Reducing the complexity of understanding cryptology using CrypTool

Sibylle Hick
Deutsche Bank AG, Alfred-Herrhausen-Allee 16-24,
Eschborn, 65760, Germany
sibylle.hick@db.com
and

Bernhard Esslinger
Institute of Information Systems, FB5, University of Siegen
Siegen, 57072, Germany
esslinger@fb5.uni-siegen.de
and

Arno Wacker
Software mechanisms for Ubiquitous-Computing-Applications, University Kassel,
Kassel, 34121, Germany
arno.wacker@uni-kassel.de

ABSTRACT

Cryptography and cryptanalysis are characterized by a great number of algorithms, parameters, and protocols where at least some of them can be considered as complex mathematical structures. As cryptography is a crucial part for securing many modern applications, it is important on the one hand that developers understand what are efficient and correct ways to implement those security mechanisms. On the other hand, users of such mechanisms need to get a better understanding why and how to apply them properly. In this paper electronic learning such as self-regulated learning, computer-based trainings, and learning by doing are presented with the help of a modern e-learning program – CrypTool (CT) – which supports both afore described approaches for developers and users. For users who are interested in cryptography, CT provides an easy way to experiment with cryptographic algorithms and protocols by augmenting complex algorithms with visualizations and context-related explanations. This paper gives an overview how to implement and apply cryptographic concepts and how to teach them. Additionally, this paper fortifies these claims with an evaluation using the feedback from classes where CT was used to teach cryptographic algorithms.

Keywords: E-learning, CrypTool, cryptography, cryptanalysis, visualization, visual programming.

1. INTRODUCTION

A sustainable approach to understand something which might look at first glance quite complex is to teach it yourself because then you have to research all relevant information from different sources and work through it step by step to get the desired knowledge. This approach will be connected with different questions which have to be solved and answered. A didactic method to achieve this objective is the concept of e-learning by using modern technologies. In this paper we give an overview of some principles of e-learning and show in a case study how these principles have been taken into account when the free e-learning tool CrypTool has been designed and implemented.

E-Learning can be used in contrast to traditional teaching concepts to study a specific topic. A brief overview of e-learning approaches is presented in Section 2. In this paper e-learning is considered against the background of cryptography and cryptanalysis. Even though cryptographic mechanisms can be found in many modern applications, a lot of users consider the mathematic principles to be complex. At the same time, there is a great demand for cryptographic mechanisms because electronic applications should often process or transfer data in a secure and trustful way. E.g. a person might want to book a vacation in the Internet, would like to buy a book online, or wants to do online banking or store passwords on a smart phone. In all these cases, security is a base requirement which is established based on cryptographic mechanisms. In Section 3 we provide an overview of the free and open-source software CrypTool which invites different target groups to learn more about cryptology. This can either be achieved by using the application CrypTool to visualize and explain cryptographic mechanisms or adding new features and protocols to the software by implementing those as outlined in Section 4. Afterwards, we describe in Section 5 how this has been realized in the second version of the software "CrypTool 2". By using the application within the scope of schools and universities, first experiences with this concept have been made and are shown in Section 6. Finally, a conclusion is drawn in Section 7.

2. APPROACHES FOR E-LEARNING

Different didactic methods have been described in the literature to impart knowledge. In schools knowledge is often provided to students by face-to-face lectures that present a specific topic, allow for questions, and mostly use exercises to demonstrate the topic in more detail. With the introduction of new media such as computers and smart devices additional ways have been presented to teach or educate a complex topic.

One approach described in [1] and [2] is called self-*regulated learning* and encourages a person to understand something in a practical way with the help of digital media. In [1] self-regulated learning is described by three components: At the beginning of the learning process a person is guided through the process but is encouraged over time to inherit more and more responsibility and flexibility what and how information is received. Secondly, it is outlined that working as a team can be used to collaborate and bring together different perspectives. Thirdly, learning results and ideas can be shared.

Many companies use the concept of *Computer Based Training* (CBT) or *Web-Based Trainings* (WBT) to provide different kinds of information to their employees in a modern and interactive way e.g. by using sounds, figures, and/or interactive quizzes (compare [3] and [4]). Thereby, different kinds of technologies allow the user to get more and more involved in the e-learning process. While CBTs are considered as offline applications WBTs allow connecting more information and people with each other online.

E-Leaning can also include game-based approaches (compare e.g. [5]). In [4] this is repatriated to the concept of "trial and error" which challenges the user and allows improving knowledge and behavior over time.

As mentioned before multimedia is an important part of e-learning. It is stated in [6] that interactivity and *visualization* can help to understand mathematical concepts easier because "spatial rationalness and virtual imagination are trained".

3. THE OPEN-SOURCE PROJECT CRYPTOOL

CrypTool is a free e-learning software for cryptography and cryptanalysis which can be downloaded from the CrypTool portal at www.cryptool.org. Originally, it was developed for an awareness and IT security initiative within Deutsche Bank [7]. In the course of time, it became an open-source project with more than 60 volunteers offering more than 200 cryptographic functions [8] within an e-learning environment and which is today used by many users all over the world: pupils, students and others interested in cryptography.

The first version, called CrypTool 1 (CT1) was implemented as a Windows application following the concept of CBT in the programming language C++ and is available in five languages: English, German, Spanish, Polish, and Serbian. The next release 1.4.31 of CrypTool is planned for Q3/2012 – localizations also for Greek and Russian are in progress [9]. Today, CT1 is mainly enhanced only by smaller changes like bug fixes and afore mentioned new localizations due to the fact that two successors – which can be understood as the next generation of this application - have been released: CrypTool 2 (CT2) and JCrypTool (JCT).

CT2 is also implemented as a Windows application but based on the concept of visual programming. The application uses a plug'n'play interface which allows a user to connect different kinds of components to build individual workflows via drag&drop. Such workflows can also contain loops and conditions, allowing the user to create complex programs. This enables the user to specify cryptographic functions or a complex combination of cryptographic primitives from a visual perspective. CT2 is written in C# using Visual Studio 2010. While CT2 is currently available in English, the implementation of a German version has already been started. Developers can easily add new functionality via the .NET Framework and a plug-in infrastructure. The second successor is JCT. This variant has been implemented in Java by using Eclipse. Thus, it executes on different platforms such as Windows, MAC OS, and Linux. Like CT2, JCT also supports the concept of plug-ins so that new cryptographic algorithms and protocols can easily be added. JCT consists of two main parts: "JCrypTool Core" which is supported by a small team to provide the base functionality and "JCrypTool Crypto" which is open to all interested developers to implement crypto plug-ins.

Besides the afore described three versions which are full-blown programs to be installed on your computer, the browser-based CrypTool-Online (CTO) supporting the concept of a WBT has been made available on www.cryptool-online.org. This allows CrypTool to run on a smart phone which could be used at any location in daily life or leisure (e.g. to solve geocaching riddles).

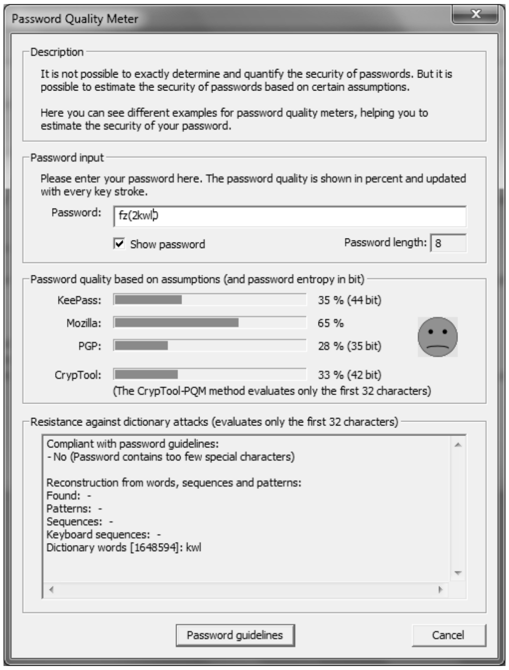
CrypTool helps to establish awareness and a better understanding for IT security not only in universities, but also within companies and public authorities. Thus, awareness trainings have been conducted with the application CrypTool e.g. in Deutsche Bank, Boeing, Microsoft, the Federal Office for Information Security, and the Federal Criminal Police Office in Germany.

4. E-LEARNING – THE CONCEPT OF TEACHING YOURSELF CRYPTOGRAPHY WITH CRYPTOOL

The Application Perspective

Using an application that requires security mechanisms does not automatically mean that every person has to implement this mechanism by her- or himself. Nevertheless, it can be advantageous and is recommended to understand the principle behind this mechanism better because this can e.g. help against attacks within the scope of social engineering.

For example, if you have subscribed to a service like online banking, where you have to present a login name and a PIN or password, you should be aware what exact information is required by you in order to successfully log in to this service and afterwards use it as desired. Otherwise, you could become a victim of a phishing attack, where someone tries to direct you to a fraudulent homepage to receive some valid TANs that allows – together with some eavesdropped information – to do unauthorized transactions with your account. As another example, a user in the Internet has often to specify a password to register for an online service. In many cases, there is the requirement to choose a password consisting of upper and lower case letters and it should consist at least of 8 digits. However, for the user this yields to the question how to find a password to use the respective service in a secure and trustful way. Figure 1 shows an example for a Password Quality Meter in CT1 providing the user instantly information on the estimated security of a password entered.


Password Quality Meter

Description

It is not possible to exactly determine and quantify the security of passwords. But it is possible to estimate the security of passwords based on certain assumptions.

Here you can see different examples for password quality meters, helping you to estimate the security of your password.

Password input

Please enter your password here. The password quality is shown in percent and updated with every key stroke.

Password: fz(2kwl)

☑ Show password

Password length: 8

Password quality based on assumptions (and password entropy in bit)

KeePass: 35 % (44 bit)
Mozilla: 65 %
PGP: 28 % (35 bit)
CrypTool: 33 % (42 bit)
(The CrypTool-PQM method evaluates only the first 32 characters)

Resistance against dictionary attacks (evaluates only the first 32 characters)

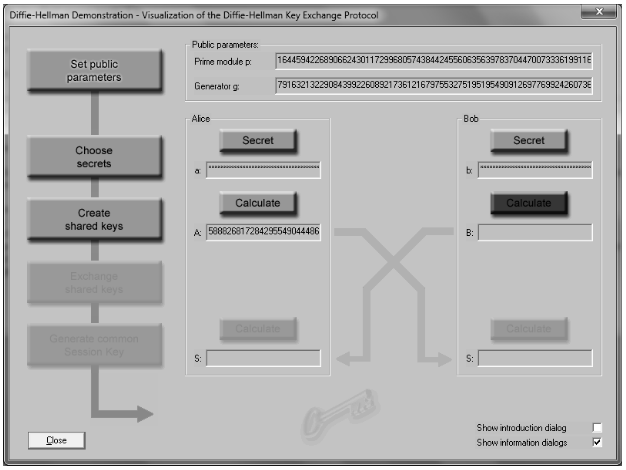
Compliant with password guidelines:
- No (Password contains too few special characters)

Reconstruction from words, sequences and patterns:
Found: -
Patterns: -
Sequences: -
Keyboard sequences: -
Dictionary words [1648594]: kwl

Figure 1: CrypTool 1 - Password Quality Meter

An e-learning tool like CrypTool provides the opportunity to guide the user step by step through a specific security mechanism by supporting the concept of self-regulated learning. Thereby, the user is enabled to change single properties that might lead to different results. An online help can be opened to provide additional information to the requested topic. However, the user cannot only learn how a security mechanism works based on the configured parameters. The cryptanalysis part also illustrates how and why the mechanism could be attacked. This allows for a comprehensive view for security protocols. The strength of a cryptographic algorithm is, in many cases, related to a time measurement indicating how long the algorithm can be used based on current calculation mechanisms of current computers.

Visualization as described in [6] is a good way to provide insight into a security mechanism. Often, a cryptographic protocol might require two communication partners to exchange information even if they have never met before. After they have both performed some calculations on their own sides, communication partner A (Alice) sends one of her results over to the other communication partner B (Bob), who then does additional calculations and sends the result of his calculations back to A. This is done e.g. by the Diffie-Hellman key exchange protocol [10]. Even though this might sound very theoretical, this is a mechanism that is included in a lot of applications. E.g. it can happen when you log on to an Internet page by using Transport Layer Security [11]. As another example, the Diffie-Hellman key exchange protocol is applied when a European ePassport is electronically checked within the scope of Chip Authentication [12] at the border in the moment the ePassport is read by an electronic inspection system. As described before, the protocol requires the exchange of information between two different communication partners (here: inspection system and ePassport). As a result, protocols are a very good candidate for an animation or visualization because the user can be guided step by step through the protocol viewing both perspectives (compare Figure 2).


Diffie-Hellman Demonstration - Visualization of the Diffie-Hellman Key Exchange Protocol

Public parameters:

Prime module p: 16445942268906624301172996805743844245560635639783704470073336199116

Generator g: 7916321322908439922608921736121679755327519519549091269776992426073€

Alice

a: ************************************

A: 58882681728429554904448€

Bob

b: ************************************

B:

S:

S:

☐ Show introduction dialog
☑ Show information dialogs

Figure 2: CrypTool 1 - Diffie-Hellman Demonstration

In the open-source project CrypTool, users of the application can easily give feedback or act as testers of the software. If an additional feature or information could be helpful or an error is recognized, the CrypTool portal offers several possibilities, e.g. with a ticket system to request a change in the software.

The Implementation Perspective

Although the application perspective already provides detailed information on a cryptographic mechanism, even more insight can be obtained when implementing the mechanism by yourself. For example, this could be interesting for a system administrator who needs to enhance a service with IT security mechanisms or a developer who implements software and wants to add encryption, integrity or authentication services.

The successors CT2 and JCT include a plug-in interface which means that if someone wants to add a new feature, one does not have to implement everything from scratch. Additionally, modern programming languages already provide security mechanisms through a defined architecture. E.g. in Java the Java Cryptographic Architecture (JCA), which includes the Java Cryptographic Extension JCE), allows the use of cryptography based on provider(s) that can be integrated through the JCE (compare [13]). Therefore, a developer does not need to implement everything from scratch, but can leverage functionalities from specific libraries and can focus on the parts that are not already included in those libraries. A presentation with detailed information on how to build such a plug-in for JCT can be found in [14]. Although programming languages already offer a baseline for cryptographic functionality, the

CrypTool project does not limit the user to just one way in implementing an algorithm. On the contrary, there is a great flexibility when it comes to the freedom on how an idea for a cryptographic mechanism can be realized. As described before, one example for this flexibility is visualization and this has actually been carried out in all three versions of CrypTool by allowing the user to test mechanisms with different options and/or parameters.

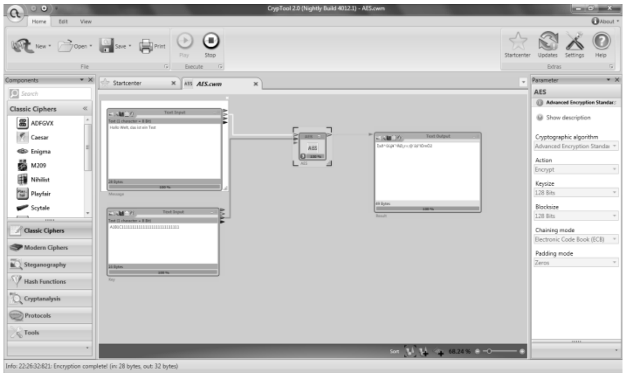
5. SELF-REGULATED LEARNING – AN EXAMPLE WITH CRYPTOOL 2

The Application Perspective

Visual programming is a key element in CT2 offering the user a flexible way to reassemble all components of a cryptographic workflow. On the application side, a user has different options to start off by using:

- the wizard,
- predefined templates, or
- the workspace manager.

The wizard allows a user to choose from the features in the tool and to combine them by choosing step by step from different options, with explanations provided in each step. At the end of the wizard process, the result is shown as a workflow in a new project. The second possibility to start with is to choose from the predefined templates. If the user selects one of the templates, a complete workflow for that mechanism is opened up as a specific project. A workflow consists of different components that are in some way connected to each other.



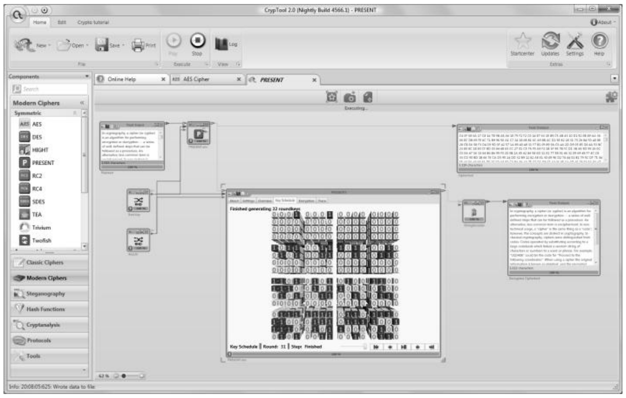
Cryptographic algorithm: Advanced Encryption Standard
Action: Encrypt
Keysize: 128 Bits
Blocksize: 128 Bits
Chaining mode: Electronic Code Book (ECB)
Padding mode: Zeros

Info: 22:26:32:821: Encryption complete! (in: 28 bytes, out: 32 bytes)

Figure 3: CrypTool 2 – Simple Workspace with AES Cipher

For example, Figure 3 displays a workflow showing an encryption method including a "Text Input" component with the plain text message that should be encrypted, a component visualizing the encryption method (here AES), and a "Text Output" component which contains the encrypted text as a result after the workflow has been executed. More complex workflows based on loops and conditions can be created. As an example, we provide a template which loops through all possible Caesar decryptions and stops when enough words from the decryption are found in a given dictionary. With this, the general concept of brute-forcing an algorithm is visually demonstrated. A more experienced user can start with the menu item workspace manager, which then opens up an empty project where the user can choose individual components from the left hand side in order to design own processes.

Independently of how the user starts, a workflow will be displayed: It could be as easy as the AES cipher in Figure 3 or more complex like the PRESENT cipher [15] in Figure 4, which shows, that even within a component of the workflow, visualizations can be displayed.



Finished generating 32 roundkeys

Info: 20:08:05:625: Wrote data to file

Figure 4: CrypTool 2 – Workspace with PRESENT Cipher

A component can have one or more interfaces each specified by a set of arguments and their data types either as input (left side of the component) or output (right side of the component). This allows data in different formats to flow from one component to another. The user can choose components from the component editor on the left side and connect an output interface with an input interface if matching data types exist. If a component is selected in the main project window, the user can switch between different views: visualization, parameters, log, data and help. Each view provides different information about the component (if implemented by the developer): The visualization view can show an arbitrary animation for the specific algorithm. Parameters, e.g the key used, can be set in the parameters view (additionally, parameters can also be set by the user on the right hand side when a component is selected (see Figure 3) – this is especially useful, when watching a live visualization of a component and the user wants to see immediate effects while changing some parameters. Furthermore, data in the components, e.g. text in a "Text input" component, that should be encrypted, can be adapted interactively at any time, and the effect can be seen instantly. In order to execute a project (let data flow through the complete workflow), the user has to select the button *Play* which can be found in the *Home* ribbon. As in other versions, CT2 provides the opportunity to visualize not only cryptography, but also cryptanalysis.

The Implementation Perspective

In order to help a user starting to implement an own plug-in for

CT2, an exhaustive "Plugin Developer Manual" [16] can be found on the CrypTool portal. It includes not only information on how to develop a plug-in for CT2, but also provides information on how to get started, i.e. how to check out the sources from the Subversion repository and compile the code within Visual Studio 2010 or Visual C# 2010 Express (read access to the repository is provided for anyone). This is in particular helpful because a developer needs to understand how to work with the developing environment before starting with the plug-in itself. To receive write access to the repository, the user can contact the CrypTool team over the CrypTool portal.

Besides the implementation of a plug-in, the user needs to research the information for the cryptographic feature. This is usually done in accordance with standards or published research papers. Afterwards, the developer decides how the feature should be visualized. This is not only important for the implementation, but also for writing the integrated online help.

Finally, when the user has implemented the plug-in, including the respective documentation, additional testing must be carried to ensure quality assurance. Usually, testing is not only performed by the developer, but can also be achieved by other users, as the plug-in could be integrated in an interim version called '*nightly build*'. The CrypTool portal offers both stable CT2 versions (beta and release versions) as well as nightly builds, which are automatically compiled over night based on what has recently been checked in into the Subversion repository. So the software development process of CT2 follows an idea of agile and extreme programming – always have running code [17].

6. EXPERIENCES WITH TEACHING CRYPTOGRAPHY

Cryptography is one area in schools and universities that offers students an exciting and motivating way to work on complex mathematical questions. One example for an event to teach cryptography to high school students and school children is described in [18]. In Germany, the event "Schuelerkrypto" (English: student crypto) has been organized since 2009. During this event, an overview of the principles of cryptography as well as cryptanalysis is provided. Presentations are followed by small exercise classes on the computer where CT1 and CT2 have been installed in advance. The exercises are designed as a special agent story i.e. a game-based approach to find the thieves of some famous old paintings. The following statistics have been compiled based on course feedback from students who took part in the event in 2011 (24 participants) and twice in 2012 (29 participants and 22 participants). The participants of the pupil crypto event were between 14 and 19 years.

For a better understanding how CrypTool has helped to change the knowledge in cryptography the students were asked how much experience they had in this topic before the event. In 2011 70% of the pupils expressed that they did not have any experience with cryptography before. 100% of the pupils stated that the event helped them to improve their knowledge in cryptography. Also in 2012 the pupils participating in the event were asked about their knowledge regarding cryptography before the event. This time 79% (59%) responded that they did not have any experiences with this topic. 90% (86%) stated that the event helped them to improve their understanding of cryptography while 10% (14%) responded that the event helped them partly to improve their knowledge.

In order to get a better understanding on how the usability of CT1 and CT2 was perceived the pupils were also asked to rate the two applications. In 2011 CT1 was rated by 29% with very good and 63% with good usability by the students. At the same time CT2 was rated 50% very good and 29% good usability. The students from the Schuelerkrypto 2012 event rated CT1 with 10% (36%) very good and 59% (45%) good. The usability of CT2 was rated 24% (45%) very good and 76% (36%) good. The results are also shown in Figure 5.

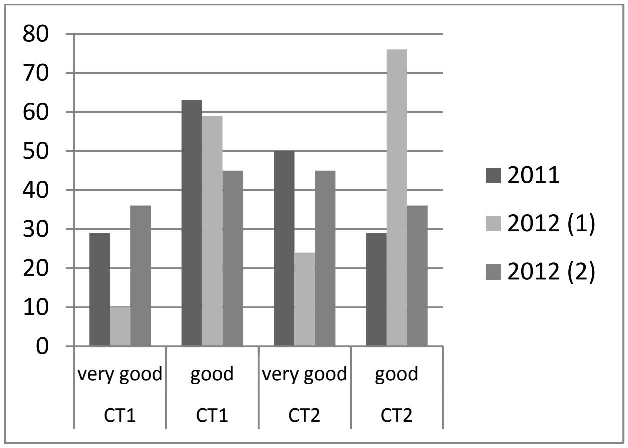


Figure 5: Usability of the CT Software Version 1 and 2

Being asked if CrypTool as an e-learning tool has helped to provide a better understanding on cryptography and crypt analysis in 2011 83% of the students agreed to this statement and 17% partly agreed. As a comparison in 2012 90% (86%) of the students agreed to CrypTool being helpful as an e-learning tool while 7% (14%) stated that the application is partly helpful.

As most of the students had no or less experience and pre-knowledge with cryptography they rated the learning-curve and the usage of the e-learning tools CT1 and CT2 very positively.

7. CONCLUSIONS

The open-source software CrypTool supports, in three different versions, an interesting and exciting way for e-learning within the scope of cryptography and cryptanalysis by considering concepts in particular such as self-regulated learning, CBTs and WBTs, and visualization. CrypTool is not only focused on developers that want to implement a security mechanism by themselves but also addresses a user that just wants to understand the concept on an application layer. Not only small number examples to reduce the mathematical complexity, but in particular the concept of visualization and collaboration of different people adding more information and functionality to CrypTool are key components to achieve this objective.

Flexibility is very important to allow the user or developer to try different approaches how something should be realized. Of course, flexibility should always be balanced with a good architecture approach of such an e-learning software. Nevertheless, the concept for flexibility is not yet finalized. In CT2, it is planned for a future release to allow a developer to insert code over the application view during runtime. Thus, typed-in C# code could be executed directly. Another planned feature is to easily support building up ad-hoc peer-to-peer networks of multiple CT2 instances by different users.

This paper has shown different concepts used to educate people on cryptography and cryptanalysis with CrypTool in order to raise security awareness. But following the concept of collaboration – CrypTool is not limited that users can only learn with the application or enhance the software by adding new functionalities. The concept of this e-learning software is completed by an international Crypto Cipher Contest called MysteryTwister C3 (MTC3) [19] which allows to test the self-trained skills directly on ciphers which have been defined with different severity levels. Again a participant is invited to just solve the riddles that have been posted or to share the new knowledge that has been gained by using CrypTool by adding a new self-made riddle to the contest.

7. REFERENCES

[1] A. Hohberg, P. Gohlke: Selbstorganisiertes Lernen 2.0. Ein neues Lernkonzept für die berufliche Weiterbildung. In: IT-gestütztes Lernen & Wissensmanagement. Hrsg. J. Hofmann, J. Jarosch. Heft 277, Februar 2011.

[2] Bundesministerium für Bildung und Forschung (BMBF): Kompetenzen in einer digital geprägten Kultur. 2010. Available at: http://goo.gl/ZNU3u

[3] A. Kuhlmann, W. Sauter: Innovative Lernsysteme: Kompetenzenentwicklung mit Blended Learning und Social Software. X.media.press. Springer 2008.

[4] M. Pivec: Informationsdidaktik: E-Learning. In: Hrsg. W. Weber: Kompendium Informationsdesign. Part 3, p. 273 – 302. Springer 2008.

[5] J. Ryoo, A. Techatassanasoontorn, D. Lee, J. Lothian: Game-based InfoSec Education Using OpenSim. Pennsylvania State University. 15th Colloquium for Information Systems Security Education Fairborn, Ohio June 13-15, 2011.

[6] V. Enss, M. Holschneider, S. Jeschke, T. Paehler, R. Seiler: MathletFactory: Komponentenframework und Autorenumgebungen für mathematische Applets interaktiver eLearning-Plattformen. In: Hrsg. K. P. Jantke, K.-P. Fähnrich, W. S. Wittig. 13. Leipziger Informatik-Tage, LIT 2005. Available at: http://goo.gl/2r77F

[7] B. Esslinger: CrypTool – an open source project in practice. Lessons learned from a successful open source project. Published in Datenschutz und Datensicherheit. Edition March 2009.

[8] Overview of the functions in the different CrypTool versions: http://www.cryptool.org/en/ctp-documentation/ctp-functions-en

[9] Roadmap of CrypTool 1 is available at: http://www.cryptool.org/en/ct1-documentation/ct1-documentation-roadmap

[10] RFC2631. IETF: Diffie-Hellman Key agreement method. E. Rescorla. June 1999.

[11] RFC 5246. IETF: The Transport Layer Security (TLS) Protocol Version 1.2, August 2008. Available at: www.rfc-editor.org/rfc/rfc5246.txt.

[12] Technical Guideline TR-03110. Advanced Security Mechanisms for Machine Readable Travel Documents. Version 2.05, 14.10.2010. Available at: http://goo.gl/2TUp0

[13] JavaTM Cryptography Architecture API Specification & Reference. Available at: docs.oracle.com/javase/1.4.2/docs/guide/security/CryptoSpec.html#ProviderArch

[14] D. Schadow: Getting started with JCrypTool. February 2012 Edition. Available at: http://goo.gl/xiGwJ

[15] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, C. Vikkelsoe: PRESENT: An Ultra-Lightweight Block Cipher, 2007. Available at: http://goo.gl/wO0N0

[16] S. Przybylski, A. Wacker, M. Wander, F. Enkler, P. Vacek: Plugin Developer Manual – How to build your own plugins for CrypTool 2.0. Version 0.7, July 16, 2011.

[17] http://goo.gl/Rcq4

[18] N. Koblitz: Cryptography as a Teaching Tool. A Shortened version of the article in Cryptologia, Vol.21, No. 4 (1997) is available at: http://www.math.washington.edu/~koblitz/crlogia.html.

[19] MysteryTwister C3 (MTC3). Available at: http://www.mysterytwisterc3.org.