Email for You (only?) – Design and Implementation of a Context-based Learning Process on Internetworking and Cryptography

Andreas Gramm
1. Schulpraktisches Seminar
Charlottenburg-Wilmersdorf
Otto-Suhr-Allee 100
10585 Berlin, Germany
+49 (0)30 90 29 13 280
gramm@gymnasium-tiergarten.de

Malte Hornung
Freie Universität Berlin
Didactics of Informatics
Königin-Luise-Straße 24-26
14195 Berlin, Germany
+49 (0)30 83 87 51 87
malte.hornung@fu-berlin.de

Helmut Witten
Gesellschaft für Informatik (GI)
Working group "CS education in Berlin and Brandenburg (IBBB)"
Brandenburgische Straße 23
10797 Berlin, Germany
+49 (0)30 88 68 25 69
helmut@witten-berlin.de

ABSTRACT

The didactical approach of teaching computer science in context aims at enabling learners to understand concepts of computer science better through the help of concrete illustration and meaning. This paper describes a learning arrangement in which students in lower secondary school education are motivated to engage with cryptographic algorithms ranging from *Caesar* to *RSA* by making them discover the challenges of a private and trustable communication over public networks. We also describe experiences we made by developing and testing the context-based learning process in several classes of different age. While designing and implementing context-based teaching material proved to be demanding, we were rewarded with a high level of both interest and understanding on the part of the students suggesting that context-based learning will prove a promising didactical tool for computer science teachers.

Categories and Subject Descriptors

K.3.2 [Computer and Information Science Education]: Computer science education

General Terms

Algorithms, Reliability, Security, Human Factors.

Keywords

Context-based computer science education, computer science in context, IniK, communication protocol, email protocol, communication security, cryptology, RSA

1. COMPUTER SCIENCE IN CONTEXT

In the light of new insights into learning through neurobiological research and with regard to a constructivist understanding of learning, it has recently been argued that while educators wish to implement largely transferable, abstract knowledge, concrete contexts are necessary for learners to understand the nature, meaning and use of an abstract idea or concept. Cooper and Cunningham showed how this approach has been applied to the design of introductory courses at university level using 3D animations (Alice), media computation, computer graphics or robotics as contexts [6]. According to their analysis, suitable contexts are a source of illustrative examples, project ideas, motivation and meaning. Guzdial points out that the use of contexts improves retention of students: Students are more likely to continue CS studies when they understand the usefulness and value of what they are learning [17]. But he also warns that learners might overgeneralise a concept when only seen in a single context. He therefore calls for concepts to be presented in various different contexts to allow for an adequate decontextualisation.

While the international debate of learning CS in context primarily discusses introductory courses to computer science at universities, the discussion in Germany focuses on secondary school education. An open project group called "Informatik im Kontext – IniK" (translates to "computer science in context") was established to develop and promote the approach of context-based computer science secondary school education. The project's website [21] describes the concept of IniK and provides teaching material and didactic instructions for teachers for a number of contexts and possible teaching units.

IniK is partly modelled on projects promoting context-based learning of sciences (chemistry, physics and biology), which have received substantial funding from both the Federal Government of Germany as well as the different federal states (known as *Bundesländer*). Out of these three projects "Chemie im Kontext - ChiK" [19] was the first and has been the project most influential for IniK (cf. [10]). In contrast to the three science projects, IniK is more of the character of a grass root development with almost no funding. To our knowledge, Berlin is the only federal state to grant a small amount of resources to a group of teachers committed to the IniK approach. Apart from teachers interested in modern teaching and learning approaches it is mainly researchers from several universities who further develop the concept of IniK.

While the idea of a context-based approach to computer science has been discussed in Germany before (see e.g. [12] and [7]), the IniK approach was first fully described in 2009 by Koubek, Schulte, Schulze and Witten in [24]. They identify three prerequisites for a learning process to qualify as being context-based in the field of secondary school computer science education:

1. Context-based education in computer science makes use of a context that is relevant to students. The context is a concrete situation, in which aspects of different dimensions are significant to understand the situation and to find adequate solutions.
2. The competencies acquired through the learning process are compliant with standards for teaching computer science in secondary school education. The Society for Informatics in Germany *Gesellschaft für Informatik (GI)* has published principles and minimal standards for computer science in lower secondary school education [1] (for an English summary of these standards see [4] and [25]). Referring to the definition of Weinert, the competencies described in the standards are to be understood as “the cognitive abilities and skills possessed by or able to be learned by individuals that enable them to solve particular problems, as well as the motivational, volitional and social readiness and capacity to use the solutions successfully and responsibly in variable situations” ([20], p. 65).
3. Context-based education in computer science shows a variety of teaching and learning methods to activate different types of learners and enhance cooperative student-student interaction in class.

Since the founding of IniK, several further ideas on how to design and implement context-based learning of computer science have been developed. Diethelm, Borowski and Weber suggested a way to find contexts which are relevant to learners by indirectly asking them what interests they have in certain informatics devices [8]. Furthermore, Diethelm and Dörge described a way to derive a list of competencies that can be developed by making use of a specific context [9]. In 2011, Diethelm, Koubek and Witten summarised the development, features and perspectives of IniK and state criteria for the selection of suitable contexts and as to how units could best be documented [11].

One goal of the IniK working group is to design several context-based learning environments with a high permeability to everyday school practice and reflect on its implementation in secondary schools both to prove the feasibility of the concept to other CS teachers as well as to gain experience and further develop the concept. The method used for design and reflection has an alignment to research methods like Action Research [2] and Design Based Research [3]. Note that therefore our conclusions are mostly based on in-service reflection. “Email for You (only?)” is the title of a context-based learning arrangement on internetworking and cryptography which is presented and discussed in this article.

2. DESIGN AND IMPLEMENTATION OF THE LERANING PROCESS

We have developed a context-based learning process on security issues of internetwork communication (for a more detailed documentation of the learning process in German see [22], [16], [14] and [13]). In doing so, we intended to show the feasibility of the IniK concept for a context-based learning of computer science in everyday computer science classrooms as well as to gather experience that can be helpful for the further development of the concept and other context-based learning processes.

The indented age group is from year 9 to year 12 (puplis aged 14 to 18). From prior experience we can say that students often enjoy the topic cryptography, because transmitting secrets is associated with adventure and decrypting or even cracking a cipher resembles a strategic game. However, just discussing cryptographic algorithms such as *Caesar* doesn’t help raising awareness for the fact, that we as users make use, or at least are well-advised to make use of cryptography whenever communicating over public networks such as the Internet. Most students communicate over the Internet on a daily basis, be it via instant messengers, social networks or email. The recent annual survey among German teenagers “Jugend, Information, (Multi-) Media (JIM) 2011” [18] shows that they spend almost half of their online time communicating. The context of security issues in private communication over public networks is therefore part of students’ daily life, only that in most cases they are not aware of it. We believe that rising awareness of both advantages and challenges of using computer systems is a central aim of secondary school computer science education.

The context can be explored with regard to different dimensions. On the one hand, assessing security risks requires a thorough understanding of the underlying network technology and the communication protocols employed. On the other hand, security always requires additional resources. In the case of private communication, it requires the exchange of public keys. Today, many users don’t use encryption even though the technology is available for free. This clearly indicates to human factors such as too little awareness of security issues and possible consequences.

We structured the learning process by five questions:

1. How is an email communicated from my computer to the computer of the addressee?
2. What dangers challenge a private communication over a public network, such as the Internet?
3. How can I achieve privacy?
4. How can I assure the integrity of a message and the authenticity of a sender?
5. Why should I communicate privately?

The answers to questions 2 to 4 respectively build on the knowledge students acquire to answer the questions asked before. This way, they shall experience this knowledge as being useful to solve tasks such as providing security in an unsafe environment. The last question could alternatively be asked at the beginning of the learning process, but in our view it is more rewarding to discuss it when students have got clear ideas of the challenges of communication over public networks, which they might not have at the beginning of the learning process. It also seems more attractive to us that students discover the possible dangers in a hands-on activity rather than in an academic discussion.

Of course, showing students ways to disturb email communication raises ethical questions. In fact, the *Bundesverfassungsgericht* (Germany’s High Court) has discussed the question whether using a network analysing tool is in breach of German law. The court decided that the law in question only applies when such tools are used with malevolent intentions. We prefer that our students know about existing dangers and take protective measures rather than leaving this knowledge to possibly more malevolent experts who might use their lack of knowledge to infect their computer with malware. However, we took the discussion as an occasion to look for a tool which restricts the accessed network traffic to those packets that are directly addressed to the client computer, which suits our purposes sufficiently.

Table 1 shows the underlying model of competencies on three different levels. It is up to teachers, the age of the learners and the time available to decide on how deeply the insights into security provided by cryptography should be developed. It is possible to just do parts of the activities in one year and continue with a more profound examination of e.g. the RSA key generation algorithm in a following year. By pointing out these levels of competency, we enable teachers to adjust the material to their needs more flexibly.

Table 1. Competency model for the understanding of the level of security provided for communication by cryptography.

Level 1	Students describe dangers of communicating via public networks (eavesdropping, manipulation of messages, false sender's identity) and state prerequisites for a secure communication (privacy, integrity of the message, authenticity of the sender). Students use tools to create a pair of keys, exchange public keys, encrypt and digitally sign emails and verify incoming emails. Students explain on a general level how a computer verifies the integrity of a message and the authenticity of its sender.
Level 2	Students assess the level of security provided by RSA using keys of different length based on computer experiments for a reconstruction of a private key and an internet research on the RSA Challenge.
Level 3	Students create pairs of keys using the RSA key generation algorithm with small prime numbers and use them to encrypt and decrypt messages manually. Students reconstruct private keys with small prime numbers manually, explore algorithms to find large prime numbers and define the relation between the length of keys and the level of security provided using the problem to factor large semi-prime numbers as a mathematical argument. They assess the level of security provided by RSA using keys of different length based on mathematical reasoning.

In the German documentation of the learning process, links to relevant sections of the GI standards for teaching computer science are provided for each section of the learning process. Relevant competencies are mostly from the content standards *informatics systems* and *informatics, man and society* as well as the process standards *reason and evaluate*, for example:

- Students understand the basic structure and functionality of informatics systems.
- Students react appropriately to risks arising from the use of informatics systems.
- Students ask questions and state hypotheses on matters of informatics.
- Students measure different criteria and assess their adequacy for their own actions.

Relevant links to the *K–12 Computer Science Standards* published by the *Computer Science Teachers Association (CSTA)* in 2011 [5] are not pointed out but can be easily found, for example:

- Students explain the multiple levels of hardware and software that support program execution (e.g., […] networks), […] describe how the Internet facilitates global communication. [Computers and Communications Devices]
- Students exhibit legal and ethical behaviors when using information and technology and discuss the consequences of misuse, analyze the positive and negative impacts of computing on human culture. [Community, Global, and Ethical Impacts]
- Students explain the principles of security by examining encryption, cryptography, and authentication techniques [Computing Practice and Programming].

While the material is in German, we have translated selected extracts of the material into English, to enable readers to understand how the materials are intended to work. The full material as well as a didactic comment can be found at [22].

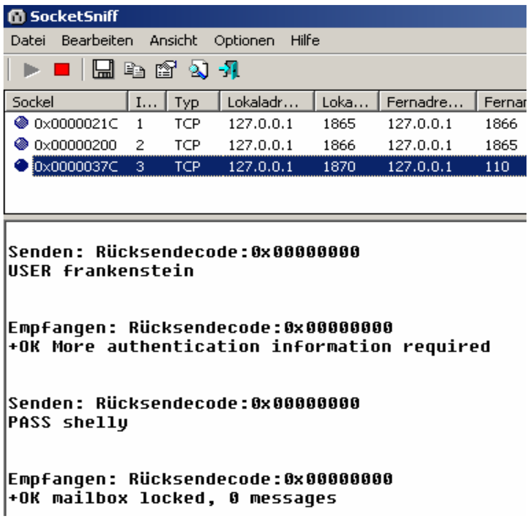
In the following sections, we describe the activities and the material developed to stimulate and enable the activities in the context-based learning process.

2.1 Discovering Email Protocols

The first question asked is: "How is an email communicated from my computer to the computer of the addressee?" To understand internetwork communication, students need to find out about the structure of interconnected computernetworks such as the Internet as well as end-to-end communication protocols between client computer and email server such as the *Simple Mail Transfer Protocol* (SMTP) for sending email and the *Post Office Protocol* (POP) for retrieving email from a mail server.

To understand the characteristics and function of communication protocols, students are first asked to communicate a single word through a door by pulling a cord that passes underneath a door. Students will soon come up with ideas for coding letters as long or short, strong or weak pulls or pulls to one of two possible sides. Inevitably from time to time, students will make mistakes in their coding or decoding and thus will agree on signs to cancel the transmission of a letter and start again from the beginning. They will also decide for a sign to show the end of a letter, e.g. a long break. Some groups agree on a sign for the receiver to signal an acknowledgement. This way, without any explicit knowledge of what a communication protocol is, they have defined such a protocol including the separation of operational and data signals. When comparing the protocols of different groups they will find different quality of service criteria such as speed and reliability.

The next step will be to have an insight into real life email network traffic. To this end, we set up a new email server in the classrooms. We use the email server software *Hamster* which needs no installation and runs from a USB flash drive, that the teacher provides during the computer science classes. Students will sign up for an individual account one after another setting a secret password, which should not be one they are already using for another account.



Sockel	I...	Typ	Lokaladr...	Loka...	Fernadre...	Fernar
0x0000021C	1	TCP	127.0.0.1	1865	127.0.0.1	1866
0x00000200	2	TCP	127.0.0.1	1866	127.0.0.1	1865
0x0000037C	3	TCP	127.0.0.1	1870	127.0.0.1	110

```
Senden: Rücksendecode:0x00000000
USER frankenstein


Empfangen: Rücksendecode:0x00000000
+OK More authentication information required


Senden: Rücksendecode:0x00000000
PASS shelly


Empfangen: Rücksendecode:0x00000000
+OK mailbox locked, 0 messages
```

Figure 1. Network traffic for authentification via POP.

After configuring their email account in the email client application *Mozilla Thunderbird*, they start sending each other emails and watch the generated email network traffic with a network traffic analyzer called *Socket Sniff* (cf. Figure 1).

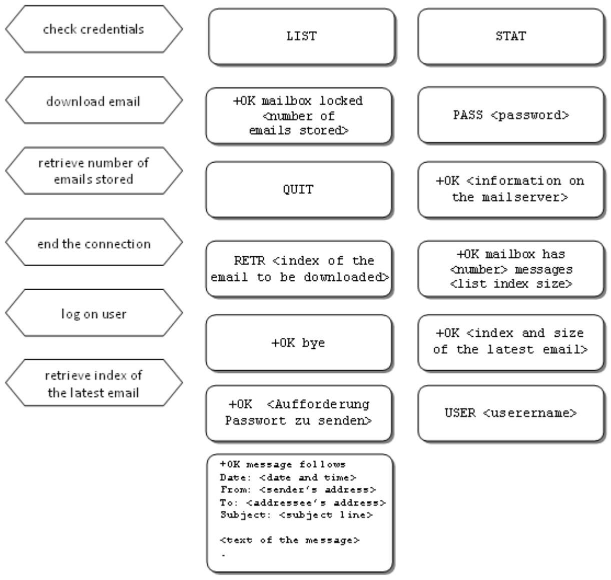

check credentials
download email
retrieve number of emails stored
end the connection
log on user
retrieve index of the latest email
LIST
+OK mailbox locked <number of emails stored>
QUIT
RETR <index of the email to be downloaded>
+OK bye
+OK <Aufforderung Passwort zu senden>
+OK message follows
Date: <date and time>
From: <sender's address>
To: <addressee's address>
Subject: <subject line>
<text of the message>
.
STAT
PASS <password>
+OK <information on the mailserver>
+OK mailbox has <number> messages <list index size>
+OK <index and size of the latest email>
USER <userername>

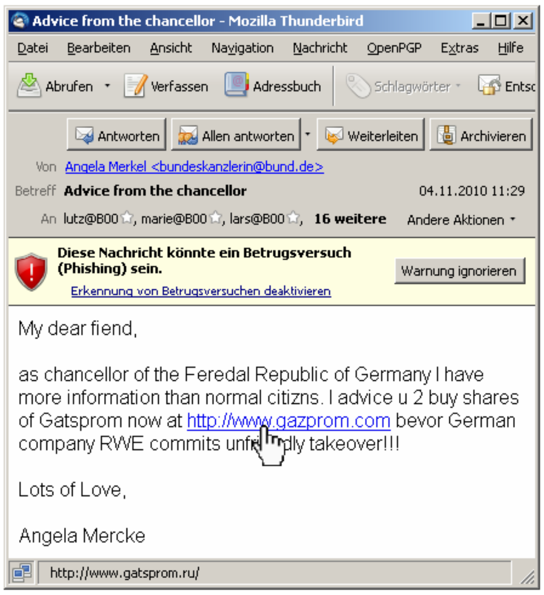
Figure 2. Pupils are to rearrange the messages in the correct order using the network traffic analyzed.

Students are asked to analyze their network traffic to reconstruct either the *Simple Mail Transfer Protocol* or the *Post Office Protocol*. Figure 2 shows a scrambled collection of generalized messages. Using the example of their concrete network connection, students will one by one identify these messages in their concrete network traffic and thus bring the general messages into the order specified in the emailing protocol in question. The left side shows terms for some more general steps, which are to be assigned to a number of messages exchanged to identify typical phases of communication such as initiating a communication channel, authorizing the communication partners, and closing the communication channel.

In a following cooperative exchange phase students will explain the protocol they have reconstructed to a student who has reconstructed the other protocol and then discuss similarities and differences of the two communication protocols.

2.2 Discovering possible Dangers

The second question structuring the learning process is: "What dangers challenge a private communication over a public network, such as the Internet?" This question is not explicitly asked in the class room. Instead, while students are still working on reconstructing the emailing protocols they experience the teacher exploiting the dangers of an unencrypted, plain-text communication over a public network in the didactic environment of the class room's local area network, the different dangers are then collected in a discussion based on the students' experiences. This is actually the reason why we use a separate mail server for the computer science classes and not the student's private email accounts. Therefore, students should be advised at the beginning to choose a password other than any passwords they already use.


Advice from the chancellor - Mozilla Thunderbird
Datei Bearbeiten Ansicht Navigation Nachricht OpenPGP Extras Hilfe
Abrufen Verfassen Adressbuch Schlagwörter Entsc
Antworten Allen antworten Weiterleiten Archivieren
Andere Aktionen
Warnung ignorieren

Von Angela Merkel <bundeskanzlerin@bund.de>
Betreff **Advice from the chancellor** 04.11.2010 11:29
An lutz@B00, marie@B00, lars@B00, **16 weitere**

Diese Nachricht könnte ein Betrugsversuch (Phishing) sein.
Erkennung von Betrugsversuchen deaktivieren

My dear fiend,

as chancellor of the Feredal Republic of Germany I have more information than normal citizns. I advice u 2 buy shares of Gatsprom now at http://www.gazprom.com bevor German company RWE commits unfriendly takeover!!!

Lots of Love,

Angela Mercke

http://www.gatsprom.ru/

Figure 3. Unsolicited mail from the German Chancellor?

First, the teacher sends an email with a false sender address. We used an email which claims, that the German Chancellor Angela Merkel suggests buying stocks of the Russian gas company Gazprom (see Figure 3). While many email providers check the correctness of a sender's address, this is not required by the SMTP protocol or the Hamster email server, The email shows several features that identify it as spam: It shows frequent spelling mistakes, shows no specific knowledge about the addressee (Be honest, who is really friend with a head of state?), shows an inappropriate choice of colloquial language and the link leads to a Url that differs from the text shown for the link. The last aspect is also the reason why modern versions of email client applications identify the mail as a possible fraud mail such as phishing mails. These features can be collected in the discussion to raise the students' awareness of spam and phishing mails.

While spam and phishing mails can be easily detected when users are aware of the features discussed above, users are more likely to react to emails that seem more personal. If we go back to Figure 1, we can see that anyone between the client computer and the mail server can read the email. With this knowledge, a malevolent communication participant could send a more individual email, in which he or she relates to something personal he or she knows from reading intercepted email. Not only can such a person retrieve personal information. The Post Office Protocol requires users to send their password in plain text. Intercepting password and username, a malevolent communication participant could access the mailbox and send mail from this mailbox as if it were his or her own. To demonstrate intercepting email on the communication path between client an mail server, the server is connected to the school network via a computer which is extended by a second network card and configured to bridge the two network cards. While this computer is technically a network bridge, it is perfectly fit to simulate an inter-network router. We have decided to provide students with a network traffic analyzer with very limited potential. It only reads the traffic assigned to a process running on the machine that the analyzer is running on. To intercept messages, teachers need to make use of a more powerful analyzer tool such as *Wireshark*.

Thirdly, a person with access to the mail server's memory could modify the content of emails before they are downloaded by the user. Here, the teacher manipulates emails by opening them from the computer's file system using a simple text editor, changes some facts such as the date of a suggested appointment and saves the changes made to the file. Students follow the manipulation on a video projector attached to the computer on which the mail server is running.

The discussion of the dangers to a private communication over a public network should lead to describing the requirements for secure communication over public networks shown in Table 2:

Table 2. Dangers to and requirements for secure communication over public networks.

danger	**requirement**
intercepting messages	privacy
manipulating messages	integrity of received messages
faking the sender's address	authenticity of senders

These three requirements structure the steps striving to find mechanisms that provide the desired security feature as described in the next section.

2.3 Achieving Privacy through Encryption

By retracing the genesis of cryptographic algorithms, students should not only learn of certain algorithms but develop an awareness of important criteria for a cryptographic algorithm and learn to question the level of security provided by a certain security feature. They should also learn that security always requires a certain amount of trust into people and technology and can never be fully guaranteed.

The desire to hide information on the communication path to prevent it from being intercepted by enemies is not at all a new requirement. The development of cryptographic algorithms dates back to ancient times. The most prominent historic algorithm is probably the *Caesar* algorithm, where Roman emperors substituted the characters of a message by other characters, retrieved by shifting an alphabet by a certain amount of letters. The key here is how far the two alphabets are shifted. This algorithm can easily be applied by children of all ages by shifting two stripes showing two alphabets each, as shown in Figure 4.

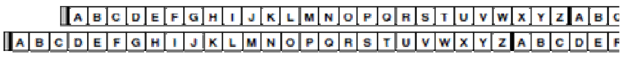
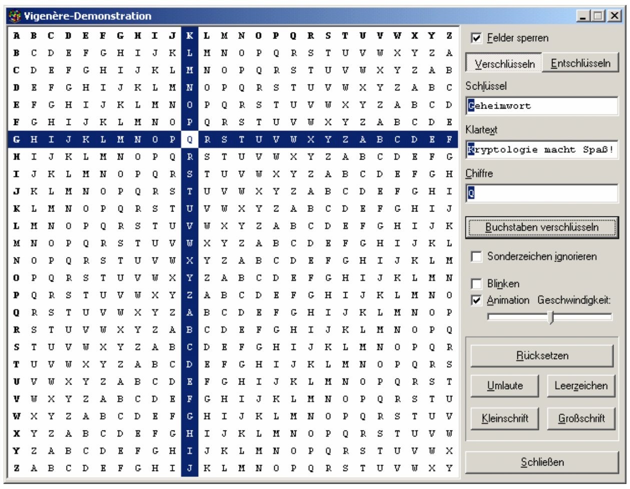

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F

Figure 4. Alphabets to apply the *Caesar* algorithm.

In order to remind students of the overall aim to achieve means for secure emailing, they are asked to exchange messages encrypted with the *Caesar* algorithm via email. They are then also challenged to crack a Caesar cipher of which the key is unknown. By providing a tool called *Krypto 1.5* by Michael Kühn to de- and encrypt *Caesar* ciphers, we enable students to quickly test the 25 possible keys for a Caesar encryption.

This way it becomes clear that despite having been in use for centuries the *Caesar* algorithm is not safe at all. One could argue that if the letters in the target alphabet were to be distributed to arbitrary positions and not in the alphabetic order, there would be 26! = 403.291.461.126.605.635.584.000.000 possible keys. While it is true, that it seems impossible to test all possible keys, longer texts can still be cracked by comparing the frequency of signs to that of the typical frequency of letters in the assumed language of the original message. This mechanism is well explained in the story "The Gold Bug" by *Edgar Allen Poe*.

A more promising way seems to work with multiple alphabets, as the *Vigenère* algorithm does, probably the best known symmetric polyalphabetic algorithm. Here several keys are employed, described by the letters of a keyword. The letter at the current position of the keyword determines how far the alphabet for a *Caesar* encryption of this letter is shifted.



```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
L M N O P Q R S T U V W X Y Z A B C D E F G H I J K
M N O P Q R S T U V W X Y Z A B C D E F G H I J K L
N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
Z A B C D E F G H I J K L M N O P Q R S T U V W X Y
```

[✓] Felder sperren
Schlüssel: Geheimwort
Klartext: Kryptologie macht Spaß!
Chiffre: Q
[ ] Sonderzeichen ignorieren
[ ] Blinken
[✓] Animation Geschwindigkeit:

Figure 5. Animation of *Vigenère* with Krypto 1.5.

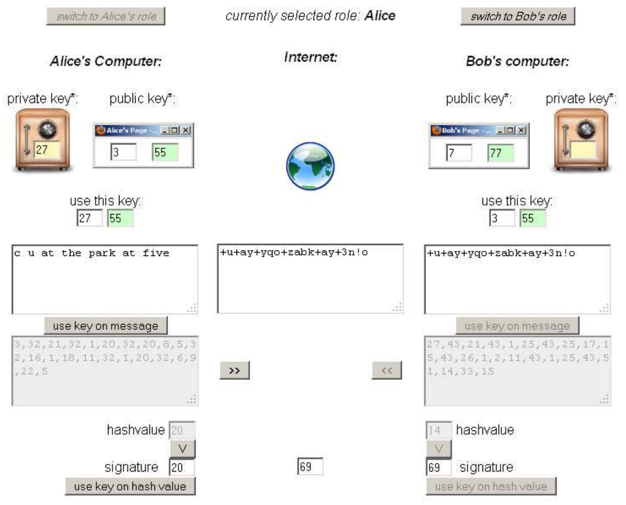
The tool *Krypto 1.5* provides an excellent animation of the algorithm. A screenshot of this animation is shown in Figure 5.

With this animation students can watch the program slowly encrypting a text.

A weak point of the *Vigenère* algorithm is that when the key is used repeatedly, typical frequent short sequences of letters such as “and”, “in”, and “but” are likely to translate to the same sequences in the cipher. The least common multiple of the positions of such parallel sequences hints to the length of the key. If the length of the key is guessed, a frequency analysis can be conducted and the key can be reconstructed. To secure 100% security, the key used must be of the same length as the message. This lead to the creation of code books called “one time pads”.

While the cipher generated from an arbitrarily generated one-time pad cannot be cracked, sender and receiver still need to hold a copy of the same one-time-pad. The exchange of secret keys can only be avoided by means of asymmetric cryptography such as the RSA algorithm. While the underlying mathematical concepts of RSA are not trivial, the idea of using pairs of keys where each communication participant holds a secret and a public key can be understood even by young learners.

The image of a padlock and a key can be used to a certain extend. We could imagine handing out a number of unlocked padlocks to communication partners, who can now lock a box by closing the lock. If we keep the key, we are the only person able to unlock the box. Of course, we cannot communicate padlocks digitally, so we need a function on numbers to do a comparable job. Here the modulo function comes into play.



Alice's computer: private key 27; public key 3, 55; use this key: 27, 55

Bob's computer: public key 7, 77; use this key: 3, 55

```
c u at the park at five
```

```
+u+ay+yqo+zabk+ay+3n!o
```

```
+u+ay+yqo+zabk+ay+3n!o
```

```
3,32,21,32,1,20,32,20,8,5,3
2,16,1,18,11,32,1,20,32,6,9
,22,5
```

```
27,43,21,43,1,25,43,25,17,1
5,43,26,1,2,11,43,1,25,43,5
1,14,33,15
```

Alice: hashvalue 20, signature 20 — Internet: 69 — Bob: hashvalue 14, signature 69

Figure 6. Animation of asymmetric cryptography.

To introduce students to the idea of asymmetric cryptography we have developed an online form, where students can exchange messages and encrypt them with public and private keys. Figure 6 shows the form in use. Very small numbers are used as keys so that students can better follow the flow of information. The form is accompanied by instructions on how to apply the keys. If they understand the instructions correctly, they will be able to successfully decrypt messages encrypted with their public key.

Once the idea of asymmetric cryptography has been established, the algorithm for generating RSA keys is presented and students manually en- and decrypt their birthday using very small keys. Students are then asked to challenge the encryption with small numbers by trying to reconstruct a possible private key for a given public key. They will find out, that using larger keys makes it more difficult to reconstruct a private key. This raises the question how large a key must be so that modern computers will not be able to crack an RSA cipher in a realistic time span. We use the tool *CrypTool* to challenge larger RSA ciphers and ask students to gather information on the *RSA Factoring Challenge*.

2.4 Verifying a digital Signature

Figure 6 shows, that from the texts a hash value is generated, that is unique for a given sequence of characters. When the sender sends the hash value as well, the receiver can detect changes made to the message, because the hash value calculated from the received text does not match the one sent by the sender. Unfortunately, a person could manipulate both the message and the hash value. This means, that the hash value has to be protected. The sender encrypts the hash value using his or her own private key. Now anyone can decrypt the signature using the sender’s public key to verify the signature. But nobody except the sender can produce a signature that can be correctly decrypted with the sender’s public key. This way, both the message’s integrity and the sender’s authenticity can be verified.

Students are often confused that now the order in which the keys are employed differs from the one used to encrypt messages to achieve privacy. Teachers are best advised to point out and discuss this difference and stress the fact, that this mechanism suits a very different purpose.

To acquire also practical competencies in using digital signatures students should first encrypt and sign emails using the animation from the project website. Then, teachers should motivate them to configure the *PGP*-plugin *Enigmail* for *Mozilla Thunderbird*, generate a set of keys, exchange public keys with their fellow students and encrypt and sign, decrypt and verify emails and signatures with *Enigmail*.

2.5 Motivation for secure Communication

In a jigsaw puzzle students first study texts to become experts for one of the following four topics:

1. Freedom of communication as a request in countries under autocratic rule such as Iran or China.
2. The Echelon project as an example of an intelligence system analyzing email traffic.
3. De-Mail as an example of attempts for commercial email cryptography with trust centres.
4. Pretty Good Privacy (PGP) as an example of royalty free cryptography with a web of trust.

Students then exchange the expert knowledge they have acquired in mixed groups.

3. LESSONS LEARNED

One question frequently discussed is: Should teachers find subject matter that suits an interesting context or should they find a context that is suitable to teach a certain subject matter? Traditionally, curricula point out fields of subject matter and assign them to certain years of learning. Teachers therefore are used to decide on the subject matter first, and then select a suitable contextualisation or application of an abstract scientific concept. More recently, curricula also point out fields of competences as well as fields of content. Teachers now have to decide on which competences they want their students to acquire, which content from the curriculum is suitable to develop these competences and which context can be used to show the concept's relevance. Within the IniK working group, we have started out designing context-based units both from a context, brainstorming on subject matter that could best be acquired, as well as from a certain subject matter, brainstorming on what context could best demonstrate the concept's relevance to students. In the case of "Email for You (only?)" we first grouped and rearranged parts of activities developed earlier and then modified and added material to bridge gaps where needed. In practise, both ways seem possible, practical and worth exploring.

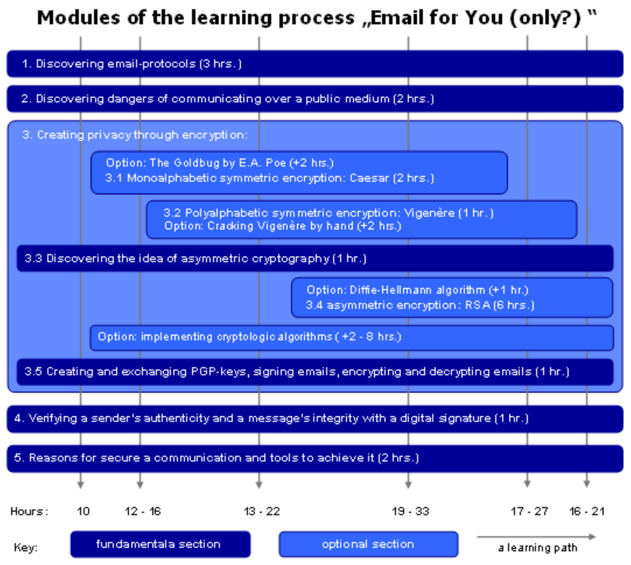

Modules of the learning process „Email for You (only?) "
1. Discovering email-protocols (3 hrs.)
2. Discovering dangers of communicating over a public medium (2 hrs.)
3. Creating privacy through encryption:
Option: The Goldbug by E.A. Poe (+2 hrs.)
3.1 Monoalphabetic symmetric encryption: Caesar (2 hrs.)
3.2 Polyalphabetic symmetric encryption: Vigenère (1 hr.)
Option: Cracking Vigenère by hand (+2 hrs.)
3.3 Discovering the idea of asymmetric cryptography (1 hr.)
Option: Diffie-Hellmann algorithm (+1 hr.)
3.4 asymmetric encryption: RSA (6 hrs.)
Option: implementing cryptologic algorithms (+2 - 8 hrs.)
3.5 Creating and exchanging PGP-keys, signing emails, encrypting and decrypting emails (1 hr.)
4. Verifying a sender's authenticity and a message's integrity with a digital signature (1 hr.)
5. Reasons for secure a communication and tools to achieve it (2 hrs.)
Hours: 10 12 - 16 13 - 22 19 - 33 17 - 27 16 - 21
Key: fundamentala section optional section a learning path

Figure 7. Fundamental and optional modules.

In the student-oriented learning process students spend a lot of time actively and individually engaging with the different topics. It is at times challenging to select just some of the many possible aspects and dimensions of a topic. Several very different units are possible for a single context. If a learning process turns out to require a lot of time, it makes sense to mark several aspects as optional and others as mandatory parts, to encourage teachers to first try out parts of the material. We have grouped the different learning activities into modules as shown in Figure 7.

While some modules are necessary for an understanding of following modules, others only lead to a more thorough understanding of aspects such as the concrete implementation of algorithms or the level of security they provide.

The proposed learning arrangement involves a lot of software. All the software proposed can be used free of royalties. While most of them are available for *Windows*, *Mac* and *Linux* operating system, some are only available for the *Windows* platform. Here, alternative applications should be found for teachers working with other platforms. For a simple start we have gathered portable versions of the software into a Zip-File, which can be extracted to a USB flash drive. Programs can then be started directly from the USB flash drive without any need for setup routines. This is extremely practical for teachers working in an environment, where software installation on clients is laborious. A disadvantage of providing the Zip-File is, that the applications are quickly outdated, especially in the case of *Mozilla Thunderbird*. This is also true for the directions given to students. Several times, these directions had to be adjusted after the dialogue for setting up an email account in *Thunderbird* was changed. (As a matter of fact, it has been improved in the sense that users are now warned of connecting to a mail server without SSL or TLS encryption. We have found a way to work around the warning so that students can still discover the dangers of plain text communication with an email server.)

While we believe we have made good use of the context for purposes of providing illustration and meaning, we must admit that with using the material as it is little is done to decontextualise newly gained knowledge by applying it to similar problems in other fields of application, e. g. designing a protocol for the communication of different parts or layers of services within a single device or discussing security issues in other situations, e. g. large permanent storages such as data bases. It is our impression that most of the other learning processes proposed at the projects website [21] lack opportunities for decontextualisation. Since the curricula only give orientation and concrete courses in computer science vary a lot between different schools and teachers, it is difficult to suggest connections to what students have learned before. Thus, it is left to teachers to realize these connections – so far no guidance is given on how to achieve decontextualisation.

Finally, designing, implementing, testing and readjusting "Email for You (only?)" has taken a lot of time. If the project is to be successful in establishing the concept in everyday CS classes, it will be necessary to acquire further resources to give teachers and university staff the time needed to develop such material. A promising approach could be to include university students and teacher trainees in the design and reflection of context-based learning processes.

4. FUTURE WORK

Further contexts should be explored to estimate their potential for teaching computer science in secondary school. The IniK project group has started a list of contexts [23] which is grouped into ideas for contexts with a potential but need for developing suitable teaching materials, ideas for contexts where it is unclear, which concrete aspects of the contexts are useful to the teaching of computer science in secondary school education, and ideas for contexts which at a first glance seem attractive but after a more thorough examination don't seem useful to the teaching of computer science in secondary school education. It would be helpful to have a more detailed understanding as to what potential different contexts have. A suggested learning process will always discuss only a selection of possible aspects, so eventually we could have different units exploring the same contexts with different focus. More teaching materials for further topics should be developed. Today, there is e. g. no unit that motivates the design and use of databases – visualization using open data seems an interesting approach here.

But next to the development of further material, experiences made in the design and implementation of context-based computer science courses should be collected, shared and discussed in order to develop guidelines and advice for the development of context-based learning processes in secondary school computer science education. An example would be concepts for a sustainable decontextualisation of concrete knowledge to abstract knowledge which can be recontextualised to be applied in new and different situations. Today, examples and guidance for de- and recontextualisation are still missing.

5. CONCLUSION

"Email four You (only?)" shows that a context-based learning of computer science is possible and helps students understand the relevance of knowledge they acquire in computer science classes.

We have learnt that we can both look for contexts where a certain topic is of relevance as well as screen an interesting context as to which topics of computer science education are relevant for it. Either way, it is a long way of selecting suitable aspects out of a wide choice of possible aspects and developing material which makes sense at a certain point of the learning process. If in doubt, one should decide on whether aspects are necessary for the following sections of the learning process. If this is not the case, they can be marked as optional. Whenever software is involved, the software should be easy to use, if possible without installation.

We have by now presented our proposed learning arrangement in several hands-on workshops for computer science teachers throughout Germany and also in the journal LOG IN [16], which is widely distributed amongst computer science teachers throughout Germany. Since then, we have received a lot of positive feedback from colleagues stating that their students are highly motivated to understand both the underlying technology of email as well as the different cryptographic algorithms. While we are happy to see our own impressions affirmed, we are indeed aware that this is only a first but nevertheless promising hint as to the potential of context-based learning of computer science in secondary school education. Reaching a stage where we now have several learning processes readily prepared and tested in class, it now seems important to focus on theory-building to get a deeper insight into teaching and learning CS in context based environments. This theory building should be based on a thorough research approach with a high alignment to the specific preconditions of the IniK-project, e.g. a Design Based Research process as proposed in [3].

6. REFERENCES

[1] Arbeitskreis »Bildungsstandards« der Gesellschaft für Informatik, 2008. Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I. Empfehlungen der Gesellschaft für Informatik e. V. vom 24. Januar 2008. In *LOG IN* vol. 150/151 (2008), supplementary. Retrieved October 10, 2012: http://informatikstandards.de

[2] Altrichter, H., Posch, P. and Somekh, B. 1993. *Teachers Investigate their Work: Introduction to the Method of Action Research*. Routledge, London, New York.

[3] Barab, S. and Squire, K. 2004. Introduction: Design-Based Research: Putting a Stake in the Ground. *The Journal of learning sciences*. 13,1 (2004), 1-14. DOI= 10.1207/s15327809jls1301_1

[4] Brinda, T., Puhlmann, H., and Schulte, C., 2009. Bridging ICT and CS – Educational Standards for Computer Science in Lower Secondary Education. In *ITiCSE '09: Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education*, 288-292.

[5] Computer Science Teachers Association (CSTA) 2011. K-12 Computer Science Standards. Revised 2011. Retrieved October 10, 2012: http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf

[6] Cooper, S. and Cunningham, S. 2010. Teaching computer science in context. *ACM Inroads* 1 (March 2010), 5 - 8. DOI= http://doi.acm.org/10.1145/1721933.1721934.

[7] Coy, W. 2005. Informatik … im Großen und Ganzen. In *LOG IN 136 (2005)*, 17-23.

[8] Diethelm, I., Borowski, C., and Weber, T. 2010. Identifying relevant CS contexts using the miracle question. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (Koli Calling 10). ACM, New York, NY, USA, 74 - 75. DOI= http://dx.doi.org/10.1145/1930464.1930477

[9] Diethelm, I. and Dörge, C. 2010. From Context to Competencies. In N. Reynolds and M. Turcsányi-Szabó, eds. *Key Competencies in the Knowledge Society*, IFIP Advances in Information and Communication Technology. Springer Boston, USA, 67–77.

[10] Diethelm, I., Hildebrandt, C., and Krekeler, L. 2009. Implementation of Computer Science in Context - a research perspective regarding teacher-training. In A. Pears and C. Schulte, eds. *Koli Calling 2009. 9th International Conference on Computing Education Research*, Uppsala University, Uppsala, Sweden, 96-99.

[11] Diethelm, I., Koubek, J. and Witten, H. 2011. Informatik im Kontext (IniK) – Entwicklungen, Merkmale und Perspektiven. In Thomas, M. ed. *Praxisberichte zur 14. GI-Fachtagung Informatik und Schule (INFOS 2011)*. Münster, Germany, 97-105.

[12] Engbring, D. 2005. Informatik im Kontext. In *LOG IN*, vol. 136 (2005), 28-33.

[13] Esslinger, B., Gramm, A., Hornung, M. and Witten, H. 2011. Asymmetrische Kryptographie für die Sek I - RSA (fast) ohne Mathematik? In Thomas, M. ed. *Praxisbeiträge zur INFOS 2011 - 14. GI-Fachtagung Informatik und Schule* (Münster, Germany, 2011, September 12-15). Gesellschaft für Informatik, Bonn, Germany, 225-235.

[14] Esslinger, B., Gramm, A., Hornung, M. and Witten, H. 2012. Kann man RSA vertrauen? In *LOG IN* vol. 172/173 (2012), 79-92.

[15] Gramm, A. 2012. Animation of Asymmetric Cryptography. Retrieved October 10, 2012: http://it-lehren.de/asymcrypt

[16] Gramm, A., Hornung, M. and Witten, H. 2011. E-Mail (nur?) für Dich - Eine Unterrichtsreihe des Projekts Informatik im Kontext. In *LOG IN* vol. 169/170 (2011), supplementary.

[17] Guizdal, M. 2010. Does contextualized computing education help? *ACM Inroads* 1, 4, 4 - 6, DOI= http://dx.doi.org/10.1145/1869746.1869747.

[18] mpfs – Medienpädagogischer Forschungsverbund Südwest (ed.), 2011. JIM-Studie 2011 – Jugend, Information, (Multi-)Media. Basisuntersuchung zum Medienumgang 12- bis 19-Jähriger. Stuttgart: Medienpädagogischer Forschungsverbund Südwest, 32. Retrieved October 10, 2012. http://www.mpfs.de/fileadmin/JIM-pdf11/JIM2011.pdf

[19] Parchmann, I., Gräsel, C., Baer, A., Demuth, R. and Ralle, B. 2007. Chemie im Kontext - a symbiotic implementation of a context-based teaching and learning approach. In *International Journal of Science Education 28, 09 (2007)* 1041-1062. DOI= http://doi.acm.org/10.1080/09500690600702512

[20] Klieme, E. et al. 2004. *The Development of National Educational Standards. An Expertise*. Bundesministerium für Bildung und Forschung, Berlin.

[21] Koubek, J. 2012. Website of the “Informatik im Kontext” project. Retrieved October 10, 2012: http://informatik-im-kontext.de

[22] Koubek, J. 2012. Website of the “Informatik im Kontext” project – section “Entwürfe » Email (nur?) für Dich”. Retrieved October 10, 2012: http://informatik-im-kontext.de/index.php/entwuerfe/email-nur-fuer-dich/

[23] Koubek, J. 2012. Website of the “Informatik im Kontext“ project – section “Kontexte und Ideen”. Retrieved October 10, 2012: http://informatik-im-kontext.de/index.php/kontextideen/

[24] Koubek, J.; Schulte, C.; Schulze, P. and Witten, H. 2009. Informatik im Kontext (IniK) – Ein integratives Unterrichtskonzept für den Informatikunterricht. In Koerber, B. ed. *Proceedings of INFOS 2009 – 13. GI-Fachtagung Informatik und Schule* (Berlin, Germany, September 21–24, 2009). Gesellschaft für Informatik, Bonn, Germany, 268–279.

[25] Saeli, M. and Schulte, S. 2013. Applying Standards to Computer Science Education. In Kadijevic, D., Angeli, C. and Schulte, C. eds. *Improving Computer Science Education*. Routledge, London, New York, in press.