. Djordjevic, "SDLDS—System for digital logic design and simulation," *IEEE Trans. Educ.*, vol. 56, no. 2, pp. 235–245, May 2013.

[2] G. Rößling, M. Schüer, and B. Freisleben, "The ANIMAL algorithm animation tool," *ACM SIGCSE Bull.*, vol. 32, no. 3, pp. 37–40, Jul. 2000.

[3] C. A. Shaffer, M. L. Cooper, A. J. D. Alon, M. Akbar, M. Stewart, S. Ponce, and S. H. Edwards, "Algorithm visualization: The state of the field," *ACM Trans. Comput. Educ.*, vol. 10, no. 3, article 9, pp. 1–22, Aug. 2010.

[4] J. Urquiza-Fuentes and J. Á. Velázquez-Iturbide, "A survey of successful evaluations of program visualization and algorithm animation systems," *ACM Trans. Comput. Educ.*, vol. 9, no. 2, article 9, pp. 1–24, Jun. 2009.

[5] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, "A meta-study of algorithm visualization effectiveness," *J. Vis. Languages Comput.*, vol. 13, no. 3, pp. 259–290, Jun. 2002.

[6] T. Naps, G. Roessling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. Velazquez-Iturbide, "Exploring the role of visualization and engagement in computer science education," *SIGCSE Bull.*, vol. 35, no. 2, pp. 131–152, 2003.

[7] D. Schweitzer and W. Brown, "Interactive visualization for the active learning classroom," *ACM SIGCSE Bull.*, vol. 39, no. 1, pp. 208–212, Mar. 2007.

[8] D. Schweitzer, D. Gibson, and M. Collins, "Active learning in the security classroom," in *Proc. IEEE 42nd Hawaii Int. Conf. Syst. Sci.*, Jan. 2009, pp. 1–8.

[9] X. Yuan, R. Archer, J. Xu, and H. Yu, "A visualization tool for wireless network attacks," *J. Educ., Inf. Cybern.*, vol. 1, no. 3, pp. 53–58, 2009.

[10] D. Schweitzer and L. Baird, "The design and use of interactive visualization applets for teaching ciphers," in *Proc. IEEE Inf. Assurance Workshop*, Jun. 2006, pp. 69–75.

[11] D. Schweitzer, L. Baird, M. Collins, W. Brown, and M. Sherman, "GRASP: A visualization tool for teaching security protocols," in *Proc. 10th Colloquium Inf. Syst. Security Educ.*, Jun. 2006, pp. 75–81.

[12] J. Tao, J. Ma, M. Keranen, J. Mayo, and C. K. Shene, "ECvisual: A visualization tool for elliptic curve based ciphers," in *Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ.*, Feb. 2012, pp. 571–576.

[13] W. Wang, A. Lu, L. Yu, and Z. Li, "A digital lego set and exercises for teaching security protocols," in *Proc. Colloquium Inf. Syst. Security Educ.*, 2008, pp. 26–33.

[14] J. Tao, J. Ma, J. Mayo, C. K. Shene, and M. Keranen, "DESvisual: A visualization tool for the DES cipher," *J. Comput. Sci. Colleges*, vol. 27, no. 1, pp. 81–89, 2011.

[15] W. Stallings, *Cryptography and Network Security*. 4th ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2005.

Zarko Stanisavljevic received the BSc and MSc degrees from the School of Electrical Engineering, University of Belgrade, Serbia, and is currently working toward the PhD degree at the School of Electrical Engineering, University of Belgrade, Serbia. He is currently a teaching assistant in the Department of Computer Engineering and Information Theory, School of Electrical Engineering, University of Belgrade. His research interests include eLearning tools, computer architecture and organization, network security, cryptography, and Internet programming.

Jelena Stanisavljevic received the BSc and MSc degrees from the School of Electrical Engineering, University of Belgrade, Serbia. She is currently a senior software developer at the Asseco SEE, Serbia. Her research interests include programming, software system analysis, and data processing.

Pavle Vuletic received the BSc, MSc, and PhD degrees in computer systems and network architecture from the University of Belgrade, School of Electrical Engineering. He was at the Computer Centre, University of Belgrade, as a senior network engineer and a head of the networking team. While with the Computer Centre, he was responsible for the development of the Serbian Research and Education Network (AMRES). He is currently an assistant professor in the Department of Computer Engineering and Information Theory, School of Electrical Engineering, University of Belgrade, teaching Advanced Computer Networks course, and a deputy director of AMRES, Serbian National Research and Education Network. His research interests span from network security to traffic characterization, network performance, monitoring and modern network management principles. He was involved in several EC projects in the area of research and education networking. From 2009 till 2013, he was a leader of a research task exploring multidomain control and management principles in GN3 project. Currently, he leads the extension of the same task in the GN3+ project.

Zoran Jovanovic graduated in electrical engineering, with the MSc degree in digital telecommunications and the PhD degree in computer engineering from the University of Belgrade in 1988. After 11 years with the Vinca Institute in Belgrade and as a visiting scholar at the Computer Science Department at UCLA in California, he became a professor in the School of Electrical Engineering, University of Belgrade. Since 2001, he has been the director of the National Research and Education Network of Serbia. His research interests include parallel and distributed computing and networking.