Journal Plus Education, ISSN: 1842-077X, E-ISSN (online) 2068-1151 Vol XX (2018), No. 2. pp. 13-23

# HOW TO TEACH THE HISTORY OF CRYPTOGRAPHY AND STEGANOGRAPHY

**Gábor Kiss[a], Carlos Arturo Torres Gastelú[b]**
**[a]Óbuda University, [b]Universidad Veracruzana**

**Abstract:** *Students of the undergraduate course Computer Science get acquainted with the most prevalent methods of Steganography and Cryptography in history as well as with up to date applications. To teach Cryptography and Steganography is complicated, because you need to plot figures and tables at the blackboard plus calculating the frequency of characters in the text is consuming a lot of work and time. Concealing information within pictures or voices is impossible to show on the blackboard and you can only partially explain how to apply the LSB (least significant bit) technique. Therefore, we have devised some applications which are appropriate to show all these algorithms. We wanted to know the usefulness of all the programs we wrote. The safety technology engineering students attended my lectures held at different times in two groups. In the first group of lectures I presented and used the programs described above while the lectures for the second group were delivered without these materials. Our hypothesis was that a group where we used the developed multimedia applications got better marks in the papers written than the other. An analysis of the results (Man-Whitney test) showed significant difference in paper results of the two groups (p<0,05). This showed: the paper results of students attending the multimedia lectures were better by one mark than the results of the other group where lectures were delivered without multimedia presentations. We can say the using of multimedia applications when teaching cryptography and steganography is productive, and the students understand the methods easier, and get better result when writing papers.*

**Keywords:** cryptography; didactical methods; higher education; teaching*;*

## 1. Introduction

Students of the undergraduate course of Computer Science get acquainted with the development of the computers, printers, data storage devices and of course the most prevalent methods of steganography and data encryption in history (Kahn, 1996). Most of these are difficult to demonstrate at the blackboard not to speak of modern steganographic methods which are generally not applicable in the classroom at all. Concealing information within pictures or voices is impossible to show on the blackboard and you can only partially explain how to apply the LSB (least significant bit) technique. Therefore we decided to create multimedia applications suitable to conceal text in BMP or WAV files not visible or detectable for human eyes and ears, respectively. First we wrote a Borland Delphi program appropriate to show up to date steganographic (data concealing) methods using picture and sound files. This is not only demonstrating the use of modern steganography, but the result is visible and/or audible while helping students to take in new knowledge. To the topic of cryptography we prepared model applications presenting historic cryptographic algorithms as well as applications demonstrating how to decipher these encrypted data (e.g. using and deciphering the Ceasar code, monoalphabetic encryption and it's deciphering, etc.). Another program demonstrating the Vigenére encryption shows continuously which rows and columns are used in the process. Illustrating the Cardano grid a program was written which prepares a rotatable grid adequate to put the characters of the message to the appropriate place in the grid in order to conceal it. Now we wanted to know the usefulness of all the programs we wrote and an excellent opportunity presented itself by the big number of admitted new students (76 students). These safety technology engineering engineering students attended my lectures held at different dates in two groups. In the second group of lectures (42 students) we presented and used the programs described above while the lectures for the first group (34 students) were delivered without these materials.

First of all, we need to look at how human memory is working (Bloom, 1969), the taxonomy of learning, teaching, assessing (Anderson, 2001) the levels of learning to guide the students through the process of learning (Hoffmann, 2011). The multi-representational learning environments can support learning in many different ways (Ainsworth, 1999).

The hypothesis was that the group where we used the developed multimedia applications would achieve better results in the papers. Start typing the body of your paper here. Papers will outline the issue addressed and research questions, the literature and background to the topic, the analytical frame, the methodology and the research results.

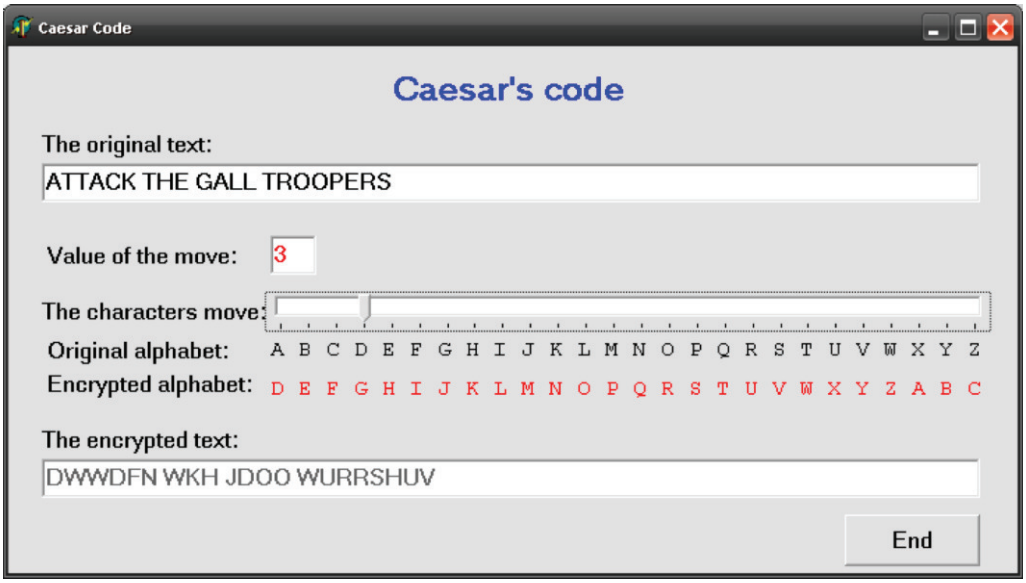
## 2. Cryptography

### *2.1. The Caesar's code*

First of all I want to give some information about the developed applications and about the background, how the methods works (Kahn, 1996).

The Caesars' code had been Julius Caesar who used it first in the time of the Gall wars and this way he asked help from Cicero.

Using the software students can see how the Caesar algorithm works, how the decryption goes, or how to brake the encrypted text (Figure 1.).



Caesar's code

The original text: ATTACK THE GALL TROOPERS

Value of the move: 3

The characters move:

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

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

The encrypted text: DWWDFN WKH JDOO WURRSHUV

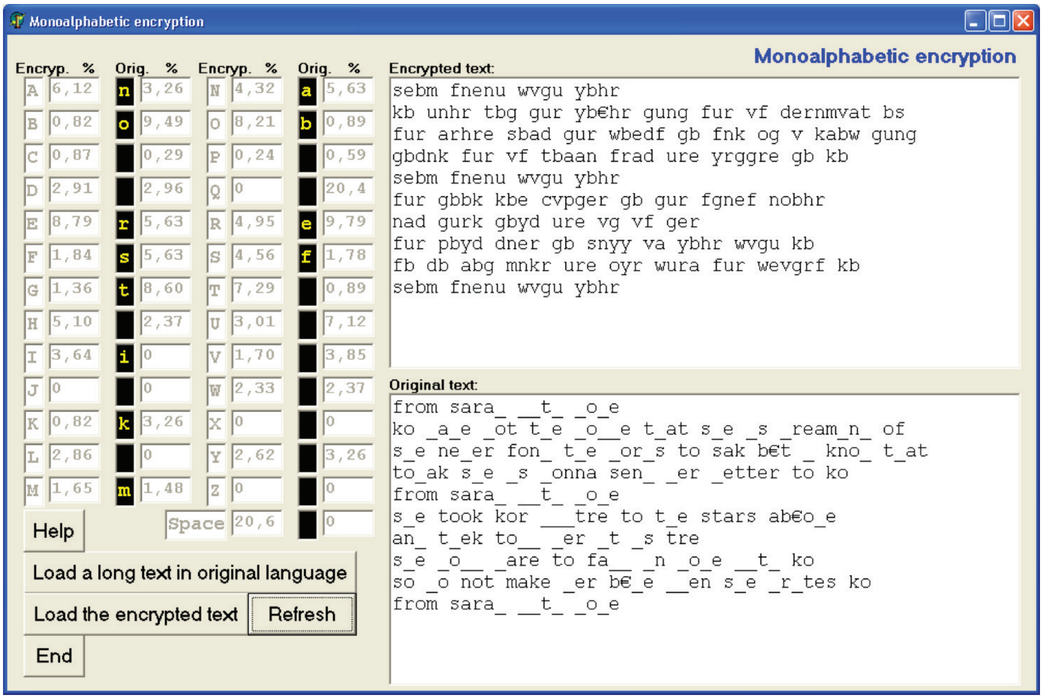
Figure 1. The Caesars' code.

So students understand easier the method because they see on the screen the text and how to produce the encrypted text. They type in the text and give the number of characters to move in the alphabet. (E.g. if this value is 3, then replace A with D, B with E, etc.). To brake a text encrypted with Caesar code is easy, because we must try 25 move in the alphabet at most, so this is not a safe encryption.

### *2.2. The Monoalphabetic Substitution*

The Monoalphabetic Substitution algorithm is much better, because we can construct freely the unique pairs of characters. The value of pairing possibilities is 26!, which is quite a big number, it's value being about $400x10^{24}$. If we would like to brake and try one in a second, it would take $128x10^{17}$ years. Through the middle ages in Europe people believed this encryption method unbreakable and was widely used in correspondence. The pen-pals have to know exactly which character stays for which in the original.

Although this algorithm seems to be safe, it is easily breakable either. Historically arab mathematicians in the 9th century found that the occurrence of some characters in a language is higher than others and they worked out a method how to brake a text encrypted with monoalphabetic encryption.


Monoalphabetic encryption
Monoalphabetic encryption
Help
Load a long text in original language
Load the encrypted text
Refresh
End

| Encryp. | % | Orig. | % | Encryp. | % | Orig. | % |
|---|---|---|---|---|---|---|---|
| A | 6,12 | n | 3,26 | N | 4,32 | a | 5,63 |
| B | 0,82 | o | 9,49 | O | 8,21 | b | 0,89 |
| C | 0,87 |  | 0,29 | P | 0,24 |  | 0,59 |
| D | 2,91 |  | 2,96 | Q | 0 |  | 20,4 |
| E | 8,79 | r | 5,63 | R | 4,95 | e | 9,79 |
| F | 1,84 | s | 5,63 | S | 4,56 | f | 1,78 |
| G | 1,36 | t | 8,60 | T | 7,29 |  | 0,89 |
| H | 5,10 |  | 2,37 | U | 3,01 |  | 7,12 |
| I | 3,64 | i | 0 | V | 1,70 |  | 3,85 |
| J | 0 |  | 0 | W | 2,33 |  | 2,37 |
| K | 0,82 | k | 3,26 | X | 0 |  | 0 |
| L | 2,86 |  | 0 | Y | 2,62 |  | 3,26 |
| M | 1,65 | m | 1,48 | Z | 0 |  | 0 |
|  |  |  |  | Space | 20,6 |  | 0 |

Encrypted text:

```
sebm fnenu wvgu ybhr
kb unhr tbg gur yb€hr gung fur vf dernmvat bs
fur arhre sbad gur wbedf gb fnk og v kabw gung
gbdnk fur vf tbaan frad ure yrggre gb kb
sebm fnenu wvgu ybhr
fur gbbk kbe cvpger gb gur fgnef nobhr
nad gurk gbyd ure vg vf ger
fur pbyd dner gb snyy va ybhr wvgu kb
fb db abg mnkr ure oyr wura fur wevgrf kb
sebm fnenu wvgu ybhr
```

Original text:

```
from sara_ __t_ _o_e
ko _a_e _ot t_e _o__e t_at s_e _s _ream_n_ of
s_e ne_er fon_ t_e _or_s to sak b€t _ kno_ t_at
to_ak s_e _s _onna sen_ _er _etter to ko
from sara_ __t_ _o_e
s_e took kor ___tre to t_e stars ab€o_e
an_ t_ek to__ _er _t _s tre
s_e _o__ _are to fa__ _n _o_e __t_ ko
so _o not make _er b€_e __en s_e _r_tes ko
from sara_ __t_ _o_e
```

Figure 2. The Monoalphabetic Substitution.

For using the method of frequency analysis: we need to know which language the text was written in, and we need a long (cca. 3-4 pages) text in this language. From the long original text we can calculate the frequency of characters and this will be the typical attribute of language. Now we can start to brake the encrypted text. We need to know which character frequency is the highest in the encrypted text and we replace it with the highest one in the original language.

If the encrypted text has got no spaces between words, than the most frequent character will be the space, simply because this is true in all of the known languages. If it works, we inspect how long the words are, which is a good starting point. In the next step we are looking for the second most frequent character in the encrypted text and exchange it with the second most frequent in the original language, and so on. After having changed a few characters, with some luck we will recognize part of words and guess the absent characters (Figure 2.).

At the end we can determine which characters are exchanged for which.

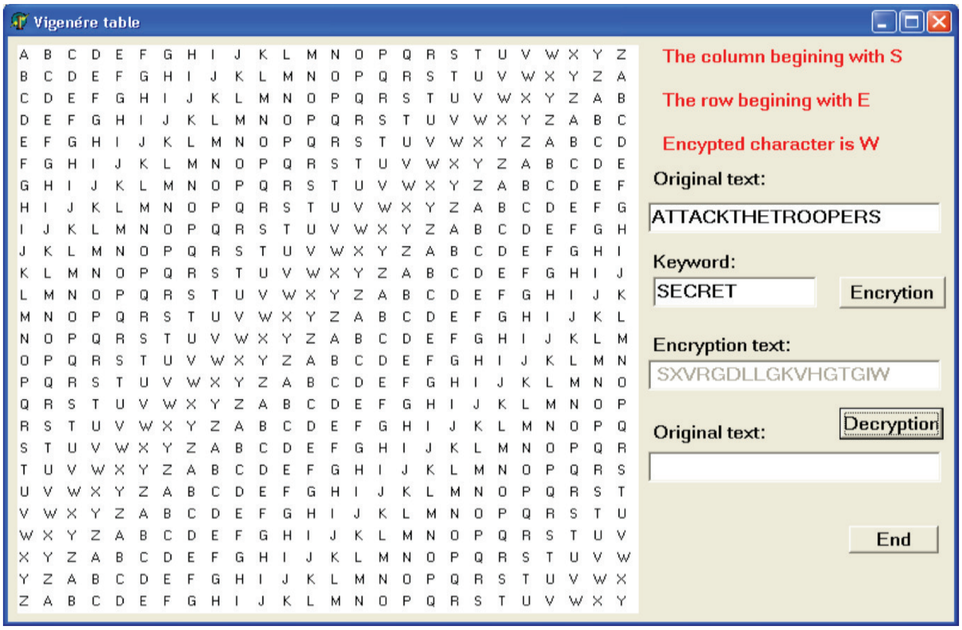
This method of decryption was not known in Europe up to the 12-13th century, so up to 400-500 years arabs could easily read european letters.

With my software students load the original text, calculate the frequency of characters and load the encrypted text. Then they can change the characters and read the decrypted text. They can also easily enrcypt texts with this software too.

### *2.3. The Vigenère table*

As we have seen earlier, the monoalphabetic substitution is not safe. Using always the same character in the encrypted text instead of the original can be easily braken with frequency analysis.

In Europe Blaise de Vigenère worked out a new encryption method. To use it, we need a Vigenère table with 26 rows. Each row holds the alphabet, but always moved by a character (Figure 3.).



Vigenére table

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

The column begining with S

The row begining with E

Encypted character is W

Original text: ATTACKTHETROOPERS

Keyword: SECRET

Encryption text: SXVRGDLLGKVHGTGIW

Original text:

Figure 3. The Vigenère table.

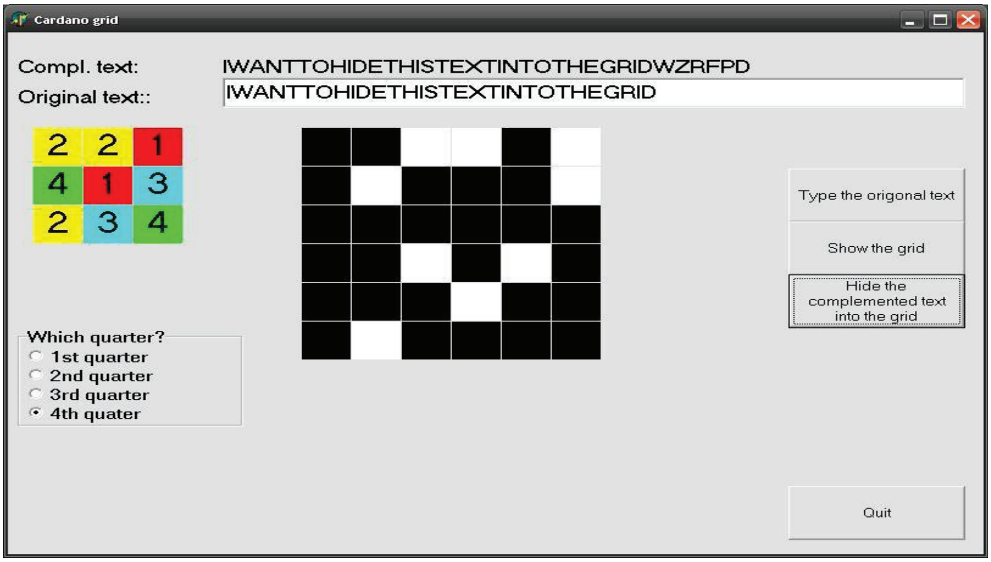
If we want to encrypt a text, we need a few freely chosen rows from it. It's easier to remember a keyword constructed from the first characters of these rows (for example KEY stays for the 10th, 4th and the 24th rows). This keyword written over the original text we get which row encrypts which original character. The "D" will be encrypted with the row beginning with "K". The encrypted character is sitting at the intersection of this row and the column beginning with "D" which is giving the character "N". This will be the first character of our encrypted text. In the next step we take the column beginning with "E" and use the row beginning with "E". This will give us the encrypted character "I". Step by step we will get the encrypted word: NICORB.

Looking closer at this word we will notice the two "E" from "DEFEND" are replaced by different characters in the encrypted text so frequency analysis breaks down. In order to decrypt the text we need the keyword.

With this software students can encrypt and decrypt texts and see decryption with other keywords is impossible.

### *2.4. The Cardano grid*

The Cardano grid is rotatable grid. It is possible to put the characters of the message to the appropriate place in the grid in order to conceal it. This technique is the part of geometric cryptography, because the characters of the original text will lose the original position and in the encrypted text we can find them in another position. We need to know first, how many characters we want to encrypt. In the second step we have to complement the original text with random characters, because we will use a square grid to encrypt the text so the length of the text must be squarenumber. After that we can mark some positions in the grid, where we can put the characters of the original text (Figure 4.).



Compl. text: IWANTTOHIDETHISTEXTINTOTHEGRIDWZRFPD

Original text:: IWANTTOHIDETHISTEXTINTOTHEGRID

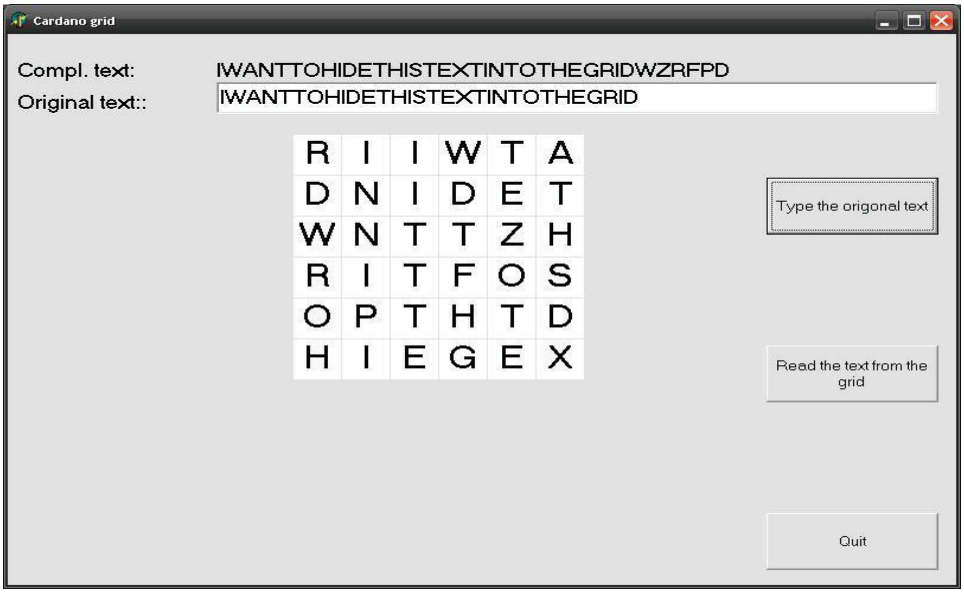
| | | |
|---|---|---|
| 2 | 2 | 1 |
| 4 | 1 | 3 |
| 2 | 3 | 4 |

| | | | | | |
|---|---|---|---|---|---|
| ■ | ■ | □ | □ | ■ | □ |
| ■ | □ | ■ | ■ | ■ | □ |
| ■ | ■ | ■ | ■ | ■ | ■ |
| ■ | ■ | □ | ■ | □ | ■ |
| ■ | ■ | ■ | □ | ■ | ■ |
| ■ | □ | ■ | ■ | ■ | ■ |

Which quarter?
○ 1st quarter
○ 2nd quarter
○ 3rd quarter
● 4th quater

Figure 4. The Cardano grid.

If we have used all of the marked positions, we can rotate the grid with 90 degrees and we will see other free positions to write characters into. We can to rotate the grid again and again with 90 degrees to use all position of the grid and encrypt the original text (Figure 5.). We can decrypt the text, if we know the positions where we started to write the original text.



Cardano grid

Compl. text: IWANTTOHIDETHISTEXTINTOTHEGRIDWZRFPD

Original text:: IWANTTOHIDETHISTEXTINTOTHEGRID

| | | | | | |
|---|---|---|---|---|---|
| R | I | I | W | T | A |
| D | N | I | D | E | T |
| W | N | T | T | Z | H |
| R | I | T | F | O | S |
| O | P | T | H | T | D |
| H | I | E | G | E | X |

Figure 5. Cardano grid with the marked positions.

## 3. Steganography

Steganography means concealing information. For example we can mark some characters in the text of a newspaper with a little point, then reading the marked characters we get the secret information. Or we can write the important characters 1 point higher in a word processor and so on.

The LSB (least significant bit) technique is the base of the modern steganography. We can to conceal information with this method in a .WAV audio file or in a .BMP in picture file. This two file types are not compressed so it is easier to modify their contents. In a 24 bit BMP files all picture points have three bytes to hold the colour of these points (Red, Green, Blue). In WAV files we also find bytes to show the pitch in the left or right side. We can use the LSB technique to conceal a text using the last 1-2 bits of these bytes. This modification of the file is not visible or detectable for human eyes and ears. We can use 4 or 8 bits too, but in this case the file would change perceptible.

If we want to conceal a text in a picture or in an audio file, we need first the ASCII code of the characters in the text. After that we decide how many bits will we use to hide the text in the file. We can easily calculate how many characters can we conceal in a BMP file or in a WAV file. This depends on the size of the file used. If the resolution of a BMP picture is 640x480 and it uses 24 bits to store the colour (3 bytes for the RGB colour) of the picture points, the picture will have 640x480x3 bytes which we can use. If we use the last bit to write the secret information into the picture we need 8 bytes to conceal a character (Table 1). Which means we can to hide 115.200 characters in the picture.

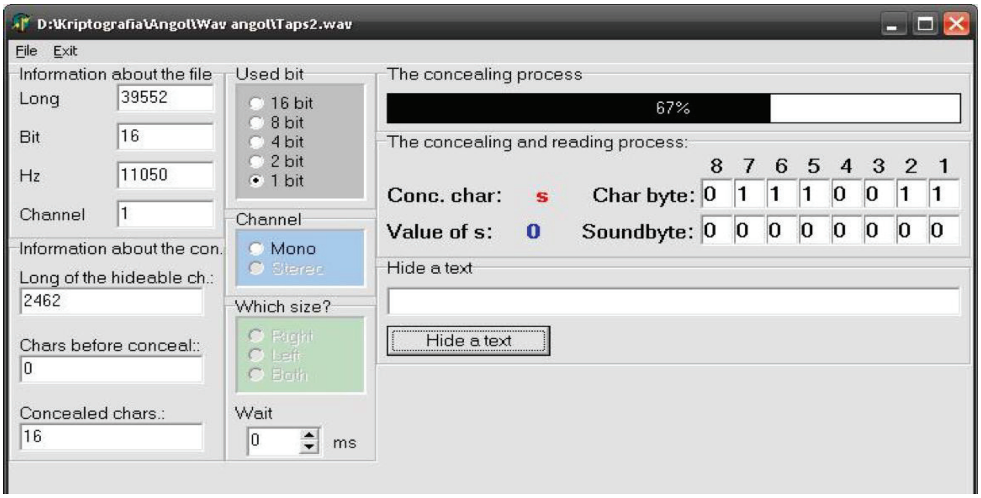
Table 1. Using the last bit to conceal information

| Original byte | Modified byte | ASCII code of „A“ |
|---|---|---|
| 01010100 | 01010100 | 0 |
| 11100110 | 1110011**1** | 1 |
| 01101101 | 0110110**1** | 0 |
| 11011010 | 11011010 | 0 |
| 01101000 | 01101000 | 0 |
| 11000101 | 1100010**0** | 0 |
| 11001110 | 11001110 | 0 |
| 10010010 | 1001001**1** | 1 |

According to the table generally about half of the bytes was changed on average. For example: if we want to save a 0 and the original bit is zero too, we would not make any changes.

If we use 2 bits from a byte to conceal the information we can “write” a text twice longer into a picture or the audio file. Of course we can use 4 bits to conceal a longer text, but in this case the change of the bytes’ value will be big enough to detect it with human eyes and/or ears. Though, in this case we can "write" a whole book in a picture if the resolution is high enough.

The developed applications are able to read the information hided in WAV or a BMP files, and show how many character can be "written" in (Figure 6).



Information about the file

Long: 39552

Bit: 16

Hz: 11050

Channel: 1

Information about the con.

Long of the hideable ch.: 2462

Chars before conceal:: 0

Concealed chars.: 16

Used bit: ○ 16 bit ○ 8 bit ○ 4 bit ○ 2 bit ● 1 bit

Channel: ○ Mono ○ Stereo

Which size? ○ Right ○ Left ○ Both

Wait 0 ms

The concealing process: 67%

The concealing and reading process:

|  |  | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| Conc. char: s | Char byte: | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| Value of s: 0 | Soundbyte: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Hide a text

Figure 6. Concealing the information in a WAV file.

We can choose the channel (audio) or the colour (picture) in which we want to conceal the text and when modifying the files. After the process we can look at or listen to both of the original and the modified file trying to detect the changes. Of course we can also read out the text from the modified files.

## 4. Statistical analysis of the paper results in two groups

In the second group of lectures (group 2) I presented and used the programs described above while the lectures for the first group (group 1) were delivered without these materials. We needed some mathematical analyzing to decide if the use of these programs helpful or not in understanding the lectures.

The students had to write two papers in the semester. The learning material of the first one was the history of Computer Science and the second one based on the history of Cryptography and Steganography. According to the table (Table 2) the mean of the results of papers of group 2 is higher.

Table 2. Group statistics

| Paper | Group | Mean | Std. deviation |
|---|---|---|---|
| 1 | 1 | 1.97 | 0.90 |
| 1 | 2 | 2.16 | 0.85 |
| 2 | 1 | 2.03 | 0.99 |
| 2 | 2 | 2.62 | 1.01 |

My null hypothesis was that the group where we used the developed multimedia applications would achieve better results in the papers. We have two independent samples so we can apply the Mann-Whitney-Wilcoxon test for 2 samples (Mann, 1947).

So we used the Mann-Whitney independent sample U test of SPSS to tell if the means of these groups are differing or not. Monitoring was held on p=5 percent significancy level in the whole analyzing process.

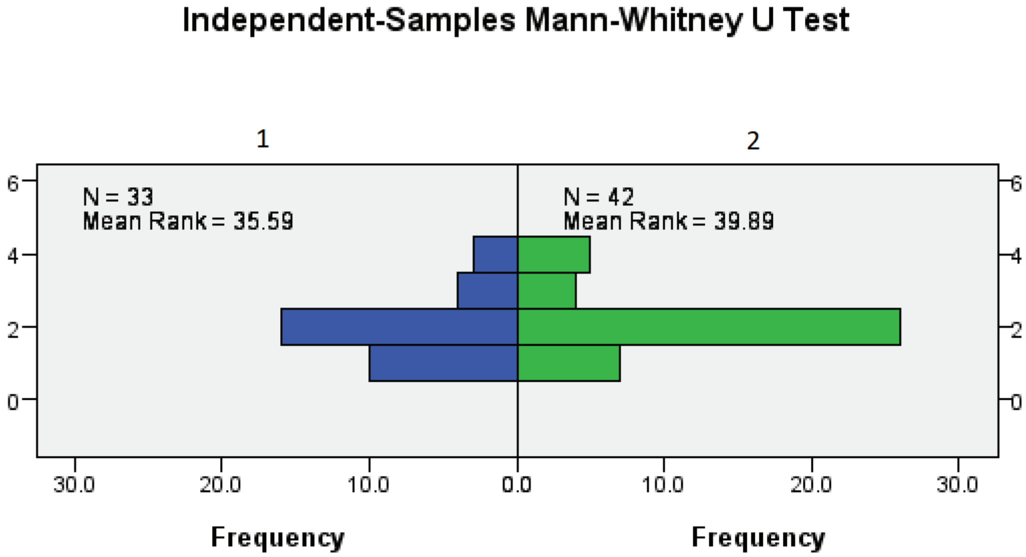


Figure 7. The frequency of the first paper results.

The result of the Mann-Whitney U test has not shown difference between the 2 groups by the first paper, p=0.346. This means the randomly built groups did not show any difference in knowledge level in topic history of Computer Science, the frequency of the paper results are the same (Figure 7).

The situation is different in topic history of Cryptography and Steganography. In this case the result of the Mann-Whitney U test has shown significant difference between the 2 groups by the second paper, p=0.007. This means the results of students attending the multimedia lectures were significant better by half mark than the results of the other group where lectures were delivered without multimedia presentations. The using of the self-developed programs had influence on the results of papers of the applied computer science engineer students (Figure 8).

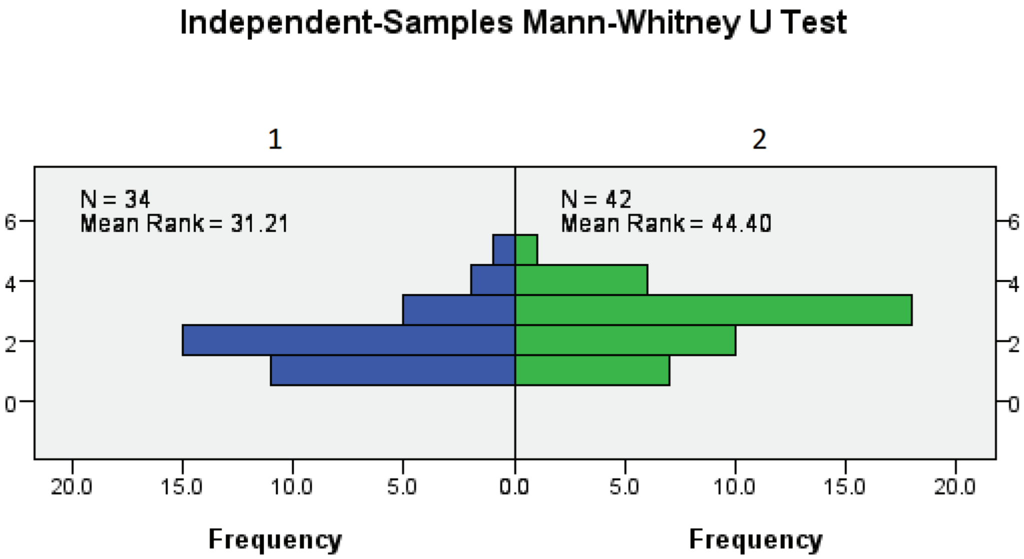


Figure 8. The frequency of the second paper results.

## 4. Conclusion

The hypothesis in the introduction was that a group where we used the developed multimedia applications got better marks in the papers written than the other. Now we know our hypothesis was correct.

We can say the using of the self-developed multimedia applications when teaching cryptography and steganography for the safety technology engineering students is productive, and the students understand the methods easier, and get better result when writing papers.

## References

Ainsworth, S. (1999): A Functional Taxonomy of Multiple Representations, *Computers & Education*, **33**, 2, 131-152

Anderson, L.W. , Krathwohl, D. R. and Bloom, B. S. (2001): *A taxonomy for learning, teaching and assessing: A revision of Bloom's taxonomy of educational objectives*, Pearson

Bloom, B. S. (1969): *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Addison-Wesley Longman Ltd.,

Hoffmann, M.H.W. (2011): When to Assess Knowledge, Skills and Competences, *Proceeding of the 22th EAEEIE Annual Conference*, Maribor, Slovenia, 1-5.

Kahn, D. (1996): *The Codebreakers*. Scribner, New York.

Mann, H.B. and Whitney, D.R. (1947): On a Test of Whether one of Two Random Variables is Stochastically Larger than Other, *Annals of Mathematical Statistics*}, **18**, 1, 50-60