# The mathematics of public key cryptography

Bartosz Kuprasz

February 19, 2015

| | |
|---|---|
| Candidate name: | Bartosz Kuprasz |
| Candidate session number: | |
| School name: | Lessing-Gymnasium |
| School number: | 003089 |
| Session number: | 003089-0014 |

# Contents

# 1 Introduction

Modern societies heavily rely on information flow, but not all information should be publicly accessible. Valuable data, such as credit card numbers, passwords and private messages, is transferred online and thus may be vulnerable to interception. Therefore information needs to be secured. Ciphers and codes are able to conceal the content of messages, hence reducing the risk of an information leak.
Surprisingly, cryptosystems are based on mathematical concepts, including algorithms and number theory. The mathematical structures underlying the operations carried out in order to code data are responsible for their security.
But how secure are presently used systems compared to previous ones? Are they truly incorruptible?
In this exploration, the development of cryptography throughout history will be presented chronologically, using selected ciphers as examples. Every technique will be tested upon advantages and weak points. Furthermore, it will be shown how the difficulties in the extraction of the content of an enciphered message arise from the mathematics used.

I personally feel drawn to this topic, because it demonstrates how abstract mathematics like number theory are practically applied in real life in a scale which affects everybody on a daily basis. Moreover, the secrecy aspect acts very alluring. Another aspect is my curiosity regarding earning money with mathematics. Security systems are told to be very valuable, so as someone who plans a career in a mathematics-related field I would love to learn what makes people pay for your mathematics.

# 2 Preceding cryptosystems

## 2.1 The Caesar cipher[1]

One of the earliest cases of cipher use is the Caesar cipher, named after Julius Caesar, who used it mainly to encrypt relevant military messages. The mechanism behind is straightforward:
The alphabet is transformed into a number series, so that
$a = 0, b = 1, c = 2$, etc. The message is transcripted letter by letter: A chosen integer $k$, which is the key, is added to the number of the letter one

[1] http://en.wikipedia.org/wiki/Caesar_cipher, 16.02.2015

wishes to encipher, then the letter which is represented by the sum is written down.
The consequence is a shift in the alphabet by the number $k$, which is why this technique also is called the Caesar shift. Shifts of more than 25 return to the beginning of the alphabet and continue from there on. The operation for $k = 0$ does not have an enciphering effect in the case of the Caesar cipher.
The receiver of the message just needs to subtract the number $k$ from the numbers of the letters in the message and write down the resulting letters.
Mathematically, the process of both encryption and decryption can be represented with help of the modulo operator. In the example, $x$ is the number of the letter we en- or decipher and is defined as $0 \leq x \leq 25$ and $k$ is the known key.

Encryption:

$$E_k(x) = (x + k) \mod 26$$

Decryption:

$$D_k(x) = (x - k) \mod 26$$

The common alphabet is substituted by only one modified alphabet (the modification is the shift by $k$), due to this the Caesar cipher is a monoalphabetic cipher.

An advantage of this method is that it does not require immensely complex calculations to translate the content, which reduces both the number of mistakes in the final version and the time it takes to transform the content when knowing the key.
Nonetheless, the disadvantages prevail. The key needs to be known by both parties prior to transmission and has to be delivered safely. Without a safe channel, which is not present until the key is established, the key can easily be seized by unauthorized individuals. This condition is crucial, since a reader accesses the information effortlessly with the key.
But the main reason why this cipher cannot solidly secure the information is the fact that there are only 25 possible shifts, all in the range from 0 to 25. Hence even without the key the interceptor needs maximally 25 trials to encipher the message, just by going through all possibilities.
This cipher could be cracked by a human in a matter of minutes, which disqualifies it from serious application.
However, the cipher can be reinforced by altering the key, for example depending on the position of a letter in a word. The first letter is shifted by

$k$, the second by $k+1$, etc. This would be the polyalphabetic version of the Caesar shift.
The cipher created is as a matter of fact a special case of the next cipher type.

## 2.2 The Vigenère cipher[2]

A further development in cryptography was made with the introduction of keys longer than one character. In the Vigenère cipher one uses a keyword to encipher a message. The process is best explained with an example. One still operates with an alphabet and assigned positions 0-25 for the letters.

| keyword | m | a | t | h |
|---|---|---|---|---|
| keyword number | 12 | 0 | 19 | 7 |

Next, the keyword is written over the message as many times as needed. The letter on top indicates the shift, the message letter obviously is the letter shifted. That operation is performed by the formula used for Caesar shift encryption.

| mathmathmat | 12 | 0 | 19 | 7 | 12 | 0 | 19 | 7 | 12 | 0 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| exampletext | 4 | 23 | 0 | 12 | 15 | 11 | 4 | 19 | 4 | 23 | 19 |
| qvttblxaqxm | 16 | 23 | 19 | 19 | 27 | 11 | 23 | 26 | 16 | 23 | 38 |

Thus we can see that the reinforced Caesar cipher was indeed a Vigenère cipher with the whole alphabet as the keyword, starting at a selected point. After the encryption, one letter is represented by different symbols and the same symbols can actually stand for different letters.

When encoding using a five character key with no repeated letter, one has five possible non-identical representations of a three characters word, such as "the".

| a | b | c | d | e | t | h | e |
|---|---|---|---|---|---|---|---|
| t | h | e | | | (a+t) | (b+h) | (c+e) |
| | t | h | e | | (b+t) | (c+h) | (d+e) |
| | | t | h | e | (c+t) | (d+h) | (e+e) |
| e | | | t | h | (d+t) | (e+h) | (a+e) |
| h | e | | | t | (e+t) | (a+h) | (b+e) |

[2]http://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher, 16.02.2015

All in all there are five word possibilities, 26 encryption possibilities per letter (the shift of 0 shall be regarded as a shift, since even when a letter once remains unchanged it does not matter when all others are changed). This method offers $5 \cdot 26^3 = 87880$ potential representations for a three character word using a keyword with no repeating characters.
Compared to the 25 possibilities in the Caesar shift, the Vigenère cipher's 87880 would, assuming both are worst case scenarios, in which the solution is eventually calculated , take approximately 3515 times as long to compute by testing all potential ciphers. The time needed to find the length of the key is disregarded.
With a fixed key, the number of representations a word has is equal to the number of the keyword's characters.

# 3 Mathematical concepts

## 3.1 Extended Euclidean algorithm[3]

Before discussing the extended version, first the Euclidean algorithm shall be presented. The Euclidean algorithm is used to find the greatest common divisor (from now on referred to as GCD). The procedure is based on division with rest. Useful definitions concerning the greatest common divisors will be deduced first. Variables $a, b, f$ are integers. One makes use of the fact that the $GCD(a, b)$ also is the $GCD(a - b, b)$. To prove that, one first proves that $GCD(a, b)$ divides $(a - b, b)$ and then that there is no greater divisor than $GCD(a, b)$ to $(a - b, b)$.

(1) $GCD(a, b)$ divides both $a$ and $b$
(2) $GCD(a, b)$ divides $sa + tb$ when $t$ and $b$ are integers, since the term can be factorized, leaving an integer in the bracket. Since s can be 1 and $t$ can equal $-1$, it also divides $a - b$
(3) $GCD(a, b)$ divides $(a - b, b)$, according to (1) and (2)

Now the proof that it is the greatest divisor follows.
For the $GCD(a, b)$ is a divisor of $(a - b, b)$, according to (1) it must divide both a and b . It is assumed that there is a number f greater than GCD$(a, b)$ which can divide $(a - b, b)$. The number $f$evidently must divide $a - b$ and $b$. Using (2), it also must divide $(a - b) + b = a$. The consequence is the condition that it must divide $a, b$ and $a - b$, but there is no greater

[3]Christian Spannangel: "Erweiterter Euklidischer Algorithmus" Teil 1: http youtu.beQORmBQo8j0, Teil 2: http://youtu.be/bNa_fLCs5GA, Teil 3: http://youtu.be/nD6psV2vkRU, 5.6.2012

common divisor to $a$ and $b$ than $GCD(a, b)$. Hence the existence of the number $f$ is disproven and it can be stated that:
(4) $GCD(a, b) = GCD(a - b, b)$
The following definitions are stated with apodictic certainty.
(5) $GCD(a, b) = GCD(b, a)$
(6) $GCD(1, a) = 1$
(7) $GCD(0, a) = a$, for $a \neq 0$
When calculating the $GCD(a, b)$ with $a > b$, one is interested in first reducing the figures $a$ and $b$. An example with $a = 739$ and $b = 211$ will be used to demonstrate the process.
$GCD(739, 211) \Leftrightarrow GCD(739 - 211, 211) = GCD(528, 211) \Leftrightarrow$
$GCD(528 - 211, 211) = GCD(317, 211) \Leftrightarrow GCD(317 - 211, 211) =$
$GCD(106, 211) \Leftrightarrow GCD(211, 106) \Leftrightarrow GCD(211 - 106, 106) =$
$GCD(105, 106) \Leftrightarrow GCD(106, 105) \Leftrightarrow GCD(1, 105)$
The process can be stopped here, since it is known considering (6) $GCD(1, a) = 1$. When the GCD is equal to one, $a$ and $b$ are said to be *relatively prime* or *coprime*.
It is obvious that the subtraction of 211 does not need to be repeated $q$ times, but one can subtract $q \cdot 211$ instantly. This leads to the Euclidean algorithm, which can be noted as follows for known integers $r_1$ and $r_2$:

$$r_1 = q_1 \cdot r_2 + r_3$$
$$r_2 = q_2 \cdot r_3 + r_4$$
$$...$$
$$r_{n-2} = q_{n-2} \cdot r_{n-1} + 0,\ r_{n-1} \text{ is the } GCD(r_1, r_2)$$

One calculates divides $r_1$ by $r_2$ mentally, notes the rest $r_3$, and follows the algorithm. The modulo operator is especially practical to use in this algorithm:

$$r_3 = r_1 \bmod r_2$$

or generally

$$r_k = r_{k-2} \bmod r_{k-1}$$

The extended Euclidean algorithm is used to calculate the algorithm backwards, it is for solving the equation $ax + by = c$, the linear Diophantine equation. It will be used to express $c = GCD(a, b)$ as a linear combination of $a$ and $b$. Returning to our example, noted with the Euclidean algorithm:
(1) $739 = 3 \cdot 211 + 106$
(2) $211 = 1 \cdot 106 + 105$
(3) $106 = 1 \cdot 105 + 1$
(4) $105 = 105 \cdot 1 + 0$

Starting from the $GCD(739, 211) = 1 = 106 - 1 \cdot 105$, the algorithm is reversed in order to express the last remainder in terms of $a$ and $b$:

| | |
|---|---|
| $1 = 106 - 1 \cdot 105$ | (2) is rearranged for 105 |
| $1 = 106 - 1 \cdot (211 - 1 \cdot 106) = -1 \cdot 211 + 2 \cdot 106$ | (1) is rearranged for 106 |
| $1 = -1 \cdot 211 + 2 \cdot (739 - 3 \cdot 211) = 2 \cdot 739 - 7 \cdot 211,$ | resulting in $x = 2$ and $y = -7$ |

## 3.2 Congruence modulo[4]

When one writes $a \equiv b \mod c$ , it means that $a$ and $b$ leave the same rest when divided by $c$, where $a, b$ are integers and $c$ is a natural number. The relationship is called congruence and is denoted with the symbol seen above. The term is true when the difference between $a$ and $b$ is divisible by $c$. It is important to mention that the $\mod c$ is not the modulo operator used on a side on the expression, but just an information in which modulo the numbers are congruent.

## 3.3 Euler's totient function[5]

The function noted as $\phi(n)$, alternatively named the Eulerian phi function, counts the number of totatives of an integer $n$.
Totatives are natural numbers relatively prime to $n$ found in the interval $[1; n[$.
To check whether two natural numbers $a, b$ are co-prime, one needs to solve the linear Diophantine equation $ax + by = 1$. If the equation can be satisfied with integers $x, y$ then $a$ and $b$ are coprime. As shown above, the process may be very time-consuming.
However there are particular cases where $\phi(n)$ can be calculated without the algorithm, thus saving time.
Definitions for special cases:

$$\phi(ab) = \phi(a) \cdot \phi(b)$$

For a prime number $p$:

$$\phi(p) = p - 1$$

[4]Christian Spannagel: "Definition Kongruenz", http://youtu.be/eD2_Q2KsYj8, 10.1.2012

[5]http://en.wikipedia.org/wiki/Euler%27s_totient_function, 16.02.2015

### 3.4 The Fermat-Euler theorem[6]

A relationship incorporating the Eulerian totient function into Fermat's little theorem, valid for relatively prime numbers $a, n$:

$$a^{\phi(n)} \equiv 1 \mod n$$

## 4 The RSA algorithm

### 4.1 The method of operation

In the previous ciphers both the transmitter and the receiver applied an identical key on the message. Ciphers with this property are called symmetrical ciphers, since the same key is used to go both ways between the plain text and the cipher text.
The innovative part of the RSA algorithm is that it is an asymmetrical algorithm: It uses two keys, the encrypting public key and the decrypting private key. This mechanism implies that the ability to write a message in a cipher does not entail the ability to reverse the encryption, at least not in an acceptable time frame. This condition arises from the mathematical basis of the algorithm: It is not mathematically challenging to multiply two primes or perform a modular exponentiation, but the inverses are immensely time-consuming and possibly expensive.

### 4.2 Generating the key

Initially two prime numbers $p$ and $q$ need to be found. This is untroublesome, since checking for primality is an relatively easy task. In practice both $p$ and $q$ are larger than $2^{512}$ [7].

Next, the primes are multiplied, the product is called $N$. The factors $p$ and $q$ are secret, while $N$ is public.

$\phi(N)$ is then found using $\phi(N) = (p-1) \cdot (q-1)$. The fact that the factors $p, q$ are secret forces an interceptor to calculate $\phi(N)$ by checking all numbers from 1 to $N$ for common factors with $N$.

The public exponent $e$ (for encryption) is generated. It can be chosen from the set $1 < e < \phi(N)$, with the conditions that it must be an integer and has to be relatively prime to $\phi(N)$, meaning that the $GCD(e, \phi(N)) = 1$. The

[6] http://en.wikipedia.org/wiki/Fermat%27s_little_theorem#Generalizations, 16.02.2015

[7] "The RSA cryptosystem.mp4", http://youtu.be/guYNbJkiGUI

condition regarding relative primality are crucial: If they are not fulfilled, it may be impossible to en- or decrypt the message.
The public key consisting of $e$ and $N$ is now ready.
The private key $d$ (for decryption) is determined as the multiplicative inverse of $e \mod \phi(N)$, which stands for $e \cdot d \equiv 1 \mod \phi(N)$.

## 4.3 RSA in practice

In order to encrypt a message, an example key must be computed.
To prevent the mathematics from becoming unnecessarily complicated, small prime numbers $p = 7$ and $q = 17$ are selected. Now the key generation steps are followed.

$$N = p \cdot q \Rightarrow 7 \cdot 17 = 119$$

$$\phi(N) = (p-1) \cdot (q-1) \Rightarrow \phi(119) = 6 \cdot 16 = 96$$

$e$ is chosen according to the conditions. For the example it is defined to be $e = 13$.

To calculate $d$ one needs to solve the equation

$$e \cdot d \equiv 1 \mod \phi(N) \Rightarrow 11 \cdot d \equiv 1 \mod 96$$

Since in modular arithmetic all terms $e \cdot d + k \cdot \phi(N) \equiv 1 \mod \phi(N)$ are congruent for all $k$ values, one needs to find an expression which outputs 1 as a linear combination of $e$ and $\phi(N)$. For the reason that $e$ and $\phi(N)$ were chosen to be relatively prime, it is known that their greatest common divisor is 1 and they can fulfill the linear Diophantine equation.Thus one applies the extended Euclidean algorithm:

$$\begin{aligned} 96 &= 7 \cdot 13 + 5 \\ 13 &= 2 \cdot 5 + 3 \\ 5 &= 1 \cdot 3 + 2 \\ 3 &= 1 \cdot 2 + 1 \end{aligned}$$

The extended Euclidean algorithm is used to express 1 as a linear combination of $\phi(N)$ and $e$:

$$\begin{aligned} 1 &= 3 - 1 \cdot 2 \\ 1 &= 3 - 1 \cdot (5 - 3) = 2 \cdot 3 - 1 \cdot 5 \\ 1 &= 2 \cdot (13 - 2 \cdot 5) - 1 \cdot 5 = 2 \cdot 13 - 5 \cdot 5 \\ 1 &= 2 \cdot 13 - 5 \cdot (96 - 7 \cdot 13) = 37 \cdot 13 - 5 \cdot 96 \end{aligned}$$

When one uses the operator $\mod 96$ on the last expression, it becomes

$$37 \cdot 13 - 5 \cdot 96 \equiv 1 \mod 96$$

$$37 \cdot 13 \equiv 1 \mod 96$$

Knowing that $e = 13$, $d$ is apparent when the expression is compared to

$$e \cdot d \equiv 1 \mod \phi(N)$$

$$\Rightarrow d = 37$$

Finally, the key is finished. The public part $(11, 119)$ can be published. Now one is able to encrypt and decrypt a message using the following formulas for a message $m$, which must be smaller than $N$ and relatively prime to $p$ and $q$, and the concealed message $c$ :

$$c \equiv m^e \mod N$$

It can be seen that only the public key is used to encrypt the message.

$$m \equiv c^d \mod N$$

The private key unravels the message.

To avoid needless translation of letters into numbers, the message is given as a number, in this example it is 5.

$$54 \equiv 5^{13} \mod 119$$

The encrypted message representing 5 is 54. The second formula is used to reverse the process:

$$5 \equiv 54^{37} \mod 119$$

Most calculators have failed to compute these calculations, the calculator *Web 2.0 Taschenrechner*[8] shall be credited for completing this colossal task. The key is proven valid, the output is identical to the input.

[8] http://web2.0rechner.de/

## 4.4 Analysis

How can such an elegant procedure work and grant security? And why were the conditions for $e$ and $d$ constructed the way they are?
From the formulas used it can be deduced that:

$$c = m^e \mod N$$

$$m = c^d \mod N$$

$$m \equiv c^d \equiv m^{e^d} \equiv m^{e \cdot d} \mod N$$

Since $e \cdot d + k \cdot \phi(N) = 1 \Leftrightarrow e \cdot d = -k \cdot \phi(N) + 1$, as was said when using the extended Euclidean algorithm, it can be applied to change the expression

$$m^{e \cdot d} \equiv m^{-k \cdot \phi(N)+1} \equiv m \cdot m^{-k \cdot \phi(N)} \equiv m \cdot m^{\phi(N)^{-k}} \mod N$$

Using the Fermat-Euler theorem the anticipated equivalence is shown

$$a^{\phi(n)} \equiv 1 \mod n$$

To fulfill this equation, $m$ must be relatively prime to $N$. This is the reason $m$ was defined as relatively prime to $p$ and $q$, the only factors of $N$ except 1 and $N$. Since in practice the numbers $p$ and $q$ are of the order $2^{512}$ and $N > 2^{1024}$, these conditions are not very likely to cause problems. The final step in the recovery of the message is

$$m \cdot m^{\phi(N)^{-k}} \equiv m \cdot 1^{-k} \equiv m \mod N$$

The reliability of the procedure on the secrecy of $p$ and $q$ also becomes transparent in these steps. Knowing $p$ and $q$ results in knowledge of $\phi(N)$, which is closely linked to $d$ and can be easily used to calculate it.

## 4.5 Evaluation

The RSA algorithm greatly relies on the insurmountable obstacle which factorization embodies. The greatest number factored so far is the 232-digit number RSA-768 [9], which took two years of computation with professional algorithms and supercomputers. Factoring the standard 1024-bit numbers used to encode was estimated to take 1,000 times more time than the 768-bit

[9] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Joppe W. Bos, Pierrick Gaudry, Alexander Kruppa, Peter L. Montgomery, Dag Arne Osvik, Herman te Riele, Andrey Timofeev, and Paul Zimmermann: "Factorization of a 768-bit RSA modulus", https://eprint.iacr.org/2010/006.pdf, 18.02.2010

RSA-768 using the same setup. Considering the cost of running supercomputers for two or even 2,000 years straight the profit chances of such endeavor are questionable: The data may not be worth that much from the beginning or may lose value with time, making the whole effort pointless.
A realistic threat to the algorithm is the Shor's algorithm [10], an algorithm which can factorize integers in a considerable amount of steps and thus in a relatively short time. The only reason it has not been used yet is that it must be run on a quantum computer, which are not yet constructible, but the technology will be inevitably developed. The first quantum computer will mean the downfall of public key cryptography in the sense it is presently known: The computation power will be sufficient to conduct one way functions in two ways.

# 5 Reflection

For this exploration I have chosen a topic that I have not dealt with in mathematics courses at school even approximately and I am very glad about it. The concepts were unknown to me and I had to start from the basics, such as learning what the modulo operator is and the basic operations and conceptions of modular arithmetic. Learning by oneself may be challenging but in the end the understanding is much deeper than if one is exposed to already processed content.
I also see a vast future advantage in knowing how to use the extended Euclidean algorithm, especially since it is used in connotation with the abstract idea of linear combinations, which has made clear how general and extensive basic mathematical operations are.
When constructing the key, I have struggled with the calculation of $d$, getting either a negative $d$ or an identical one to $e$. I had to identify the problem myself and come up with another solution via trial and error and so I deeply internalized the extended Euclidean algorithm after going through it 15 times.
The idea which public key cryptography is based on, namely that multiplying is an easy task while factorizing is relatively unfeasible has become clear to me, but the reason for it is not really transparent. It may be because that fact is so deeply rooted in the structure of numbers that I yet lack background knowledge on what numbers are to see through the problem.
The method of the RSA algorithm was surprisingly simple at first sight, but the simplicity transitioned into confusion while constructing the key by

[10] http://en.wikipedia.org/wiki/Shor%27s_algorithm

myself. The conditions for the key are derived from different mathematical concepts, which are merged into the algorithm and it was challenging to win insight into the particular subparts of the structure while also gaining an overview. It was the first time in my life that I have learned this many distinct concepts and discovered the connections in the ideas independantly.
In retrospect I think that I might have included more concepts related to the RSA, like the Chinese Remainder Theorem, but the reason I chose not to is that it is not necessary and , since it is only serves as an optimization to the algorithm and is not essential to it. I felt no need to further elongate the exploration.
Alternatively, I could have dealt with elliptic curve cryptography (ECC), another widely used modern cryptosystem based on similar mathematical concepts, I did not find it as appealing as the RSA algorithm.
Of course not all that I learned is entailed in this exploration: Some concepts were too specific, another I could not yet grasp. I have the feeling that I only scratched the surface of the matter, but I acquired a basic understanding of a very worthwhile topic which I look forward to delve into.