Lodz University of Technology

Faculty of Electrical, Electronic, Computer and Control Engineering

Institute of Applied Computer Science

Engineer’s thesis

**The web application supporting encryption algorithms   
–**

**Does cryptography provide security?**

Paweł Grzegorzewski

Album number: 190116

Thesis written

under supervision of

**Justyna Żuk, PhD**

Łódź, January, 2017

**Contents**

[1. Introduction 4](#_Toc471680541)

[2. Involved technologies 7](#_Toc471680542)

[**2.1.** **HTML5** 7](#_Toc471680543)

[**2.2.** **Cascading Style Sheets (CSS3)** 8](#_Toc471680544)

[**2.3.** **JavaScript (JS)** 8](#_Toc471680545)

[**2.4.** **JQuery** 9](#_Toc471680546)

[**2.5.** **Cryptico.js** 10](#_Toc471680547)

[3. Software 12](#_Toc471680548)

[**3.1.** **Git** 12](#_Toc471680549)

[**3.2.** **Adobe Photoshop CS6 (PS CS6)** 13](#_Toc471680550)

[**3.3.** **Notepad ++** 14](#_Toc471680551)

[**3.4.** **Eclipse** 14](#_Toc471680552)

[4. Information included in the thesis 16](#_Toc471680553)

[**4.1.** **General cryptography issues** 16](#_Toc471680554)

[4.1.1. Cryptology 16](#_Toc471680555)

[4.1.2. Cryptography 18](#_Toc471680556)

[4.1.3. Cryptanalysis 18](#_Toc471680557)

[**4.2.** **Cryptography methods** 19](#_Toc471680558)

[4.2.1. Caesar 19](#_Toc471680559)

[4.2.2. Vigenere cipher 20](#_Toc471680560)

[4.2.3 Merkle-Hellman Cipher 22](#_Toc471680561)

[4.2.3. RSA cipher 23](#_Toc471680562)

[**4.3.** **Digital Signature** 25](#_Toc471680563)

[**4.4.** **Electronic signature** 26](#_Toc471680564)

[5. Practical solution – webapplication ‘Magic of cryptography’ 28](#_Toc471680565)

[5.1. Graphical user interface 28](#_Toc471680566)

[5.2. Practical part – cipher and signature implementations 29](#_Toc471680567)

[5.2.1 Caesar cipher implementation 29](#_Toc471680568)

[5.2.2 Vigenere cipher implementation 30](#_Toc471680569)

[5.2.3 Merkle-Hellman cipher implementation 31](#_Toc471680570)

[5.2.4 RSA cipher implementation 33](#_Toc471680571)

[5.2.4.1 Possible algorithm modifications 34](#_Toc471680572)

[5.2.5 Digital signature implementation 34](#_Toc471680573)

[6. JavaScript Cryptography Library 35](#_Toc471680574)

[**6.1.** **xxx** 35](#_Toc471680575)

[**6.2.** **xxx** 35](#_Toc471680576)

[**6.3.** **xxx** 35](#_Toc471680577)

[6.3.1. xxx 35](#_Toc471680578)

[6.3.2. xxx 35](#_Toc471680579)

[6.3.3. xxx 36](#_Toc471680580)

[6.3.4. xxx 36](#_Toc471680581)

[**6.4.** **xxx** 36](#_Toc471680582)

[6.4.1. xxx 36](#_Toc471680583)

[6.4.2. xxx 36](#_Toc471680584)

[6.4.3. xxx 36](#_Toc471680585)

[7. The web application supporting encryption algorithms 37](#_Toc471680586)

[**7.1.** **Graphical user interface** 37](#_Toc471680587)

[**7.2.** **xxxx** 37](#_Toc471680588)

[7.2.1. xxx 37](#_Toc471680589)

[7.2.2. xxx 37](#_Toc471680590)

[7.2.3. xxx 37](#_Toc471680591)

[**7.3.** **xxx** 37](#_Toc471680592)

[7.3.1. xxx 37](#_Toc471680593)

[7.3.2. xxx 37](#_Toc471680594)

[7.3.3. xxx 37](#_Toc471680595)

[8. Summary 38](#_Toc471680596)

[9. Appendix 39](#_Toc471680597)

[Bibliography 40](#_Toc471680598)

# Introduction

Topic of data security is known for centuries. There are many reasons to hide true meaning of the text against inappropriate receivers. The first documented appearance of secured information comes from ancient Rome. It is easy to predict that the reason was to send secret military orders and to make them readable only for proper allied commanders. Disclosure of this kind of information was very dangerous. Here we came to the roots of cryptography. We do not know who exactly came with the idea of manipulating the characters of the message using the secret algorithm. This algorithm known only for sender and the receiver we name “a key”. Thanks to that key sender is able to encrypt plain text and get a cryptogram. Key gives also the opportunity to retrieve true meaning of the cryptogram. The algorithm used in that process is a cipher. At the very beginning no one expected that message which at the first sight consists of totally random characters can hide the secret but within the years people learnt how to break the cryptogram without the key. And here we have to introduce cryptanalysis – the branch of science which is about analysing cryptograms and finding its corresponding plain text. One is the opposite of the other, and both of them affect each other. Cracking the cipher force the people to invent more complex encrypting algorithms to provide higher level of security.

Demand for better and more ciphers is still present. Over the years knowledge and technology progressed enormously. Nowadays we do not use piece of paper and a pen to solve problems connected with cracking the data. Computers enabled us to make huge amount of computations in a short period of time. We can say that computers started totally new age for cryptology. Having in mind the speed of computations we have to ask the question which known ciphers may be considered as these which may provide sufficient level of security for our message? What are the constraints of these ciphers? Which factors may positively affect the encrypting algorithm and which may make our cipher useless? This will be the main goal of this thesis – analyse existing encrypting methods and judge if messages encrypted with them may be easily broken or not. Unfortunately topic of data protection and cryptology itself is not well known among the people. Because of that fact the special approach have been taken. There are few available webservices and applications designed for learning and presenting the process of encryption of few cryptography methods. Few of them are prepared for learning basics of cryptography and others are used for academical purposes. There are lots of problems connected with them. The most common is that they are dedicated for specific group of users, either users who have no previous contact with cryptography or for users who have at least intermediate experience with this topic. There are also few websites which enable user to encrypt and decrypt information but gives no theoretical background. The great example is French site *http://www.dcode.fr/* which implement a huge variety of ciphers, however there is no explanation how the encrypting algorithm works. Another example is *http://rumkin.com/* whichimplements less ciphers than previous website but also gives only laconic description of each presented cipher. There are also some web applications which concentrate only on one particular cipher like *http://www.xarg.org/tools/caesar-cipher/* (there is only one cryptography method among many other small programs like calculators).

The most advanced web application of that kind is CrypTool. It is not only a web application, there are also versions of that program designed for Windows, Linux and MacOs operating systems. It is a great application which provides both theory and practical implementations of huge variety of ciphers. It also gives information about digital and electronic signatures, key exchange protocols, random key generators and many more. It covers almost all topics connected with the cryptology. The author of CrypTool is Bernhard Esslinger. There were many university workers involved in the project from universities of Darmstadt, Duisburg-Essen, Kassel and Siegen. The first version of the program was released1998. The newest application CrypTool 2 released two years ago. It provides modern graphical user interface. It realises all ciphers with graphical block workflows which enables user to see every step of encryption or decryption process. However this approach makes encryption pipeline more difficult than filling in required coefficient in a cipher’s form in the previous version of CrypTool. Both versions have some advantages and drawbacks but none of them provides theory part. Cipher descriptions are available on web version of CrypTool – CrypTool Online. It is suited for any kind of users, no one should have any problems with using it. It has almost every feature which such a service should be equipped with. Thanks to accessible characterizations and clear, and professional approach to the practical presentation of ciphers online version outwent other versions. This diploma thesis application was inspired by CrypTool software.

The final solution is the web application supporting the encrypting algorithms. One of the most important assumptions connected with the solution is that ordinary user may not be familiar with the cryptology or have sufficient knowledge of mathematics or computer science that is why the theoretical background was provided. The application is divided into two main parts: theoretical and practical. The theoretical part of the application explains the basis related to the subject of cryptology and gives the introductions to each cipher used in the application. These sites provide all required information to understand and enable user to try how the cipher works. All the information is presented in such a way that everyone will understand operation of the cryptography system: people acquainted with the topic as well as these who read about ciphers for the first time. The second part of the application is focused on the application of the cryptography systems. Within these sites user is able to use the ciphers. Each cipher is editable, there is a possibility to change important coefficients and options. The application has got functionalities of encryption and decryption of arbitrary text. One of the most important aspects concerning the practical part is the section describing safety of the cryptosystem. Application not only gives the general information about the safety of the cryptography system but also reacts to the provided input. The output about the safety depends closely on all provided data. This functionality gives the real filling about the cipher, user can try lot of examples, modifications of the cipher and compare achieved outputs. This is the feature which set the software apart from other available programs. After research of many web sites and tests of few desktop applications the conclusion is that none of them have the connection between encryption and immediate feedback about the security level of the cryptogram.

# Involved technologies

To create an application which fulfils all stated requirements there should be involved few different technologies. To obtain proper appearance of the application, support of the inputs and operation on the provided data there was combined five different programming languages.

* 1. HTML5

HTML5 – Hypertext Markup Language is the newest version of the programming language used for creation internet sites. Initial release of that standard was published on 28th of October 2014. It takes it roots from the previous version HTML4 and its XML version XHTML1. HTML5 like the previous versions uses the special mark-ups to make the content of the document structured and make it visually distinguishable. There are three main categories of markup language: presentational, procedural and descriptive markup language. The first type is responsible for preprocessing document in such a way that both printed document and its electronic version on the computer screen will look exactly the same. Procedural markup supplies the document with the tags which give the instructions to the processor how to edit the text. HTML belongs to the descriptive family of the markup languages. The aim of the descriptive markup is to label and divide the document. Semantics allows the special presentation of the created document. It does not provide any specific instruction about the processing of the file. However it states the role of the text. All the information about edition and formatting is usually stated in the separate place by a Cascade Style Sheets. The modification stated by the tag affects part of the text placed between opening and ending tags.

HTML5 makes the websites less static, it influences the level of interactivity of the site. It provides new semantic elements like: <section>, <article>, <header> and <footer>, graphic elements like <canvas> and multimedia elements <audio> and <video>. New features of the HTML for storing data in the browser are “localstorage” and more powerful “indexedDB”. One of the most important aspects of the new standard is the improved error handling. Browsers supporting current version of HTML check the correctness of the syntax. HTML5 is also compatible with older versions of the webbrowsers which just ignore new unknown tags.

That technology was used to design and create all websites of this thesis application. It is very easy to use and provide few very applicative properties. HTML5 was helpful with preserving whole structure of the document. Using the tags makes the administration of the project very easy and fast. Changes for particular part of the project such as for example sections or headers may be applied in few seconds. Using the same frame in the all documents of the project makes them consistent and so ensures clarity.

* 1. Cascading Style Sheets (CSS3)

CSS3 is the latest version of the language used for describing and providing user defined presentation of the websites. CSS was inspired by the Document Style Semantics and Specification Language (DSSSL) which was the first language created for the purpose of defining style and look of the documents. The inventor of the Cascading Style Sheets is Norwegian computer scientist Hakon Wium Lie. He described his idea in the book titled “Cascading Style Sheets: Designing for the Web”. It was published in 1994. Cascading Style Sheets works perfectly along the HTML mark-ups. CSS3 is fully compatible with its previous versions. We can split whole language into many modules where old version of CSS is also treated as a module. The most important among all are: Text Effects, Animations, Selectors, Background, Borders. CSS file consists of the list of rules and directives bound with specified part of the documents. The greatest module which enables proper choose of the elements to edit is Selectors. It has been developed in the CSS3 by adding more constituents which allow to make more precise selection. Next new module introduced by the latest standard that is worth to pay more attention is “Media Queries”. Media Queries enables user to control the behaviour of the website on different resolutions of the screen. We can choose range of screen sizes and state adequate scaling of websites elements. Thanks to that there is no need to separate versions for mobile devices and appearance of the website is adjusted within the media queries.

To easily administrate and manage the whole project CSS and HTML files should be separated. However there is still possibility to inject CSS code into HTML file. Adding reference to the CSS file in HTML document user has to remember about the priorities of processing the directives. In that kind of situation there may occur overlay of rules, that is why it is vital to place the CSS directives in the separated file.

* 1. JavaScript (JS)

It is the most popular scripting language aimed at web development. JavaScript language was created by the Netscape company in May 1995. As a main author and originator of the language should be considered American programmer and hacker JavaScript is based on the ECMAScript which was created by Sum Microsystems. JS is considered as a high-level, dynamic, untyped and interpreted programming language. Three technologies: HTML, CSS and JavaScript are present in the majority of World Wide Web projects. Within the years many different implementations of the JavaScript were created. There were no official standardization. What is more different browsers had their own interpretation of Document Object Model (DOM). That is why it was not possible to create website which would look and behave identically on all the most popular webbrowsers. Luckily W3C organization crated standardized model DOM and in 2009 finally the common version of JS was published.

Language cannot be treated as a classical object-oriented programming language. Instead of classes JavaScript uses prototypes. The instance of the prototype is an associative array. The other property of the JS is the way of typing. In JavaScript we do not state the type of the variable explicitly. The type is set with the assignation of the value to the variable, however it may be retyped at any time appropriately to the needs. JavaScript relies on a run-time environment to provide interaction between the script and a website DOM. This feature is essential to enable importing the scripts to the HTML documents. Whole operation of the JS is based on the call stack. JS is able to process one message form a queue at the time. Each time the message is processed, the adequate function is called. It is placed with proper arguments on the stack and waits till the moment all the previously called functions end their operation. The function stack resizes during the run within the website.

The purpose of using JavaScript is to add dynamic to HTML sites. The code may manipulate all elements of Document Object Model. The most popular applications of the code are: manipulation of the browser bookmarks, adding animations, creation of dialogs and control predefined style sheets and form validation. The last thing can be easily achieved by regular expressions which are supported by JavaScript. Programming code can be included in the HTML document in two different ways. It may be placed between two special tags ‘<script>’ which are created to introduce code in any scripting language. But the more elegant way is to again separate code and place them in a file with ‘js’ extension. Inside the HTML document should be then attached only function calls.

In the practical solution of this diploma thesis JavaScript was used mainly for cryptography methods implementations. There have been also added some functions to manage Cascade Style Sheets for some simply animations and handling all inputs. As a main scripting technology JavaScript was used for a practical workflow for all used cryptography methods.

* 1. JQuery

JQuery is the next technology appearing in the thesis which is library designed for JavaScript. It is the most popular JS library. JQuery first release took place in August 2006. It is still under process of development, last stable release was in September of a current year. It is an open-source, free software. There are many applications of that library but one of the most important is the improved way of selecting elements of the Document Object Model for which JQuery prepared totally new syntax. Second very important functionality of JQuery is the ability of animation creation. Library is also equipped with simple API for realisation AJAX requests. It realises them with two built-in functions ‘jQuery.get’ and ‘jQuery.post’. Like in the case of JavaScript itself JQuery is compatible with all popular webbrowsers however the effects may slightly differ on each of them.

There are two ways of using JQuery. First of them is using ‘$’ function which is very useful design pattern – factory method which enables to use chainability of the code. All functions with ‘$’ return objects of type JQuery. Second method is using ‘$.’ utility function which does not operate upon JQuery object straight away.

Very useful feature of library which I mentioned above is the chaining of the code. JQuery function returns object of type JQuery that is whe we can easily bind many functions together. Thanks to that we are able to shorten our code by lots of lines preserving at the same time clarity of the code. Functions which occupy several lines of code in JQuery often may be rewritten into two or three lines. This functionality is usually used for manipulation of CSS. It is achieved by passing selector to the function on call the proper methods on it.

JQuery library was used few times to add some simply animations. It influenced the appearance of the web application and made it more pleasing to the eye and user friendly.

* 1. Cryptico.js

Cryptico is an open-source JavaScript library dedicated for RSA encryption. It implements one of the most popular and very safe cryptography algorithms. Cryptico.js enables user not only encrypt or decrypt message with the RSA but also to sign whole message. Library uses few important secondary libraries. The most significant are “jsbn.js” and “aes.js”. First one introduces necessary type for the implementation of RSA cryptography method which is big integer and mathematical operations with these numbers. It is required to use very high primary numbers to make the cipher hard to break and it was the only possibility to use that implementation of big integers. As the RSA is asymmetric cipher it needs pair of asymmetric keys. Cryptico.js uses Tom Wu’s RSA encryption library which implements random key generation. Keys are also encrypted with the AES cryptography method. RSA cipher provided by the Cryptico.js enables user to choose between five different available lengths of the key. These are 512, 1024, 2048, 4096 and the longest possible 8192 bit key length. All of them provides very high security level, all encrypted messages cannot be cracked on theory by a personal computer.

Implementation of the RSA cipher is not an easy task that is why the library was used to introduce this cryptography method for users. In the thesis application its interface was used in two different places, that is site about the cipher itself and describing a digital signature. In the second place there are mixed both processes of encryption and signing the message to show the more complex way of creation secured message.

# Software

Well suited software is very important during process of development. Software as a tool should help user with creation the code, management and edition of all project files. The more advanced the software is the more optimal is the work. Programming process becomes much faster, and effective, there is a lower possibility to make errors, finding sources of problems and resolving them is also easier. All these aspects have to be taken in mind at the first phase of planning.

During the thesis project realisation a lot of time had to be spent on preparation of images and animations. That is why another type of software − graphical program, alongside these strictly connected with programming, was used. After some consideration the following softwares were involved.

* 1. Git

Git is one of the most popular softwares for version control. It is a distributed revision control system with own Source Control Management (SCM) and is aimed for speed, data integrity and non-linear workflows. The author of the Git is Linus Torvalds Finnish software engineer, who is also known as an one of the major developers of Linux kernel. Another important person attributable with a git software is Junio Hamano, the programmer responsible for maintenance of the software. The first release of this free software was in April 2005. As a distributed control system each created Git directory is an outright repository. There is no main repository like in client server type of version control systems. Every repository has got full revision history and tracking abilities. There are three main stages of the Git may be distinguished. First one is a Working Directory where user is working on his/her version of code. Before publishing the revision it is unique code which is only available on users computer. The second stage is named Staging Area which is a pre-commit stage. All the changes which are made since the last fetch of a code may be administrated and manipulated at this stage. User may choose which changes should be pushed and which should be discarded for some reason. The last stage is a Git Commit, part responsible for publishing the code and resolving all emerged conflicts.

What is huge advantage of Git there is no need to access network or a central server to be able to work on repository. It is one of the essential kind of softwares in every information technology projects. Git is very fast in comparison to competitive softwares and very scalable. It finds great application in any kind of project, for projects administrated by only few people as well as for huge commercial projects which involves tens or hundreds of Information Technology specialists. Git’s strong point is the process of branch creation and merging which is also visualised in special tool. It has got also great approach to the branches – they are only references to a commit, the logical structures which makes branch very light. There are many ways to publish the repository, it may be accomplished by Hypertext Transfer Protocol (HTTP), Secure Shell (SSH), File Transfer Protocol (FTP). Each revision is not the whole copy of the code because after some time of development whole project becomes huge and storing code at each revision memory cost would be enormously high. That is why revision remembers only the change in regard to last version. This difference is called by Git the delta.

Git was used via the most popular Git repository hosting service GitHub. It enables user to oneself some of the features on the web. As the diploma thesis is quite big project it is very reasonable to save whole revision history. Using a Git repository was kind of self-protection against loosing or breaking the software. GitHub played also another very important role during the process of creation the thesis. GitHub provides not only the whole history of revisions but also shares the graph presenting the amount of changes within the time. This chart was a great motivator to further work and to make the graph only ascending. It is incredible how huge impact to the diploma project made so trivial thing like this progress graph.

* 1. Adobe Photoshop CS6 (PS CS6)

Adobe Photoshop CS6 is the latest version of the most famous software for raster graphics edition. Raster graphics is the way of presenting picture by the grid of pixels. Unlike vector graphics the quality of the raster graphics picture depends on the resolution of the picture. Adobe Photoshop can be concerned as a flagship software of the Adobe Systems company. Adobe Photoshop was created by brothers Thomas and Jon Knoll in nineteen ninety, but first commercial edition was published in two thousand and three. Since four years there is the newest sixth version of the Adobe Photoshop. The source code of an application is written in C++, however first releases were written in Pascal. Files created in Photoshop have .psd extension. These files are the project files which are divided by layer and each created element is fully editable. Files have also the latest history of changes. CS which is part of name of the software comes from “Creative Suite” special extension which as merged with the major program. Creative suite added the possibility to create own shortcuts, tool for searching, filter gallery, colour matching and many other very useful tools. Photoshop is not only the most popular but also the most advanced software of that kind. It supports huge range of graphical files, also animation files like for example .gif. Unfortunately its biggest competitor GIMP has got substantially less functionalities and is less user friendly than PS CS6. There are lot of great functionalities which makes this software amazing for amateur users as well as professional graphics designers. The most important tool which I also used for preparing graphics for my thesis is “planar selection”, tool which enables user to select and then cut the part of the image.

All the pictures used in the practical part of my diploma thesis were edited in the Adobe Photoshop CS6. Some clip arts or ready elements from the internet were also used but all of them were edited and changed in the program. There are no pictures (despite pictures presenting authors of cryptography methods) which were just downloaded and put without any change within a thesis. There are also two short animations which show the operation of Vigenere and Caesar ciphers. This kind of presenting an example is more appealing to the users of the application. It helps to imagine whole process of encryption.

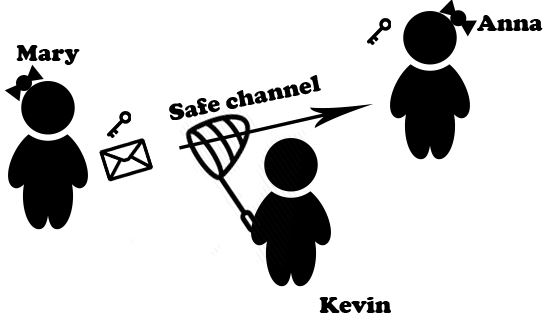


Figure 1 – Example of graphics created in PS CS6 used in the application

* 1. Notepad ++

Notepad++ is an advanced text editor based on the Scintilla project. Software is created in C++ language with the Win32 API by programmer whose pseudonym is “Don Ho”. Notepad is more advanced substitute of basic Windows tool – Notepad. It was released on two thousands and three and is still under a development. Notepad++ supports a lot of file types. The next great advantage is recognition and marking the syntax of the code with different colours. That makes the file presentation very clear for users. Notepad++ offers also auto completion of the code. Each edited file within the program stays in its memory till the close it in the interface of Notepad++.

I have used this program for small changes of the project. It was only used as substitute of the main software for development of my web application which was Java Eclipse Mars, definitely more advanced and better suited program for creation of the application.

* 1. Eclipse

Eclipse is one of the most popular integrated development environments (IDEs) and the most often used by Java developers. It is free, open-source software, written in Java programming language. The first version comes which comes from IBM rational code. It was created by Eclipse Foundation in 2001. Under the name of Eclipse foundation there were initially grouped many different well known brands like: Borlands, IBM, Red hat, Rational Software and many more. The last stable release was month ago so the software is being still at the process of development and improvement. Eclipse like other advanced programming environments enables user to create applications in many different languages. Each programming language may be added to the Eclipse by special plugin. There are also lot of functionalities available in form of plugins. The most important are tools for graphical user interface or IML graphs creation. Eclipse may be also extended with plugins responsible for connection with servers or databases.

Eclipse has to be considered as a main developing software which was involved in the thesis. Despite small added or edited in the previously mentioned Notepad++ all the application code was written in Eclipse. This IDE is very user friendly and helpful in the development process. Numerous helpful features provided by the software like code colouring, code suggestion, tree presentation of project files, easy and fast code refraction make writing of the code very rapid, simple and convenient.

# Information included in the thesis

The most important assumption connected with the software which must be highlighted is to provide information understandable for any kind of user. With that in mind there was provided few sites devoted with general questions connected with the cryptology. Among them there are sites which explain what the cryptology is about, what are the differences between cryptography and cryptanalysis and many others. It is vital to read these theoretical parts before moving to the cryptography methods and encrypting to have complete view about the topic of cryptology.

The main part of the application are sites dedicated to cryptography methods. It was not easy task to choose proper ciphers for the application. There were many aspects under consideration before including cipher in the application like: level of complexity, type of cryptosystem, key type, safety of cryptosystem and so on. User should learn, investigate and try ciphers which belong to different available types.

* 1. General cryptography issues

Cryptology itself is very broad topic. It is very important to understand some of basic information before going deeply into matter of cryptosystems and safety provided by them. Web application proposed as a solution of this thesis provides some sites which purpose is to explain in a very clear understandable way basic topics connected with the cryptology.

* + 1. Cryptology

To understand all information presented in the thesis user should at the begging realise what the cryptology is about. Cryptology is a branch of science very closely connected with mathematics. Cryptology is generally related with the data, the process of data manipulation which main purpose is to hide its true content from unwanted users. This data manipulation is a cipher – way of changing a plain text into another text, which should look like nonsense for people we want to hide information from. We can consider a cipher as an algorithm, set of steps which must be performed in order to change a text into a secret and secured information. The most important part of the cipher is a key. The key is usually a string of letters which enables receiver to retrieve the plain text from cryptogram. It must be remembered that there are two main types of keys and as a result two sorts of cryptosystems: symmetric and asymmetric. There are also two expletive and competing parts of cryptology: cryptography and cryptanalysis. The first part is responsible for hiding and securing information and the second one breaks the cryptogram and retrieves plain text. Cryptology is very broad topic. It is not only about ciphering data and cracking them, it covers also random key generation, key exchange protocols, digital and electronic signatures. We can state that the cryptology takes its roots in ancient Rome. All historical resources state that Roman army as a first tried to hide the meaning of the important military orders from the enemies. Roman were using cipher which name came from currently ruling dictator, one of the most important people from whole history Gaius Julius Caesar. At the same time the opponent and complementary part of cryptology – cryptanalysis was born. Since that times both branches of this science were developing alongside. One influenced the other and vice versa. People were creating more complex and providing higher level of safety cryptography methods. There were created two types of ciphers substitution in which every character from the plaintext is replaced by ciphertext character according to the fixed rule – key. Another type of cryptosystem is known as a transposition cipher where used in the plain text characters are changing their positions within a ciphertext. All mentioned above ciphers belong to one family of cryptosystems – symmetric cryptosystems. Symmetric cryptosystem uses just one key in both processes: encryption and decryption. It has got advantages as well as few major drawbacks. The biggest one is that both sender and receiver of encrypted message are obliged to state the key before transmission of the messages. These cryptosystems have no protection in the case when attacker somehow steals the secret key.

The first milestone in cryptology took place during the second world war when one of the biggest inventors in the history Alan Turing as a first constructed mechanical computer. A.Turing and his invention not only changed the course of a war and saved millions of lives but also began the age of computer science. That moment, since computers started to create cryptograms and break them was an enormous change for cryptology. From these days computers took the humans role. Amount of computations which computers are able to perform in a short period of time enabled us to analyse, change and test giant amount of data. Evolution in computers architecture still lasts and accelerates more and more. Speed and amount of computation performed by today machines grow very fast. That influences also both cryptography as well as cryptanalysis and motivate to development.

The second – huge milestone in cryptology took place in 1976when the idea of public key was proposed by Whitfield Diffie and Martin Hellman. This kind of cryptosystem is based on the pair of different keys. Sender uses one key for encryption and the receiver uses second one to decrypt the message. Both keys are very closely attributable to each other but it is really hard to retrieve or guess one from another. This feature solves most of the problems which are related to symmetric cryptosystems. Most of modern ciphers belong to this relatively new family of ciphers. The idea of digital signatures is also based on the asymmetric cryptography.

Nowadays cryptography overtook cryptanalysis in their ‘race’. There are few cryptography methods which provide sufficient level of safety and even the most powerful machines are not able to retrieve plaintext from the cryptograms. According to many scientific publications these lead will remain for next ten or fifteen years however we cannot be sure about that. The industrial and technological progress is enormously fast and there is no possibility to predict how huge power computers will have in few years.

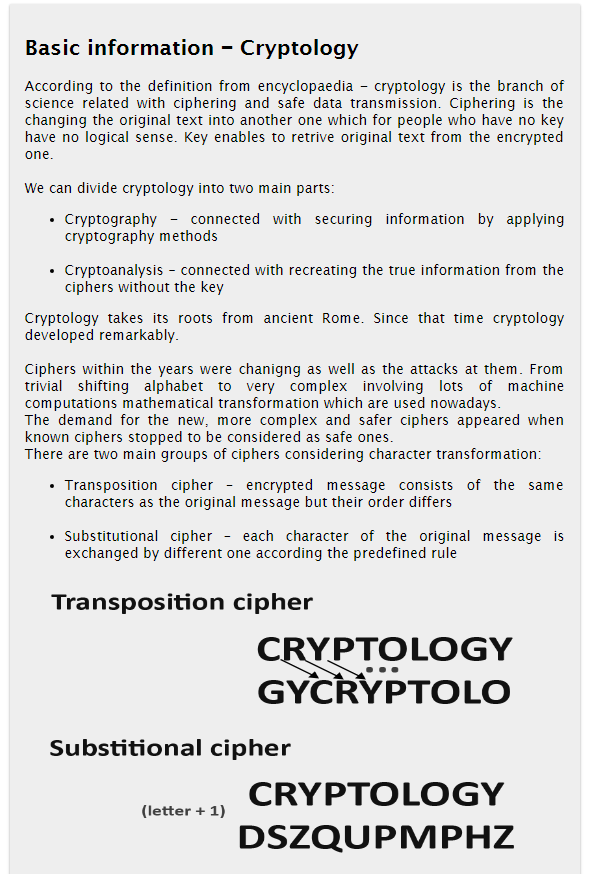


Figure 2 – Cryptology (description provided in the application)

* + 1. Cryptography

Cryptography is a branch of cryptology responsible for securing information. The main goal of cryptography is to cipher plaintext to provide sufficient level of safety which means that no third parties will be able to retrieve secret message. Cryptography strives to create such cryptogram which will be conditionally easy and unconditionally as difficult as possible. Conditionally easy cryptogram is such a cryptogram for which the time and amount of arithmetical operations needed to retrieve plaintext is low in a case when user owns a key. On the other side retrieving the plaintext from cryptogram without any knowledge about the used key should be as hard as possible – this is what ‘unconditionally difficult’ means.

There are many diffusions of cryptography methods concerning different aspects, in regard to used alphabets, way of changing plain text characters but the most important is in regard to the key type. Since 1976 an idea of asymmetric cryptosystem was created. All previous ciphers use symmetric key which means that encryption and decryption can be done by using one key. These kind of ciphers had many advantages. Computational complexity of encryption and decryption is relatively low, which is a big virtue. Drawback of symmetric cryptosystem is the users are forced to fix the key before the transmission. Having in mind that it is vital to change keys to preserve high level of security, the symmetric key becomes a huge disadvantage.

The totally new, asymmetric cryptosystem introduced huge change – pair of keys. In asymmetric cryptosystems users generate pair of keys: encrypting and decrypting one. Sender produces keys, and publishes key responsible for encryption. That kind of approach resolves totally problem of stating key which was present in symmetric cryptosystems. However the computational complexity is much higher than in previous case. Process of encryption and decryption lasts longer and need more resources than the symmetric ciphers. What is more interesting asymmetric cryptography found application in other important area of cryptology – digital signature.

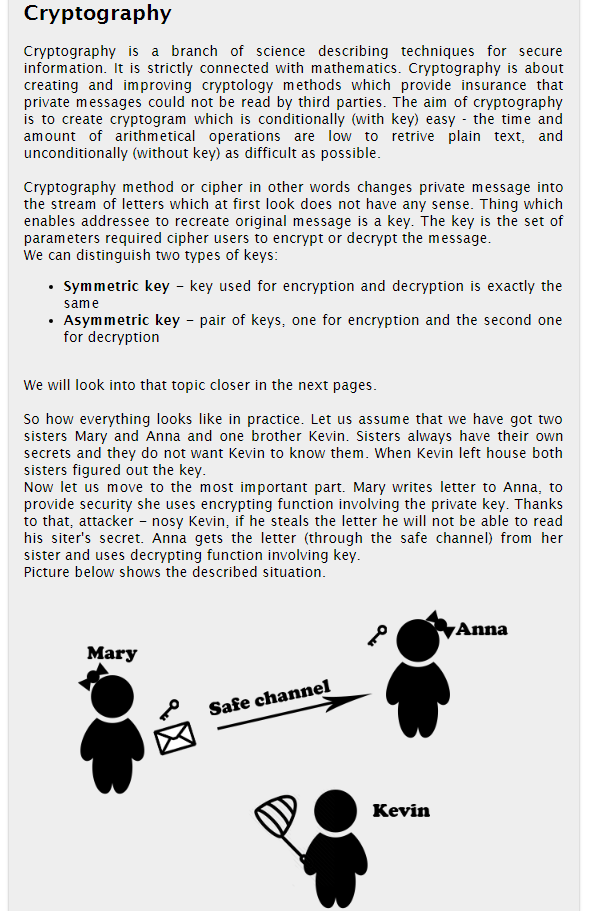


Figure 3 – Cryptography (part of the description provided in application)

* + 1. Cryptanalysis

Cryptanalysis is a second branch of cryptology which is totally opposite to cryptography. It appeared in the history almost parallel to the cryptography. As cryptography’s opponent this kind of science arose right after the first cipher. Cryptanalysis is responsible for finding and using vulnerabilities of cryptography methods. After finding and using weakness of cipher a cryptanalyst is able to retrieve plaintext from cryptogram. Each attack at cipher requires special amount of resources. There are three main which need to be considered during attack: time needed to retrieve message, computers memory needed to perform all required computations, and amount of data which cryptanalyst intercepted. All these three resources are strictly connected and they affect each other. Considering the last, the most important resource – data, there are many different situations in which cryptanalyst can collect it. There are at least four main worth to mention: situation when cryptanalyst owns only ciphertext, more convenient when cryptanalyst has got both – ciphertext and corresponding plaintext and situation when cryptanalyst has got temporary access to either encrypting or decrypting machine and collect date from it.

Cryptanalyst may be considered as a danger not only for encrypted message itself but also for a transmission channel. This channel can be attacked passively or actively. In active wiretap sender is impersonated by attacker which means that the authenticity of the message is threaten. Passive wiretap is the vulnerability against the confidentiality of the message. In this kind of wiretap attacker focuses on transmission channel and tries to intercept private message.

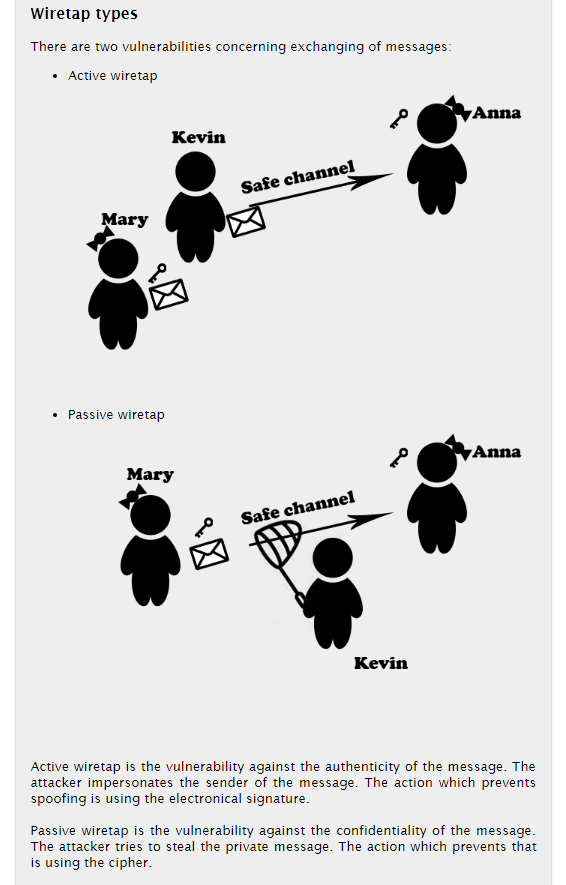


Figure – Cryptanalysis (part of desription provided in the application – wir etap types)

* 1. Cryptography methods

One of the most significant issues concerning practical solution of the thesis was the proper choice of cryptography methods which should appear in the web application. Application should cover ciphers from both symmetric and asymmetric cryptography system families to show the user major differences between them. Beginning user should also start his or her adventure with the cryptology with an easy example of the cipher, that is why there are examples of the classical cryptosystems in the application. On the other hand it was necessary to put an example of modern and very powerful cryptosystem. To give the user complete view at exchanging secured messages the digital signature should be also available. After a research and having in mind all mentioned requirements the following cryptosystems were described and implemented in the thesis.

### Caesar cipher

One of the most basic and probably the oldest cipher in human history. This classical cipher was firstly used in ancient Rome for military purposes. Caesar cipher belongs to the symmetric, transposition family of ciphers. The cryptogram is created by replacement of the every plain text letter by letter shifted by fixed value. The key determines the value of the alphabet shift. There are two different ways of presenting key in the Caesar cipher. We can present key as a number or as a letter. The second option goes to show the letter which the first letter of original alphabet will be replaced by. Originally Roman were using only the shift three, nowadays we can find in many places implementation with shift thirteen. Apart from the value of shift, there is also possibility to state the direction of the shift.

Encryption in the Caesar cryptography system is described by the following formula:

,

where

*E*(*i*) − *i*-th character of encrypted text,

*D*(*i*) − *i*-th character of plain text,

*k* – shift value.

Decryption looks almost exactly the same, the only difference is the opposite sign in the formula

.

Unfortunately Caesar cipher does not provide any acceptable level of safety. Because of the fact that all characters of the cryptogram are only shifted by one value from the original message the secured message may be cracked in no more than few minutes by everyone who has got pen and piece of paper. This simple cryptography method is weak against all available types of attack. Its space of keys which means how many possible keys may be used are determined by the length of used alphabet.

### Vigenere cipher

Vigenere cipher is strongly connected with the Caesar cipher. This cipher belongs to a polialphabetic, substitution family of ciphers. It is also based on the idea of using shifted alphabet but is more advance than its ancestor. Vigenere cipher can be realized with the tool invented by Johannes Trithemius − Tabula Recta. Tabula Recta is a two dimensional square array. Each row contains all characters from the alphabet in proper order however each consecutive row has higher shift than previous one. Tabula consist of 26 rows and includes all possible shifts of the alphabet and with vertical and horizontal headers which contains not shifted alphabets. Trithemius basing on his own invention created also very basic cipher. During the encryption each consecutive character of the cipher was taken from the shifted alphabet where value of the shift was equal to the index of the plain text character. Around fifty years later in 1553 true inventor of Vigenere cipher Giovan Battista Bellaso described his invention in the book ‘La Ciphra del. Sig. Giovan Battista Bellaso’. Bellaso added one improvement to Trithemius cryptography system which was remarkable amelioration − the key. Key was a string of characters used during encryption. When user of Vigenere cipher wants to encrypt a message with the Tabula Recta he/she just need to look at the row (column) where header shows the letter to encrypt and the column (row) with the header with key character. Cipher text character can be found at the intersection of row and column. In the case when the lengths of plaintext and the key are not equal, key need to be shorten or copied to adjust to amount of plaintext characters. Blaise the Vigenere also added something to this famous cryptography method – new type of key which is called autokey. Autokey is just a letter or short phrase to which user has to add almost whole original message to make the plain text and the key equal in terms of length.

From the mathematical point of view process of encryption can be described by the following formula:

,

where

*C*(*i*) − *i*-th character of ciphertext,

*D*(*i*) − *i-*th character of plaintext,

*K*(*i*) − i-th character of key.

Decryption process looks almost exactly the same like the encryption process. It is described according to the formula:

.

Vigenere cipher may be considered as a cryptography system which provides very high protection level. What is more, under three conditions, the Vigenere cipher can count as a perfect one. If the key is as long as a plain text, it is totally random and used for only one encryption, then the cipher may be considered as unbreakable. Encryption which fulfils all these three constraints is known as an one-time-pad encryption. This method was invented and described at the first time in 1882 by Fran Miller. However it was reinvented and patented in 1919 by American cryptologist Gilbert Vernam who was also inventor of polyalphabetic cryptography system known as Vernam cipher.

There are few different methods of attack at the Vigenere cipher. The first one invented by Friedrich Kasiski is about finding repetitions in the ciphertext. Kasiski assumed that the same occurrences of part of ciphertext may be encrypted with the same key. Finding the distance between these repetitions can denote the possible lengths of the key. After find the length of key the frequency analysis can be applied. This kind of attack is about counting the number of occurrences of certain character and matching them to possible plaintext letters knowing the fact how probable the character is within a language.

Other attack known as a Friedman test is based on index of coincidence (IC). This index measures how similar a frequency distribution is to the uniform distribution. Index of coincidence can be counted from the following formula:

where

*IC* – index of coincidence,

*fi*– count of the letter,

*N* – amount of characters in the encrypted text.

### Merkle-Hellman Cipher

Merkle-Hellman cipher belongs to asymmetric cryptosystems family with public key. It is one of the first asymmetric ciphers in the history. It was invented in 1978 by Ralph Merkle and Martin Hellman. Cipher is also one out of many ‘knapsack’ ciphers. The name comes from the mathematical problem which has got the same name. The problem is defined as follows: for a given set of numbers A the goal is to find any subset of numbers B which gives the sum S. Merkle-Hellman cryptosystem is based on the special case of sum of subset problem – whole set of numbers has to be superincreasing, That means the each consecutive number in the set is higher than sum of all previous numbers from the set. It is very important aspect because in traditional knapsack question there are usually more than one subsets which fulfil the problem. That would mean that there are more than one solutions which could give plaintext or there are more than one possible plaintexts which is also a nonsense.

Keys are the two different sets of numbers and two additional elements. Private key is a superincreasing set of numbers. To generate public key, which is also set of numbers, user has to find the sum of private set and choose arbitrary number which has to be coprime with the sum. Each element of public key is the result of modulo operation of the sum from the result of multiplication of element from private set and chosen multiplier. Both multiplier and sum of private set are part of public and private keys.

To perform encryption user is obliged to convert the message into the binary string. After that all consecutive numbers need to be multiplied by the consecutive elements of the key set. The cipher text is the sum of all results of multiplications. Depending on lengths of plain text and key, ciphertext may consist of one or more numbers. The process of encryption can be described by the formula:

where

c – ciphertext character,

*αi* – *i*-th bit of the plaintext,

*βi* – *i*-th element of the public key set.

Process of decryption is a bit more complicated than encryption. Receiver of encrypted message at the beginning has to perform modulo operation of sum from result of multiplication of number representing ciphertext by modulo reverse of multiplier. The result of the operation is the number which has to found in knapsack problem. Each element which is part of the ciphertext sum gives bit ‘1’ and the other gives ‘0’. The last step is just to convert given bits into characters string. Process of decryption based on solving superincreasing sum of subset problem is described by the following formula:

where

*m* – plaintext character,

*wi*– *i*-th element of the private key set.

In terms of safety Merkle-Hellman cryptography system like other knapsack ciphers provides very high level of secure against brute force attack. Brute force attack is computational ineffective method for this cipher. Its computational complexity is θwhich for big *n* gives very time consuming problem to resolve. However there was found a vulnerability connected with the knapsack cipher. Adi Shamir co-author of the one of the most popular and safe cryptography system used the dynamic programming and wrote the algorithm which solves the knapsack problem in pseudo-polynomial time. Because of this loophole Merkle-Hellman cipher does not find application nowadays.

### RSA cipher

RSA is one of the most popular asymmetric cryptography systems with public keys. It was invented in 1977 by three cryptologists Ron Rivest, Adi Szamir and Leonard Adleman. The cipher is very universal, it finds application not only in message encryption but also may be used for creating digital signatures. Message encrypted with RSA with signature with the same cryptosystem fulfils all important safety aspects. It provides confidentiality, integrity, authenticity, and non-reputability. RSA cipher is based on the difficulty of factoring of result of multiplication of two very large prime numbers. Multiplication of numbers is not a problem even for very large integers, however difficulty arises for opposite side operation which is factoring. That kind of computation is very hard and resources consuming even for very powerful computers. There are no algorithm which would significantly speed up the whole process. That makes RSA very powerful cryptography tool which under few constraints may be considered as a perfect cipher.

The security level depends mostly on the length of the key. The longest known broken key used in RSA was 768 bit. It was achieved at the beginning of 2010. The whole process of cracking the key was performed by a huge cluster of computers which consists of hundreds of machines. The time spent on cracking took around two years of computations. There are no published information about cracking longer key. Because of the fact how hard and resources consuming the cryptanalysis process for RSA is, the most common implementations of this cryptosystem have between 1024 and 4096 bit long keys. According to many predictions in around ten to twenty years 1024 bit key length can be considered as not enough safe. However the technology development is enormously fast and computers’ computational power is growing year over the year what may result that cracking longer keys may be achieved faster than the all predictions say.

To generate key user at the beginning need to choose two prime numbers. The higher these numbers are the higher safety will be assured. Next step is to multiply both these coefficients and the result *n* becomes one of the common factor for both private and public key. Next important factor *Ф*(*n*) is the result of multiplication of both previously chosen factors decreased by one. To obtain public key user has to choose one more coefficient *e.* This coefficient need to be coprime with the *n* coefficient. Just to remind, two numbers are coprime when the only positive common divisor for them is equal one. The last step of key generation is to get second parameter for private key *d*. To determine *d* coefficient user has to perform the Extended Euclidean algorithm. Parameter *d* is the greatest common divisor of *e* and *Ф*(*n*). Finally, the public key creates the pair (*n*, *e*) and the private one is (*n*, *d*).

RSA encryption process can be described by the following formula:

where

*c* – cipher text character,

*m* – plain text character,

*e* – parameter of the public key,

*n* – parameter of the public key.

Decryption is described by very similar formula which is as follows:

where

*d* – parameter of the private key.

RSA as very effective cryptography system finds many different applications. It is regularly used in internet browsers, in applications based on exchanging data like email, chat applications, virtual private networks and for user identification.

* 1. Digital Signature

Digital signature is an another method used for data security. It is strictly connected with the cryptography systems, it uses mathematics to achieve authenticity of electronic documents. Digital signature was firstly described in the 1976 by Whitfield Diffie and Martin Hellman as a possible application of their great invention which was asymmetric cryptography. It was only theoretical solution, they did not present any practical implementation of the signature. The first proof that asymmetric cryptography system can implement digital signature made inventors of RSA cipher Ron Rivest, Adi Szamir and Leonard Adleman. Digital signature is based on the asymmetric cryptography, however it is realised in an opposite way to the ciphers. During the ciphering sender uses public key of the receiver to encrypt the message and receiver uses his private key to decrypt the document. In the case of signature, sender uses his private key to generate his signature and receiver of the message verifies the authenticity by using public version of the sender key.

Similarly to the ciphers there can be distinguished three main phases connected with digital signature scheme. First phase is a creation of the pair of keys, firstly random private key is generated from the all possible keys. Basing on the private key the other from the pair is calculated. Second phase is signing the document. User performs signature function on the document in order to sign it. Last phase is on the receiver side – signature verification. In the process of verification involved is public key of sender which states if the signature belongs to the proper person or not. Process of verification gives also more important information which will be explained later. There is one very important rule connected with the securing data with both encryption and the signature. Sender of the information should encrypt signed message. If user signs encrypted message signature may be easily verified and exchanged if the case of theft of the document.

There are some cryptography systems which can be applied to both encryption and signing the electronic documents. RSA is a great example. Previously discussed ciphers provide data confidentiality, which means that message is readable only for people who are allowed to do that. Only people who have access to the key can read the true content of the encrypted message. Digital signature does not provide data confidentiality, however it fulfils other very important aspects of data protection. First and foremost digital signature, like traditional written signature, verifies the author of the signature. This feature is called authentication. The second very important feature – non-repudiation states that sender of the message cannot deny that message comes from him or her. The last important feature tells the receiver that the original message was not altered by third parties during the data transmission.

Digital signature finds application mostly in electronic signature implementation. However electronic signature is not used in all countries because of the legal reasons. Very important fact is that digital version of signature is more reliable than traditional, handwritten one. It is much harder to forge digital signature than traditional one.

* 1. Electronic signature

Electronic signature is a cryptography mechanism which allows to verify the owner of signed document. Electronic signature is a digital correspondent of the handwritten signature. Like its traditional version it provides the legal standing to the electronic document. It is in most cases realised by the digital signature. Many people use these digital and electronic signature as synonyms which is a mistake. Digital signature should be considered rather as a process of signing the document and electronic signature is rather the result of this process. Electronic signature is concerned as a legal confirmation of authenticity. The common regulations concerning signature which are named eIDAS for countries of the European Union were stated in nineteen nineties. The implantation and realisation also varies within countries. There are some features which are always present. These are: possibility of author verification, signatory is the only person who have access to his/her private key, no one else should have access to it, electronic signature is a function of message which means that signature depends on the content of the message, change of the content alters the signature, so electronic signature play also a security role.

Electronic signature has got different levels of security. At the bottom of this hierarchy is basic – ‘electronic signature’. It only confirms the identity of person who signs the document. Next level of signature security is provided by ‘Advanced electronic signature’. The signature is strictly connected with the author and particular document. Mathematical transformations provided by cryptography system makes the signature almost impossible to forge. The highest safety standard gives ‘Qualified electronic signature’ – the special kind of the electronic signature which has to fulfil all the legal, technical and organizational requirements. This specific signature is ensured by the Certificate Authority Office. User gets a qualified certification, the private key, a cryptographic card on which the key is stored. Qualified electronic signature is relatively expensive signature and it must be renewed.

The next important area where electronic signature may be applied is the biometrics. There are many biological qualities which are unique for every person. Examples of these qualities are: fingerprint or iris patterns. This kind of data can be collected by the special kind of sensors and to the bit form and used in the electronic signature. The biggest drawback of using biological identity in the signature is that this makes signature much less trustable as long as this information may be easy spoofed.

# Practical solution – webapplication ‘Magic of cryptography’

The Magic of cryptography application consists of two main parts: sites concerning theoretical descriptions of main topics connected with the cryptology and descriptions of implemented ciphers. All the information available on the informative sites are very accessible and comprehensible even for users for who contact with the application would be the first contact with cryptology. Application offers also supportive pictures and two animations. Description, images, animations and well described examples should bring all available topics to users. The second part enables user to try the ciphers and also a digital signature in practice. There are four different cryptography systems supported with the application. Implementations allow to change available parameters of cryptography systems. So user can compare not only the ciphers itself but also ciphers executed with different parameters. One of the most important feature which makes Magic of cryptography different then all available in the internet is the feedback about of safety which appears after each encryption. This enables user to have a real feeling about the ciphers and shows the most important aspect of encryption process.

### 5.1. Graphical user interface

The template of the site of the application is rather traditional one. It has got two side menus: left one covers all links to sites with theoretical content and right one leads user to practical part. Nowadays the most popular is the one main vertical menu however in this kind of webapplication traditional approach is more legible. At the top there is the banner with the title of application, in the middle there is main part for the content and the last important part is the footer which contains the information about the author of the application. Whole template is responsive and all parts are resizing and changing their layout in regard to screen resolution. However whole application was design to use it on computer screen and for smaller resolutions using it especially practical part. may be less convenient. The colours used in the application are three different shades of grey, black and white. Two most extreme shades: black and white are for the fonts, grey colours are used to differentiate segments of the site. There are only few places where some other colours may appear like animations of ciphers or two small icons for special buttons. These are: blue graphic of information sign which shows the animation and the second one again blue key links the user from theoretical site strait to the practical correspondence. This colour palette of colours does not distract user and this few exceptions highlights important places. Other important aspect was the proper readable font. There was used standard ‘Lucida Sans’ font which belongs to sans serif group of fonts.

### Practical part – cipher and signature implementations

Magic of cryptography gives an opportunity to try and test four different ciphers and a digital signature. All implementations are suited to particular cryptography system. Each cipher allows to set and manipulate different coefficients which are used within an encrypting algorithm. User can observe not only different cryptograms on the output but also is able to see information about safety level of the encryption. Most of inputs are mostly textboxes and dropdown menus which make the usage very user friendly. Distribution of these elements are prepared in a similar way in all the practical sites. The cipher workflows are designed to be as intuitive as possible. Moreover some textboxes and buttons are disabled until user provides all required inputs. For example button responsible for encryption process stays disabled until plaintext and key are chosen by the user. Each practical site remind also main information about provided cryptosystem. All these aspects make testing cryptography systems very easy and immediate.

* + 1. Caesar cipher implementation

Caesar cipher is the first cipher presented in the Magic of cryptography application. After short description of the cipher user can try this classical cipher in two different ways. First presentation covers two the most famous versions of the cipher. Each inputs calls the adequate set function which saves user input into variables. At the beginning user chooses between historical used by Roman army and present implementation of Caesar cipher which can be found in many chat applications or forums. First one uses the key 3 and the second one is based on the key 13. User chooses one of these options from the dropdown menu. To make the encryption two remaining steps are just to fill the textarea with the plaintext in and click an ‘Encrypt’ button. This button calls the function which performs the encryption algorithm, which iterates through all characters of the plaintext and applies proper shift. In the second version available on the site user is able to choose any available key between 0 and 25. Key in Caesar cipher is also often represented by a letter that is why user can also fill second input by the letter. These two inputs are bounded together and once the first is set second is also changed adequately. The last aspect connected with key is the shift direction available in the dropdown menu. The only thing left is to write message for encryption and click ‘Encrypt’ button to get proper cryptogram. Like in the previous case all textareas and button calls proper functions provided in js file. As long as the decryption process does not differ from encryption there is no separate workflow for that operation. This is also mentioned in the short reminder at the beginning of the site so user is able to decrypt cryptogram with the form provided by the application, the only change user need to make is to set different shift direction. At the bottom of the page there is a section about the safety. As the safety does not change with different keys user has access only to one description. Caesar cipher practically does not provide any safety, cryptogram may be cracked in fem minutes by everyone on the paper. For computer finding the plaintext lasts not even a 1/1000 of second. Despite that fact it is still used in the previously mentioned communicators.

* + 1. Vigenere cipher implementation

Like in the case of previous cipher, practical site presenting Vigenere cipher resembles the most important information concerning that cryptography system. This practical site has the highest amount of forms, so to make this part more readable there is a division between encryption and decryption part. Both parts are accessible through two appropriate buttons. Clicking on ‘Encryption’ or ‘Decryption’ button shows the adequate forms.

Encryption button gives the access to three different implementations of Vigenere cipher. First available form is connected with classical utilization of that cryptography system. User fills two different textareas, one for plain text and second for key phrase. These textareas calls key and message setters on onblur event. After setting both variables the button ‘Create key’ becomes enabled. This key call function responsible for adjusting key provided by the user to the message length. If the key is longer than the plaintext, key is cut, however if the key is shorter than a message to encrypt, key has to be copied as many times to obtain proper length. New matched key is set in the new ‘Key result’ textarea. Having these things filled last ‘Encrypt’ button becomes available. Onclick event calls the function which changes plaintext into cryptogram, which is set in the last textarea. What is more important encryption function has got also reference to another function responsible for analysis of the key and plain text to give user feedback about safety. Each encryption shows the safety section. Feedback includes information about key space, and points at the key length and suggests user to make key longer while it does not match the message length.

The second implementation covers Vigenere with autokey encryption. This form has exactly the same type and arrangement of inputs. ‘Create key’ calls the function which converts chosen before plain text and key onto autokey invented by the Blaise de Vigenere. Function takes first character from the key and concatenates plaintext without last letter to it. This key appears in the ‘Autokey’ textbox. Like in the previous implementation at this moment ‘encrypt’ button becomes available. Encrypt function called by the button takes plaintext and the autokey and iterates in the loop through all characters and applies encrypting transformation. Cryptogram shows on the text box next to ‘encrypt button’

The last method of encryption uses tool on which the cipher was based at the beginning-Tabula Recta. There is a huge square made of buttons with letters on it. Only letters placed in the Tabula Recta headers are enabled. Because of the fact that that tool is symmetric, key or plaintext can be placed both vertically or horizontally, and both will give the same true result of encryption. That is why it should be decided at the beginning what will user place on columns and rows. On the site letters chosen vertically are considered as key characters and plaintext is placed on the other side. Each time user clicks on button respectively whole column or row of buttons is coloured on red and the result of encryption is highlighted on green. Clicking on button triggers not only highlighting but also clicked letter is saved and when both key and plaintext character are set, three textareas are updated with the user’s input. There is also additional protection provided, whenever user clicks two times in the row on key buttons or plaintext ones letter chosen as a first is rejected by the application.

The last form is responsible for the process opposite to the encryption. Decryption consists of four textareas, two of them are for user input. User sets encrypted text and key. After billing both of them there is function triggered which transforms key into a reversed one. As the decryption may be performed according to two different algorithms, where one involves normal and the second one involves reversed key, it was important to show the user how the reversed one looks like. To get plaintext from the cryptogram user needs to call decrypting function which is bound with the ‘Decrypt’ key. Algorithm which performs decryption uses reversed key, so it is exactly the same which is responsible for encryption.

* + 1. Merkle-Hellman cipher implementation

Merkle-Hellmann cipher is the first example of modern, asymmetric cryptography system which is supported by Magic of cryptography application. It is one out of many available knapsack ciphers. This cryptography system has got one more action which has to be performed than previous traditional examples. That process is random key generation. Apart from that there are also two essential sections: encryption, decryption and two informative parts – cipher description and comment about safety.

Asymmetric cryptosystems require to generate and publish one from the pair of keys. The most important elements of keys in Merkle-Hellman ciphers are two sets of numbers. In the key generation section user at the beginning specifies in input of numeric type the size of the key set. Setting this coefficient in the textarea runs three functions. First one is just a setter of key number set size. Next function generates superincreasing set which becomes a part of private key. This is put also to the special textbox on the site. Below the textbox second text area is filled with the total sum of the previously mentioned set which is second part of the key. At this moment there is only one thing left to choose – multiplier which is cooprime to the sum of the set. User fills second text area and on blur the function creating public key starts. This function checks at the beginning necessary constraint – if the chosen multiplier is cooprime with the sum of generated set of numbers. If the cooprime test return true, last textarea in the key generation section is populated with elements of public key set.

Next two sections covering processes of encryption and decryption consists only of four elements each. Encryption section requires from user only setting message for encryption and running the encryption algorithm by clicking the ‘Encrypt’ button. Despite that fact, functions supporting these input and output elements are more complicated than it may seem. After chose of the plaintext, all characters are converted to ASCII numbers and then into 8 bit strings. This bit sting is shown to the user in a separate textbox. Encryption algorithm needs to adjust key length to number of bits from the plaintext. During encryption cipher creates one or more numbers which comes from multiplication of consecutive bits and elements from key set which are the cryptogram.

Decryption part does not provide opportunity to set own cryptogram. This section makes use from cryptogram from the previously created ciphertext. ‘Decrypt’ button calls few different function. First one transforms cryptogram into numbers which can be easily recreated back to the plaintext bits stream. Transformation applies modulo of private key sum from each element of cryptogram multiplied by a modular inverse of the chosen multiplier. This converted cryptogram is printed in the special textarea. Next function takes each element of this preprocessed cryptogram and iterates through private key to find elements which summed up will be equal to the ciphertext numbers. All numbers which are composing cryptogram give 1 as a plaintext bit, rest give 0s. Last function takes all bits, groups them into 8 bit strings and converts back into ASCII numbers and characters. Converted cryptogram, bit string and at the end plain text, all these things are presented to the user in the separate output places.

Safety feedbacks contain constant information about general safety knapsack cryptography system. Cipher is very resistant against brute force attack, the higher the numbers in the key set, the higher level of safety against this kind of cipher break. Cipher has got unfortunately important vulnerability. Merkle-Hellman cryptography system like other ciphers based on knapsack problem are weak against dynamic programming. Other well-known cryptologist Adi Shamir wrote dynamic algorithm which breaks the cryptogram in pseudo-polynomial time. Feedback gives also information about key space for the key which was stated by the user in the first section on this site. Weakness pointed by Shamir makes the Merkle-Hellman cipher useless nowadays.

* + 1. RSA cipher implementation

RSA cipher is the second cryptography system after Merkle-Hellman cipher which uses asymmetric keys. That cipher is the most important cipher supported by the diploma thesis webapplication. It is used in a wide range of applications. Apart of easy to understand and learn cryptography systems it was necessary to provide cipher more complex cipher which at the same time provides really high level of safety and under some limitations is concerned as unbreakable. Implementation of RSA is very complex task so in this thesis it is used open source JavaScript library Cryptico.js which provides all necessary functions for using the cipher. Cryptico.js uses also few side libraries which implements biginteger type used for storing key, support AES cryptography system used for hashing and few more.

After short presentation of the cipher, user moves to the encryption form. First two textareas user fills with the message and passphrase. Passphrase is transformed according to the SHA256 algorithm. Hash result is then used by a random number generator which becomes an argument of Tom Wu’s RSA key generator. To generate key user must also choose between five possible key lengths. Having passphrase and key length set user can generate a pair of keys. Unfortunately user is not able to choose *p* and *q* coefficients, all is automatically generated by the provided generators. Button ‘Generate Key’ fills three disabled textareas: *n* and *e* parameters of public key. As the all three coefficients *n*, *p* and *q* are represented by bigintegers – stored in the array, there is no reason to show to the user all of them. After key generation also two textareas in the decryption section are set. These represent two parameters of the private key *n* and *d.* With the keys generated user is ready to make the encryption. Encryption not only applies RSA encryption but also hashes cryptogram with AES cryptography system.

Like in the case of Merkle-Hellman implementation, user cannot specify cryptogram, he or she is able to use ciphertext generated in the encryption process before. User should copy encrypted message from the textarea above and put into textbox in the decryption form. To get back plain text user must run the decryption function by clicking ‘Decrypt’ button. Algorithm from the function retrieves original text message and also puts them into separate textbox.

RSA practical site generates five different information about safety which depends on the chosen key length. All five key lengths may be successfully used but shorter ones may be broken now or in the near future. The longest ones are considered as one hundred percent safe. Section safety appears at the bottom of the site after each encryption.

### Possible algorithm modifications

In the further development RSA implementation May be extender by few improvements. First of all the biggest drawback in the solution provided in the webapplication, user is not able to provide own *p* and *q* parameters. Using complex random key generation algorithm which does not allow to choose both parameters is unfortunately imposed. On the one hand provided solution shows to user cipher which can be used to provide enormous level of security but on the other hand there should be a possibility to show very trivial example which could be checked with a pen and piece of paper. It is not possible to modify given RSA algorithm. There should be added totally new version adjusted to that purpose. Implementation given by the Cryptico.js library has got also some limitation which can be improved. First of all using additionally key and message hashing by two SHA256 and AES cryptography methods should be optional. They give additional level of security but there should be a possibility to make ciphering without them. Next drawback of used solution is hardcoded value of *e* coefficient in public key. In Tom Wu’s algorithm this value is always constant and is equal 65537. Making it non-constant it would assure higher flexibility to this cryptography method.

* + 1. Digital signature implementation

Digital is the last topic in the practical part of the application. Digital signature was next important area of cryptography which had to be covered in the thesis. The idea of signature emerged at the time when asymmetric cryptography was proposed by Whitfield Diffie and Martin Hellman as the possible application of their invention. It is realised again with the RSA cryptography method and the same Cryptico.js library. Moreover in this site user can combine both processes: encryption and message signing, to show very important aspect – proper order of both processes. Digital signature is extended version of practical site about RSA cipher.

To perform both processes there is a need to create two pairs of keys for sender to create signature and second pair for receiver to perform a ciphering. As the first should be signature added to the plaintext there is sender key generation form put at the top of the site. Algorithm for key generation requires passphrase. This is taken as a parameter in Tom Wu’s algorithm which creates pair of keys. Next important parameter is length of the key which can chosen from five possible values. With these two things set triggers flag which enable button ‘Generate key’ which runs the algorithm. Signature is shown in the special textarea. Next part is the exact repetition of the form used in the RSA cryptography system presentation site. Next difference are two extra textareas which are very significant in the process of decryption which this time is extended with signature verification. ‘Decrypt’ button calls function responsible for both these processes. At the beginning Encrypted message is being decrypted and after that signature is taken from the message and compared to key public id. First textarea presents signature from the message and the last textbox signals the user if the signature is verified and message was not altered during the transmission or someone stole and changed the secret text. Thanks to that site user is able to get overall view at message securing.

# JavaScript Cryptography Library

Functions and implemented algorithms

……………………………………………………………. ………………… ……………….

Possible modifications

……………………………………………………………. ………………… ……………….

* 1. xxx

……………………………………………………………. ………………… ……………….

* 1. xxx

……………………………………………………………. ………………… ……………….

* 1. xxx

……………………………………………………………. ………………… ……………….

### xxx

……………………………………………………………. ………………… ……………….

### xxx

……………………………………………………………. ………………… ……………….

### xxx

……………………………………………………………. ………………… ……………….

### xxx

……………………………………………………………. ………………… ……………….

* 1. xxx

……………………………………………………………. ………………… ……………….

### xxx

……………………………………………………………. ………………… ……………….

### xxx

……………………………………………………………. ………………… ……………….

### xxx

……………………………………………………………. ………………… ……………….

# The web application supporting encryption algorithms

……………………………………………………………. ……….…………………

* 1. Graphical user interface

Graphical user interface (GUI), is a type of user interface that allows users …

* 1. xxxx

……………………………………………………………. ……….………………… ……………….

### xxx

……………………………………………………………. ………………………… ……………….

### xxx

……………………………………………………………. ….. ………………… ……………….

### xxx

……………………………………………………………. ………. ………………… ……………….

* 1. xxx

……………………………………………………………. …….. …………………

### xxx

……………………………………………………………. ……….. ………………… ……………….

### xxx

……………………………………………………………. ……… ………………… ……………….

### xxx

……………………………………………………………. ….. …………………… ……………….

# Summary

……………………………………………………………. ………………… ……………….

# Appendix

Source code…

# Bibliography

Badley, Linda. *Writing Horror and the Body. The fiction of Stephen King, Clive Barker and Anne Rice.* Westport: Greenwood Press, 1966.

Benefiel, Candace E., “Blood Relations: The Gothic Perversion of the Nuclear Family in Anne Rice's *Interview with the Vampire*.” *The Journal of Popular Culture,*Vol. 38, No. 2 (2004): 261-273.

Hoppenstand, Gary, and Ray B. Browne. “Vampires, Witches, Mummies, and Pother Charismatic Personalities: Exploring the Anne Rice Phenomenon.” *The Gothic World of Anne Rice.* ed. Hoppenstand, Gary, and Ray B. Browne. Bowling Green: Bowling Green State University Popular Press, 1996.

Ingebretsen, Edward J. “Anne Rice: Raising Holy Hell, Harlequin Style”. *The Gothic World of Anne Rice.* ed. Hoppenstand, Gary, and Ray B. Browne. Bowling Green: Bowling Green State University Popular Press, 1996.

Schneier, Bruce „Kryptografia dla praktyków”, WNT 2002 2.

Schneier, Bruce „Applied Cryptography. Protocols, Algorithms and source code in C".

Menezes, Alfred J., van Oorschot, Paul C., Vanstone, Scott A. „Handbook of Applied Cryptography”, CRC Press,1997. (Kryptografia stosowana, WNT 2005.

Stallings, William „Ochrona danych w sieci i intersieci w teorii i praktyce”, WNT, Warszawa 1997.

Stallings, William „Cryptography and network security: principles and practice”, Pearson/Prentice Hall, 2011.

Helen Fouche Gaines' Cryptanalysis: A Study of Ciphers and Their Solution.

Klander, L. „Hacker proof czyli jak się bronić przed intruzami" Mikom Warszawa 1998.

Mitnick, Kevin D., Simon, William L., and Wozniak, Steve „The Art of Deception: Controlling the Human Element of Security”, Wiley, 2003.

Mitnick, Kevin, and Simon, William „Sztuka podstępu. Łamałem ludzi, nie hasła”, Helion 2003.

Stallings, William „Cryptography and network security: principles and practice”, Pearson/Prentice Hall, 2011

Gaines, Helen Fouche „Cryptanalysis: A Study of Ciphers and Their Solution.

Swenson, Christopher „Modern Cryptanalysis: Techniques for Advanced Code Breaking”

Ferguson, Niels „Cryptography Engineering: Design Principles and Practical Applications”

**Web resources**

**Abstract**

**Keywords: cryptology, cryptography, safety, cipher, webapplication**

Załącznik

do Regulaminu Studiów w PŁ

z dnia 29 marca 2006 roku

Łódź, dnia …………………… r.

Paweł Grzegorzewski

(IMIĘ I NAZWISKO STUDENTA)

190116

(NR ALBUMU)

Computer Science

(KIERUNEK STUDIÓW)

Inżynierskie, stacjonarne

(RODZAJ I FORMA STUDIÓW)

OŚWIADCZENIE

Świadomy odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że przedkładana praca magisterska / ~~inżynierska~~ / ~~licencjacka~~ (\*)na temat:

*„The web application supporting encryption algorithms   
– Does cryptography provide security?”*

*Aplikacja internetowa wspierająca algorytmy szyfrowania   
– czy kryptografia zapewnia bezpieczeństwo?*

została napisana przeze mnie samodzielnie.

Jednocześnie oświadczam, że ww. praca:

* nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz. U. z 2000 r. nr 80, poz. 904, z późniejszymi zmianami) oraz dóbr osobistych chronionych prawem cywilnym, a także nie zawiera danych i informacji, które uzyskałem w sposób niedozwolony,
* nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów wyższej uczelni lub tytułów zawodowych.

Jestem także świadomy, że praca zawiera rezultaty stanowiące własność intelektualną Politechniki Łódzkiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Uczelni.

……………………………………………………………

(PODPIS STUDENTA)

(\*) - niepotrzebne skreślić