Adamovic, S., Sarac, M., Veinovic, M., Milosavljevic, M., & Jevremovic, A. (2014). An Interactive and Collaborative Approach to Teaching Cryptology. *Educational Technology & Society*, *17* (1), 197–205.

An Interactive and Collaborative Approach to Teaching Cryptology

Sasa Adamovic*, Marko Sarac, Mladen Veinovic, Milan Milosavljevic and Aleksandar Jevremovic

Singidunum University, Department of Informatics and Computing, Serbia // sadamovic@singidunum.ac.rs // msarac@singidunum.ac.rs // mveinovic@singidunum.ac.rs // mmilosavljevic@singidunum.ac.rs // ajevremovic@singidunum.ac.rs

*Corresponding author

(Submitted October 4, 2012; Revised March 7, 2013; Accepted May 10, 2013)

ABSTRACT

This work proposes and describes the implementation of a novel module for the standard Cryptool educational software that provides the ability to communicate over the local computer network and the Internet. The development environment consists of the C# programming language and the open interface for Cryptool modules. The solution we propose facilitates interactive and collaborative work of students on solving cryptology problems and enables a more even learning pace across the entire group. We present and discuss practical results of our approach tested in the classroom setting during 2010 and 2011. In addition to better final grades, we have observed an increase in student interest for this area manifested in better class and lab attendance, as well as more active and creative participation. We describe two lab exercises based on the proposed solution. We evaluate the impact of our solution by means of a statistical analysis.

Keywords

Cryptology, Education, Interactive, Collaboration

Introduction

Cryptology represents an increasingly important discipline of our time. Development of modern information technologies and accompanying educational materials has led to migration of various resources to the Internet and its services. One of the most important questions that presents itself is how to protect those resources from malicious actions; in other words, how to educate security professionals capable of taking security risks and challenges?

The main problem with learning cryptology is its complexity and its foundation on complex mathematical principles and formulae. The vast majority of security solutions today does not require detailed understanding of the said mathematical apparatus, but rather a basic grasp of it, and the ability to understand its role in practical applications. The shifting of the focus from the domain of mathematical laws inherent to security solutions to the domain of practical IT applications has a significant impact on the profile of a student taking a course in this discipline. For this very reason, it becomes necessary to support the changing paradigm with appropriate approaches to learning, accompanying materials and, generally speaking, learning environment.

Our experiences with using the traditional approach to teaching cryptology have lead us to believe that a new, more interactive and collaborative approach to gaining knowledge in this area is needed. Instead of using the bottom-up approach – building from the necessary mathematical foundations towards their cryptographic application, we opted for a top-down approach – representing cryptological principles in the context of their practical application with the aim of provoking the interest for mastering the underpinning mathematical principles. In order to apply the said approach, we set out to develop educational software solution that offers the appropriate level of abstraction for solving the problems we dealt with.

This paper describes a model for interactive learning of cryptography, used in the Cryptology course at the Singidunum University during the 2010/2011 academic year. Model is based on the use of the Cryptool package with subsequently built and added network functions (details are discussed later in paper) that facilitate cooperation between students in learning more complex cryptographic functions. Paper gives examples of two exercises utilizing the model described: Caesar code and *Man-in-the-middle* attack. In addition, we present the evaluation of the proposed model from the standpoint of class attendance and final student grades with the discussion of improvement in comparison to the previous year.

Related work

In this section, we discuss other existing models for interactive and collaborative approaches to learning cryptography.

Rachid, Kevin and Georgios (2008) were among the first to explore the subject. In their work, they consider the fact that algorithmic animation has been the focus of intense research in many disciplines and its impact on the educational process has been marked by increasing learner autonomy. Research in this field is driven by the belief that algorithm animation can be a more effective means of instruction than manual or verbal modes of delivery. Encryption algorithms in particular, offer an interesting domain for the application of animation principles in learner/content interaction. The main challenge, however, has been how to design effective animations with a pedagogical value. This paper is concerned with the presentation of an animation of the DES algorithm that exhibits many of the features of a useful instructional material. Its pedagogical value is expressed in terms of intrinsic qualities and, in particular, the degree of interactivity and the granularity of abstraction.

Matthaus, Arno & Torben (2010) describe the development of a set of tools which allows for running large cryptanalytic jobs on a peer-to-peer (P2P) system. While P2P systems are known to scale well (BitTorrent, Skype), they are much harder to deal with than server-based systems, based on grids or cloud computing offerings. The reason to build on top of P2P, nevertheless, is to circumvent the inherent cost of any server-based solution. In this paper they show that P2P systems – while fulfilling our requirements – pose new challenges which do not exist in server-based solutions in this form. They iterate over different algorithm designs and discuss the implications of executing algorithms of these classes on a P2P system. Based on their analysis they discuss two cryptanalytic algorithms and their suitability for P2P-based computation. Additionally, they present a new fully decentralized approach for distributing the discussed algorithms in a P2P system. Their approach is specifically tailored towards scalability and different failure classes caused by malicious or unreliable peers.

Jingtao, Yiming and Lei (2009) share their experiences on the practice of interactive teaching in an information security course. The three major methods, seminar-style teaching in classroom, topic presentations and discussions, and course projects for promoting hands-on learning are described. The positive results in terms of successful learning have been witnessed on the course evaluation and the feedback from the students.

Feng, Cheng, MengXiao and YiRan (2009) consider the fact that combining the content, features and the development trend of cryptography, as well as practical experience in teaching and research, produces a teaching mode of cryptography courses based on "theory–algorithm–practice–application ". According to authors, “...years of teaching and research show that the model can get better teaching results, and help train students to solve practical problems encountered in the engineering practice by using cryptography.”

In their work, Xiulli and Hongyao (2009) assume that cryptography course is designed for undergraduate students interested in this area. In order to overcome student's fear of difficulties and arouse enthusiasm in learning, authors resort to telling informative cryptography narratives to students. In addition to ensuring simplicity and ease of understanding, interactive education software was used to effectively demonstrate cryptography algorithms during the class. These made up for student lack of mathematical knowledge and application of cryptography. They encourage students to explore boldly and discuss actively on a particular subject. Through using these flexible and diverse teaching approaches, cryptography algorithms become easy to understand for the students with less mathematical background. As authors report: “undergraduate students find lots of interest in the course and are amused by the active content of the textbook.” This further motivates them to use theory and technologies of cryptography in future research projects.

Proposed solution

The first step towards improving cryptography teaching is to raise the level of interaction between students and the course material. The traditional approach to this problem consists of implementing cryptographic functions by means of classical programming languages such as Java, C or C++. The main downside to this approach is the low level of abstraction; in other words, the need for a time consuming implementation of the code that demonstrates a certain cryptographic routine. Moreover, we noticed that this option is coupled with burdensome debugging of the code due

to the fact that the expected results of cipher operations are such that it becomes difficult to determine whether an error has occurred in them or not. One also needs to take into account that the prerequisite programming skills of students within each group differ, thus resulting in different levels of ability to implement certain cryptographic routines which consequently leads to asynchronous work pace of the group.

Cryptool, on the other hand, provides higher level interaction with cryptologic components. There are not many alternatives on this field, so we selected this software as primary tool for practical laboratory classes. Lack of alternatives also limited possibilities to compare its effectiveness against other known implementations.

Using the Cryptool add-on allows for a more interactive student work with cryptographic methods compared to independent implementation of these methods by means of the above said programming languages. This tools provides the appropriate level of abstraction and visualization in working with cryptographic methods. All popular cipher algorithms are available as built-in components. One connects different routines by simply connecting their graphical representations.

Additional Cryptool functions that allow for communication over the local network and the Internet have been developed in order to facilitate student collaboration on problem solving. Also, these functions cause the group to advance at a more even pace. While utilizing the Cryptool add-in we observed considerable improvements in students results (grades, attendance, activity, creativity) which we discuss in the section devoted to evaluation of the solution.

Adding networking capabilities to Cryptool

One of the most fundamental shortcomings of applying Cryptool in a classroom setting in a standard way is the inability to solve problems jointly, by having several students work together. The interface itself provides for a single-user work mode, whereas attempts to have two or three students solve a problem at a single workstation have failed to produce positive results. This problem becomes particularly obvious in scenarios with three participants (e.g., Man-in-the-middle attack), but also in problems where two separate sides take part in encrypting and decrypting data.

Primary contribution of this work is the addition of networking functions to Cryptool that are aimed at enabling collaborative work of students. Developing a complete Cryptool-like platform from scratch was deemed both prohibitively time consuming and laborious if one were to achieve the same level of functionality. Given the modular structure of Cryptool and the possibility for adding self-developed add-ins, the main contribution of this work is the creation of a communication capability between two running instances of this program on different computers.

The solution model

The proposition we discuss here is implemented in the form of two novel Cryptool components which enable this tool to communicate over a network on a client-server architecture level (Figure 1). Components were developed in the C# programming language and the standard Cryptool module interface. Source code is published and available free of charge on the project web site.

The first component, *TCP server*, requires only one parameter to be entered by the user, namely the (number of the) port that is to be used by the server. Once this is done and the simulation is initiated, this component requests from the host operating system to redirect the traffic from the specified port to the component's inner structure. Incoming network traffic at the same time becomes the output of this component and is subsequently redirected towards other components – cryptographic modules within the particular scenario.

The second part of the solution is the *TCP client* component. This component requires two input parameters: IP address of the computer with a running Cryptool instance and running TCP component, and the designated port of the server in question. Input to this component is the regular expression from other components in the scenario which is sent over the network in the original form.

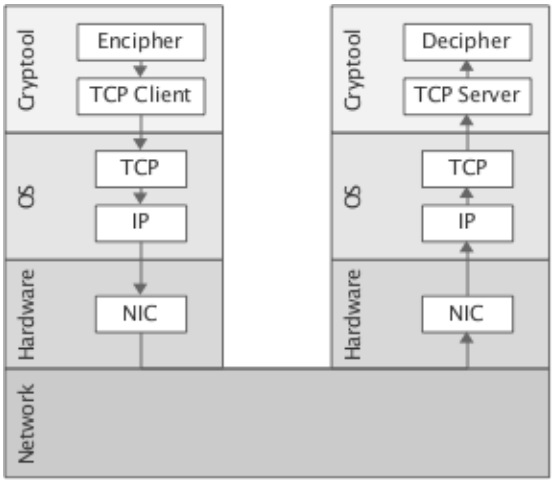

Figure 1. System architecture of the solution

Using the two components we describe above makes it possible to simulate a wide array of network scenarios. In cases where a scenario might require a communication intermediary, this role is accomplished by using both components – TCP server receiving original traffic and TCP client forwarding it to the final destination (Figure 2). Other components dedicated to decrypting and alteration can be inserted in between these two.

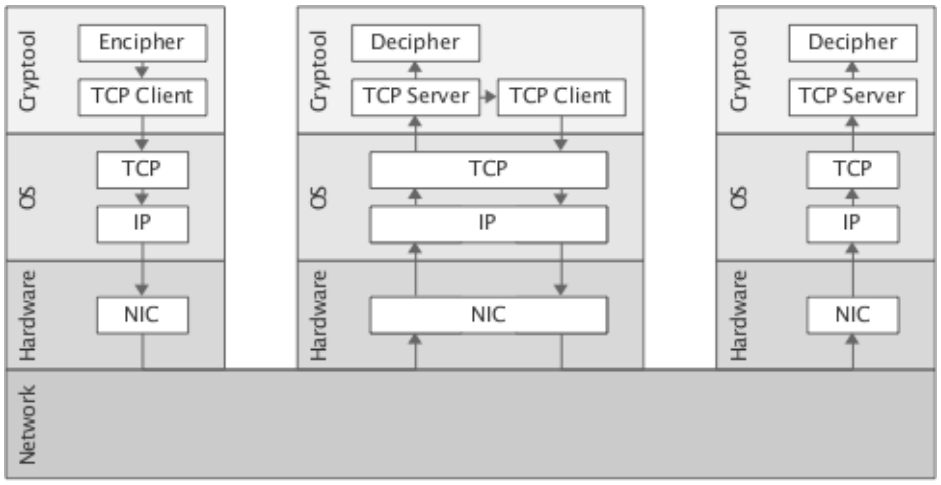

Figure 2. A model with several Cryptool instances coupled

One should keep in mind that the operation of components is not strictly limited to communication with other Cryptool instances. Instead, they may be used to connect this tool with other network systems and services. Moreover, components are in no way constrained to operation on LAN, but can be used for student collaboration over the Internet.

Lab examples

This portion of the paper presents two examples of lab exercises utilizing our solution. The first exercise serves as the introductory example for the course and pertains to the use of the Caesar cipher, which is one of the simplest tools employed for encryption of text messages. This exercise is carried out with students working in pairs.

The second exercise we describe is concerned with the *man-in-the-middle* type of attack. This exercise is of significntly greater complexity and requires participation of three students for collaborative learning (Lewis & Lunsford, 2010).

Caesar cipher

Caesar cipher is one of the simplest methods for encryption of text messages. It consists of shifting the letters of a particular alphabet for a previously agreed number of positions. Although it nowadays bears only a historical significance, Caesar cipher readily lends itself to demonstration of basic cryptologic concepts, be it cryptographic (algorithm and key) or cryptanalytic (attack techniques). Cryptool single-user mode provides students with the opportunity to acquaint with the Caesar cipher principle. Modules for encrypting and decrypting with Caesar cipher have been built into Cryptool.

Figure 3. Using Caesar Cipher in Cryptool

Using proposed and developed Cryptool extension has made collaborative student work on understanding Caesar cipher possible. In this way, better insight into practical use of the cipher is gained since, in addition to the basic communication channel, students must agree on the shift (key) they will be using. In this scenario, students are arranged in pairs, where one participant is tasked with entering the open message that should be transmitted and with defining the module of the Caesar cipher in the encryption mode. Encryption result, which is the output of this module, is redirected to the TCP client, which is supplied with the IP address and the port number of the other participant's computer. This is the mechanism for exporting the cipher from the local Cryptool environment and for transfer over the local network to the remote computer environment.

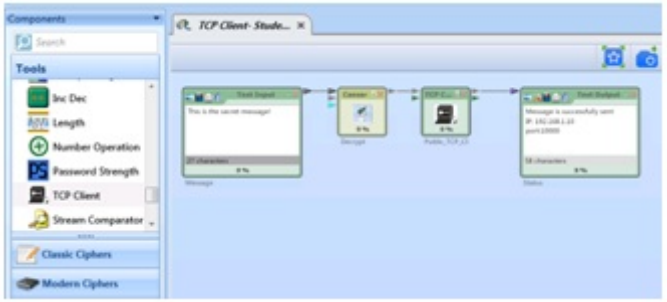

Figure 4. Modules on the side of the encrypting participant within the networking Cryptool environment

Figure 5. Modules on the side of the decrypting participant within the networking Cryptool environment

The Cryptool TCP server component must be initiated on the recipient side with the aim of receiving the data sent by the encrypting party. Incoming data is redirected to the Caesar cipher module in the decryption regime. When exercise has been successful, the receiving side gets the original message. In case procedure has not been followed through correctly, the message received is not intelligible.

A necessary addition on the receiving side is the Gate component, inserted between the TCP server and the Caesar component. This component is dedicated to synchronization of recipient's and sender's environments. Namely, it delays decryption until data has arrived. Naturally, in order for the system to function properly and to establish the network connection between the parties, it is necessary to first start the environment of the recipient.

Man-in-the-middle

In the realm of cryptography and computer networks *Man-in-the-middle* is a complex attack scenario in which attacker actively eavesdrops on the communication channel between two persons. Attacker is entirely independent of the parties who believe they are exchanging messages over a private connection, while the entire conversation is controlled by the attacker.

Attacker may take on either a passive or active role. In case attacker is passive, he/she will be eavesdropping on the entire conversation in order to obtain information. In order for the attacker to be active, he/she must be able to intercept all messages, change their meaning and pass on the message.

In Cryptool single-user mode, students cannot simulate *Man-in-the-middle* attacks faithfully. For example, an attacker within reception range of an unencrypted Wi-Fi wireless access point can insert himself as a man-in-the-middle.

Utilization of originally developed Cryptool extensions allows for collaborative work of students on understanding the scenario of the Man-in-the-middle attack as well as devising new ways to defend from this type of attack utilizing cryptographic methods for the intended communication environment.

In this scenario, students are divided into groups of three. Two students, student A and student B, represent participants in the communication who exchange cryptologic keys by the Diffie-Hellman method (a method for exchanging cryptographic keys with symmetric key ciphers), whereas the third party is the student C assuming the attacker role (Man-in-the-middle).

In the first phase of the scenario, student A calculates information for student B based on his secret *a* with the help of the module intended for calculation of large prime numbers. The result is passed on to the module for conversion of data types that serves to synchronize modules. Output from this module is directed to the TCP client that is provided the IP address and the port number on the workstation operated by student B. This is the sequence of steps that enables exporting of a message from local Cryptool environment and its transfer to student B over a computer network.

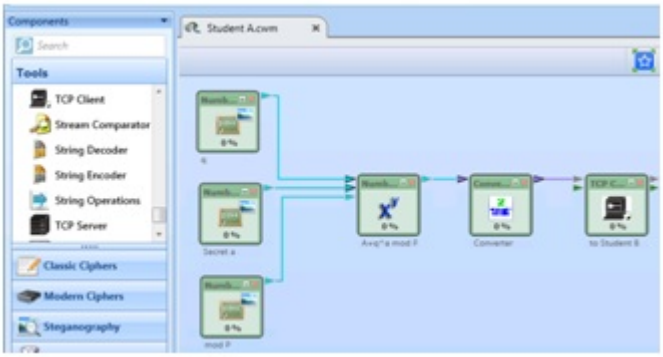

Figure 6. First phase of the Diffie-Hellman algorithm on the side of student A

In the second phase of the scenario, the Cryptool TCP server component must be started on the side of student B with the purpose of receiving data sent by student A. In this phase students A and B are oblivious to the fact that they became victims of the Man-in-the-middle attack. Received data is redirected to the module for calculation of the cryptologic key. Next, based on his secret *b*, student B calculates the information for student A and transmits the message. In case exercise is properly carried out, the receiving side, student B, gets a counterfeit message from student C playing the attacker role and posing as student A. Student B, unaware of the deception, replies to student C and establishes a joint cryptological key with the attacker.

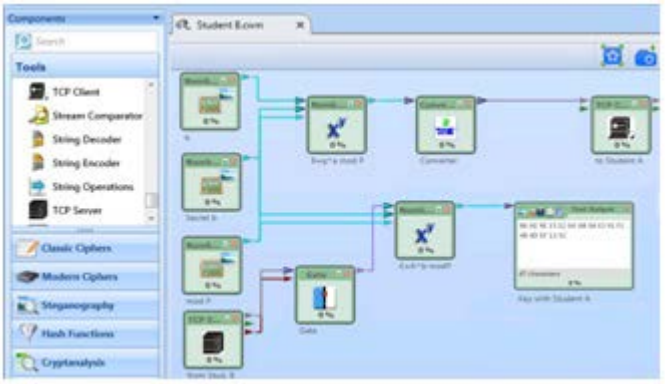

Figure 7. Second phase of the Diffie-Hellman algorithm on the side of student B

Cryptool environment on the side of student C, the Man-in-the-Middle, includes the TCP server component that is tasked with receiving data sent by student A. Incoming data is redirected to the module for calculation of the cryptologic key. Based on the data received, a joint cryptologic key between student C and student A is determined. Next, based on his/her secret *c* student C calculates information for student B and sends it to him. When exercise is properly executed, the recipient, student B, receives a message from student C, the attacker, posing as student A. Believing that he is communicating directly with the intended person, student B responds to student C and sets up a cryptologic key with the attacker.

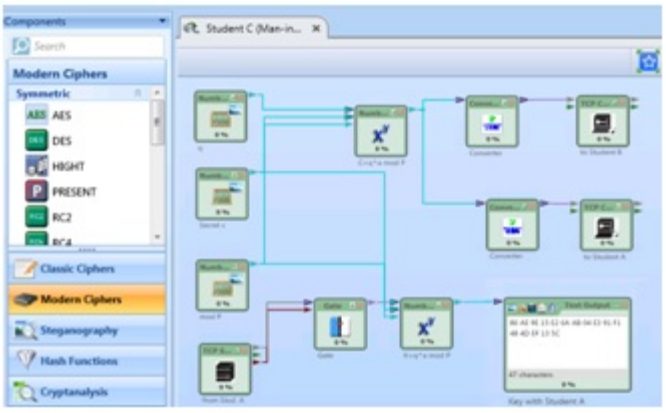

Figure 8. Third phase of the Diffie-Hellman algorithm on the side of student C

When students completely understand the scenario behind the Man-in-the-middle attack they can further design a cryptographic system in which a communication between students A and B would be protected with other cryptographic algorithms.

Evaluation

Introduction of tools and extensions described herein into the teaching process has resulted in a good overall reception by the students and obvious positive effects. One of the most significant effects we observed is the

exchange of knowledge and insights between students. Those who were quicker to understand and adopt presented concepts had a positive impact on students that needed more time. One of the reasons for this was the wish of advanced students to have competent collaborators for scenarios that require more participants. Consequently, this helped to prevent polarization of students on the basis of different learning speeds and provided an opportunity for the group to advance faster as a whole. Also, even though attendance in lectures and exercises is not mandatory, it has risen sharply leading to an increase from 59% to 75% in more than half of the classes as compared to the previous academic year.

Significant improvement was observed in final student grades, i.e., final grades in the first three exam dates following the end of the course. During 2010/2011 distribution of student grades was approximately uniform. In year 2011 students had better grades, where a considerable portion of them attained close to the maximum number of points (Figure 9).

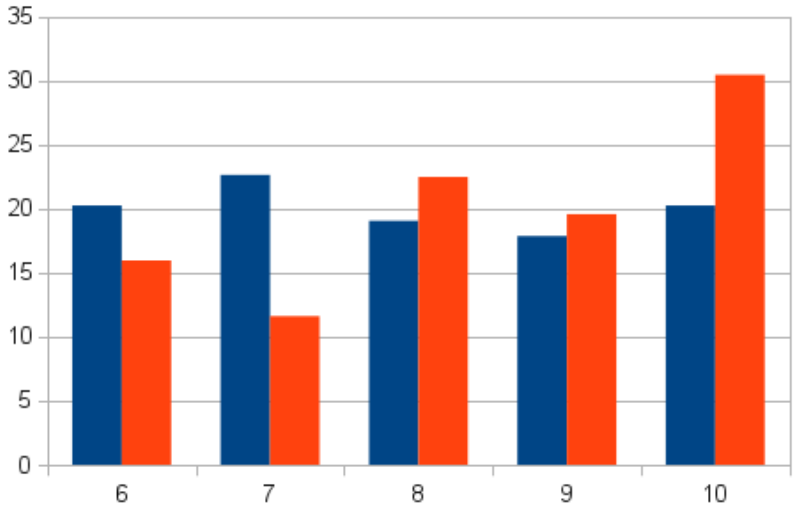

Figure 9. Distribution of grades in 2010/2011 and 2011/2012

We have conducted the statistical analysis of final student grades. To this end we used the T-test (Table 1). Results of the 2010 group were adopted as a control group, whereas the 2011 class – the class that has actually used the system – was chosen as the treatment group.

Table 1. T-test result and intermediate values

Intermediate values used in calculation			
Control Group		Treated Group	
Mean	7.95	Mean	8.37
SD	1.43	SD	1.43
SEM	0.16	SEM	0.12
N	84	N	138
Standard error of difference			0.198
Degrees of freedom			220
T value			2.1079
Confidence interval			
CG mean - TG mean			-0.42
Confidence interval (95%)			-0.81 to -0.03
P value and statistical significance			
P value			0.0362

Control group mean and standard deviation were 7.95 and 1.43, respectively. In case of the treatment group, mean was 8.37 and standard deviation was 1.43. Based on the comparison of results with the corresponding values in T-table (for a statistically significant p - value of 0.0362, calculated on the basis of degrees of freedom for both groups), there exists a statistical significance of the results.

Conclusion

This work presents a solution for interactive and collaborative work of students on adopting the cryptology material. The solution is based on the use of Cryptool, a software package that offers a graphical interface for modeling and simulation of cryptological scenarios with the ability to use popular algorithms.

Appropriate program extensions were developed in order to facilitate collaborative work of students – namely, *TCP server* and *TCP client*. Utilization of these modules enables communication between operating environments of students working on different computers within the local network.

Deployment of the proposed solution offers an adequate level of abstraction in handling cryptographic and cryptanalytic principles and algorithms. This approach has resulted in tremendous savings of time needed to implement cryptographic procedures in classical programming languages, with particular emphasis on time saved on debugging. Focus has shifted from understanding mathematical fundamentals of cryptographic algorithms and their implementation in programming languages to understanding of architecture and weak points of complex systems. Stress was also put on using best practices to implement described mathematical systems.

In the paper, we proposed and discussed the results of our approach applied in practice. Besides better final student grades, we have observed a significant increase in student interest for the area manifested in greater dedication to lectures and exercises, as well as a more active and creative participation. In addition, better continuity of material adoption was also achieved on an individual level.

Acknowledgments

This work was supported by the Ministry of Science and Technological Development of Serbia through the projects TR32054, III 44006 and ON 174008. Project website: http://cryptool.singidunum.ac.rs

References

Rachid A., Kevin P., & Georgios T. (2008). *An animated cryptographic learning object*. Paper presented at the Fifth International Conference on Computer Graphics, Imaging and Visualization: Modern Techniques and Applications, Penang, Malaysia. doi: 10.1109/CGIV.2008.15

Matthaus W., Arno W., & Torben W. (2010). *Towards peer-to-peer-based cryptanalysis*. Paper presented at the 35th Annual IEEE Conference on Local Computer Networks, LCN 2010, Denver, Colorado, USA. doi: 10.1109/LCN.2010.5735672

Jingtao L., Yiming Z., & Lei S. (2009). *Interactive teaching methods in information security course*. Paper presented at the Eighth International Conference on Embedded Computing, Dalian, China. doi: 10.1109/EmbeddedCom-ScalCom.2009.94

Feng Y., Cheng Z., MengXiao Y., & YiRan H. (2009). *Teaching cryptology course based on theory-algorithm-practice-application mode*. Paper presented at the First International Workshop on Education Technology and Computer Science, ETCS 2009, Wuhan, China. doi: 10.1109/ETCS.2009.366

Xiulli, S., & Hongyao, D. (2009). *Taking flexible and diverse approaches to get undergraduate students interested in cryptography course*. Paper presented at the First International Workshop on Education Technology and Computer Science, ETCS 2009, Wuhan, China. doi: 10.1109/ETCS.2009.371

Lewis, J., & Lunsford, P. (2010). *TLS man-in-the-middle laboratory exercise for network security education*. Paper presented at the 11th Conference on Information Technology Education, Midland, MI, USA. doi: 10.1145/1867651.1867681