Teaching Public-Key Cryptography in School*

Lucia Keller, Dennis Komm, Giovanni Serafini,
Andreas Sprock, and Björn Steffen

Department of Computer Science, ETH Zurich, Switzerland
{lucia.keller,dennis.komm,giovanni.serafini,andreas.sprock,
bjoern.steffen}@inf.ethz.ch

Abstract. These days, *public-key cryptography* is indispensable to ensure both confidentiality and authenticity in numerous applications which comprise securely communicating via mobile phone or email or digitally signing documents.

For all public-key systems, such as RSA, mathematically challenging and technically involved methods are employed which are often above the level of secondary school students as they employ deep results from algebra. Following an approach suggested in 2003 by Tim Bell et al. in *Computers and Education, volume 40, number 3*, we deal with the question of how to teach young students the main concepts, issues, and solutions of public-key systems without being forced to also teach rather complicated theorems of number theory beforehand.

1 Introduction

Cryptography is one among the topics in computer science that raise the interest of most students of all ages. Naturally, it allows the teacher to create interesting lessons including experiments and games. Complementing the well-known classical cryptosystems, such as Caesar and Skytale [1], which are rather easy to understand, in this paper we want to focus on communicating the ideas of mechanisms that are actually used today. Some of the most widely used and popular cryptosystems (e. g., RSA, El-Gamal, . . .) are asymmetric methods based on the idea of *one-way functions with trap-doors*. For many students it is fascinating that there are functions in which computation in one way is easy and the computation of the inverse function is much harder. At this point, it is the challenge of didactics to introduce the key issues behind this concept in a fashion as simple as possible. To explain the intuitive idea, there is a famous example given by Salomaa [14] using a phone book. Since nowadays this example is not really applicable, we use a more sophisticated model for our lectures. We took the idea from "CS unplugged" [2] and attempted to duplicate the positive results concerning the progress of students [3] similar to the work of Nishida et al. in Japan [12]. In contrast to their work, we only focused on one specific topic from "CS unplugged".

To describe our concept in greater detail, we first need to give a quick overview of the most important facts about public-key cryptography. The need for

* This work was partially supported by the Hasler Foundation, project "Kantonales Fortbildungszentrum Informatikunterricht", and by FILEP grant 351 of ETH Zurich.

J. Hromkovič, R. Královič, and J. Vahrenhold (Eds.): ISSEP 2010, LNCS 5941, pp. 112–123, 2010.

asymmetric cryptosystems arises from the following problem: when using symmetric-key cryptosystems, at least two parties *share* a secret (i. e., a key) which they use to encrypt and decrypt messages from plaintext to ciphertext and the other way around. In the real world, however, a serious problem arises when trying to share the key, because clearly, in general, the two entities (in this scenario, say two persons Alice and Bob) do not have a secure channel to communicate yet. Many strategies have been proposed to solve this problem, among which one of the most famous certainly is the Diffie-Hellman key exchange protocol [8]. However, on the downside, using this protocol still requires the creation of approximately n^2 keys if n parties want to securely communicate with each other. In 1978, one of the first and (up to today) most widely used public-key cryptosystem, namely RSA, was introduced [13]. Here, it suffices to create n key pairs for a scenario as above. For more details, we refer to the standard literature [1, 7].

The main idea behind public-key (or asymmetric) cryptosystems is the following: one entity has (in contrast to symmetric cryptosystems) a pair of keys which are called the *private* key and the *public* key. These two parts of the key pair are always related in some mathematical sense. As for using them, the owner of such a key pair may publish her public key, but it is crucial that she keeps the private key only for herself. Let (sk, pk) be such a key pair where sk is Alice's secret private key and pk is the corresponding public key. If a second person Bob wants to securely send Alice a message M, he computes $C = encrypt(M, pk)$ where *encrypt* denotes the so-called *encryption function* which is also publicly known (see Fig. 1). This function is a one-way function with a trap-door. In other words, the trap-door allows for the creation of the secret key sk which in turn enables Alice to easily invert the encryption function. We call C the ciphertext. Obtaining M from C can be done easily using the (publicly known) decryption function *decrypt* and Alice's private key (sk). On the other hand, it is much harder to decrypt without having any knowledge of the private key. As already mentioned, the great advantage of this approach is that no secure key exchange is necessary before a message is transmitted.

Since public-key cryptography is commonly used in many applications and settings, it is very appropriate and promising to teach cryptography. However, since the low level ideas and calculations require a lot of mathematical background, it is very hard to discuss cryptosystems like RSA with students of secondary schools.

We therefore use a very simple and straightforward theoretical cryptosystem from "CS unplugged" by Tim Bell et al. [2] that only involves some very easy

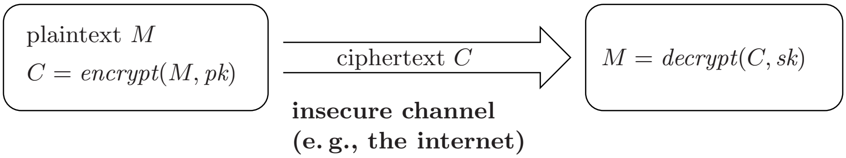


Fig. 1. Scheme of using public-key cryptography

mathematical ideas, but highlights the important principles of public-key cryptography very nicely.

The paper is organized as follows. In Section 2, we describe the cryptosystem used and explain its theoretical background. After that, we explain how to use this approach to teach students the ideas of public-key cryptography and also describe our experiences in Section 3 and 4. Section 5 discusses a concrete lesson held in July of 2009 at a local secondary school in Switzerland. We conclude with a reflection of the main ideas and open questions in Section 6.

2 Technical Details for the Teachers

The cryptosystem used is based on a graph-theoretical problem which we describe in the following. First, we need some basic notations and definitions.

For $n \in \mathbb{N} \setminus \{0\}$, let $V = \{v_1, v_2, \ldots, v_n\}$ denote a set of *vertices* and $E \subseteq \{\{v_i, v_j\} \mid i, j \in \{1, 2, \ldots, n\}, i \neq j\}$ a set of *edges*. We then call $G = (V, E)$ a *graph* with n vertices and $|E|$ edges. Furthermore, if for two vertices v_i and v_j we have $\{v_i, v_j\} \in E$, we call v_i and v_j *adjacent* (i. e., connected by an edge). For $i \in \{1, 2, \ldots, n\}$, let $\mathit{weight}\colon V \to \mathbb{Z}$, $v_i \mapsto w_i$ be a vertex *weight function* where the integer w_i is called the weight of vertex v_i. We call the pair (G, weight) a weighted graph. For our investigations we need the following definition.

Definition 1 (Dominating set). *Let $G = (V, E)$ be a graph. A* dominating set *– DS for short – is a set of vertices $V_{\text{DS}} \subseteq V$ such that, for every vertex $v' \in V \setminus V_{\text{DS}}$, there exists a vertex $v \in V_{\text{DS}}$ with $\{v', v\} \in E$.*

A special variant of a DS has the additional property that, for every vertex from $V \setminus V_{\text{DS}}$, there exists exactly one *vertex in V_{DS} such that $\{v', v\} \in E$ and that no two vertices from V_{DS} are adjacent. We call this special case an* exact dominating set *– EDS for short – and the corresponding set V_{EDS}.*

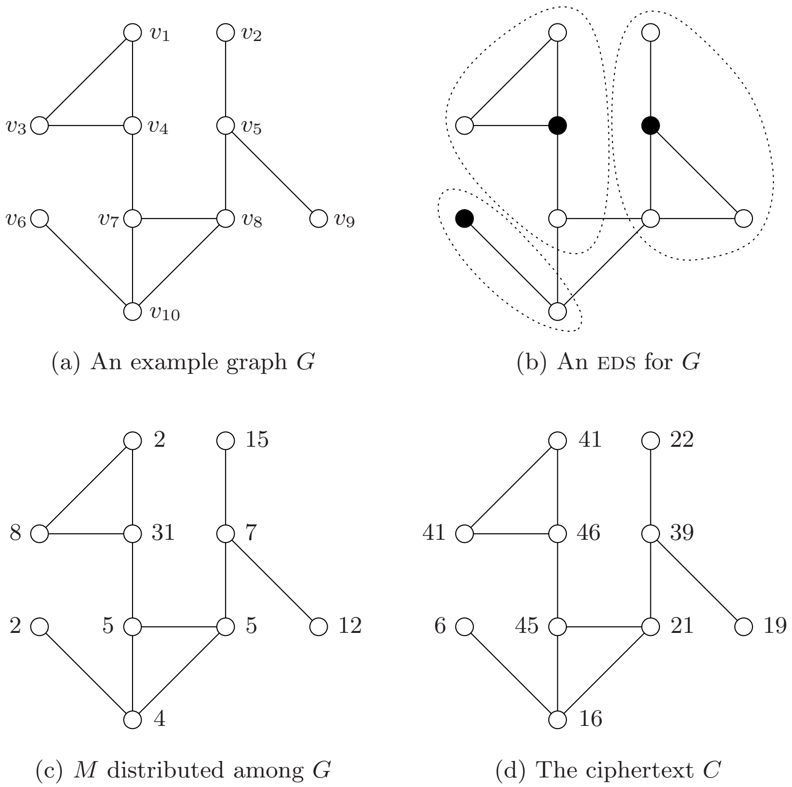
In the following, we construct a graph with an EDS. The graph itself is used as the public key and the set V_{EDS} as the secret key. Note that determining whether a graph has an EDS is an $\mathcal{NP}$-complete problem [6]. Thus, finding an EDS in a graph which is known to have an EDS is $\mathcal{NP}$-hard, too. Again, consider two persons Alice and Bob and assume that Alice wants to securely send a message M to Bob. Bob therefore needs to create a key pair $(\mathit{sk}, \mathit{pk})$ to enable them to encrypt and decrypt messages and therefore he takes the following steps.

a.1 Choose two arbitrary natural numbers n_{EDS} and n_{dom} and let $V_{\text{EDS}} = \{v_1, v_2, \ldots, v_{n_{\text{EDS}}}\}$ and $V_{\text{dom}} = \{v'_1, v'_2, \ldots, v'_{n_{\text{dom}}}\}$ be two pairwise distinct sets of vertices.

a.2 Set $E = \emptyset$. Then, for *every* vertex $v' \in V_{\text{dom}}$, choose *exactly* one arbitrary vertex $v \in V_{\text{EDS}}$ and set $E = E \cup \{\{v, v'\}\}$, i. e., connect every vertex from V_{dom} to exactly one vertex from V_{EDS}.

a.3 Choose an arbitrary number of pairs (v'_i, v'_j) of vertices from V_{dom} and set $E = E \cup \{\{v'_i, v'_j\}\}$, i. e., connect v'_i to v'_j.

a.4 Set $V = V_{\text{EDS}} \cup V_{\text{dom}}$ and $G = (V, E)$.



(a) An example graph G

(b) An EDS for G

(c) M distributed among G

(d) The ciphertext C

Fig. 2. An example of a graph G for encrypting the plaintext $M = 91 = 2 + 15 + 8 + 31 + 7 + 2 + 5 + 5 + 12 + 4$

As already mentioned, the graph $G = (V, E)$ is Bob's public key *pk* (as shown in Fig. 2 (a)). Obviously, the set V_{EDS} forms an EDS which is immediately clear by the way the edges of G are constructed in a.2 (as shown in Fig. 2 (b)). Bob keeps V_{EDS} as the private key *sk* for himself and publishes G.

Now suppose that Alice wants to encrypt the plaintext message M using the public key of Bob. Let $M \in \mathbb{N} \setminus \{0\}$ be a natural number.[1] She acts as follows.

b.1 Write M as the sum $\sum_{i=1}^{n} w_i$ where w_i are integers which are randomly chosen (except the last one).

b.2 Then define a weight function $plain(v_i) = w_i$ for every $v_i \in V$. The "plain" graph $(G, plain)$ is shown in Fig. 2 (c).

b.3 For every vertex $v \in V$, let $\text{neighbor}(v) = \{v' \mid \{v, v'\} \in E\} \cup \{v\}$ be the set of all neighbors of v in G, i. e., all vertices which are adjacent to v and v itself. We then define the function $ciph : V \to \mathbb{Z}$ as

[1] Clearly, we may represent any text by natural numbers using, e. g., ASCII.

$$\mathit{ciph}(v) = \sum_{v' \in \mathrm{neighbor}(v)} \mathit{plain}(v').$$

The weight function *ciph* is the ciphertext of M (as shown in Fig. 2 (d)). Obviously, *encrypt* is the algorithm described by b.1 to b.3. The plaintext M, on the other hand, can be easily calculated if the function *plain* is known. However, Bob may simply use *sk* to receive the plaintext with

$$\mathit{decrypt}(C, \mathit{sk}) = \mathit{decrypt}(\mathit{ciph}, V_{\mathrm{EDS}}) = \sum_{v \in V_{\mathrm{EDS}}} \mathit{ciph}(v).$$

Since V_{EDS} forms an EDS in V, it holds that

$$\sum_{v \in V_{\mathrm{EDS}}} \mathit{ciph}(v) = \sum_{v \in V} \mathit{plain}(v) = M.$$

An example for this cryptosystem and the plaintext $M = 91$ is shown in Fig. 2 for a graph with 10 vertices.

However, if *sk* is unknown, it is still possible to calculate the weights of all vertices (i. e., the function *plain*) by solving a system of n linear equations which can be done in $\mathcal{O}(n^3)$ which clearly is extensively more time[2] than the computation of M takes if V_{EDS} is known.[3]

More particular, for a given ciphertext, the message $M = \sum_{v \in V} \mathit{plain}(v)$ can be decrypted by solving a system of linear equations given for all $v \in V$:

$$\sum_{v' \in \mathrm{neighbor}(v)} \mathit{plain}(v') = \mathit{ciph}(v)$$

For the example shown in Fig. 2 (d), we obtain the following system of linear equations.

$$\begin{aligned}
\mathit{plain}(v_1) + \mathit{plain}(v_3) + \mathit{plain}(v_4) &= \mathit{ciph}(v_1) = 41\\
\mathit{plain}(v_2) + \mathit{plain}(v_5) &= \mathit{ciph}(v_2) = 22\\
\mathit{plain}(v_1) + \mathit{plain}(v_3) + \mathit{plain}(v_4) &= \mathit{ciph}(v_3) = 41\\
\vdots \qquad &\quad \vdots \qquad = \vdots\\
\mathit{plain}(v_6) + \mathit{plain}(v_7) + \mathit{plain}(v_8) + \mathit{plain}(v_{10}) &= \mathit{ciph}(v_{10}) = 16
\end{aligned}$$

Please note that such a system might have one or infinitely many solutions, but the sum is still unique. A reasonable definition of a secure cryptosystem for teaching purposes is:

[2] Note that the fastest known algorithm can solve a system of n linear equations theoretically in $\mathcal{O}(n^{2.367})$ [5].

[3] That means that we still need to solve an $\mathcal{NP}$-hard problem to find *sk*, but for a given public key and ciphertext (G, ciph), the plaintext M can be computed in polynomial time.

> A cryptosystem is secure if there does not exist any efficient algorithm that decrypts the cipher text without knowing the secret key used, but knowing the way in which the cryptosystem works. [11]

This implies that the cryptosystem introduced is not *secure* in this sense. However, it is still very adequate to explain asymmetric-key cryptography since we can easily show our students the big discrepancy concerning the effort between decrypting with and without knowing *sk*.

3 How We Teach Public-Key Cryptography

In what follows, we give an exemplary way of how to teach students the ideas and concepts of public-key cryptography using the cryptosystem introduced in Section 2.

Usually the students had some previous classes, where we taught them the basic concepts of classical (symmetric) cryptography.

3.1 Teaching Goals

We define the following teaching goals – they describe the minimal knowledge the students should achieve after completing the teaching unit:

1. Students understand why in *public-key cryptography* two different keys are needed for encryption and decryption. One is public, the second one must be kept secret and is only known to the owner (is private).
2. Students are able to explain that everyone is able to encrypt the plaintext with the public key, but only the owner of the related private key has the means to decrypt the ciphertext.
3. Students are able to depict the decryption process as an easy task when a secret information is known and as a practically infeasible one if this information is not available.
4. Students are able to encrypt and decrypt messages correctly with the presented cryptosystem.

3.2 Introducing the Concept of Public-Key Cryptography

First, we motivate the need for public-key cryptography by showing that the key management overhead of symmetric cryptography is high. We point out the weaknesses of symmetric cryptosystems and propose the concept of having two keys (a private and a public one) which clearly overcomes the disadvantages of symmetric cryptosystems. We informally introduce one-way functions with trapdoors and then ask the students whether they are familiar with anything like it in everyday life.

As a first candidate of such a function, we describe the public-key cryptosystem that uses a phone book [14]. Given a phone book (the public key), we can encrypt a letter by choosing at random a name starting with this letter and then send the corresponding phone number instead. Clearly it is easy to find a name starting with a given letter. On the other hand, it is hard to find the name which belongs to a given phone number. But if the recipient has a special phone book (the private key) which is ordered according to phone numbers, then she is able to efficiently decrypt the message.

3.3 Teaching the Graph-Theoretical Cryptosystem

To give a more qualified cryptosystem we now explain the students the graph-theoretical public-key cryptosystem discussed in Section 2.

To do this, we first informally introduce them to graphs. For the cryptosystem it is only important that the students know the notions of *vertices*, *edges* and the concept of a *neighborhood*. Note that we do not describe what an exact dominating set is, yet.

We now show the students an example graph G with n vertices that has an EDS which is, of course, only known to us. This graph represents a public key and we explain how to encrypt an integer (message) with this graph (as shown in Fig. 2):

Step 1. We draw the graph G on the blackboard to encrypt the message M.

Step 2. We write M as a sum of n integer summands.

Step 3. We write each of these numbers next to exactly one of the n vertices (using some color, e. g., green) and ask the students if this is already a secure ciphertext.

Step 4. We then show the students how to obtain the ciphertext by adding up the numbers in the neighborhood of a vertex and writing the sum down in some other color (e. g., red) next to the corresponding vertex. After calculating all red numbers, we clean out the green ones and say that the graph G with the red numbers is our ciphertext.

Asking the same question as above, the students realize that it is now not easy anymore to derive the plaintext given only the red numbers.

After all students know how to encrypt a message using this cryptosystem we may carry out a little contest within the class. We form teams of two students each. Every team is given two copies of a graph different from G on two pieces of paper and they have to encrypt a message which they have chosen for themselves. Now, the students can write the summands of the plaintext on one graph, each number next to the according vertex. After calculating the numbers for the ciphertext, they then write these numbers on the second piece.

When all teams are finished, we collect their ciphertexts and give each team the same ciphertext that we have created beforehand. Their task is to find the plaintext of the given ciphertext.

In the meantime, we demonstrate how easy the recipient can decrypt the message. We do this by decrypting the messages of the teams very fast and show them that we really found the corresponding plaintext. The students are usually very impressed how fast we can do that, as they have not been able to decrypt the given ciphertext, yet.

After that, EDS are introduced on the board (usually, the students have gotten an idea of how the secret key might work during the competition).

The next step is to demonstrate that it is a hard task to find an exact dominating set in a graph. Achieving this is rather easy. We simply hand out a drawing of a big graph ($n \geq 30$) and ask the students to find its EDS.

Afterwards, we need to show that, on the other hand, it is an easy task to design a graph which has an EDS which is only known to the creator of the graph. Therefore, we draw a set of vertices to the board and mark them (again, we might use some special color) as *dominating vertices.* After that, we draw another set of *dominated vertices* in a different color. We then draw exactly one edge from each dominated vertex to one dominating vertex. Finally, we add various edges between the dominated vertices. It is obvious that the dominating vertices now really form an EDS for this graph. After this is also clear to the students, we color all vertices with the same color to “cover up our tracks”.

What should follow is a discussion on how difficult it is to decrypt the plaintext with and without knowing the EDS. At this point, the students themselves probably discovered the alternative way to decrypt the message (i. e., a system of linear equations as described in Section 2). However, if not, we make them familiar with this idea and write down such a system for the graph on the board. Our students are then asked to solve it themselves. They will immediately realize that this way is a lot harder than using the EDS.

The investigation of the running time of algorithms for solving a system of linear equations may be as detailed as the student’s knowledge allows it to be. In any case it is very important to show and to compare the two function graphs so that the difference between decrypting with and without the secret becomes clear, i. e., the linear function $f(n) = n$ for the decrypting knowing the EDS (we simply need to do less than n additions) and the function $g(n) = n^3$ for decrypting via solving a system of linear equations.

4 General Experiences

We have been able to gather a multitude of experiences by applying our teaching concept at various Swiss secondary schools which we describe in the following.

4.1 Introducing the Concept of Public-Key Cryptography

The students will certainly agree that the method in the example of the phone book makes sense, but they are not very convinced that this method works in practice, since, nowadays, with the help of digital phone books, anyone is able to find the name to a given phone number efficiently. We also have to exclude the option of calling the number and asking for the name.

This leads to the observation that this cryptosystem is rather artificial. But none the less, the example is very qualified to introduce the basic concepts of public-key cryptography and it gives a first idea of the difference between decrypting with knowing the secret key and decrypting without knowing it.

4.2 Teaching the Graph-Theoretical Cryptosystem

This example is a lot more serious and realistic and it has the big advantage that we do not have to restrict the students in the same drastic way as for the first candidate.

Of course, most students immediately realize that the message in step 3 is not encrypted at all, since one can easily add up the green numbers. But this question prepares the students for the following step and keeps their attention on the distribution of the numbers.

The students enjoy the competition and are highly motivated to find the secret of the decryption. In many classes that we taught, at least some groups were successful, only a few needed hints.[4] They also get the feeling that this cryptosystem is more "serious" than using phone books, because now they see that it is not immediately clear how to find the plaintext.

At this point, some students discovered that they can get M using a system of linear equations. In this case, we simply told them that this still takes very long and there is a way which is a lot faster. We then postponed a detailed discussion of the running time of both methods to a point when the second technique (i. e., using the EDS) is also discovered.

5 An Example Lesson at a Secondary School

In this section, we informally describe an implementation of the presented method at a secondary school in Zurich at the beginning of July 2009. Classes other than languages and literature are taken either in German or in Italian [9].

The Swiss secondary school system is extremely heterogeneous. A common Swiss-wide frame of regulations [4] is implemented in many slightly different ways by each of the 26 cantons.

The secondary school involved focuses on preparing students for academic studies in the field of arts. Since computer science was no official subject at the time of our lesson, the students did not have any prior knowledge in algorithmics and programming. More specifically, in contrast to many other classes which, at this point, already knew basic things like classical symmetric ciphers, this was the students' first lesson on cryptography.

One of the authors has a teaching appointment for mathematics at this secondary school and tried the presented method in one of his classes. Mathematics lessons during the first two secondary school years are given in small classes consisting of up to 12 students. Students at the end of the second year are able to deal with basic algebra and Euclidean geometry, linear and quadratic equations, trigonometry as well as systems of linear equations.

The chosen class consists of students who are between 16 and 18 years old. The lesson was carried out as a game during the last lesson of the school year just after the students received their grades.

[4] It suffices to say that only a linear number of operations are necessary, e. g., "Five operations are enough".

5.1 Experimental Setting

Although we did not implement a formal empirical study, some typical methodical constraints were fulfilled during our test lecture: we decided to assess the available specific prior knowledge in cryptography (especially in public-key cryptography), then to carry out the lecture, and to assess the students knowledge again. We want to focus on measuring the students' proficiency in the basic principles of public-key cryptography after completing the teaching unit described.

The assessments were carried out anonymously. Each student received a personal tag. We gave the lecture in a compact (no longer than 45 minutes), concentrated and unstressed way.

5.2 Overview of the Lesson

The lesson was designed as simple as possible. We chose examples and metaphors based on the everyday life of students inside and outside school. The teaching goals correspond to the ones we described in Section 3.1.

Pre- and Post-Test. Both the pre- and the post-test were conducted based on the same form containing the following 5 questions:

1. The concept of *symmetric cryptography* means ...
   - □ that in order to encrypt and subsequently decrypt a message, two different pieces of information ("keys") are necessary.
   - □ that in order to encrypt and subsequently decrypt a message, only one (secret) information is necessary.
   - □ I don't know the concept of *symmetric cryptography*.
2. The concept of *asymmetric cryptography* means ...
   - □ that in order to encrypt and subsequently decrypt a message, two different pieces of information ("keys") are necessary.
   - □ that in order to encrypt and subsequently decrypt a message, only one (secret) information is necessary.
   - □ I don't know the concept of *asymmetric cryptography*.
3. In *asymmetric cryptography* ...
   - □ everyone is able to encrypt a message using the so-called public key.
   - □ everyone is able to encrypt a message using the so-called private key.
   - □ I don't know the concept of *asymmetric cryptography*.
4. In *asymmetric cryptography* ...
   - □ everyone is able to decrypt an encrypted message rapidly using the so-called public key.
   - □ only the authorized recipient is able to decrypt an encrypted message rapidly using the so-called private key.
   - □ I don't know the concept of *asymmetric cryptography*.

5. In *asymmetric cryptography* ...

    □ the decryption of an encrypted message is a very short process, if we know a secret information, otherwise it is a very long one. Without this secret information it is practically impossible to decrypt the encrypted text.
    □ the decryption of an encrypted message is a very long process, independent of whether you know a secret information or not. For everyone, it is practically infeasible to decrypt the encrypted message.
    □ the decryption of an encrypted message is always a very short process, independent of whether you know a secret information or not. Everyone is able to decrypt the encrypted message without any effort.
    □ I don't know the concept of *asymmetric cryptography.*

Protocol of the Lesson. We started the lesson with a short talk about security in the internet and mentioning the need for security when, e. g., buying books online. The need for confidentiality while communicating was illustrated on the basis of the situation in which two students wish to exchange encrypted messages. Symmetric and asymmetric cryptography issues and terms were introduced on the basis of the lock metaphor (where the public key is a padlock and the corresponding private key is a key that opens it).

In the following phase, we introduced the cryptosystem illustrated in the prior sections. The students were really impressed by the topics and by the way how messages are encrypted or decrypted. The challenging task to find out the trick, permitting to decrypt a ciphertext rapidly (as the teacher did) was really motivating for the class. Here, no one suggested a solution based on solving a system of linear equations. One student depicted an approach similar to the expected solution (i. e., using the EDS), without being able to explain the details of her strategy.

The fast decryption method was shown to the class and explained carefully. After that, the post-test was done showing that almost all students reached the teaching goals and were able to talk about the high-level ideas of public-key cryptography as intended.

6 Conclusion

In this paper, we discussed a simple way of introducing the basic principles of *public-key cryptography* to students of secondary schools. A more detailed discussion of the cryptosystem will be part of an upcoming book [10]. We gave concrete suggestions how to use the theoretical cryptosystem designed by Bell et al. [3] to teach students of young age public-key cryptography in an entertaining way. Additionally, we discussed our experiences while teaching these ideas. Finally, we highlighted a specific lesson at a school in Switzerland. The lesson we described was not planned as a formal experiment. However, it was possible to verify that, after completing the unit, the majority of the students mastered the goals we set.

It therefore remains open to formalize this experiment and test this method on a larger set of students. However, the results of this small class match our general experiences when teaching cryptography.

References

1. Bauer, F.L.: Decrypted Secrets: Methods and Maxims of Cryptology, 4th edn. Springer, Secaucus (2006)
2. Bell, T., Fellows, M., Witten, I.H.: Computer Science Unplugged - Off-line activities and games for all ages (1999), `www.csunplugged.org` (last accessed: October 22, 2009)
3. Bell, T., Thimbleby, H., Fellows, M., Witten, I., Koblitz, N., Powell, M.: Explaining cryptographic systems. Computers & Education 40(3), 199–215 (2003)
4. Bundesrat and EDK. Verordnung des Bundesrates/Reglement der EDK über die Anerkennung von gymnasialen Maturitätsausweisen (MAR) (1995), `http://www.sbf.admin.ch/evamar/reglemente/VO_MAR_1995_d.pdf` (last accessed: October 22, 2009)
5. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. In: Proc. of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC 1987), pp. 1–6. ACM, New York (1987)
6. Cull, P.: Perfect codes on graphs. In: Proc. of the 1997 International Symposium on Information Theory, p. 452 (1997)
7. Delfs, H., Knebl, H.: Introduction to Cryptography: Principles and Applications. Springer, Heidelberg (2002)
8. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory IT-22(6), 644–654 (1976)
9. Elmiger, D.: Die zweisprachige Maturität in der Schweiz (2008), `www.sbf.admin.ch/htm/dokumentation/publikationen/bildung/bilingue_matur_de.pdf` (last accessed: October 22, 2009)
10. Freiermuth, K., Hromkovič, J., Keller, L., Steffen, B.: Kryptologie, Lehrbuch Informatik. Vieweg+Teubner (to appear, 2009)
11. Hromkovič, J.: Algorithmic Adventures. Springer, Berlin (2009)
12. Nishida, T., Idosaka, Y., Hofuku, Y., Kanemune, S., Kuno, Y.: New methodology of information education with “computer science unplugged”. In: Mittermeir, R.T., Sysło, M.M. (eds.) ISSEP 2008. LNCS, vol. 5090, pp. 241–252. Springer, Heidelberg (2008)
13. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
14. Salomaa, A.: Public-Key Cryptography. Springer, Berlin (1996)