VOJNOTEHNIČKI GLASNIK/MILITARY TECHNICAL COURIER, 2014., Vol. LXII, No. 1

# MATRIX-BASED ALGORITHM FOR TEXT-DATA HIDING AND INFORMATION PROCESSING

*Nenad* O. Vesić, *Dušan* J. Simjanović
University of Niš, Faculty of Science and Mathematics

DOI: 10.5937/vojtehg62-3935

FIELD: Mathematics
ARTICLE TYPE: Original Scientific Paper

*Abstract:*

An application of differential geometry in hiding text-data is presented in this paper. This application is a generalization of similar previously presented methods. Matrix valued functions of more than one variable are used in text hiding. This method is covered by the corresponding programs in the MATHEMATICA software package and compared with the previously presented method.

Key words: *program; decryption; encryption; text; message; public key; private key; algorithm; matrix valued function.*

## Introduction

The expanding usage of the Internet and the world wide web in general rises two important issues. How to provide secrecy and security of the data positioned in computer folders, as well as of those transmitted via the network?

Secrecy refers to the right of an individual and organization to decide whether, and in what amount, the information they have would be available to others.

Security of data refers to their protection from illegal detection, destruction or modification.

In order to efficiently estimate and choose appropriate protection means, network administrators have to systemize and define the requests and accesses referring to protection.

The approach based on the following three security aspects usage is most often used (Bohen, 1999):

• attack protection (refers to an action compromising safety of the information posessed by an institution, e.g. bank, police)

ACKNOWLEDGEMENT: This paper is financially supported by project 174012 of the Serbian Ministry of Education, Science and Technological Development.

e.mail: vesic.specijalac@gmail.com

- protective mechanism (detecting, protecting and recovering from an attack)

- protective service (utility making safety better during data processing and transmitting)

Attacks may be divided into passive and active ones. Passive attacks are basically interception and following transmission of information. The aim of the attacker is to reach information which is being transmitted, not changing its content while doing that. Active attacks refer to information flow change or creation of a false flow and may make large damage. Nowadays, two technics for providing network safety are most often used, and those are cryptography and authentification. Cryptography is a scientific discipline dealing with a study of methods for sending messages in a form readable by a receiver. The basic task of cryptography is to enable a sender and a receiver to communicate while insecure channels make them incomprehensible to the third person. Considering different kinds of danger, cryptography manages very well to protect messages from detection, modification, infiltration of messages and repetition of the old ones.The process of standard coding is presented in the following figure. An original comprehensible message (plaintext),marked as an original text (open text) by a sender, is transformed, using a previously agreed upon key, into an apparently incomprehensible message (ciphertext), marked as an encrypted text. Then the sender via some communication channels sends the ciphertext to the receiver who knows the key and is able to decode the ciphertext amd determine the original text. Unlike him, the opponent (oposite side) even if he finds the code has no key and thus cannot read the original text.

The encrypting process depends on the algorithm and the key. The algorithm is the procedure by which the way of getting the original text from the encrypted one is determined. The key is the value which is independed from the original text. Depending on the key to be used, the algorythm is going to generate a certain result. By changing the key, the result generated by the algorithm is also changed. When the original text is encrypted, it could be further forwarded to the receiver. On the receiving side, the encrypted text is transformed into the original text by the algorithms for decrypting and the identical key used for encrypting. The algorithm for decrypting is inversive to the encrypting algorithm. The encrypting algorithm must be powerful enough so that it is practically impossible to decrypt the message only by the encrypted text and the knowledge of algorithms. This means that standard encrypting safety depends on the key secrecy.

The basic aims of cryptography are (Buchmann, 2001):

- data secrecy – data are accesable only to those who are allowed to access to them

- data integrity – attacks discovery and false message identification

- identity check – proof that the parts in communication are really those they claim they are
- nondenying – prevents from false denying about taking part

As already stated, cryptography is the science dealing with metods for keeping informational secrecy. Cryptographic systems can be classified based on three criteria:

- Based on the operations used in the encryption process

All encrypting algorithms are based on two general principles: the principle of substitution and the principle of transposition. In the algorithms based on the principle of substitution, each individual element of the original text (bit, letter, group of bits, group of letters) is substituted with another element. If the principle of transposition is used in the algorithm, the elements of the original text are reorganized or rearranged. The basic aim of both principles is prevention from losing all the information the message consists of.

- Based on the number of the keys used

If the sender and the receiver use the same key, the system is symmetrical (system with one key, system with a secret key). Otherwise, if the sender and the receiver use different keys, the system is asymmetrical (system with two keys, system with a public key). The encryption key can be public while the decryption key must be secret.

- Based on the manner in which the original text is processed

In the process of blocks encryption, only one block at a time is processed, generating thereby one exit block for each entered block. In the process of sequence encryption, elements are sequentially processed.

Cryptography has experienced the largest development when Diffie and Hellman published their work (Diffie, Hellman, 1976).

The cryptanalysis is a process of attempting to discover the original text or the key or the both of them. Which strategy will be used in the cryptanalysis depends on the encryption scheme and available information. A cryptanalytic attack can be performed on an encrypted text, a known original text, a selected general text, a selected encrypted text and a selected text. For each of these attacks, the cryptanalyst has a different perimeter of information needed to be performed. In case that the cryptanalyst knows only the encrypted text and the algorithm used for the encryption, one of the possible attacks is testing all possible keys. If there is a enormous number of keys, decrypting becomes impractical. To use this approach, the attacker must have a general idea about the type of the original text that is hidden.

Defense based only on knowing the encrypted text is the simplest because the attacker has the least information for the attack. When the cryptanalyst has a larger amount of information, he could be able to obtain one or more messages with the original text with their encrypted versions or could know (or guess) that certain parts of the original text will appear in the message. Using the metod by which the original text is transformed, the cryptanalyst will be able to reveal the key and decrypt the message easily. An attack on the selected original text is performed if the cryptanalyst is able to access to the original system and insert the desired message. If he is able to chose the message that will be encrypted, for detecting the structure of the key he can chose sentences that may be expected, and, by taking only them, find out the key. Attacks on selected encrypted texts and attacks on selected texts are rarely perfomed, but it is stil possible that the cryptanalyst will realise them.

The encrypting scheme is unconditionally secured if the encrypted text is generated by the scheme that does not contain enough information to uniquely determine the appropriate original text, regardless of the accessibility of the encrypted text. There is no unconditionally secure encrypting algorithm, so thereby it is necessary for the encrypting algorithm to fulfill at least one of the following criteria:

- Price of the code breaking exceeds the value of the encrypted message
- Time needed for the code breaking exceeds the time for which the informations is valid

In (Kuljanski, 2010, pp.65-77) we saw a clear and simple use of the RSA (named by the author's initials) algorithm. It is logic that the attack on the RSA could be conducted easily if the exponent $d$ is known. With a succesful factorization of the number $n=pq$, $\varphi(n)= (p-1)(q-1)$ can be obtained, and then, using $de \equiv 1 \ (mod\varphi(n))$ (using Euclidean algorithm), the exponent $d$ as well.

The strength and safety of the RSA algorithm (Menezes, 2001, Menezes, et al., 1996) lies in the size of the number n (around 200 digits) for whose decryption a little fewer than **googol** years will be needed.

- The main idea of this paper is based on the matrix factorization which is an equally difficult (if not harder) task as the RSA algorithm number factorization.

## Main result

In this part of the paper, an algorithm for the encryption and decryption of different types of texts, like the RSA, will be presented. It will be an improvement of the encryption of an algorithm presented in (Vesić, Simjanović, 2012). In that encryption, a code is hidden by multiplying numerical and matrix valued functions. For this code breaking, finding one numerical matrix that will be the key is sufficient.

Let a text T contains N characters. The aim of this algorithm is hiding the text T using the $k$-coordinate matrix surface $M=M(u_1,u_2, \dots ,u_k)$ of a type $p\times q$ (Vesić, 2013). The matrix ordered a set of characters $Ch$, as a private key, and the coordinate matrix curves ($1$-coordinate matrix surfaces) $L=L(t)$ of the type $p\times p$, $R=R(t)$ of the type $q\times q$ and $\Gamma=\Gamma(t)$ of the type $p\times q$ ,are known to both the sender and the receiver. The additional and necessary condition is the existence of the integer numbers $r_l$ and $r_r$ so that det $L(r_l)\neq 0$ and det $R(r_r)\neq 0$. The coordinate matrix curves *L, R* and *Γ* with the set *Ch* are known only to the sender and the receiver of the message.

## *Encryption of a text*

The encryption of the text is represented in the following steps.

**E$_1$:** Transform the text T into an ordered set of pairs of positive integers ($a_d$, $b_d$), where $a_d$ is the row and $b_d$ is the position in the $a_d$-th row in the *Ch* where the d-th character of the text is. In this way, the set of pairs $S_1$ is obtained.

**E$_2$:** Transform each pair ($a_d$, $b_d$) into the pair

($\mathbf{a}_d$, $\mathbf{b}_d$)=($a_d$+RandomInteger[{1,100}]*Dimensions[*Ch*][[1]], $b_d$+RandomInteger[{1,100}]*Dimensions[*Ch*[[$a_d$]]][[1]]).

In this way, the set of pairs $S_2$ is obtained.

**E$_3$:** For the given positive integers *p* and *q* and an integer *k* such that

$$(k-1)pq < N \leq kpq$$

set $S_2$ should be supplemented with the pairs (1,0) to the set $S_3$ with *kpq* elements.

**E$_4$:** Choose the *k* different variables $u_1, u_2, \dots, u_k$ placed in an ordered set U and an element of the set $S_3$ at the position *vpq+w*,

$$v=0, 1, \dots, |S_3|/(pq), \ w=0, 1, \dots, pq-1,$$

should be replaced by the polynomial $\Phi_{vpq+w}(u_v)$ of degree 2 with random real positive roots between 1 and 100 if $\mathbf{b}_{vpq+w}$=0 and roots $\mathbf{a}_{vpq+w}\pm i\cdot\mathbf{b}_{vpq+w}$ if $\mathbf{b}_{vpq+w}\neq 0$. In this way, the ordered set $S_4$ is obtained. The position (d) is the d-th position in the set U.

**E$_5$:** Using the polynomials of the set $S_4$ , a new set

$$S_5=\{M^{(1)}(u_{(1)}), M^{(2)}(u_{(2)}), \dots, M^{(k)}(u_{(k)})\}$$

which consists of the coordinate matrix curves $M^v=M^v(u_v)$ of the type *p×q* is obtained. In the curve $M^v$ at the place (*r,c*) is the polynomial $\Phi_{(v-1)pq+rc}(u_v)$.

**E$_6$:** Using the coordinate matrix curves from the ordered set $S_5$, the *k*-coordinate matrix surface

$$M=M(u_{(1)},u_{(2)}, \dots, u_{(k)})=M^{(1)}(u_{(1)})*M^{(2)}(u_{(2)})*\dots*M^{(k)}(u_{(k)})$$

is formed

**E$_7$:** Determine the rational numbers $r_l$, $r_r$ and $r_c$ such that

$$\det L(r_l)\cdot\det R(r_r)\neq 0.$$

Form the coordinate matrix surface

$$E=E(u_1,\ u_2,\ \dots,\ u_k)=L(r_l)\cdot(M(u_1,\ u_2,\ \dots,\ u_k)-\Gamma(r_c))\cdot R(r_r)$$

which is the public key.

## *Decryption of a public key*

Because of the necessary condition for the numerical matrices $L(r_l)$ and $R(r_r)$, based on a message message, the decryption of the public key

$$E=E(u_1,u_2,\ \dots,\ u_k)$$

is

***D$_1$:*** $F=L^{-1}(r_l)\cdot E\cdot R^{-1}(r_r)+\Gamma(r_c)$

**D$_2$:** Factorize the polynomials in the coordinate matrix surface F. The polynomial at the position (p,q) in the coordinate matrix surface F should be transformed into the set of pairs $\gamma_{pq}$ with appropriate variables as the first and the polynomial dependent on the first coordinate as the second coordinate.

**D$_3$:** Transform the second coordinates of the elements of the sets obtained in the previous step into empty strings if the roots of the polynomials are real, or into the character at the position (**x**, **y**) if the roots are the complex numbers $z_{1,2}=x\pm i\cdot y$,
where

***x****=Mod1n[x, Dimensions[Ch][[1]],*

***y****=Mod1n[y, Dimensions[Ch[[****x****]]][[1]]*

and Mod1n[a, b]=b if b|a or Mod1n[a, b]=r, where r is a positive integer such that for an integer q, the equation $a=b\cdot q+r$, for $0\leq r<b$ holds.

**D$_4$:** Depending on the variables position, based on the transformation showed in the step **D$_3$**, the text should be formed in the way that, going over the message, the strings which gather the characters hidden by the factors dependent on the variables $u_k$, k=1,2, ... N, are got from the elements form the surface F at the positions

$$(1,1),\ \dots,\ (1,q),\ (2,1),\ \dots,\ (2,q),\ \dots,\ (p,1),\ \dots,\ (p,q),$$

in that order.

**D$_5$:** The previous procedure should be finished creating a string s obtained by the concatenation of strings $s_k$,

$$s=s_{1'} <> s_{2'} <> \dots <> s_{N'}.$$

We should prove that this decryption is executed. Indeed, regarding the way the public key is created, the coordinate matrix curve F from the step **D₁** satisfies the equation

$$F=L^{-1}(r_l)\cdot E\cdot R^{-1}(r_r)+\Gamma(r_c)=$$
$$L^{-1}(r_l)\cdot(L(r_l)\cdot(M(u_1,u_2,\ \dots\ ,u_N)-\Gamma(r_c))\cdot R^{-1}(r_r)+\Gamma(r_c),$$

from which it clearly follows that $F=M(u_1,\ u_2,\ \dots\ ,\ _N)$. Further, considering the way the coordinate matrix surface $M(u_1,u_2,\ \dots\ ,\ u_N)$ is created, we conclude that the decryption was correctly performed.

Since the coordinate matrix curves $L$, $\Gamma$ and $R$ are of finite dimensions, this algorithm as well as the algorithm presented in (Vesić, Simjanović, 2012) is linearly dependent on the number of characters in the message (Cormen, et al., 2009). More accurately, the complexity of this algorithm is $O(n)$ dependent on the number $n$ of characters.

The previous algorithm is graphically presented in Figure 1.

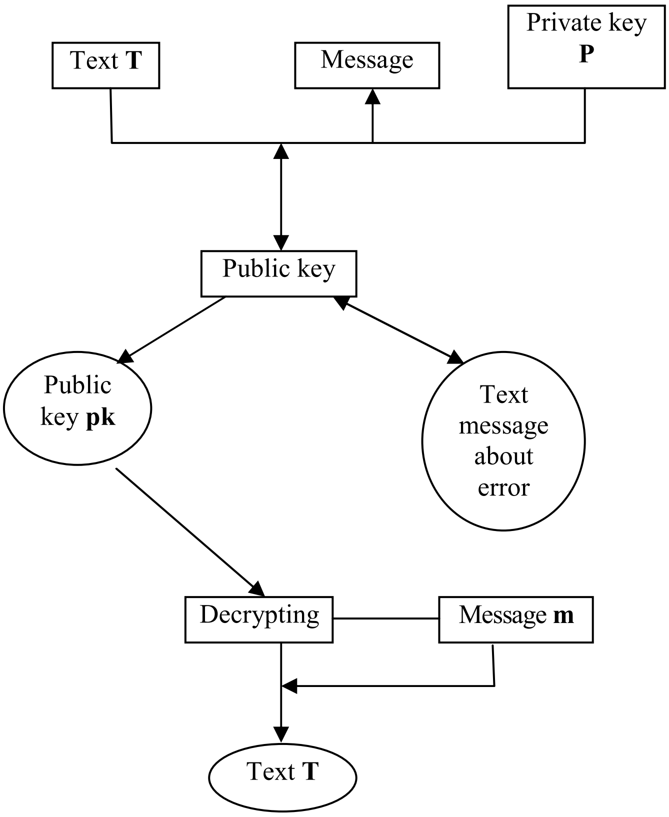

*Figure 1* – Encrypting-decrypting process
*Slika 1* – Proces šifrovanja i dešifrovanja

## *Programs*

In this part of the paper, the previously presented algorithm will be followed by programs in the Mathematica 8.0.software package (Wolfram, 1999) For the program encryption and decryption, matching subprograms will be used, the merging of which will create main programs. These subprograms will be used in the realization of some steps represented in the algorithm. Note that in these programs the coordinate matrix curve, signed as Γ in the algorithm, will be signed as CC.

### *Program covered encryption*

The encryption and decryption of a text with the previous presented method are covered by the programs in this part of the paper.

```
TextPairs[text_,Ch_]:=
     Module[{tp={}},
     For[i=1,i<=StringLength[text],i++,

     tp=Append[tp,Position[Ch,StringTake[text,{i}]][[1]]]];
tp];
```

This program transforms the text into a set of pairs of natural numbers which indicate the positions of the matching characters in the set Ch.

```
HidePairs[pairs_,Ch_]:=
        Module[{hp=pairs},
        For[i=1,i<=Dimensions[pairs][[1]],i++,

        hp[[i,2]]+=
        Dimensions[Ch[[pairs[[i,1]]]]][[1]]*
                        RandomInteger[{1,100}];
        hp[[i,1]]+=Dimensions[Ch][[1]]*

                        RandomInteger[{1,100}]];
hp];
```

This program hides the pairs by increasing the coordinates which is one of the reasons for this algorithm – breaking difficulty.

```
CompletePairs[pairs_,p_,q_]:=
        Module[{cp=pairs},
        For[i=1,Mod[Dimensions[cp][[1]],p*q]!=0,i++,
        cp=Insert[cp,{1,0},
                RandomInteger[{1,Dimensions[cp][[1]]}]]]];
```

This program supplements a set pairs of pairs to the set of pairs with K·p·q elements for a positive integer K.

```
PairsMatrices[pairs_,p_,q_]:=
        Module[{pm={}},
        d=Dimensions[pairs][[1]]/(p*q);
        For[i=1,i<=d,i++,pm=Append[pm,{}]];
        For[i=1,i<=Dimensions[pm][[1]],i++,
        For[j=1,j<=p,j++, pm[[i]]=Append[pm[[i]],

        Take[pairs,{(i-1)*p*q+(j-1)*q+1,(i-1)*p*q+j*q}]]]]];
```

This program orders a set of pairs of natural numbers. Accurately said, it transforms a set of pairs obtained using the CompletePairs program into a set of matrices of the order p×q whose elements are ordered pairs indicating to the positions of the characters that make up the text.

```
PairFunction[pair_,v_]:=
        Module[{pf},
        If[pair[[2]]==0,
        pf=(v-RandomInteger[{1,100}])*
                (v-RandomInteger[{1,100}]),
        pf=v^2-2*pair[[1]]*v+pair[[1]]^2+pair[[2]]^2];
Expand[pf]];
```

This program transforms a pair of numbers into a polynomial dependent on the variable *v*.

In the case when the second coordinate of the pair *pair* is different from 0, the roots of that polynomial are complex numbers whose real part indicates to the row in the matrix *Ch* where the character is, while the absolute value of the imaginary part indicates the matching column in the matrix *Ch*. Otherwise, it is a polynomial with real roots.

```
MatricesPairsFunctions[pairs_,variables_]:=
        Module[{mpf},
        If[Dimensions[variables][[1]]<Dimensions[pairs][
[1]],
        mpf="More variables are needed. Correct
that,please!",
        mpf=pairs;
        For[i=1,i<=Dimensions[pairs][[1]],i++,
        For[j=1,j<=Dimensions[pairs][[2]],j++,
        For[k=1,k<=Dimensions[pairs][[3]],k++,
        mpf[[i,j,k]]=PairFunction[mpf[[i,j,k]],variables[[i]]
]]]];
mpf];
```

This program transforms each of the matrices obtained by the PairsMatrices program using the PairFunction program into the coordinate matrix curves of the order *p*×*q* by transforming the *k*-th matrix of the set into the coordinate matrix curve dependent on the *k*-th variable in the set of variables. If there are an insufficient number of variables, the program indicates on this error.

```
HadamardProduct[pvf_]:=
        Module[{hp=pvf[[1]]},
        For[i=2,i<=Dimensions[pvf][[1]],i++,
        hp*=pvf[[i]]];
Expand[hp]];
```

This program transforms a set of matrices of the same type into the Hadamard product of these matrices (at the *(u,v)* place in the Hadamard product of the matrices $U_1, \dots , U_k$ is the product of the elements of these matrices at the places *(u,v)*).

```
Hiding[pvf_,L_,CC_,R_,l_,c_,r_]:=
        Module[{h},
        dl=Det[L/.t->l];
        dr=Det[R/.t->r];
        If[(dl==0)&&(dr==0),
                h="Incorrect left and right values.
        Correct that, please!",
        If[(dl==0)&&(dr!=0),
                h="Incorrect left value. Correct that,
        please!",
        If[(dl!=0)&&(dr==0),
                h="Incorrect right value. Correct that,
        please!",
        If[(dl!=0)&&(dr!=0),
        Ll=L/.t->l;Rr=R/.t->r;Cc=CC/.t->c;
```

This program hides the matrix by multiplying it with the numerical matrices obtained as coordinate matrix curves at a given point. If the determinant of the coordinate matrix curve *L* at a point *l* or the determinant of the coordinate matrix *R* at a point *r* is equal to 0, it is impossible to uniquely determine the coordinate matrix surface which is hidden. In that case, the result of the program is a text message which exactly determines the variable which causes the problem. Otherwise, the result of this program applied to the previous problem's results related to the same text encryption (the resulting matrices) is a public key. The message consists of the numbers *l, r* and *s* as well as the ordered set of the variables figuring in a public key.

```
PublicKey[text_,message_,ch_,p_,q_,L_,CC_,R]:=
        Module[{pk},
        u=MatricesPairsFunctions[PairsMatrices[
        CompletePairs[HidePairs[
        TextPairs[text,ch],ch],p,q],p,q],message[[2]]];
        If[StringQ[u],pk=u,u=HadamardProduct[u]];
        u=Hiding[u,L,CC,R,message[[1,1]],
        message[[1,2]],message[[1,3]]];
        If[StringQ[pk],pk=pk,pk=u];
```

This program integrates all previous programs related to encryption and as a result brings back a public key if the entered variables are correct or a text message if some of the numerical parameters are not correctly assigned.

### *Program covered decryption*

The steps in the decryption of the previously presented algorithm are covered by the programs in this part of the paper.

```
nForm[n_]:=
        Module[{nsf={}},
        For[i=1,i<=n,i++,nsf=Append[nsf,{}]];
nsf];
```

This program is the first auxiliary program which forms a set of *n* empty sets for a natural number *n*.

```
PolynomialDegree[pol_,var_]:=
        Dimensions[CoefficientList[pol,var]][[1]]-1;
```

This is the second auxiliary program. This program returns a degree of a polynomial *pol* depending on a variable var. The polynomial degrees over the field of integer numbers are of special interest in the decryption shown here.

```
Mod1n[a_,b_]:=Module[{m=Mod[a,b]},If[m==0,m=b];m];
```

This program gives back the operation's Mod1n result.

```
Dehiding[publickey_,message_,L_,CC_,R_]:=
        Expand[
        Expand[
        Inverse[L/.t->message[[1,1]]].publickey].
        Inverse[R/.t->message[[1,3]]]]+
        CC/.t->message[[1,2]];
```

This program realizes the step $\mathbf{D_1}$.

```
FactorCharacter[f_,ch_]:=
        Module[{fc},d=Discriminant[f[[1]],f[[2]]];
        If[(PolynomialDegree[f[[1]],f[[2]]]==2)&&
        (d<0), cl=CoefficientList[f[[1]],f[[2]]];
        r=Mod1n[-cl[[2]]/2,Dimensions[ch][[1]]];
        c=Mod1n[Abs[Sqrt[d]]/2,
                Dimensions[ch[[r]]][[1]]];
        fc=ch[[r,c]],
        fc=""];
fc];
```

This program transforms the polynomial into the character which that polynomial represents or into an empty string if the polynomial does not encrypt any character.

```
FactorsCharactersPositions[m_,ch_]:=
        Module[{ff={},fv=m},
        For[i=1,i<=Dimensions[m][[1]],i++,
        For[j=1,j<=Dimensions[m][[2]],j++,
        fv[[i,j]]=Delete[FactorList[m[[i,j]]],{1}];
        For[k=1,k<=Dimensions[fv[[i,j]]][[1]],k++,
        fv[[i,j,k,2]]=Variables[fv[[i,j,k,1]]][[1]];
        fv[[i,j,k,1]]=FactorCharacter[fv[[i,j,k]],ch]]]];
fv];
```

This program transforms the factors of all polynomials in the matrix surface into pairs. The first coordinate is matching the one-variable polynomial which is the divisor of the polynomial into the matrix surface, while the second coordinate is a variable which the divisor depends on.

```
PairsMessagePositions[pairs_,message_]:=

        Module[{pmp=pairs},
        For[i=1,i<=Dimensions[pairs][[1]],i++,
        For[j=1,j<=Dimensions[pairs[[i]]][[1]],j++,
        For[k=1,k<=Dimensions[pairs[[i,j]]][[1]],k++,
        pmp[[i,j,k,2]]=
        Position[message[[2]],pairs[[i,j,k,2]]][[1,1]]]]];
pmp];
```

This program transforms the pairs obtained by the previous program by replacing the variable on the second coordinate with the position of that coordinate in the part *message* in which the variables are arranged.

```
PairsMessageSets[pairs_,message_]:=
        Module[
        {pms=nForm[Dimensions[message[[2]]][[1]]]},
        For[i=1,i<=Dimensions[pairs][[1]],i++,
        For[j=1,j<=Dimensions[pairs[[i]]][[1]],j++,
        For[k=1,k<=Dimensions[pairs[[i,j]]][[1]],k++,
        pms[[pairs[[i,j,k,2]]]]=
        Append[pms[[pairs[[i,j,k,2]]]],pairs[[i,j,k,1]]]]]];
pms];
```

This program collocates the first coordinates of the pairs into the ordered sets of the polynomials which appear in the same order as in the encryption and which depend on the same variable. The previously mentioned first coordinates are appropriate characters.

```
SetsText[sets_]:=
        Module[{st=""},
        For[i=1,i<=Dimensions[sets][[1]],i++,
        For[j=1,j<=Dimensions[sets[[i]]][[1]],j++,
        st=StringInsert[st,sets[[i,j]],-1]]];
st];
```

This program transforms the sets obtained by the PairsMessageSets program into the decrypted text which is, as it is proved, identical to the encrypted text.

```
Decrypting[message_, publickey_, L_, CC_, R_, ch_] :=
        SetsText[PairsMessageSets[
        PairsMessagePositions[
        FactorsCharactersPositions[
        Dehiding[publickey, message, L, CC, R], ch],
        message], message]];
```

This program integrates all previous programs related to encryption and as a result brings back the text encrypted by a public key.

*Test example*

An example of the encryption of a text with the previous presented algorithm is given bellow. Let

– *t*=L O L I T I C A<br>
– *L={{t, t+3 },{t+2,t+2 }}*;<br>
– *R={{t,0,0,0 },{0,t+1,0,0 },{0,0,t+2,0 },{0,0,0,t+3 }}*;<br>
– *CC={{t,t,t,t },{t+1, t, t-1,* $t^2$ *}};*<br>
– *message={{1,1,1 },{a, b}}*;<br>
– *p=2, q=3*;<br>
– *Ch*= {{"A","B","C"},{"a","b","c"},{"D","E","F"},{"d","e","f"," "},
{"G","H","I"},{"g","h","i"},{"J","K","L"},{"j","k","l"},{"M","N","O"},
{"m","n","o"},{"P","Q","R","S"},{"p","q","r","s"},{"T","U","V"},
{"t","u",v"}};<br>
– pk=PublicKey[t, message, Ch, p, q, L, CC, R]<br>
– dt=Decrypting[message, pk, L, CC, R, Ch]

This example is similar to the example in (Vesić, Simjanović, 2012). The time necessary for the realization of forming the public key *pk* is $t_1=0.466s$. The result of the decryption is *dt*=L O L I T I C A. The time used is $t_2=0.776s$.

# Conclusion

The matrix-based algorithm for text-data hiding is presented in this paper.This algorithm is covered by the corresponding programs in the MATHEMATICA 8.0 software package. These programs could be used in earlier versions of this software package.

The algorithm presented in this paper is a preferment of the algorithm presented in (Vesić, Simjanović, 2012). The algorithm presented in that paper has the same idea for text-data hiding (factorization of matrices) but the algorithm in this paper hides texts with the multiplication of an encoded text with two matrices. The multiplication of a numerical matrix and a coordinate matrix surface is not safe enough hiding (systems of Diophantine equations with linear unknown variables which are not so complicated to solve are the reason for that insecurity).

For breaking a public key as a result of applying the algorithm for text-data hiding presented in this paper, it is necessary to solve a system of Diophantine equations which have products of unknown variables. These systems are more complicated to solve.

The time used for encrypting a text with the algorithm presented in this paper, for the same text as in (Vesić, Simjanović, 2012), is *0.002s* longer. The time necessary for the decryption is *0.279s* longer than in (Vesić, Simjanović, 2012).

## References

Bohen, D., 1999, Twenty Years of Attacks on the RSA Cryptosystem, *Notices of the AMS*, 46(2), pp. 203-213.

Buchmann, A. J., 2001, *Introduction to Cryptography*, New York, Springer-Verlag.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2009, *Introduction to algorithms*, Third Edition, Cambridge, Massachusetts London, England, The MIT Press.

Diffie, W., Hellman, M., 1976, *New directions in Cryptography*, IEEE Transactions on Information Theory, IT-22.

Kuljanski, R. S., 2010, RSA algoritam i njegova praktična primena, *Vojnotehnički glasnik/Military Technical Courier*, 58(3), pp. 65-77.

Menezes, A., 2001, *Evaluation of Security Level of Cryptography: RSA-OAEP, RSAPSS, RSA Signature*, CRYPTREC.

Menezes, A., Oorschot, P., Vanstone, S., 1996, *Handbook of Applied Cryptography*, CRC Press, Boca Raton.

Vesić N. O., *Differential Geometry and Matrices* [Internet], Available in: poincare.matf.bg.ac.rs/~geometricalseminar/presentations/vesic.pdf, Downloaded: February 1, 2013.

Vesić N. O., Simjanović D. J., 2012, Tensors and Cryptography, pp. 76-83, *InfoTech-2012*, September 20-21.

Wolfram, S., 1999, *The Mathematica Book*, *4th ed.*, Wolfram Media/Cambridge University Press.

## MATRIČNO ZASNOVAN ALGORITAM ZA SKRIVANJE TEKSTOVA I OBRADU INFORMACIJA

OBLAST: matematika

VRSTA ČLANKA: originalni naučni članak

*Sažetak:*

*Primena diferencijalne geometrije u skrivanju tekstualnih podataka prikazana je u ovom radu. Ta primena predstavlja generalizaciju prethodno predstavljenih metoda. Matrično vrednosne funkcije više promenljivih korišćene su u ovom radu. Metod je pokriven odgovarajućim programima u softverskom paketu MATHEMATICA 7.0.*

Ključne reči: *program; dešifrovanje; šifrovanje; tekst; poruka; javni ključ; privatni ključ; algoritam; matrica vrednosti funkcije.*

Datum prijema članka/Paper received on: 27. 05. 2013.

Datum dostavljanja ispravki rukopisa/Manuscript corrections submitted on: 20. 06. 2013.

Datum konačnog prihvatanja članka za objavljivanje/ Paper accepted for publishing on: 22. 06. 2013.