Modeling, simulation and visualization of automatic cryptoanalysis of the short monoalphabetical substituted cipher text

S. Hubalovsky, P. Hanzalova

***Abstract*—** One of the most important methods in current scientific and technological research as well as in research of strategy algorithm and programming is modeling and computer simulation of real systems and real processes. System approach, modeling and simulation are discipline with its own theory and research methodology. Modeling and simulation can be understand as one of the most important method in teaching of algorithm development and programming is to use a suitable method of developing theoretical knowledge of algorithm development and programming.

The paper focuses to the theory of modeling, simulation and visualization of solution of real process – *automatic decryption of very short monoalphabetical substituted cipher text*.

The solution is demonstrated step by step – it starts with the problems definition, then the strategy solution analyses are shown and finally computer simulation in MS Excel Spreadsheet is shown.

***Keywords*—** Algorithm development, Cryptoanalysis, Monoalphabetical substitution cipher, System approach, Trigram.

I. Introduction

COURSES of the algorithm development and programming are currently provided in the Czech Republic at secondary schools, technical colleges and universities in the subjects focusing on informatics. The ability to create algorithms develops logical thinking skills and imagination and is an inseparable part of study skills of prospective and undergraduate teachers specializing in "Informatics" at the Faculty of Education.

Algorithms are encountered in all practical activities without being realized. An algorithm generally involves defining the rules and giving the sequences of steps necessary for any activities. The most common examples of algorithms in everyday life can be found e.g. in [1], [2], [3], [4] or [5].

The expression of the algorithms in the form of schemes and figures - flowcharts can be hardly understandable for the beginners. An abundance of particular shapes can be confusing.

In this paper we introduce a case study illustrating step by step process problem solution, strategy and algorithm development and computer simulation of a real process – *automatic decryption of very short monoalphabetical substituted cipher text* using the frequency analysis of the trigrams.

Stepan Hubalovsky is working at University of Hradec Kralove, Department of informatics, Faculty of Science, Hradec Kralove 500 38, Rokitanskeho 62, Czech republic, stepan.hubalovsky@uhk.cz.

Pavla Hanzalova is master student at University of Hradec Kralove, Department of informatics, Faculty of Science, Hradec Kralove 500 38, Rokitanskeho 62, Czech republic, pavla.hanzalova@uhk.cz.

II. Modeling and simulation of the systems and processes

A. Modeling

Modeling is a method that is often used in professional and scientific practice in many fields of human activity.

The main goal of modeling is not only describing the content, structure and behavior of the real system representing a part of the reality but also describing the processes.

The process can be understood as series of transformations that changes the input values to output values. From the system point of view the process is dynamic system in which the values of the characteristic of the system elements are changed under the influence of the external elements.

The models are always only approaching of the reality, because the real systems are usually more complex than the models are. The system homomorphism is applied in the process of modeling, which means that each element and interaction between the elements of the model corresponds to one element and interaction of the modeled real system or real process, but the reverse is not true. The model is always to be understood as simplification of the original. If the relation of isomorphism is between the model and real system the original model we could not distinguish between the model and the original, which is discussed e.g. in [6] and [7].

The first step in the process of computer simulation is creation of conceptual model of the studied real system / real process. Conceptual model can be represented in different way. The most used representations are:

- Mathematical equitation;
- Process charts.

Mathematical equitations establishes mathematical model of the studied real system. The model can be obtained either theoretically based on basic physical properties of the system, or numerically by means of the measured values. Determination of parameters of theoretical model developed from empirical data is called system identification.

Process charts establishes process model of the real process. The process models can be described by different way; the most common are flowcharts that described the algorithm of the modeled process.

The conceptual model must adequately describe the dependency system outputs on its inputs. Models of real process system will be shown in the following paragraphs of this paper.

B. Simulation

The process of modeling is closely related to the simulation. Simulation can be understood as process of executing the model. Simulation enables representation of the modeled real system or real process and its behavior in real time by means of computer. The simulation enables also visualization and editing of the model.

A typical simulation model can be written both through specialized programming languages that were designed specifically for the requirements of simulations, or the simulation model can be created in standard programming languages.

From the above considerations, it is clear that simulation is a process that runs on the computer. In some publications, therefore, can be found the term ”computer simulation“. It generally is valid that computer simulation is a computer-implemented method used for exploring, testing and analysis of properties of the conceptual (mathematical or process) models that describe the behavior of the real systems or real process which cannot be solved using standard analytical tools, se e.g. [8].

The simulation models represented by executable computer program have to be isomorphic with the conceptual model that is a representation. It means that the mathematical model and simulation model have to represent the real system, its elements, internal interactions and external interaction with the environment in the same way.

In our paper the real process *Automatic cryptoanalysis of the short monoalphabetical substituted cipher text* will be presented. This real process is simulated in *Visual Basic for Application and visualized in MS Excel spreadsheet*.

C. Significant function of the simulation

Simulation has from the scientific point of view several functions – see e.g. [8].

We will focus in this paper two of them and they are:

- replacing the real process;
- development of educational process.

1) Replacement of the real process

This is an important and indispensable feature of simulations and simulation model because it allows realize a situation of the process that cannot be investigated conventionally. The main advantage of simulations is that simulations model allows providing rather big number of the process steps in relatively short time, changing of input parameters and its visualization and optimization of the process.

2) Development educational process

The simulation is very useful from educational point of view. Using the simulation model and visualization of simulation results on the screen, students can better understand the basic features of the processes and systems and develop their intuition. It is also essential that the teaching by means of simulation is much cheaper and faster than the teaching carried by real experiment. In some cases providing the real experiment cannot be feasible.

D. Model verification and validation

Verification and validation are important aspects of the process modeling and simulation. They are essential prerequisites to the credible and reliable use of a model and its results [4].

1) Verification

In modeling and simulation, verification is typically defined as the process of determining if executable simulation model is consistent with its specification – e.g. conceptual model. Verification is also concerned with whether the model as designed will satisfy the requirements of the intended application. Verification is concerned with transformational accuracy, i.e., it takes into account simplifying assumptions executable simulation model. Typical questions to be answered during verification are:

- Does the program code of the executable simulation model correctly implement the mathematical model?
- Does the simulation model satisfy the intended uses of the model?
- Does the executable model produce results when it is needed and in the required format?

2) Validation

In modeling and simulation, validation is the process of determining the degree to which the model is an accurate representation of the real system / real process. Validation is concerned with representational accuracy, i.e., that of representing the real system / real process in the conceptual model and the results produced by the executable simulation model. The process of validation assesses the accuracy of the models. The accuracy needed should be considered with respect to its intended uses, and differing degrees of required accuracy may be reflected in the methods used for validation. Typical questions to be answered during validation are:

- Is the mathematical model a correct representation of the real system?
- How close are the results produced by the simulation executable model to the behavior of the real system?
- Under what range of inputs are the model’s results credible and useful?

Validation and verification are both ultimately activities that compare one thing to another. Validation compares real system / real process and conceptual model. Verification compares conceptual model and executable simulation model.

Sometimes validation and verification are done simultaneously in one process.

Validation of the conceptual model as well as verification of the simulation model of our real process – ***Automatic cryptoanalysis of the short monoalphabetical substituted cipher text*** – are to be done simultaneously by running simulation computer model.

The whole process of transformation from a real system, the simulation model and its visualization is shown in Fig. 1.

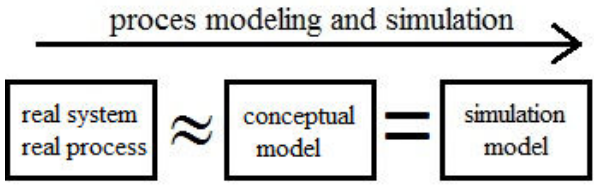

Fig. 1 Process modeling and simulation

Here again let us summarize that the mathematical model that reflects the real system / real process has some limitations and simplifying assumptions (the real system / process and conceptual model are in homomorphic relation).

In contrast, the simulation model is only the computer expression of the conceptual model (the conceptual model and simulation model are in isomorphic relationship).

E. Multidisciplinary approach

Another important benefit associated with the modeling and simulation of real processes is a multidisciplinary approach, without which the identification of the real processes using conceptual and simulation model and cannot be realized. This is also emphasized in this paper.

Multidisciplinary approach generally means that specialized disciplines are applied in a study of real process. These disciplines provide partial analysis of the process. These mono-disciplinary analyses are integrated to overall solution by integrating the solver who has basic multi-disciplines knowledge.

In our case study four disciplines are integrated, namely, algorithm development, programming and mathematics.

III. PROBLEM FORMULATION
MONOALPHABETICAL SUBSTITUTION

The monoalphabetical substitution cipher is a cipher where one-to-one mapping is used to substitute each of the characters of the plaintext by a corresponding character of the cipher text – see e.g. [9]. A plaintext is a message before encryption. A plaintext is a normal text, e.g. in English, presented in a readable and understandable form. A cipher text is an unreadable message after encryption. A cipher text looks like a random jumble of letters or other characters. One way how to obtain a cipher text from a plaintext is a character by character substitution realized according to the so-called conversion matrix. The conversion from a cipher text back into a plaintext is called decryption. The decryption is realized by using the same conversion table in the opposite direction.

A. Principles of monoalphabetical substitution

A concrete example of a historic cipher is the substitution used in the Old Testament, where re-writers of the Bible wanted to leave their mark in the text. Sesah appeared instead of Babel. This encryption emerged from the substitution of the letters from the beginning of the Hebrew alphabet by letters from its end, namely the **aleph** by **tav**, **bet** by **shin** and **kaf** by **lamed**. Since the Hebrew text is often recorded only in consonants and vowels are added from the context, this cipher is known as the **Atbash** [9], [11].

Applying the same principle to the Roman alphabet (the international alphabet with 26 letters without diacritics is used), the following Conversion table (Table 1) can be presented:

Table 1 Conversion table

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

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

The Atbash type system has another advantage - the same conversion table can be used for both decryption and encryption. An additional advantage is that the table can be reduced by a half of the original conversion table.

A Conversion table of monoalphabetic substitutions, however, may be more general. For example, mixing the letters of the cipher alphabet according to the password can be used; as in the following example (Table 2) where the password is based on the name of our university, from which repeated letter have been removed:

UNIVERSTY OF HADC KL (University of Hradec Králové)

Table 2 Conversion table with the password mixing letters

A	B	C	D	E	F	G	H	I	J	K	L	M
U	N	I	V	E	R	S	T	Y	O	F	H	A

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	C	K	L	B	G	J	M	P	Q	W	X	Z

The question is how many monoalphabetic ciphers can be created. Each cipher can be described by the table, which has in the first row the characters of the alphabet sorted in the usual alphabetical order (A, B, C, ..., Z) and in the second row there are the letters arranged in any random order. Number of different orders, i.e. permutations of the 26 letters of the alphabet is 26! = 26.25.24. ... 3.2. 1=
= 403291461126605635584000000,
approximately 403 quadrillions.

The fact that each single substitution is relatively easy to solve, was probably discovered for the first time by Arab

scholar Al-Kindí in the 9th century AD.

Al-Kindi described, in two brief paragraphs of the manuscript, the principle of the frequency analysis, which is one of the most important tools of the classical cryptoanalysis. The origin of the frequency analysis is likely to have been influenced by studying of the Qur'an, which was so thorough that it not only examined the frequency of individual words, but even the frequency of individual letters. In Arabic, the letters ***a*** and ***l*** appear very often, while the frequency of ***j*** is about ten times lower. Frequency, or percentage distribution of individual letters, however, is different in each language. The Table 3 shows the percentage distribution for English [12]:

Table 3 Percentage distribution of letters in a purely English text

A	B	C	D	**E**	F	G	H	**I**	J	K	L	M
8,2	1,5	2,8	4,3	**12,7**	2,2	2,0	6,1	**7,0**	0,2	0,8	4,0	2,4
N	**O**	P	Q	R	S	**T**	U	V	W	X	Y	Z
6,7	**7,5**	1,9	0,1	6,0	6,3	**9,1**	2,8	1,0	2,4	0,2	2,0	0,1

The letters E, T, A, O and I are the most frequent letters in English. The longer a cipher text is and the purer English is used in the text (without a mixture of foreign words), the more helpful the frequency analysis is. The procedure of the decryption will be described in the following text of the paper.

B. Principles of deciphering of the monoalphabetic substitution of very short text

The following linguistic facts, knowledge and skills have to be included in the automation of decryption analysis of very short text, shorter than 500 characters:

- Letters with the highest frequency are probably E, T, A, O and I - vowels. But for the short cipher text, it is probable that the sequence of letters above will be replaced. Therefore other linguistic facts, knowledge and skills have to be included in the decryption analysis.
- Automation decryption based on frequency analysis of the bigrams works well for long text, longer than 500 characters. A lot of manual changes have to be done in shorter text.

As a solution for automation decryption has been found the frequency of triplets of consecutive letters, which are called trigrams.

IV. PROBLEM SOLUTION
AUTOMATIC CRYPTOANALYSIS OF VERY SHORT TEXT

Before the automation analysis of the trigrams will be discussed, very well known algorithm for computer deciphering based on method of bigram analysis is to be recapitulated.

This algorithm is based on method of evaluation of frequency of pairs of consecutive letters and compares it with the frequency of bigrams of reference text. The evaluation of the compliance of the relative frequency of the bigrams in the cipher text and the reference text is obtained using the evaluation function:

$$f = \sum_{i=1}^{26} \sum_{j=1}^{26} \left| D_{ij} - E_{ij} \right| \quad (1)$$

Detailed description of the automation analysis of the bigrams programmed in VBA for Excel can be found e.g. in [13], [14].

As discussed in these papers Bigram Computer Cryptoanalysis can be used very well work for ciphered longer than 500 characters. This method does not work fully automatically for deciphering of the shorter text – manual changes are usually needed.

In this paper we will introduce the method for automation cryptoanalysis of the short text – shorter than 500 characters. This method, as have been stated above, is based on evaluation the frequency of trigrams - triplets of consecutive letters and compares it with the frequency of trigrams of reference text.

A. Algorithm

The algorithm itself was divided into three relatively independent procedures.

1) Frequency

This procedure performs an initial setup for the conversion table. The frequency of each character in the cipher text is detected and cipher text characters are sorted in the descending order of frequency. That determined the sum of the rows from the reference matrix ***E*** that corresponds to the frequencies of letters in the reference text; the reference text characters are sorted in the descending order of frequency.

Finally, the procedure pairs the first most frequent letter of the cipher text with the first most frequent letter of the reference text, then the second most frequent letter of the cipher text with the second most frequent letter of the reference text, and so on.

2) Matrix of trigrams

The procedure creates three-dimensional matrix ***D*** of relative frequencies of the trigrams of the cipher text. The part of the matrix ***D*** is shown on the Figure 2 – see e.g. [10].

Axes *x* (vector *x*) represents the first letter of the trigram, axes *y* (vector *y*) represents the second letter of the trigram and axes *z* (vector *z*) represents the third letter of the trigram. The frequency of the trigram EAD shown on the Figure 2 is 23.

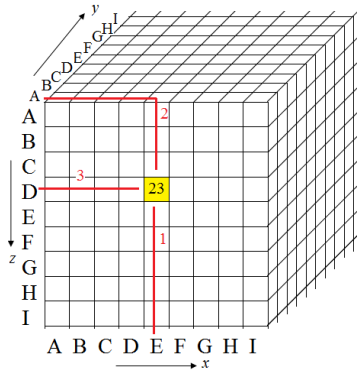

Fig. 2 Matrix of trigrams

In next step (see e.g. [10]) the procedure evaluates the compliance with the reference text by using an evaluation function:

$$f = \sum_{i=1}^{26} \sum_{j=1}^{26} \sum_{k=1}^{26} \mid D_{ijk} - E_{ijk} \mid \qquad (2)$$

3) Exchange

Step by step, 2 *x*-vectors, 2 *y*-vectors and 2 *z*-vectors of three dimensional matrix ***D*** are exchanged and thus the matrix ***D'*** is obtained. The vectors are exchanged in the order of the frequencies of the letters in the cipher text from the most frequent to the least frequent:

- the *x*-vector corresponding to the order of the first exchanged character is replaced by the x-vector corresponding to the second exchanged character;
- then *y*-vector corresponding to the order of the first exchanged character is replaced by the y-vector corresponding to the second exchanged character;
- then *z*-vector corresponding to the order of the first exchanged character is replaced by the y-vector corresponding to the second exchanged character;

The substitution of the vectors of the matrix corresponds to the replacement of the letters in the second row of the conversion table. First, the first and the second letters are replaced then the first and the third letters are replaced and so on, until finally the first letter is exchanged by the last one. The same is done with the second letter, which is replaced with the third one, then with the forth one and so on. Finally next to last letter is replaced by last one.

After each substitution a new matrix ***D'*** is obtained, and the evaluation of the compliance of the relative frequency of the trigrams in the cipher text and the reference text is obtained using the evaluation function:

$$f' = \sum_{i=1}^{26} \sum_{j=1}^{26} \sum_{k=1}^{26} \mid D'_{ijk} - E_{ijk} \mid \qquad (3)$$

After each substitution the values *f* and *f'* are compared:

- If *f'* > *f*, the procedure continues with the next pair of the letters in order.
- However, if *f'* < *f*, the procedure immediately stops the process of the letter substitution and the exchange in the conversion table is proposed, which will improve the compliance (lowering the value of evaluation function *f*).
- If the program exhausts all the possible exchanges of the characters, without any condition *f'* < *f*, the message is notified to the operator. If the cipher text is not fully deciphered, the operator can propose a manual replacement.

Finally the procedure "Exchange" will create a new matrix ***D*** of relative frequencies of the trigrams of the cipher text, and it will provide a new assessment of compliance with the reference text using the evaluation function (3).

From the description of the above procedures it is obvious that the procedure "Frequency" is run only once at the beginning of the program to set the appropriate initial conditions. The procedures "Trigrams" and "Exchange" are run alternately.

That algorithm is very for short texts with a length of up to 500 characters. For longer text this algorithm is very time consuming.

B. Computer simulation of the algorithm in MS Excel

The principle of computer simulation of the frequency analysis trigrams is very similar to the computer simulation of frequency analysis of bigrams described in paper [9].

The computer simulation program can be split to separate sub-procedures.

1) Frequency – Frequency analysis of the characters in cipher text

Load of cipher text, treatment of cipher text, frequency analysis of characters of cipher text, ordering, ordering of characters according to their frequency in descendent order.

The procedures are similar to the procedures already discussed in [5].

2) Trigram matrix – Creation of three-dimensional matrix D for frequency analysis of trigrams

For frequency analysis of the bigrams can be used two-dimensional matrix, which can be represented e.g. by table in MS Excel spreadsheet – see e.g. [9].

Matrix for analyzing of frequency of trigrams is three-dimensional – see Figure 2. This three-dimensional matrix has to be represented in the easily readable form – e.g. in the MS Excel spreadsheet table. The method of transposition of the three-dimensional matrix *D* to two-dimensional table can be for example as follows:

- The first row header contains letters arranged according to their frequency in descending order (already created by procedure 5). It represents the x-vector of the three-dimensional matrix ***D*** - the first letter of the trigram.
- The first column header contains letters arranged according to their frequency in descending order. It represents the y-vector of the three dimensional matrix ***D*** - second letter of the trigram.
- The second column header contains letters arranged according to their frequency in descending order. It represents the z-vector of the three dimensional matrix ***D*** - third letter of the trigram.

The transformation from 3D to 2D is for illustration indicated in the Figure 3. The 3D → 2D transformation can be described as a gradual cutting of the front layers of cube that are in order stacked under each other. This will provide a table which part is shown in the Figure 4. In this picture the example of the trigram spreadsheet table is shown. The marked cell represents trigram TEX with the frequency 2.

Reading trigrams from the table is simple - the first letter can be found in the row ***2***, then second letter in the column ***A*** and third letter in the column ***B***.

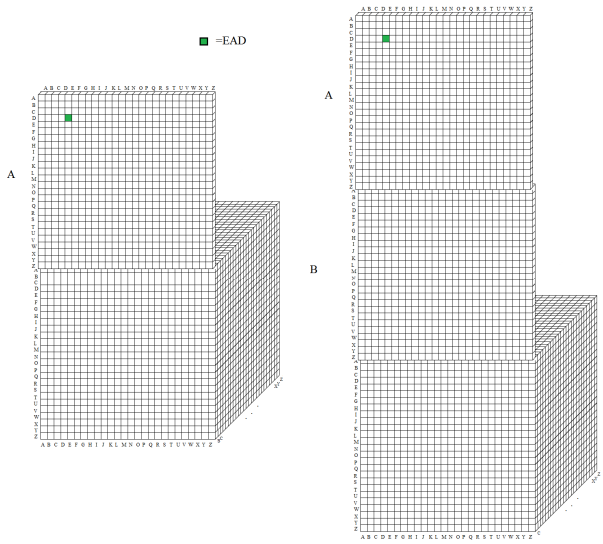

Fig. 3 Trigram MS Excel spreadsheet table

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1			177	107	103	83	82	78	63	62	51	49	48	46	44
2			E	O	A	T	I	S	R	N	L	Y	P	U	D
3	E	E													
4	E	O													1
5	E	A								1	1				
6	E	T							1				1		
7	E	I								1					
8	E	S						1		3					1
9	E	R				2				1					1
10	E	N				1		1	1	1					1
11	E	L				1		1							
12	E	Y													
13	E	P													
14	E	U													
15	E	D				1		1							
16	E	K													
17	E	V				1		1			1				1
18	E	M				1		1			1				1
19	E	C						1							
20	E	B							1	1					
21	E	Z						2		1					
22	E	H							1	1					
23	E	J													
24	E	F													
25	E	X				2									
26	E	G													
27	E	W													
28	E	Q													
29	O	E													
30	O	O													
31	O	A													
32	O	T				1						1			
33	O	I													
34	O	S				1	1			1	1				
35	O	R								1			1		

Fig. 4 Trigram MS Excel spreadsheet table

Function f depends on three parameters – $i = (1\ldots26)$, $j = (1\ldots26)$, $k = (1\ldots26)$, which represent x, y, z coordinate of the element of three-dimensional matrix D_{ijk}.

The location of element in MS Excel Spreadsheet coordinate system is as follows:

$$D_{ijk} = \text{Cells}(A + X, B + Y), \quad (4)$$

where

$X = (j - 1)*Z + k$,

$Y = 1 + I$,

$A = 2$ is header row,

$B = 1$ is header column,

$Z = 26$ is number of characters.

Similarly the element E_{ijk} of matrix $\boldsymbol{E}$ can be calculated and function f according to equation (2).

3) Exchange – Replacement of two characters in the cipher text table

This procedure creates new matrix $\boldsymbol{D'}$ by exchange the vectors of matrix $\boldsymbol{D}$ based on algorithm described in section III.).

The principle of the exchange of the vectors in MS Excel Spreadsheet is shown on the Figure 5.

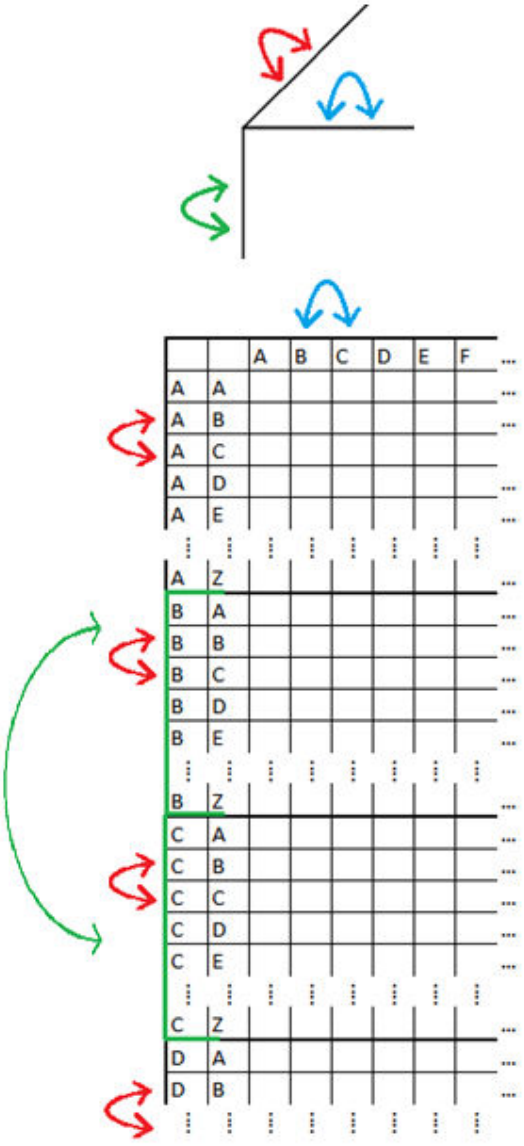

Fig. 5 Principle of the exchange of the vectors in MS Excel Spreadsheet

The programming code written in Visual Basic for Application of sub-procedure *Exchange* is as follows:

```
Sub Exchange()
A = 2 'header row
B = 1 'header column
Z = 26 'number of characters
n = 1 'change characters n places side by side
i = 1 'order of changed letter (A=1, B=2,...)

H = Columns(B + 1 + i)
G = Columns(B + 1 + i + n)
Columns(B + 1 + i) = G
Columns(B + 1 + i + n) = H

For j = 0 To Z - 1
   V = B + 1 + Z
   X = A + j * Z + i
   Y = A + j * Z + i + n
   H = Range(Cells(X, 1), Cells(X, V))
   G = Range(Cells(Y, 1), Cells(Y, V))
   Range(Cells(X, 1), Cells(X, V)) = G
   Range(Cells(Y, 1), Cells(Y, V)) = H
Next j

X = A + (i - 1) * Z + 1
Y = A + i * Z
T = A + (i + n - 1) * Z + 1
U = A + (i + n) * Z
V = B + 1 + Z
H = Range(Cells(X, 1), Cells(Y, V))
G = Range(Cells(T, 1), Cells(U, V))
Range(Cells(X, 1), Cells(Y, V)) = G
Range(Cells(T, 1), Cells(U, V)) = H

End Sub
```

V. Program Optimization for VBA

A program written based on above mentioned parts of algorithms works properly, but this program is very time consuming. The time needed for whole program execution is in the order of tens of hours. Therefore, two changes we have made to solve this problem.

The main reason of the long execution of the program is process of the exchange process directly in the Excel worksheet. The above mentioned VBA code of the subroutine Exchange is suitable for illustration, but not for the purpose of the program. Instead of Exchange provided directly in the Excel worksheet cells, the procedure is to be executed in the background. For this purpose the worksheet Excel table is replaced by two dimensional array. The programming code written in Visual Basic for Application of sub-procedure *Exchange* with the arrays is as follows:

```
Sub Array_initialization()
Dim Ref_text(1 To 26 * 26, 1 To 26) As Integer
Dim Ciph_text(1 To 26 * 26, 1 To 26) As Integer
Dim Key(1 To 26)

For e = 1 To Z 'block
   For j = 1 To Z row
      For k = 1 To Z 'order in row (column)
         Ref_text((e - 1) * Z + j, k) =
         = Cells(A + (e - 1) * Z + j, C + 1 + k).Value
      Next k
   Next j
Next e

For e = 1 To Z 'block
   For j = 1 To Z 'row
      For k = 1 To Z 'column
         Ciph_text((e - 1) * Z + j, k) =
          =Cells(A + (e - 1) * Z + j, B + 1 + k).Value
      Next k
   Next j
Next e

For j = 1 To Z
   Key(j) = Cells(A, B + 1 + j)
Next j
End Sub
```

The programming code written in Visual Basic for Application of exchange inside of arrays is as follows:

```
Sub Exchange_Array()
'block
      For j = 1 To Z
      For k = 1 To Z
         x = Ciph_text((i - 1) * Z + j, k)
         Ciph_text((i - 1) * Z + j, k) = Ciph_text((i + n - 1) * Z + j, k)
         Ciph_text((i + n - 1) * Z + j, k) = x
      Next k
      Next j
  'rows
      For e = 1 To Z
      For k = 1 To Z
         x = Ciph_text(Z * (e - 1) + i, k)
         Ciph_text(Z * (e - 1) + i, k) = Ciph_text(Z * (e - 1) + i + n, k)
         Ciph_text(Z * (e - 1) + i + n, k) = x
      Next k
      Next e
  'columns
      For e = 1 To Z
      For j = 1 To Z
         x = Ciph_text((e - 1) * Z + j, i)
         Ciph_text((e - 1) * Z + j, i) = Ciph_text((e - 1) * Z + j, i + n)
         Ciph_text((e - 1) * Z + j, i + n) = x
      Next j
      Next e
  'exchange in key
      x = Key(i)
      Key(i) = Key(n + i)
      Key(n + i) = x
End Sub
```

Note: Two dimensional array is used in above mentioned VBA code to keep form of Excel worksheet. Similarly, three dimensional array of VBA code could be used:

```
Dim Ref_text(1 To 26, 1 To 26, 1 To 26) As Integer
… Ref_text(k, e, j) = Cells(A + (e - 1) * Z + j, C + 1 + k).Value …)
```

The procedure *Exchange*, however, cannot be executed for each step of exchange. The procedure is executed only if the function f' after exchange of rows, columns and blocks is less than the original function f – see [14]). The VBA code for calculation of the function f' has to be rewritten. The Figure 6 shows the situation.

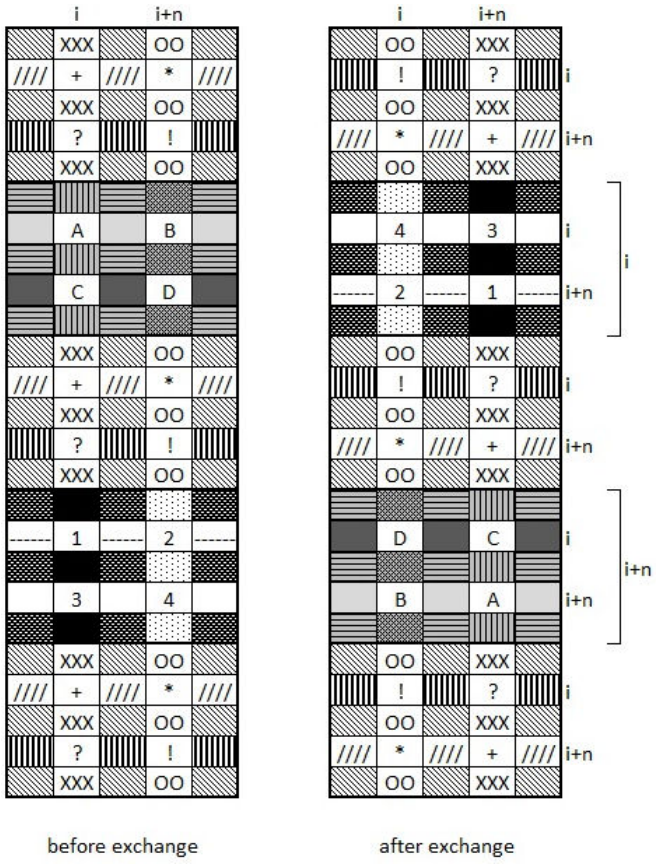

Fig. 6 Exchange in the trigram table

The Figure 6 shows the table of trigrams for the number of letters $Z = 5$ before exchange and after the exchange.

Note: $Z = 5$ is the smallest possible number of letters, on which (in the case of $i = 2$ and $n = 2$) can be seen all exchange areas (i.e., before i, between i and $(i + n)$ and after $(i + n)$).

The cells of this type ( ) in the Figure 6 shows cells that remain after the exchange in their original order. The cells of this type (and) signifies exchange of rows i and $i + n$ in all blocks except the block i and $i + n$, where the exchange of rows is slightly different.

The table can be described based on the blocks (e), the rows in the block (j) and the columns (k). Then, for each e, j, k can occur just three possibilities:

- equals i
- equals $i + n$
- does not equal either i or $i + n$.

If these options are mutually combined, the 27 different options can be obtain (i.e. 27 variously labeled cells in the image).

Now in general (by using e, j, k, i, n) the exchange of cells for each of these options can be described – see Table 4.

Table 4 Options of exchange

Position in the table (array) of reference text			Color	Order after exchange - position
e (block)	j (row)	k (column)	(figure 6)	in table (array) of cipher text
e = i	j = i	k = i	A	(i + n – 1) * Z + i + n ; i + n
		k = i + n	B	(i + n – 1) * Z + i + n ; i
		k ≠ i ∧ k ≠ i + n		(i + n – 1) * Z + i + n ; k
	j = i + n	k = i	C	(i + n – 1) * Z + i ; i + n
		k = i + n	D	(i + n – 1) * Z + i ; i
		k ≠ i ∧ k ≠ i + n		(i + n – 1) * Z + i ; k
	j ≠ i ∧ j ≠ i + n	k = i		(i + n – 1) * Z + j ; i + n
		k = i + n		(i + n – 1) * Z + j ; i
		k ≠ i ∧ k ≠ i + n		(i + n – 1) * Z + j ; k
e = i + n	j = i	k = i	1	(i – 1) * Z + i + n ; i + n
		k = i + n	2	(i – 1) * Z + i + n ; i
		k ≠ i ∧ k ≠ i + n	----------	(i – 1) * Z + i + n ; k
	j = i + n	k = i	3	(i – 1) * Z + i ; i + n
		k = i + n	4	(i – 1) * Z + i ; i
		k ≠ i ∧ k ≠ i + n		(i – 1) * Z + i ; k
	j ≠ i ∧ j ≠ i + n	k = i		(i – 1) * Z + j ; i + n
		k = i + n		(i – 1) * Z + j ; i
		k ≠ i ∧ k ≠ i + n		(i – 1) * Z + j ; k
e ≠ i ∧ e ≠ e + n	j = i	k = i	+	(e – 1) * Z + i + n ; i + n
		k = i + n	*	(e – 1) * Z + i + n ; i
		k ≠ i ∧ k ≠ i + n	////	(e – 1) * Z + i + n ; k
	j = i + n	k = i	?	(e – 1) * Z + i ; i + n
		k = i + n	!	(e – 1) * Z + i ; i
		k ≠ i ∧ k ≠ i + n		(e – 1) * Z + i ; k
	j ≠ i ∧ j ≠ i + n	k = i	XXX	(e – 1) * Z + j ; i + n
		k = i + n	OO	(e – 1) * Z + j ; i
		k ≠ i ∧ k ≠ i + n		(e – 1) * Z + j ; k

The principle is that the function provides the exchange, so the exchange is not performed either in the Excel worksheet, or in the array. The exchange is performed after the function f' is for given i and n less than the original function f. The indexes e, j, k describe the position in reference table (in the array). For this position, then the corresponding position in the cipher table (array) is determined after the fictional exchange. This can be provided in the VBA programming code by using several nested If conditions.

```
For n = 1 To Z − 1
For i = 1 To Z − n
For k = 1 To Z
For e = 1 To Z
For j = 1 To Z

If e = i Then
   If j = i Then
      If k = i Then
         Eejk = Ciph_text((i + n - 1) * Z + i + n, i + n)
      Else
         If k = i + n Then
            Eejk = Ciph_text((i + n - 1) * Z + i + n, i)
         Else
            Eejk = Ciph_text((i + n - 1) * Z + i + n, k)
         End If
      End If
   Else
..... (followed based on the table)
End If

Dejk = Ref_text((e - 1) * Z + j, k)
f' = f' + Abs(Dejk - Eejk)
```

```
Next j
Next e
Next k

'comparison of the funcions f and f'

If f > f' Then
        f = f'
        f' = 0
'replacement of the letters in the array Ciph_text and in the
array Key
        e = Z + 1
        n = 0
Else
   f' = 0
End If
Next e
Next n
```

The principle of the program can be expressed via flowchart. The flowchart is shown on the Figure 7.

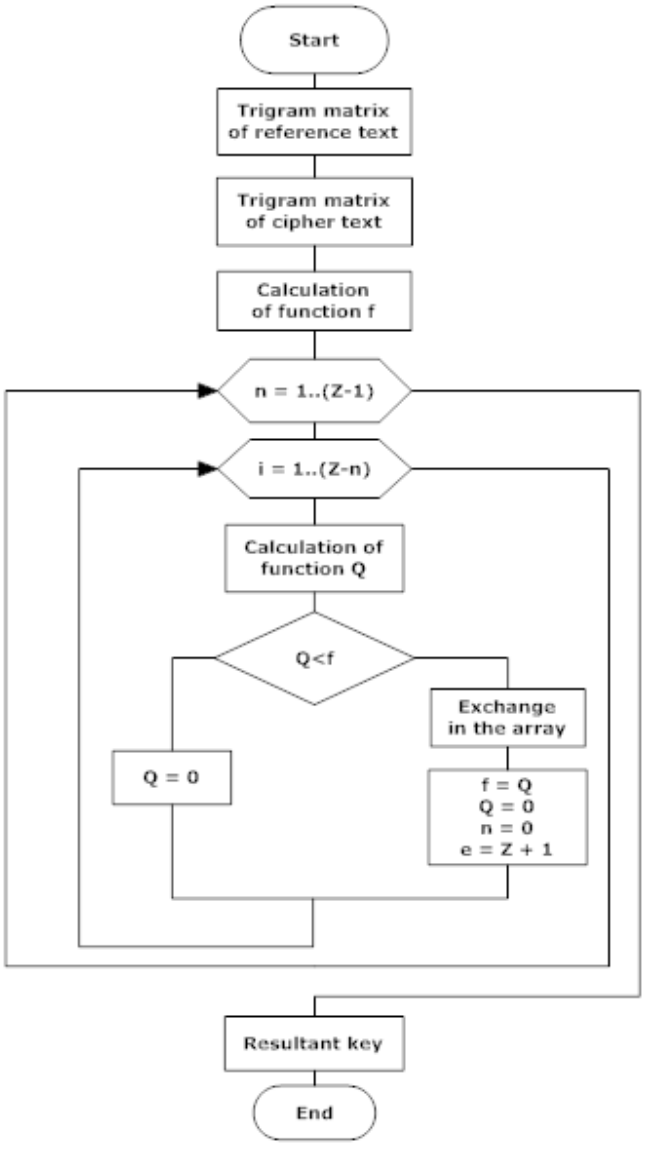

Fig. 7 Flowchart of the program

VI. The Results of the Deciphering

The modified program is executed within a few tens of seconds. This execution speed enables practical using the program.

The program enables decryption of short simple substitution cipher text almost without manually performed exchanges. It is important to note that the mentioned algorithm (described in detail in [10] and [11]) does not provide hundred percent deciphering key, but it provide rapid approach to the key. To obtain the wholly the 26! permutations of the letters has to be executed and for each permutation the function *f'* (deviation of trigram table of cipher text with respect to the trigram table of reference text) has to be calculated.

Minimum (ideal) value of the function *f'* corresponds to the hundred percent deciphering key. After the completion of the program execution, however, we often get a slightly higher value which identify, that hundred percent deciphering key is not found.

Deciphering based on trigram analysis has been studied in cipher text with the length in the range from 200 to 500 characters.

We defined new quantity to help summarize the results coming from the analysis. The quantity is accuracy of the resulting key – $A = (26 - w)/26$, where w is number of incorrect deciphered letter. The value of A are from interval $A \in \langle 0, 1 \rangle$. If $A = 0$ the key is absolutely wrong, if $A = 1$ the key is absolutely correct.

We have found three following correlations:

First correlation: between length of the cipher text *L* and the accuracy of the resulting key *A*. In the longer text the founded deciphering key less differs from the true keys – is more accurate. The correlation dependency *A*(*L*) is shown on the Figure 8.

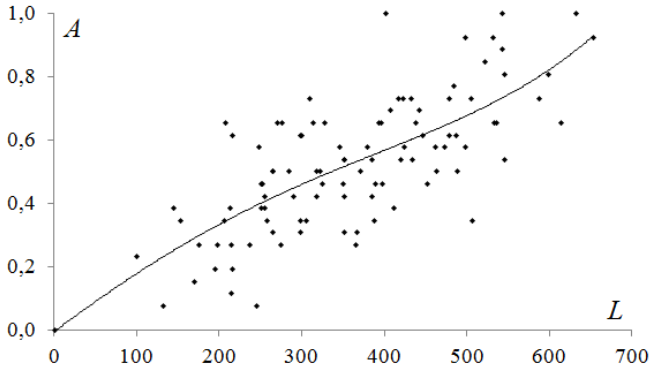

Fig. 8 The chart of correlation dependency *A*(*L*)

Second correlation: between the accuracy *A* of the resulting and true key and difference between the final value of *f'* and the minimum value of *f*. The resulting key is more accurate if difference between final value *f'* and *f* is less. The correlation dependency *A*(*f'*−*f*) is shown on the Figure 9.

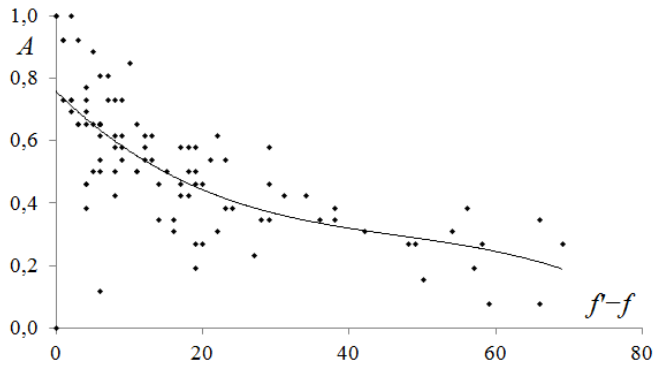

Fig. 9 The chart of correlation dependency $A(f'-f)$

Third correlation: between accuracy A of the resulting key and length of the reference text. The reference text should written in a similar form and style as we expect from the cipher text (the results will be more accurate).

VII. Conclusion

The paper summarized the basic principles of automation cryptoanalysis of monoalphabetical substituted short cipher text. The algorithm is based on the frequency analysis of the trigrams. The algorithm as well as the computer simulation program has been validated by both Czech and English short cipher text.

The paper also offered one of the kinds of the possible teaching / learning strategies using the system approach. The system approach can be set as the default paradigm for a wide integration of the principles of the algorithm development into education. The paper emphasizes the fact that the algorithm development of the real processes should be supported by computer simulation and visualization.

Acknowledgment

This research has been partially supported by the Specific research project of the Faculty of Science of University of Hradec Kralove No. 2105.

References

[1] S. Hubalovsky, J. Jelinek, J. Sedivy, „Mathematical modeling and computer simulation of optimal reaction time of the Lupine protein hydrolysis using fermented whey", *International Journal of Mathematical Models and Methods in Applied Sciences*. vol. 6, No. 2., 2012.

[2] J. Sedivy, S. Hubalovsky, "Mathematical foundations and principles in practice of computer aided design simulation", *International Journal of Mathematics and Computers in Simulation*. vol. 6, No. 1. 2012.

[3] O. Horák, L. Mitrovič, "Description of the Basic Algorithm Blocks and Structures Representation in Courses of Algorithm Development", *WSEAS Trans. on Information Science & Applications on Advances in Engineering education*. vol. 9, No. 2. 2012.

[4] J. Šedivý, **"**Multimedia support of parametric modeling", in *Proc. 9th WSEAS International Conference on Engineering Education (EDUCATION '12)*. WSEAS Press, 2012.

[5] K. Dvořák, J. Šedivý, "CAx application in the teaching of engineering subjects", in *Proc. 9th WSEAS International Conference on Engineering Education (EDUCATION '12)*. WSEAS Press, 2012.

[6] S. Hubalovsky," Modeling and computer simulation of real process – solution of Mastermind board game", *International Journal of Mathematics and Computers in Simulation*. vol. 6, No. 1. 2012.

[7] J. Bailer, M. Daniela, "Tracing the Development of Models in the Philosophy of Science", *Magnani, Nersessian and Thagard*, 1999.

[8] S. Hartmann, "The World as a Process: Simulations in the Natural and Social Sciences", In R. Hegselmann, et al., *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, Theory and Decision Library. Dordrecht: Kluwer, 1996.

[9] S. Hubalovsky, M. Musílek, "*Automatic cryptoanalysis of the monoalphabetical substitution as a method of the system approach in the algorithm development thinking*". International journal of applied mathematics and informatics. vol. 4, No. 4, 2010.

[10] P. Hanzalová, Š. Hubálovský, M. Musílek, "Automatic cryptoanalysis of the short monoalphabetical substituted cipher text". *Visualization, imaging and simulation (VIS '12)*. WSEAS Press, 2012.

[11] M. Musílek, *Cipher and codes* [online]. 2010. Available: <http://www.musilek.eu/michal/>.

[12] *Letter Frequency*. Available: <http://en.wikipedia.org/wiki/Letter_frequency>.

[13] S. Singh, *The Code Book* [CDROM]. 2010. Available: <http://www.simonsingh.net>.

[14] T. Jacobsen, "*A Fast Method for the Cryptoanalysis of Substitution Ciphers*", Cryptologia, Vol. 19, 1995.

Stepan Hubalovsky was born in 1970 in Czech Republic. He obtained master degree in education of mathematics, physics and computer science in 1995 and doctor degree in theory of education in physics in 1998 both in Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic. He works as associated professor on University of Hradec Kralove His scientific activities are system approach, modeling and simulation.