ISSN 1512-1232

UDC: 681.142.001

Cryptographic System of high Stability

Tamaz Chantladze*, Zurab Kipshidze*, Mziana Nachkebia*, Duglas Ugulava*

* Niko Muskhelishvili Institute of Computational Mathematics, Georgian Technical University, 77, Kostava str., 0175, Tbilisi, Georgia

duglasugu@yahoo.com, mzianachkebia@yahoo.com

Abstract

A symmetric algorithm of high stability of information encryption is considered, in which the best properties of American standards DES and RINDAEL are used. On the positive side, we note the complete identity of the processes of encryption and decryption. Relevant fragments of practical implementation of algorithm in the medium of MATLAB is given.

Keywords: *cryptographic system, encryption and decryption, symmetric algorithm.*

1. Introduction

The paper considers a symmetric block ciphering algorithm that is stable with respect to known cryptographic attacks. In contrast to [1], high stability is achieved by increasing the encryption block and accordingly, the encryption key to 128 bits. The encryption algorithm uses the best properties of American standards DES and RINDAEL [2]. On the positive side, we should especially note the complete identity of the processes of encryption and decryption. This is a difference from the system RINDAEL, in which the identity of the encryption and decryption is achieved with the help of additional actions, which complicates the system and is considered as its disadvantage.

In our system we use a new nonlinear element, which almost does not contain any sign of linearity.

Below, in figures 1 and 2, the schemes of encryption and decryption of derived keys are given, as well as relevant fragments of their practical implementation in the medium of MATLAB.

2. The algorithm

Encryption and decryption is carried out in 8 rounds.

The 128 bit information to be encrypted is denoted by T. The 128 bit basic key, which is known only to the encoder and receiver of the encoded information, is denoted by K.

After 8 rounds from the basic key, 8 generated keys $K_1, K_2, \cdots, K_8$ are accepted. The scheme of their reception is carried out in the following way: a sequence composed of 128 bits of K is represented as 4×8 dimensional matrices A_1, A_2, A_3, A_4 (the 128 bit information K is represented as a matrix $A(4 \times 32)$, the first line 32 bits from K, the second line by the next 32 bits, and so on. A_1 is composed of the first 8 columns from A, A_2 -of the second 8 columns and so on).

```
K(1:128) % Primary key
% representation of the key K as the matrix A(4× 32)
A=[K(1:32);K(33:64);K(65:96);K(97:128)];
% the creation of 4 matrices with 8 columns from the matrix A
A1=A(1:4,1:8);
A2=A(1:4,9:16);
A3=A(1:4,17:24);
A4=A(1:4,25:32);
```

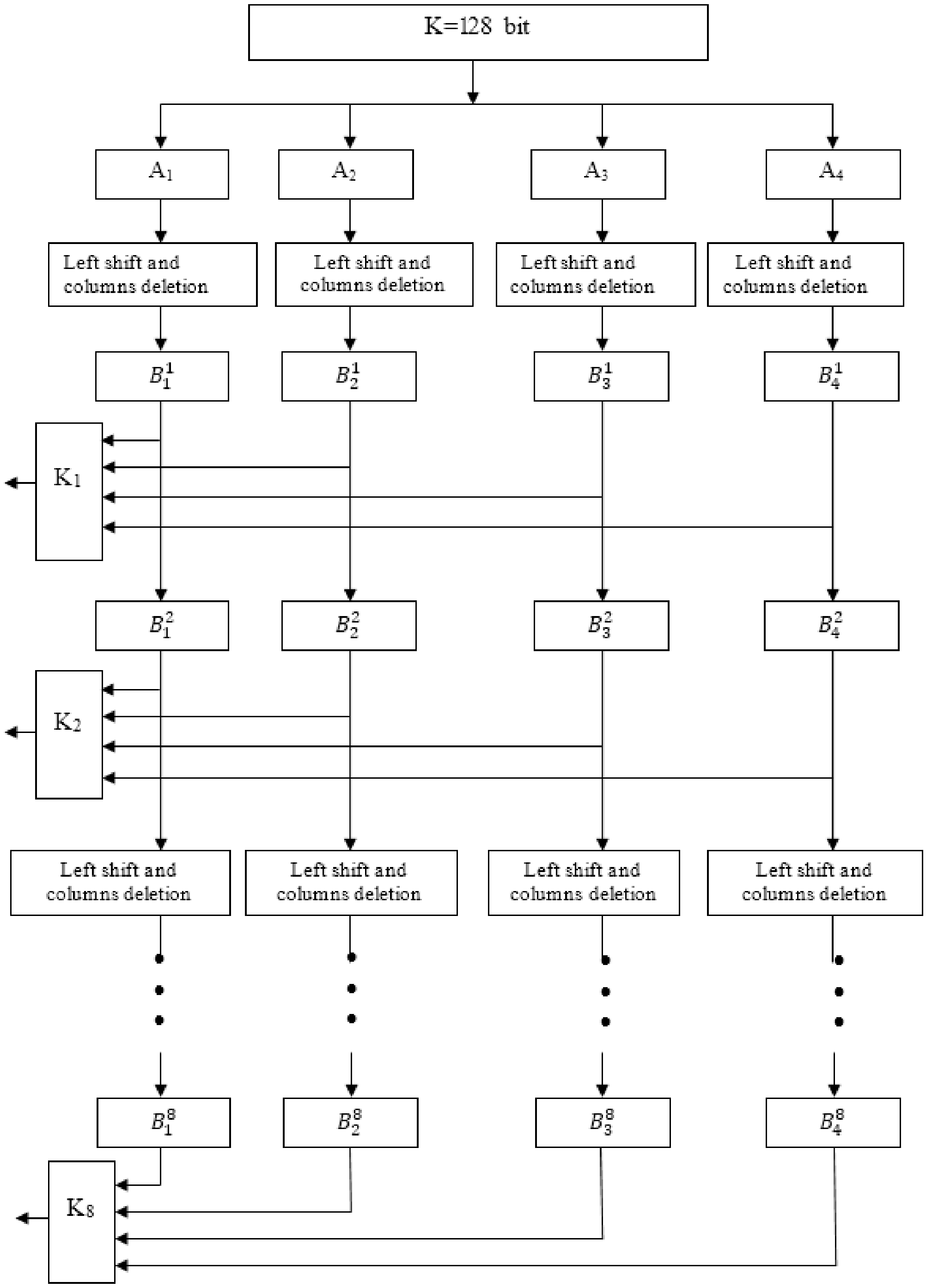

Fig.1

The resulting matrix blocks in each round are shifted to the left by one column. In the obtained matrices $A_1^1, A_2^1, A_3^1, A_4^1$ the I, III, V, VII columns are deleted.

```
function [B,C]=sveti(A)
% replacement of column and deletion of 4 column,B-replaced,C-removed
   B=A;
   B(:,9)=B(:,1); B(:,1)=[];
   C=B;
   m=[1,3,5,7];
   C(:,m)=[];
end
```

ISSN 1512-1232

After joining the obtained matrices $B_1^1, B_2^1, B_3^1, B_4^1$ (by attaching to each other) we obtain a matrix of dimension 4×16, which is the first generated key K_1. To obtain the key K_2, we shift to the left the obtained in the first round matrices $A_1^1, A_2^1, A_3^1, A_4^1$. In the obtained matrices $A_1^2, A_2^2, A_3^2, A_4^2$ the column I, III, V, VII are again removed. After joining the obtained matrices $B_1^2, B_2^2, B_3^2, B_4^2$ (by attaching to each other) we obtain a matrix of dimension 4×16, which is the second generated key K_2. Similarly, the remaining keys $K_3, \cdots, K_8$ are obtained.

```
% creating generated keys
for i=1:8
     [B1,C1]=sveti(A1);
     [B2,C2]=sveti(A2);
     [B3,C3]=sveti(A3);
     [B4,C4]=sveti(A4);
A1=B1; A2=B2; A3=B3; A4=B4;
   if i==1
     K1=[C1 C2 C3 C4] % the first generated key
   elseif i==2
   % . . .
   elseif i==8
       K8=[C1 C2 C3 C4] %the eighth generated key
   end
 end
```

To encrypt the data, we perform the operation $S = T \oplus K$, where $\oplus$ denotes addition modulo 2. The obtained 128 bit sequence is written in the form of (4 × 32) matrix, which is further divided into two (4 × 16) matrices L_0 and R_0 (the first matrix L_0 is represented by the first 8 columns, and the second one by the last 8 columns). The algorithm represents 8 rounds, which are carried out according to the following scheme

$$L_i = R_{i-1}, \quad R_{i-1} = L_{i-1} * f(R_{i-1}, K_i), \quad i = 1,2,\cdots,8,$$

where K_i is the generated key received in the i – th round.

The function $f(R_{i-1}, K_i)$ represents below a certain composition $f_1 * f_2$ of $f_1(R_{i-1}, K_i)$ and some fourth-order binary martix f_2. Here $f_1(R_{i-1}, K_i) = a_i x_{i-1} \oplus b_i y_{i-1}$, where a_i (respectively b_i), is the matrix composed of the first (respectively, the last 8) columns of the generated key K_i. x_{i-1} (respectively, y_{i-1}) is the matrix composed of the first (respectively, the last 8) column of the matrix R_{i-1}. The product $a_i x_{i-1}$ is obtained as follows: the first rows of matrices a_i and x_{i-1} are multiplied as ordinary double numbers. A double number is obtained, the number of bits of which does not exceed 16. We regard it as a 16-digit binary number, the initial bits of which, if necessary, are padded with zeros. The resulting number is considered as the first row of a matrix of dimension 4 × 16. Similarly, the second rows of matrices a_i and x_{i-1} are multiplied as ordinary double numbers. The resulting number is considered as the second row of the matrix 4 × 16 and so on. Similarly it turns out the product $b_i y_{i-1}$. Adding the matrices $a_i x_{i-1}$ and $b_i y_{i-1}$ modulo 2, we obtain the matrix of dimension 4 × 16 $f_1(R_{i-1}, K_i) = a_i x_{i-1} \oplus b_i y_{i-1}$. As f_2 we use some binary four-dimensional matrix. $f_1 * f_2$ is the double product of 4 × 4 matrices f_2 and 4 × 16 matrices f_1 that has dimension 4 × 16.

ISSN 1512-1232

```
function res = bin_mult(a,b,n,varargin)
% usage res = bin_mult(a,b[,n])
% Function multiplies two binary numbers
% a and b written as char arrays.
% Result is a binary number as char array.
% n is the number of digits in output binary number
   if nargin<2
      error('At least two parameters must be passed to bin_mult');
   elseif nargin<3
      n = 32;
   end
   q=quantizer([n 0]);
   a1=bin2num(q,a);
   b1=bin2num(q,b);
   y = a1 * b1;
   res = num2bin(q,y);
end
```

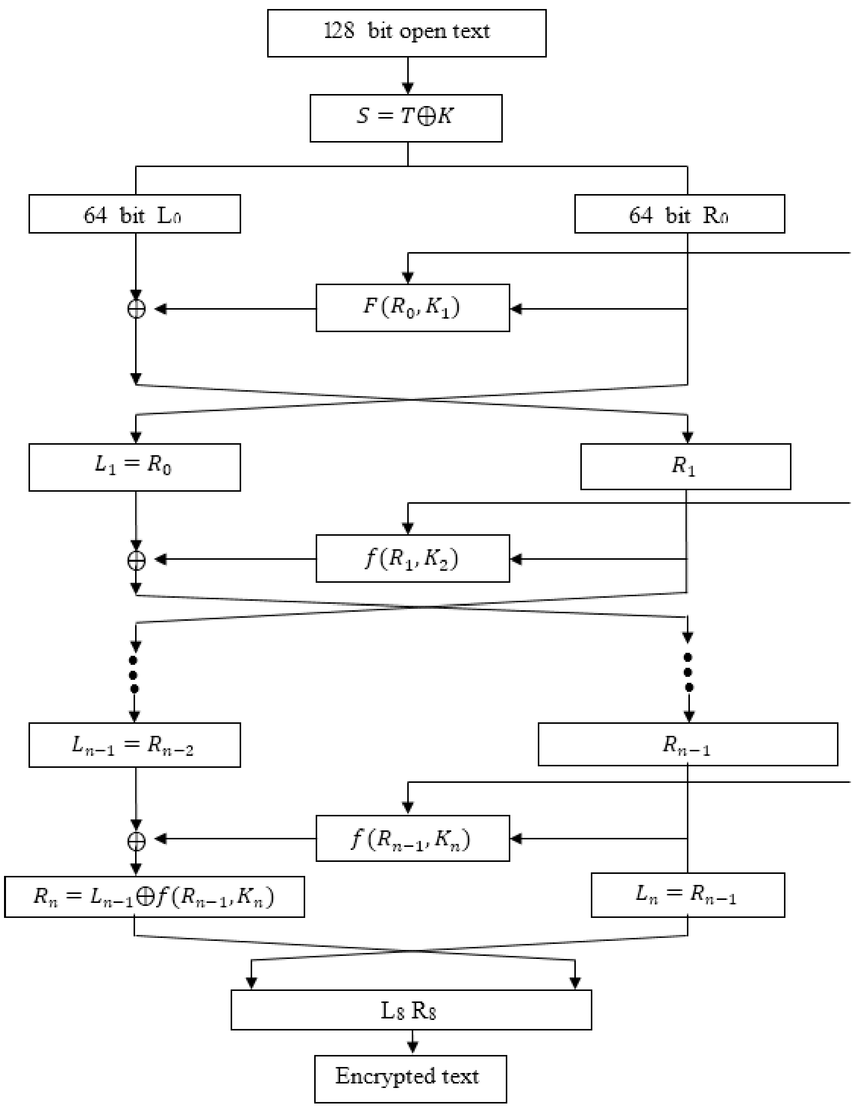

Fig.2

```
function R=f(L,K)
% for i-th round is computed f = f1 * f2,
% where f1(R_{i-1},K_i) = a_i x_{i-1} ⊕ b_i y_{i-1}, R_{i-1} = L_{i-1} * f(R_{i-1},K_i),
% K_i –generated key
   f2=['0101';'1101';'1011';'1110'];
   for j = 1:4
       a=K(j,1:8); b=K(j,9:16);
       x=L(j,1:8); y=L(j,9:16); aa=[num2str(a(1)),...,num2str(a(8))];
       bb=[num2str(b(1)),...,num2str(b(8))];
       xx=[num2str(x(1)),...,num2str(x(8))];
       yy=[num2str(y(1)),...,num2str(y(8))];
       ax=bin_mult(aa,xx,16);
       by=bin_mult(bb,yy,16);
        ff=mod(ax+by,2);
        f1=(j,1:16)=ff;
     end
       fff=f2*f1;
       R=mod(L+fff,2);
 end
```

Obtained in the 8-th last round matrices L_8 and R_8, by means of their adjunction, are combined in the form of the 4×32 dimensional matrix L_8R_8. Farther, the resulting matrix, by adjunction their rows in order, will give 128 bit single-line encrypted information, that is, the cryptogram E, which is fed to the output of the system.

```
   % encription
   % 128 bit encrypted text T
   % T + K – addition by modulo 2
      S1=mod(T+K,2);
   % representation of 128 bit S1 text in the form of the matrix (4x32)
      S=[S1(1:32); S1(33:64); S1(65:96); S1(97:128)];
          L0=S(:,1:16),  R0=S(:,17:32);  L1=R0
         R1=f(L0,K1),  L2=R1;
         R2=f(L1,K2),  L3=R2;
         R3=f(L2,K3),  L4=R3;
         R4=f(L3,K4),  L5=R4;
         R5=f(L4,K5),  L6=R5;
         R6=f(L5,K6),  L7=R6;
         R7=f(L6,K7),  L8=R7;
         R8=f(L7,K8)
      LR=[L8 R8]
      L8R8=[LR(1,1:32) LR(2,1:32) LR(3,1:32) LR(4,1:32)]
```

The decryption scheme differs from the encryption scheme in that the generated keys are used in the reverse order, that is, in the first cycle - K_8, in the second - K_7, at the end - K_1 is used.

ISSN 1512-1232

3. Numerical realization of the algorithm

We present the results of the numerical realization of the algorithm described above on one example

Initial information:

K= [1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1];

T= [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
1 0];

Results of implementation:

K1 =

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

. . .

K8 =

1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1

E= [0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 1 1 0 0 1 0 1]

4. Remarks

It should be noted that the use in the process of creating the generated keys of the left shift operation onto one column and the removal of odd columns, as well as the choice of the matrix f_2, are performed according to the preliminary agreement between the cipher and the recipient. In the process of transferring other information it is important to use a different agreement, for example, left shift onto two columns, deletion of even columns, use another matrix f_2. The fact that such agreements are not available to other persons, creates great additional difficulties for deciphering information.

Finally, we should also note one remarkable property of the system, which consists in asymmetry with respect to the operation of the complement. It is well known, symmetry reduces time for opening the cipher by 50%.

ISSN 1512-1232

REFERENCES

[1]. W. Diffie and M. E. Hellman. Privacy and Authentication: An Introduction to Cryptography. Proceedings of the IEEE, Vol. 67, March 1979, pp. 397-427.

[2]. A. Chaduneli, M. Chochrauli, Z. Kipshidze. Cryptographic System of high Stability. Proceedings of Georgian Technical University. N 1(471), 2009, (in Georgian).

Article received: 2017-09-17