Cryptography in the Basic Computer Science

Gábor Kiss

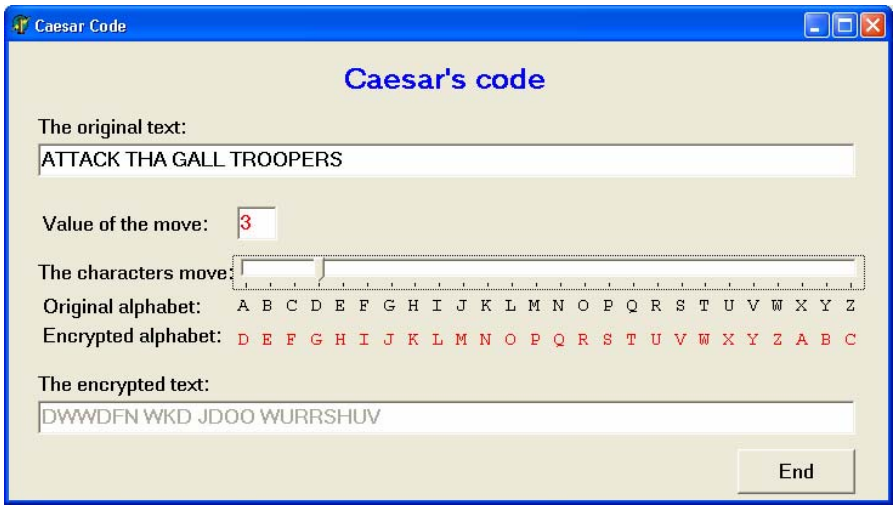
Senior Teacher of Computer Science
Budapest Tech, Budapest, Hungary
kiss.gabor@bgk.bmf.hu

Abstract: In the subject „Basic Computer Science for Managers" I teach different encrypted algorithms, which had been used in ancient times, in the middle ages and in modern history too (e.g. Caesar code, monoalphabetic encryption, Vigenere-table, etc.) and I teach also their decryption.
To teach Cryptography 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.
Therefore I have devised a software which is appropriate to show all these algorithms.

1 The Caesar's Code

Why is it called Caesar's code? Because it 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 the student can see how the Caesar algorithm works, how the decryption goes, or how to brake the encrypted text. So the students will easier understand the method because they will see on the screen the text and how to produce the encrypted text. They will type in the text and give the number the characters in the alphabet must move (Picture 1). (E.g. if this value is 3, then I can replace A with D, B with E, etc.). To brake a text encrypted with Caesar code is easy, because we must try 25 move of the alphabet at most, so this is not a safe encryption.



Picture 1

2 The Monoalphabetic Encryption

This algorithm is much better, because we can construct freely the unique pairs of characters. The value of pairing possibilities is 26!, what is a quite big number, it's value being about $400*10^{24}$. If we would like to brake and try one in a second, it would take $128*10^{17}$ years. Through the middle ages in Europe people believed this encryption method unbrakable 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 4th 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.

The method of frequency analysis:

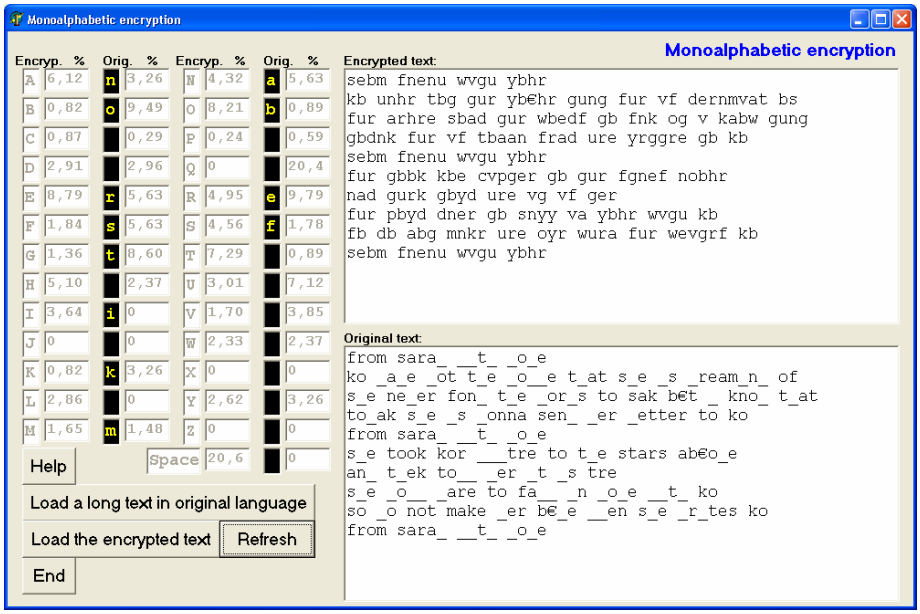
Thereunto we need to know which language the text was written in, and we need a long (cca 3-4 pages) text in this language. From the long original text we can calculate the frequency of characters and this will be the typical attribute of language. Now can we start to brake the encrypted text. We need 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 will to see 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 encypted text and exchange 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 (Picture 2).

At the end we see which charaters are exchanged for which.

This method of decryption was not known in Europe up to the 9th century, so up to 500 years arabs could to 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 decrypred text. They can easily a enrcypt texts with this software too.



Picture 2

THE VIGENÈRE TABLE:

As we have seen earlier, the monoalphabetic encryption 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:

original	**ABCDEFGHIJKLMNOPQRSTUVWXYZ**
1	BCDEFGHIJKLMNOPQRSTUVWXYZA
2	CDEFGHIJKLMNOPQRSTUVWXYZAB
3	DEFGHIJKLMNOPQRSTUVWXYZAAC
4	EFGH**I**JKLMNOPQ**R**STUVWXYZAACD
.	
9	JKLMNOPQRSTUVWXYZAACDEFGHI
10	KLM**NO**PQRSTUVWXYZAACDEFGHIJ
11	LMNOPQRSTUVWXYZAACDEFGHIJK
.	
23	XYZABCDEFGHIJKLMNOPQRSTUVW
24	YZA**BC**DEFGHIJKLMNOPQRSTUVWX
25	ZABCDEFGHIJKLMNOPQRSTUVWXY
26	ABCDEFGHIJKLMNOPQRSTUVWXYZ

If we want to encrypted 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).

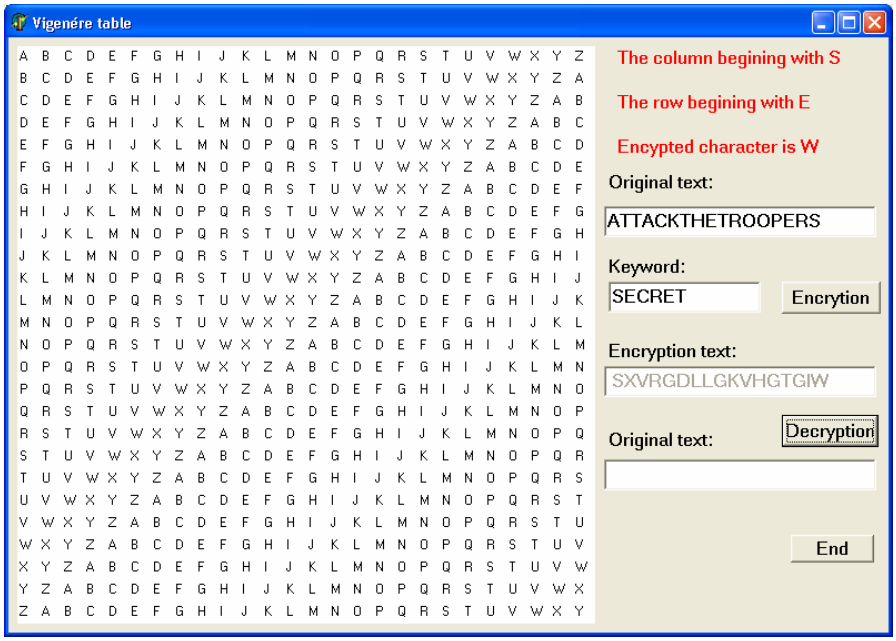
This keyword written over the original text we get which row encrypts which original character:

Keyword:	KEYKEY
Original text:	DEFEND
Encrypted:	NICORB

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 (Picture 3). 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 breakes down. To decrypt we need the keyword.

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



Picture 3

Conclusion

Using the software also helps to see how these methods work. The students will understand easier quicker the encrypting algorithms resulting in better exams.

References

[1] Simon Singh: Codekönyv. Magyar Könyvklub, 2002

[2] David Kahn: Code berakers, 1999