CONTRIBUTION OF CRYPTOGRAPHY TO MATHEMATICS TEACHING

A way to illustrate mathematics as a living science

Katharina KLEMBALSKI
Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin
klembals@math.hu-berlin.de

ABSTRACT

This paper examines the question to what extent cryptography and its history can be used to promote the idea of mathematics as a living science in mathematics education. Our investigations are substantiated in the context of the RSA cryptosystem and its mathematical background. Cryptography proves to be a suitable example for pointing out the development of the mathematical sciences triggered by development outside of mathematics – in this case the invention and usage of computers. The nature of problems solved and still open can broaden the students' view of mathematics as a living science that is still developing. The given considerations may be used as a framework for a teaching module about RSA as well as a subsequent occasion to reflect upon modern cryptography (and mathematics) for those who already know RSA.

1 Introduction

A project aimed to bring current mathematics into schools forms the background of the considerations presented in this paper. The teaching experiment corresponding to the project is an additional course in cryptography, students choose as part of their Abitur[1]. The course is a two-semester course consisting of weekly lessons a 3 x 45 min lessons, i.e., 135 minutes each week. The course was tested in two high-schools (gymnasium) in Berlin with 12 and 14 participants, respectively. The high-school students were about 18 years old and had no prior knowledge of cryptography, of number theory, or of congruences in particular. The teaching unit outlined in this paper is extracted and redesigned referring to the realisation and reflection of the original teaching experiment. It comprises 10 up to 12 units a 90 minutes and can be taught independently, e.g. as workshop.

Large parts of the German standard curriculum (e.g. [11]) do not extend beyond the scientific knowledge of the 19$^{\text{th}}$ century. Modern elements of mathematics are often limited to probability theory and some elective parts, that are not mandatory for all students. Teaching cryptography can contribute to this issue in different ways. Still "... mathematical education must follow, at least to some degree, what happens in mathematical research", as Lovász reasons in his article "Trends in mathematics: How they could change education?"[8]. He uses especially the term *algorithmic* mathematics (opposite to *structural* mathematics) to characterise the development in many branches of mathematics and applications through the use of computers: "it enriches several classical branches of mathematics with new insight, new kinds of problems, and new approaches to solve these". Mathematical activities in class that illustrate this, hence

[1] general German qualification for university entrance

give an insight into (the development of) modern mathematics in general. Thus, in this paper RSA is analysed in order to identify such possible ideas, techniques or algorithms.

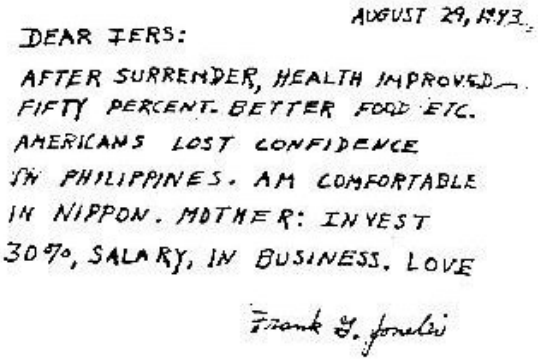
As an example for modern cryptography, RSA requires only little mathematical background (predominantly elementary number theory) in order to be understood. That does not only refer to the cryptosystem itself, but to the process of the corresponding *development* as well. In contrast to the often found image of mathematics, the development of modern cryptography demonstrates that mathematics has no fixed structure or just needs some completion in order to give some helpful applications to other sciences. It is an example of how the development in mathematics is very often triggered by questions posed by other sciences, technological advances, or even social developments.[2]

In order to shift the focus on to the process of development as well as the RSA-algorithm itself, the following outlined teaching unit concentrates on the question *Why did it take more than 2000 years to invent RSA?* First RSA is motivated by a historical description of a classical problem of cryptography. Second we introduce RSA and answer the posed question. Third the influence of computers on the development of RSA is shown in more detail. Finally, aspects that may be helpful to broaden the students' view of mathematics are highlighted.[3]

2 Historical and practical background – Motivation

Cryptography has been used for several thousand years [1]. Some well known ciphers used in different centuries are named in fig. 2. They can be found in most popular and didactic literature about cryptography or cryptography in the classroom. However, all ciphers up to the 1970's are connected by one significant problem: the key exchange problem.

An illustration of the problem is a letter of an American soldier imprisoned by the Japanese sent to his family during the Second World War (fig. 1). The written text hides additional information that can be found by reading only the first two words in each line. But there was no way to transmit this information – the key – separately and securely without the knowledge of the Japanese. In this case the hidden message was puzzled out.


AUGUST 29, 1943

DEAR FERS:

AFTER SURRENDER, HEALTH IMPROVED—
FIFTY PERCENT. BETTER FOOD ETC.
AMERICANS LOST CONFIDENCE
IN PHILIPPINES. AM COMFORTABLE
IN NIPPON. MOTHER: INVEST
30%, SALARY, IN BUSINESS. LOVE

Frank J. Jones

Figure 1: Postcard of an American soldier [7]

Obviously, for every day use of cryptography, this is not an option. Therefore until 30 years ago, information was encrypted solely according to the following principle. A message M is encoded by an invertible

[2] An example is physics and its influence on the developments of calculus (mechanics) in the 18^{th} century, which also appears in teaching practise. Other examples are functional analysis (quantum mechanics) or differential geometry (general theory of relativity).

[3] In the opening plenary of ESU-6 Jankvist presented an empirical study about the use of "history as a goal" in mathematics teaching. One of his teaching units is about RSA. Therefore some of the results are quite related [5].

parametric function E into a cipher text $C = E_K(M)$. The parameter K is called the key and needs to be kept secret. The recipient decodes / decrypts the message with the inverse function D so that $D_K(C) = M$. The parameter K for the construction of E and D must be transmitted via a secure channel (personally in advance, by a messenger, etc.).

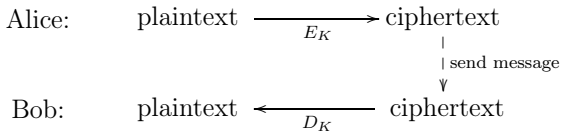


Figure 2: Data exchange with classic cryptography

Secure data exchange between strangers over an insecure communication channel is not possible this way. To overcome this problem, a cryptographic revolution was needed. With the development of the Internet this became of paramount importance as in principle all information sent over the Internet can be intercepted or monitored.

The revolution in cryptography that overcame the key exchange problem started in 1976. Diffie and Hellman published the idea of an algorithm including a pair of different keys instead of one, K_e for encryption and K_d for decryption [3]. From the knowledge of one of the keys, one cannot derive the other. Therefore, K_e can be transmitted via a public channel (hence the name: public key algorithm). An algorithm that implements this idea was published by Rivest, Shamir, and Adleman in 1978 [10], and is known as the RSA cryptosystem.[4] This leads to the following leading question for the teaching unit:

3 Why did it take more than 2000 years to invent RSA?

The development of public key cryptography can be seen as long term problem solving. To underline the similarities to problem solving in the classroom the answer is divided into three parts. We need to know, what is so revolutionary about Diffie's and Hellman's idea and later on how it was implemented. This leads to the principle of RSA, but covers less than 90% of the answer (considering the time interval of more than 2000 years). Practical aspects will complete the answer and illustrate how external driving forces influence development in mathematics.

It is less trivial than it seems to start thinking about a pair of keys instead of one key. For a function with the key parameter K it is

$$D_K(E_K(M) = M.$$

But how to find a suitable function, that offers different keys for encryption and description

$$D_{K_d}(E_{K_e}(M) = M?$$

Suitable means especially hard to invert even if one of the keys is known. To illustrate the difficulty, students were given the following exercise.

> *Exercise: Find candidates of suitable functions. Create a list of different kinds of functions and their inverses. Which of them fulfil the criteria of being difficult to invert and fast to calculate?*

[4]Independently from Rivest, Shamir, and Adleman the algorithm was found but not published for military reasons by Clifford Cocks in 1973. In contrast to the three authors above Cocks only needed one day to find a suitable algorithm, when he got to know the idea of the public key algorithm, James Ellis wrote down in 1969 (and Diffie and Hellman 1976) [12]. This fact illustrates that even in mathematical science problem solving is a very individual and not predictable process.

Some well-known functions and their inverses are subsequently listed.

$$\begin{array}{ll} f(x) = x + k & f^{-1}(x) = x + (-k) \\ f(x) = k \cdot x & f^{-1}(x) = \frac{1}{k} \cdot x \\ f(x) = x^k & f^{-1}(x) = x^{\frac{1}{k}} \\ f(x) = k^x & f^{-1}(x) = \log_k x \end{array}$$

Usually students define f over real numbers (or maximum possible intervals in $\mathbb{R}$). If students do so, they soon recognise that none of the known functions are useful candidates for cryptographic requirements.[5]

Additionally, the function needs to be easy to calculate and without rounding errors. Therefore, more complicated functions are not an option (e.g. including trigonometric functions). But what happens if you change the domain?

Comparing known possible domains will soon lead up to natural numbers or integers. Problems of calculability guide to residues modulo n. Thus, instead of determining functions within the domain of rational or real numbers, students investigated the same functions of residues modulo n. Residues behave with respect to addition and multiplication like integers, but differently concerning division and exponentiation. The second attribute gives the candidate for the function we are looking for. Compared to working with natural numbers it is quite different to extract the inverse of a residue which has been raised to a higher power, as the following example shows.[6]

Example: $19^7 \bmod 55 = 24$ and $24^{\frac{1}{7}} \bmod 55 \neq 19$, however $24^{23} \bmod 55 = 19$. As the example shows the inverse exponent 23 is not obvious from knowing the exponent 19. Background of finding exponents to invert exponentiation is *Euler's theorem*: For a and n relatively prime and φ the Euler function[7] it is

$$a^{\varphi(n)} \equiv 1 \bmod n.$$

By multiplication with a follows

$$a^{\varphi(n)+1} \equiv a \bmod n$$

and

$$a^{k\varphi(n)+1} \equiv a \bmod n.$$

for integers k. The last congruence can be used to find the pair of keys K_e and K_d we are looking for. If there are numbers e and d that give the exponent $k\varphi(n) + 1$ those are the pair of keys.[8] Because of

$$a \equiv a^{k\varphi(n)+1} \equiv a^{ed} \equiv (a^e)^d \bmod n.$$

Now e, n can be published, d remains secret. So everyone, e.g. Alice (sender) is able to encrypt a message a using the public key (e, n):

$$a^e \bmod n = c.$$

[5] Additionally the concept of functions is supplemented explicitly.

[6] A detailed teaching unit that follows the differences of operations between $\mathbb{N}$ and $\mathbb{Z}/n\mathbb{Z}$ to identify possible functions useful for cryptography is described in [9].

[7] $\varphi(n) = \#\{0 < t < n \mid \gcd(t, n) = 1\}$.

[8] Obviously, e (and d) is relatively prime to $\varphi(n)$. Therefore $\gcd(e, \varphi(n)) = 1$ and there exist $u, v \in \mathbb{Z}$ such that $e \cdot u + \varphi(n) \cdot v = 1$ (Bézout's Theorem). The integer u gives the factor d. Considering the example above it is $7 \cdot 23 \bmod \varphi(n) = 1$.

The cipher text c is decrypted by Bob (recipient) by using the private key (d, n):

$$c^d \bmod n = a.$$

Why is this secure? Why can't anybody calculate d with the knowledge of n and e? In practice e is chosen randomly with $\gcd(e, \varphi(n)) = 1$. The number d is calculated by using the extended Euclidean algorithm with the input values of $\varphi(n)$ and e.

That is possible for everyone, but only if $\varphi(n)$ is known.[9] To calculate the value of $\varphi(n) = n \prod_{p|n} (1 - 1/p)$ the factorisation is needed. That means, to make RSA secure the modulus n is chosen to be hard to factor. For RSA this is achieved by choosing n to be a product of two large primes.[10] The resulting procedure for the RSA key generation, encryption, and decryption is given in the box below together with an illustrating example.

Procedure of RSA	**Example**
Key generation	
Multiply large primes p and q	$p = 5, q = 11$
$n = p \cdot q$	$n = 55$
Compute $\varphi(n) = (p-1)(q-1)$	$n = 40$
Find integers[11] $e, d : ed \equiv 1 \bmod \varphi(n)$	$e = 7, d = 23$
Encryption of the message $a = 19$	
$c = a^e \bmod n$	$19^7 \bmod n = 24$
Decryption of the message c	
$a = c^d \bmod n$	$a = 24^{23} \bmod 55$

Figure 3: RSA – Key generation, encryption and decryption and illustrating example

Up to this point the mathematical background includes the Euclidean algorithm, modular arithmetic, and Euler's theorem (see fig. 4). Regarding the origin of the mathematical background the initial question can be changed to *Why did it take another 200 years to invent RSA?*[12]

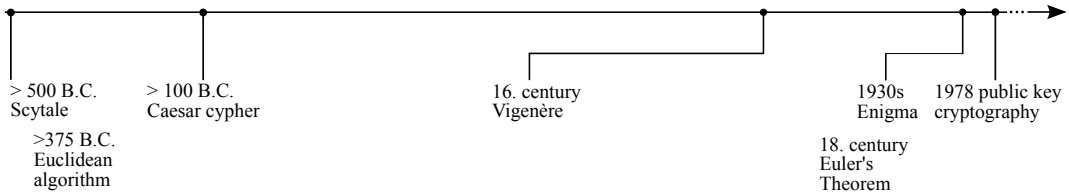


Figure 4: Timeline – History of cryptography and mathematical background of RSA

[9] A proof that the knowledge of the prime factorisation of n is equivalent to the knowledge of the private key can be found in ([2], p. 141).

[10] Obviously the factorisation is easy to calculate, e.g. by dividing n by all primes up to $\sqrt{n}$. Therefore *hard to factor* refers not to the existence but the practical way to find the factorisation, because of the number of steps and calculation time of the algorithm, which is executed by computers.

[11] While introducing RSA the Euclidean algorithm is not necessary. For small primes like this trial and error works too.

[12] Depending on the curriculum Fermat's little theorem can also be considered. For $n = p \cdot q$, p and q primes, the proof of RSA can be reduced to Fermat's little theorem. The elementary proof of Fermat's little theorem is more accessible to work with in class than Euler's theorem. The question to be posed would then be *Why did it take another 350 years to invent RSA*?

The answer can be found rather in the practical realisation of RSA than the formal description above.

The advances in technology and availability of computers were the crucial factors for the development of RSA for two reasons. First, the predominant historic use of cryptography had been the exchange of military secrets between *two parties*. In contrast, the use of computers increased, in particular, the exchange of sensitive data over *multi-party* communication networks, especially the Internet. To overcome the key exchange problem became a public concern, not only a military one.[13] Secondly, Euler's theorem, or the Euclidean algorithm were already valuable instruments in mathematics. But the use of computers made them practically applicable.
Thus, computers provided both the need for RSA as well as the means.

Similar developments can be found in different parts of mathematics. Computational mathematics, in particular numerical methods and discrete mathematics, has increased in importance in recent decades. A major contribution of computers consists in shifting problems of computability to the development and implementation of suitable (and in particular efficient) algorithms. Algorithm design has always been a classical activity in mathematics, but computers increased their visibility and respectability substantially [8]. On the other hand, computers introduced new problems, such as the need for data compression (information theory) and more philosophical questions, e.g. the nature of mathematical proofs [4].

4 Technology in Mathematics

Students use lots of algorithms in maths, e.g. multiplication algorithms and Gaussian elimination. Even procedures like identifing extreme values can be interpreted as an algorithm. Instruments like graphic calculators, spradsheed software, and CAS play an important role in school, as to visualise and perform individual calculations. Only a few algorithms are used to recognise the special values or chances (and limits) offered for mathematics and mathematics education by informatics. Therefore in the following we focus especially on these values, chances and limits when dealing with the algorithms that are integral part in the realisation of RSA.

The algorithms cover very old mathematics (Euclidean algorithms) as well as modern ideas (probabilistic algorithms). Developing and analysing the RSA related algorithms utilises classic problem solving strategies as well as computer oriented activities like analysis of running time and space. The old mathematics, its implementation, and practical limits especially of factorisation give students exemplary insight into mathematical development that has still not come to an end.

As an initial exercise students build their own example of RSA using primes as large as possible.[14] Depending on the used computer algebra system and knowledge about

[13] Because of more involved scientists this includes not only more chances to solve cryptographic problems but also more chances to verify the security of published ciphers as well. Today cryptography includes lots of applications for everyone's purpose, e.g. authentication of communication participants, the verification of the integrity of transmitted data and more – hidden in process of onlinebanking, software updates, mobiles etc.

[14] The term *large* depends on the CAS available. E.g the Voyage2000 only knows the *Mod(a,b)* command to calculate $a \bmod n$ but no *PowerMod(a,b,c)* to calculate $a^b \bmod c$ stepwise (accordingly to the square and multiply algorithm). Because of the number of digits for internal calculation the limits of calculability were found for primes with 3 digits and exponents higher than 200.

the corresponding/respective CAS commands problems arise by ...

(i) ... finding e and d (extended Euclidean algorithm),

(ii) ... rising a to the power of e and more often
... rising c to the power of d (square & multiply algorithm),

(iii) ... Factorisation of n to break RSA and

(iv) ... very interestingly: how to find primes (Miller Rabin primality test).

It is advisable to introduce the corresponding CAS commands later to motivate the algorithm development. But even if commands are known, the comparison between calculation by CAS and "by hand" leads to questions about the nature of the (black box) command (difference of calculation by hand and computer? limits of exact calculations?).

In the following it is outlined how the algorithms were introduced and which aspects were given priority.[15]

(i) Euclidean algorithm

This very old algorithm can be developed by the students and may be useful to introduce pseudo code.[16] Students have to describe the Euclidean algorithm in a general way and to prove it afterwards.

> *Exercise*
> *How can we compute the* gcd *of natural numbers a and b without knowing their factorisations?*
> *If $a > b$, then:*
> $$\gcd(a, b) = \gcd(a, a - b). \qquad (*)$$
>
> *a. Use this fact to compute* $\gcd(322; 98)$ *in steps.*
> *b. Give a recipe for doing this in general.*
> *c. Prove (*).*

In the next step students get the pseudo code description and have to connect their written text results to corresponding parts in pseudo code. This illustrates the strong connection between textual solutions and programming solutions and calls attention to missing or mistakable parts (example, see fig. 5).

[15] According to the students previous knowledge and not to cover up the mathematical essence the algorithms are described in pseudo code only. As a consequence we don't distinguish between the mathematical object algorithm and its implementation as a computer program, which depends on the machine and/or on the programming language [8].

[16] Pseudo code is a compact and informal description of a computer programming algorithm, augmented with natural language descriptions of the details, where convenient, or with compact mathematical notation. It is easier for students to understand pseudo code than conventional programming language code and omits details that are not essential for the understanding of the algorithm. Pseudo code is commonly used to describe algorithms independently from individual programming language.

"The small number is subtracted from the greater one, thus $a - b$, this gives a natural number c. This step is repeated. So, it is calculated $c - b$ or $b - c$. This is repeated until one of the numbers is zero. The number different from zero is the gcd. *When a equals b it doesn't mind which one is subtracted from the other one."*

```
if a = 0
   return b
while b > 0
   if a > b
      a := a - b
   else
      b := b - a
return a
```

Figure 5: Textual description of the Euclidean algorithm (Natalie, 15 years) and pseudocodedescription

(ii) Square & multiply algorithm

The exercise for the square & multiply algorithm can be made the other way around. After testing alternatives to simple multiplication by using power laws, students can be given the pseudo code description of the square and multiply algorithm.

Exercise

a. Compute $4^{19} =: a^m$ using the given algorithm.

b. Try to find own examples that need less multiplications than the algorithm below.

```
Power := 1
while m > 0
   if m uneven
      Power := Power a
      m := m-1
   else
      a := aa
      m := m/2
return: Power
```

In doing so the students attention is forced to the different concepts of variables used in maths and computer sciences.

(iii) Factorisation of n

The issue of security of RSA is useful to confront the students with limitations of algorithms. The initial question is the following.

Exercise

Decipher the cipher text $c = 74273$, public key $n = 77057$, $e = 211$.

Obvious approach is to factorise the modulus n and follow the steps of the standard key generation using p and q (fig. 3). To study the limit of this approach, students can experiment with (their own) moduli like
$n = 942876191136657658379477430933277830167219$ and
$n = 213598703592091008239502318434121854383503406330042754837222169102460 \cdots$
$997011045373604390158396064 79$.

Using MATHEMATICA it took 4 seconds to factorise the first modulus. The factorisation of the second one started during the lesson and was interrupted one week later (and would have needed more than 100 additional years on the same computer). After testing further examples the found exponential connection between factorisation time

and the number of digits may indicate that it is not just a problem related to the computer equipment used.[17] It shows that even if there is an algorithm available (in this case to factorise natural numbers) it is not certain that it produces results efficiently.

A trickier didactic problem is to convince students of the security of an algorithm that depends on the non-existence of a practicable solution to a problem (in this case the factorisation of integers), especially if there is no proof of this non-existence.

There are other methods than simply dividing all integers from 2 up to square root n (e.g. Quadratic sieve ([2], chapter 10). But none of them are fast enough to be a satisfying alternative and to question the security of RSA.To illustrate this and to compare it with the process of simple division, the software CryptTool[18] was used.

But not to know a suitable algorithm is no proof that such an algorithm does not exist. Thus the students are confronted with an open question: is there an efficient way to find prime factors of large numbers? Similarly, is it possible to decipher RSA directly without factorisation?

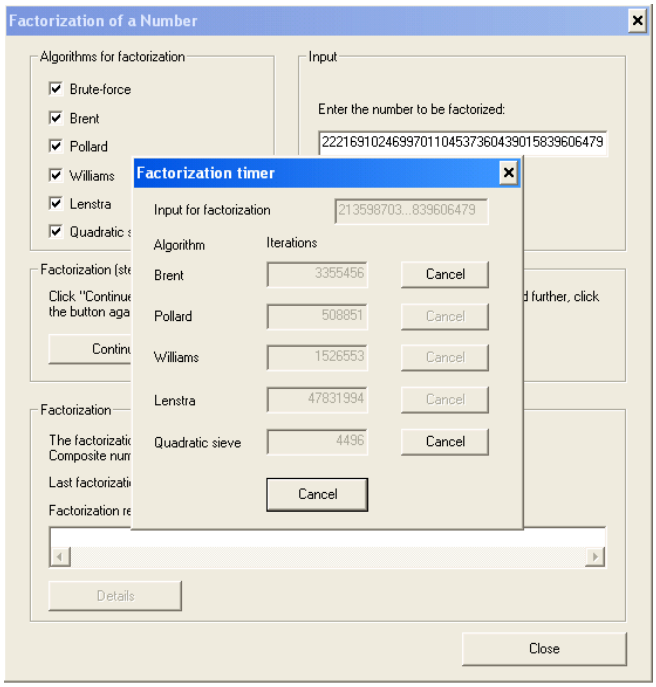


Figure 6: Comparing different algorithms to factorise large numbers using CrypTool.

(iv) Miller Rabin primality test

At the first glance the factorisation problem seems very similar to the problem of finding primes. But the latter can be solved very quickly in practice. This is at the cost of absolute certainty, because the algorithm of Miller Rabin is a probabilistic one. That is an interesting and rare connection between number theory and probability [6].
On the basis of the given literature the principle of the primality test was presented and complemented by exercises by a group of students. This was done in the context of six possible specialisations subsequent to the introduction of RSA.

[17] The key length of the modulus n used today is 1024 or 2048 bits. Numbers this large are secure against factorisation. Conceivable development in computer speed for the next 10-20 yearsThat is considered there. Therefore, its no option just to use a "better" computer (CAS instead of a calculator, 100 CAS instead of one CAS, super computer instead ...). Computers worthy of this task have not been constructed yet. Improvement that questions RSA (and other modern ciphers beyond extension of the key length where appropriate) is foreseen when the quantum computer is available. This is not yet in sight.
An impressive example of the different meaning of the term *large* when speaking about factorising large numbers can be found in [13].

[18] Free software to demonstrate and analyse classic and modern ciphers. Available in English, German, Polish, Spanish and Serbian.

5 Conclusion

The answer to *Why did it take more than 2000 years to invent RSA?* as presented above adds to knowledge and concepts from early school years including especially the key mathematical concept of numbers, functions and algorithm. Also it introduces actual development in mathematics. This actual development often gives contrast to the students' image of mathematics. It applies especially to the factorisation of natural numbers n, as the factorisation of n ...

- ... starts to be a problem, if n is large. In contrast to the simple problem solved in primary school using prime factors.[19]
- ... supports a sparsely used approach in students' school experience. Now the *lack of knowledge* can be utilised to solve a 2000 year old problem.
- ... is an integral part of an algorithm of which billions of people benefit every day without any proof of its security.

Or more from a student's point of view: Even the "well known" natural numbers are of scientific interest and far from being completely understood.[20] Even more amazing: the inability to solve a problem is useful.

The last aspect illustrates how mathematics develop much in the same way as natural sciences do, and that mathematical development does not take place in an enclosed environment but in mutual interaction with the surrounding world. In this case it was the practical problem of key exchange from outside of mathematics and the new perspective through the use of computers that made already valuable instruments (like the Euclidean algorithm or Euler's theorem) practically applicable and accessible. On the other hand, the present algorithm is inherently questionable, because its security is substantially based on practical reasons (length of the modulus n and some parameters of p and q to avoid possible attacks) and not on a mathematical proof. Therefore, teaching cryptography provides various links to possibly widen the students' view of mathematics and present mathematics as a living science. Summed up: mathematical history is an ongoing one, taking place today just like yesterday.

REFERENCES

1. Bauer, F. L., 1997, *Entzifferte Geheimnisse : Methoden und Maximen der Kryptologie*, Berlin: Springer.
2. Buchmann, J., 2007, *Einführung in die Kryptografie*, Berlin: Springer.
3. Diffie, W. and Hellman, M., 1976, "New Directions in Cryptography", Trans. IEEE Inform. Theory, IT-22, 6, 644-654.
4. Hales, T. C., 2008, "Formal Proof", Notices of the AMS 55, 11, 1382-1393

[19] Similar problems to RSA and the factorisation arise for other ciphers that solve the key exchange problem in connection with the discrete logarithm (e.g. Diffie Hellman key agreement [2]). There exists no way to calculate the discrete logarithm.

[20] Open questions to underline this may arise in the context of the distribution of primes, which are constitutional for a RSA and other ciphers too. (Is there an infinite number of prime twins? Is there always a prime between n^2 and $(n+1)^2$?

5. Jankvist, U. T., 2009, "An implementation of two historical modules: outcome and perspectives", this issue.
6. Klembalski, K., 2010, "Primzahltests als innermathematische Vernetzung von Zahlentheorie und Wahrscheinlichkeitsrechnung", in Beiträge zum Mathematikunterricht, Franzbecker.
7. Kippenhahn, R., 1999, *Verschlüsselte Botschaften*, rororo.
8. Lovász, L., 2007, Trends in Mathematics: How they could Change Education, Lisbon, pdf: http://www.cs.elte.hu/ lovasz/popular.html.
9. Puhlmann, H., 1998, "Kryptographie verstehen – Ein schülergerechter Zugang zum RSA-Verfahren", TU Darmstadt, pdf: http://www3.mathematik.tu-darmstadt.de/fb/mathe/bibliothek/preprints.html.
10. Rivest, R.L., Shamir, A. and Adleman, L., 1978, "A method for obtaining digital signatures and public-key cryptosystems", Comm. A.C.M. 2, 120-126.
11. Senatsverwaltung für Jugend, Bildung und Sport, 2007, *Rahmenlehrplan für die gymnasiale Oberstufe Mathematik*, Berlin.
12. Singh, S., 1999, *The code book: the secret history of codes and codebreaking*, Forth Estate.
13. Wiesenbauer, J., 1986, "Was sind und was sollen große Primzahlen?", ÖMG-Didaktikreihe, 27, 196-206.