Exploring the Modified IDEA

Using the `topicIDEA` Package

Academy of Management in Legnica, Legnica, Poland,
Faculty of Computer Science,
Wroclaw University of Applied Informatics, Wroclaw, Poland

e mail: c.koscielny@wsm.edu.pl

Abstract

In the worksheet it has been shown how to use the IDEA algorithm for encryption 64/32/16-bit data with keys 832-bit or less.

1. Introduction

Normally, the IDEA operates on 64-bit blocks using a 128-bit user-selected secret key. Out of this 128-bit key fifty two key subblocks of 16-bit are generated and used in the encryption process. The 52 16-bit key subblocks needed in decryption are computed from the encryption key subblocks. In the modified mode of encryption/decryption by means of the IDEA algorithm the user does not select the 128-bit key but chooses directly the 52 16-bit key subblocks for encryption and employs them as a secret key. In this way IDEA can protect a 64-bit block of data using 832-bit key. By making this new secret key partly public we can encipher/decipher by means of the IDEA using a key of an arbitrary length, but no longer than 832 bits.

To use the **`topicIDEA`** package the user should download the files **`topicIDEA.mla`** and **`topicIDEA.hdb`** from the Maple Application Center [1], should place them in the directory having the name defined by the variable **`dn`**, and ought to execute the statements:

```
> restart:
dn := "g:/idea02":
#it is assumed that you store the package in the directory
"g:/idea02":
currentdir(dn);
march('open', "topicIDEA.mla"):
```

```
with(topicIDEA);
```

```
"
G:\idea02"

libname := "C:\Program Files\Maple 10/lib","topicIDEA.mla"

[A, HexS2bl, HexS2n16b, M, X, dbhist, i16cbcd, i16cbce,
   i16ecbd, i16ecbe, i32cbcd, i32cbce, i32ecbd, i32ecbe, i64cbcd,
   i64cbce, i64ecbd, i64ecbe, iddea, idea, idispks, iksb, iksd,
   iksdm, iksgc1, iksgc2, iksgmd, ip24, ip42, ip48, ip84, mtf,
   sbdbarr, sbhist, up]
```

The worksheet **idea02.mw** should also be stored in the directory **dn**.

After carrying out the above statements, you may want to see the **topicIDEA** help page. If yes, execute:

```
> ?topicIDEA
```

We are now ready to test the modified mode of operation of the IDEA algorithm.

2. Modified Encryption/Decryption of 64-bit Block of Plaintext/Cryptogram - Examples

The first example concerns encryption with the 832-bit key:

```
> MD := 2^16:
n := 2:
printf("key length = %d%s",52*16, " bits");
key length = 832 bits
```

Now we generate at random a key schedule for encryption, playing the role of the 832-bit secret key, using the seed **123**:

```
> Z:=iksgmd(123);
```

$$Z := \begin{bmatrix} 52734 & 15725 & 28030 & 17730 & 61404 & 54370 \\ 63206 & 15377 & 7763 & 63594 & 46203 & 58937 \\ 23766 & 22241 & 96 & 55409 & 23166 & 53295 \\ 53321 & 5664 & 33710 & 55520 & 111 & 36249 \\ 17747 & 35662 & 3325 & 31140 & 14944 & 64720 \\ 17476 & 7985 & 50999 & 45507 & 26626 & 6484 \\ 49191 & 17730 & 55892 & 30255 & 6648 & 22461 \\ 23728 & 17543 & 59241 & 4195 & 2428 & 8028 \\ 3252 & 17766 & 28513 & 53461 & 0 & 0 \end{bmatrix}$$

The encryption key schedule in hexadecimal notation:

```
> idispks(Z);
Key schedule:
round No. 1: CDFE 3D6D 6D7E 4542 EFDC D462
round No. 2: F6E6 3C11 1E53 F86A B47B E639
round No. 3: 5CD6 56E1 0060 D871 5A7E D02F
round No. 4: D049 1620 83AE D8E0 006F 8D99
round No. 5: 4553 8B4E 0CFD 79A4 3A60 FCD0
round No. 6: 4444 1F31 C737 B1C3 6802 1954
round No. 7: C027 4542 DA54 762F 19F8 57BD
round No. 8: 5CB0 4487 E769 1063 097C 1F5C
round No. 9: 0CB4 4566 6F61 D0D5
```

Let us use the message **Mersenne** as our plaintext block. Since the IDEA accepts four 16-bit plaintext subblocks, we must convert 8 bytes of the message into four subblocks:

```
> convert("Mersenne", bytes):
m := ip84(%); #see help page
```

$$m := [19813, 29299, 25966, 28261]$$

After doing it we are ready to encrypt the message represented by the variable `m` and to observe the encryption process:

```
> c := iddea(m);
input:              4D65 7273 656E 6E65
after round No. 1: EBCA 55D6 F6CC C801
after round No. 2: 5390 569A 2221 B591
```

```
after round No. 3: F79D A90F E486 AA7D
after round No. 4: 7EFC 7498 212B 8AF6
after round No. 5: 6A88 B1F2 A069 561E
after round No. 6: 612A 4932 FF5A 3577
after round No. 7: 9E2B 43FD DF57 C308
after round No. 8: B083 8E0E 0F4D 5854
output:            375A 54B3 FD6F 75D7
```

$$c := [14170,\ 21683,\ 64879,\ 30167]$$

To decrypt the obtained cryptogram **c**, we need to determine the key schedule for decryption:

```
> Z := iksd(Z);
```

$$Z := \begin{bmatrix} 38109 & 47770 & 37023 & 43183 & 2428 & 8028 \\ 2472 & 6295 & 47993 & 64709 & 6648 & 22461 \\ 20989 & 9644 & 47806 & 26074 & 26626 & 6484 \\ 57343 & 14537 & 57551 & 25760 & 14944 & 64720 \\ 41419 & 62211 & 29874 & 16178 & 111 & 36249 \\ 51465 & 31826 & 59872 & 54166 & 23166 & 53295 \\ 34947 & 65440 & 43295 & 32102 & 46203 & 58937 \\ 731 & 57773 & 50159 & 13357 & 61404 & 54370 \\ 12961 & 49811 & 37506 & 22426 & 0 & 0 \end{bmatrix}$$

The key schedule for decryption in hexadecimal:

```
> idispks(Z);
Key schedule:
round No. 1: 94DD BA9A 909F A8AF 097C 1F5C
round No. 2: 09A8 1897 BB79 FCC5 19F8 57BD
round No. 3: 51FD 25AC BABE 65DA 6802 1954
round No. 4: DFFF 38C9 E0CF 64A0 3A60 FCD0
round No. 5: A1CB F303 74B2 3F32 006F 8D99
round No. 6: C909 7C52 E9E0 D396 5A7E D02F
round No. 7: 8883 FFA0 A91F 7D66 B47B E639
round No. 8: 02DB E1AD C3EF 342D EFDC D462
round No. 9: 32A1 C293 9282 579A
```

Here is the decryption process:

```
> mr := iddea(c);
input:              375A 54B3 FD6F 75D7
after round No. 1: F84D C6C0 8884 DF9D
after round No. 2: 0478 D9AE 8E74 922B
after round No. 3: 4FB8 67A0 D123 1B0E
after round No. 4: F552 2E28 FFE6 0991
after round No. 5: 6250 6834 BF2F 14F2
after round No. 6: 7C13 2281 AD7B E380
after round No. 7: 1015 151F 91E7 0657
after round No. 8: 6CF0 D2EC AFE0 912D
output:             4D65 7273 656E 6E65
```

$$mr := [19813, 29299, 25966, 28261]$$

The retrieved plaintext block is correct:

```
> ip48(mr): #see help page
convert(%, bytes);
```

```
"
Mersenne"
```

Assume now that we publish the following key schedule for encryption:

```
> Z := Matrix(9, 6,
{(1, 1) = k[1],  (1, 2) = 56312, (1, 3) = 53605,
 (1, 4) = 60862, (1, 5) = 5819,  (1, 6) = 21621,
 (2, 1) = k[2],  (2, 2) = 15830, (2, 3) = 61240,
 (2, 4) = 39567, (2, 5) = 7420,  (2, 6) = 8119,
 (3, 1) = k[3],  (3, 2) = 46749, (3, 3) = 32755,
 (3, 4) = 30469, (3, 5) = 22510, (3, 6) = 45058,
 (4, 1) = k[4],  (4, 2) = 19673, (4, 3) = 13581,
 (4, 4) = 24889, (4, 5) = 46192, (4, 6) = 34700,
 (5, 1) = k[5],  (5, 2) = 56701, (5, 3) = 25294,
 (5, 4) = 51040, (5, 5) = 37454, (5, 6) = 8996,
 (6, 1) = k[6],  (6, 2) = 35369, (6, 3) = 25589,
 (6, 4) = 64115, (6, 5) = 21144, (6, 6) = 8367,
 (7, 1) = k[7],  (7, 2) = 46961, (7, 3) = 20213,
 (7, 4) = 55905, (7, 5) = 25923, (7, 6) = 31741,
 (8, 1) = k[8],  (8, 2) = 16401, (8, 3) = 33946,
 (8, 4) = 25021, (8, 5) = 61848, (8, 6) = 36574,
 (9, 1) = k[9],  (9, 2) = 50601, (9, 3) = 36321,
 (9, 4) = 16448, (9, 5) = 0,     (9, 6) = 0}):
```

In the above case our choice of a secret key consists in determining nine 16-bit subblocks **k**. Evidently, the number of elements of the variable **k** may be arbitrary, but less than 53. Let

```
> k := [38157,55984,18626,55330,64243,1676,44249,18157,16858]:
printf("key length = %d%s",9*16, " bits");
key length = 144 bits
```

Here is the key schedule for encryption with the 144-bit secret key:

```
> idispks(Z);
Key schedule:
round No. 1: 950D DBF8 D165 EDBE 16BB 5475
round No. 2: DAB0 3DD6 EF38 9A8F 1CFC 1FB7
round No. 3: 48C2 B69D 7FF3 7705 57EE B002
round No. 4: D822 4CD9 350D 6139 B470 878C
round No. 5: FAF3 DD7D 62CE C760 924E 2324
round No. 6: 068C 8A29 63F5 FA73 5298 20AF
round No. 7: ACD9 B771 4EF5 DA61 6543 7BFD
round No. 8: 46ED 4011 849A 61BD F198 8EDE
round No. 9: 41DA C5A9 8DE1 4040
```

and the encryption process of the plaintext **2*17*59:**:

```
> convert("2*17*59:", bytes);
m := ip84(%);
c := iddea(m);
```

$$[50, 42, 49, 55, 42, 53, 57, 58]$$

$$m := [12842, 12599, 10805, 14650]$$

```
input:              322A 3137 2A35 393A
after round No. 1:  FD68 E61C 2DFD D53A
after round No. 2:  0459 F665 F106 AC5F
after round No. 3:  88EC AA23 295A B2E0
after round No. 4:  8B5E 0EF9 2187 B889
after round No. 5:  EF22 E6EA 846E 4AB1
after round No. 6:  3D48 5950 DE3A B158
after round No. 7:  3C48 39CF D8C7 4317
after round No. 8:  8A1F BA8B 9C13 C0AD
output:             59DF 61BC 486C 3AE5
```

$$c := [23007, 25020, 18540, 15077]$$

The key schedule for decryption:

```
> Z := iksd(Z);
idispks(Z);
```

$$Z := \begin{bmatrix} 58551 & 14935 & 29215 & 65023 & 61848 & 36574 \\ 3891 & 31590 & 49135 & 6983 & 25923 & 31741 \\ 56723 & 45323 & 18575 & 6321 & 21144 & 8367 \\ 6061 & 39947 & 30167 & 61343 & 37454 & 8996 \\ 39150 & 40242 & 8835 & 11668 & 46192 & 34700 \\ 12848 & 51955 & 45863 & 60663 & 22510 & 45058 \\ 15816 & 32781 & 18787 & 39345 & 7420 & 8119 \\ 52125 & 4296 & 49706 & 34000 & 5819 & 21621 \\ 50632 & 9224 & 11931 & 39925 & 0 & 0 \end{bmatrix}$$

```
Key schedule:
round No. 1: E4B7 3A57 721F FDFF F198 8EDE
round No. 2: 0F33 7B66 BFEF 1B47 6543 7BFD
round No. 3: DD93 B10B 488F 18B1 5298 20AF
round No. 4: 17AD 9C0B 75D7 EF9F 924E 2324
round No. 5: 98EE 9D32 2283 2D94 B470 878C
round No. 6: 3230 CAF3 B327 ECF7 57EE B002
round No. 7: 3DC8 800D 4963 99B1 1CFC 1FB7
round No. 8: CB9D 10C8 C22A 84D0 16BB 5475
round No. 9: C5C8 2408 2E9B 9BF5
```

Decryption process:

```
> mr := iddea(c);
ip48(mr);
convert(%, bytes);
```

```
input:              59DF 61BC 486C 3AE5
after round No. 1: 6DF5 5D61 79E0 255E
after round No. 2: 28A8 2D2F 10C1 8B11
after round No. 3: 8C7B E863 7113 1E71
after round No. 4: 8D9D 8455 EC76 22A9
after round No. 5: DBC0 5E67 F6FC 6FF2
after round No. 6: 5236 70F9 AD02 36B8
after round No. 7: EF09 1D35 23F2 7EAB
after round No. 8: E0EE FB9A 0D2F F5E8
output:             322A 3137 2A35 393A
```

$$mr := [12842, 12599, 10805, 14650]$$

$$[50, 42, 49, 55, 42, 53, 57, 58]$$

"
2*17*59:"

```
> k := 'k': #the variable k will be used later
```

3. Modified Encryption/Decryption of 32-bit Block of Plaintext/Cryptogram - Examples

To encrypt/decrypt 32-bit blocks of plaintexts/cryptograms we should execute:

```
> MD := 2^8:
printf("key length = %d%s",52*8, " bits");
key length = 416 bits
```

Now the key schedule consists of 52 8-bit subblocks and the maximal length of the modified secret key is now 416 bits.

```
> Z := iksgmd(12321);
idispks(Z);
```

$$Z := \begin{bmatrix} 225 & 39 & 172 & 133 & 3 & 14 \\ 252 & 93 & 192 & 97 & 111 & 24 \\ 86 & 255 & 100 & 43 & 216 & 244 \\ 110 & 128 & 105 & 123 & 66 & 235 \\ 226 & 68 & 17 & 166 & 119 & 215 \\ 169 & 59 & 31 & 118 & 39 & 82 \\ 214 & 255 & 186 & 107 & 63 & 25 \\ 104 & 221 & 184 & 162 & 114 & 77 \\ 77 & 40 & 19 & 114 & 0 & 0 \end{bmatrix}$$

```
Key schedule:
round No. 1: E1 27 AC 85 03 0E
round No. 2: FC 5D C0 61 6F 18
round No. 3: 56 FF 64 2B D8 F4
round No. 4: 6E 80 69 7B 42 EB
round No. 5: E2 44 11 A6 77 D7
round No. 6: A9 3B 1F 76 27 52
```

```
round No. 7: D6 FF BA 6B 3F 19
round No. 8: 68 DD B8 A2 72 4D
round No. 9: 4D 28 13 72
```

Let's follow the encryption:

```
> m := convert("1+2=", bytes);
c:= iddea(m);
```

$$m := [49, 43, 50, 61]$$

```
input:              31 2B 32 3D
after round No. 1: 32 0B D2 12
after round No. 2: D6 43 44 E0
after round No. 3: 94 A1 D7 EE
after round No. 4: D4 CD D8 10
after round No. 5: 90 17 90 D7
after round No. 6: 6B 76 12 F8
after round No. 7: 70 A5 D5 E1
after round No. 8: CC 12 9B CC
output:             1F C3 25 7E
```

$$c := [31, 195, 37, 126]$$

Key schedule for decryption and decryption process:

```
> Z := iksd(Z);
idispks(Z);
mr := iddea(c);
convert(%,bytes);
```

$$Z := \begin{bmatrix} 247 & 216 & 237 & 124 & 114 & 77 \\ 215 & 72 & 35 & 211 & 63 & 25 \\ 251 & 70 & 1 & 245 & 39 & 82 \\ 73 & 225 & 197 & 159 & 119 & 215 \\ 58 & 239 & 188 & 48 & 66 & 235 \\ 250 & 151 & 128 & 140 & 216 & 244 \\ 3 & 156 & 1 & 6 & 111 & 24 \\ 154 & 64 & 163 & 53 & 3 & 14 \\ 8 & 217 & 84 & 143 & 0 & 0 \end{bmatrix}$$

```
Key schedule:
round No. 1: F7 D8 ED 7C 72 4D
```

```
round No. 2: D7 48 23 D3 3F 19
round No. 3: FB 46 01 F5 27 52
round No. 4: 49 E1 C5 9F 77 D7
round No. 5: 3A EF BC 30 42 EB
round No. 6: FA 97 80 8C D8 F4
round No. 7: 03 9C 01 06 6F 18
round No. 8: 9A 40 A3 35 03 0E
round No. 9: 08 D9 54 8F
input:              1F C3 25 7E
after round No. 1: 53 8D 82 D5
after round No. 2: 19 CC 75 41
after round No. 3: B2 AF 52 B8
after round No. 4: 6E E9 11 56
after round No. 5: 59 40 21 E9
after round No. 6: 9D A8 42 7B
after round No. 7: 07 92 68 CC
after round No. 8: E7 DE 52 92
output:             31 2B 32 3D
```

$mr := [49, 43, 50, 61]$

"

$1+2$="

To reduce the length of a modified secret key we ought to partly announce the encryption key schedule. Thus, let

```
> Z := Matrix(9, 6,
  {(1, 1) = k[1], (1, 2) = 216,   (1, 3) = 237,
   (1, 4) = 124,  (1, 5) = 114,   (1, 6) = 77,
   (2, 1) = 215,  (2, 2) = k[2],  (2, 3) = 35,
   (2, 4) = 211,  (2, 5) = 63,    (2, 6) = 25,
   (3, 1) = 251,  (3, 2) = 70,    (3, 3) = k[3],
   (3, 4) = 245,  (3, 5) = 39,    (3, 6) = 82,
   (4, 1) = 73,   (4, 2) = 225,   (4, 3) = 197,
   (4, 4) = k[4], (4, 5) = 119,   (4, 6) = 215,
   (5, 1) = 58,   (5, 2) = 239,   (5, 3) = 188,
   (5, 4) = 48,   (5, 5) = k[5],  (5, 6) = 235,
   (6, 1) = 250,  (6, 2) = 151,   (6, 3) = 128,
   (6, 4) = 140,  (6, 5) = 216,   (6, 6) = k[6],
   (7, 1) = 3,    (7, 2) = 156,   (7, 3) = 1,
   (7, 4) = 6,    (7, 5) = k[7],  (7, 6) = 24,
   (8, 1) = 154,  (8, 2) = 64,    (8, 3) = 163,
   (8, 4) = k[8], (8, 5) = 3,     (8, 6) = 14,
   (9, 1) = k[9], (9, 2) = k[10], (9, 3) = k[11],
   (9, 4) = 143,  (9, 5) = 0,     (9, 6) = 0}):
```

Our secret key is now determined by the variable **k**. Let:

```
> k := [118, 94, 155, 34, 225, 76, 168, 131, 106, 17, 6]:
printf("key length = %d%s",11*8, " bits");
key length = 88 bits
```

Encryption key schedule:

```
> idispks(Z);
Key schedule:
round No. 1: 76 D8 ED 7C 72 4D
round No. 2: D7 5E 23 D3 3F 19
round No. 3: FB 46 9B F5 27 52
round No. 4: 49 E1 C5 22 77 D7
round No. 5: 3A EF BC 30 E1 EB
round No. 6: FA 97 80 8C D8 4C
round No. 7: 03 9C 01 06 A8 18
round No. 8: 9A 40 A3 83 03 0E
round No. 9: 6A 11 06 8F
```

Encryption:

```
> m := convert("1+2=", bytes);
c:= iddea(m);
```

$$m := [49, 43, 50, 61]$$

```
input:              31 2B 32 3D
after round No. 1:  9B 04 A0 CC
after round No. 2:  E9 86 19 06
after round No. 3:  DD F9 08 7D
after round No. 4:  B3 B9 CC 9C
after round No. 5:  68 86 C4 4F
after round No. 6:  33 5C 7E 6A
after round No. 7:  15 F3 1E 9C
after round No. 8:  B4 E3 15 A3
output:             3E 26 E9 B3
```

$$c := [62, 38, 233, 179]$$

Decryption key schedule and decryption process:

```
> Z := iksd(Z);
idispks(Z);
mr := iddea(c);
convert(%,bytes);
```

$$Z := \begin{bmatrix} 177 & 239 & 250 & 133 & 3 & 14 \\ 252 & 93 & 192 & 206 & 168 & 24 \\ 86 & 255 & 100 & 43 & 216 & 76 \\ 110 & 128 & 105 & 123 & 225 & 235 \\ 226 & 68 & 17 & 166 & 119 & 215 \\ 169 & 59 & 31 & 189 & 39 & 82 \\ 214 & 101 & 186 & 107 & 63 & 25 \\ 104 & 221 & 162 & 162 & 114 & 77 \\ 159 & 40 & 19 & 114 & 0 & 0 \end{bmatrix}$$

```
Key schedule:
round No. 1: B1 EF FA 85 03 0E
round No. 2: FC 5D C0 CE A8 18
round No. 3: 56 FF 64 2B D8 4C
round No. 4: 6E 80 69 7B E1 EB
round No. 5: E2 44 11 A6 77 D7
round No. 6: A9 3B 1F BD 27 52
round No. 7: D6 65 BA 6B 3F 19
round No. 8: 68 DD A2 A2 72 4D
round No. 9: 9F 28 13 72
input:             3E 26 E9 B3
after round No. 1: 96 C1 33 85
after round No. 2: 99 7F F8 7A
after round No. 3: 2B 44 1D 09
after round No. 4: 66 88 A8 23
after round No. 5: C7 CD DA 8A
after round No. 6: 90 B4 CC B9
after round No. 7: AC C3 62 7D
after round No. 8: 80 1F 03 6F
output:            31 2B 32 3D
```

$$mr := [49, 43, 50, 61]$$

"

$1+2$="

4. Modified Encryption/Decryption of 16-bit Block of Plaintext/Cryptogram - Examples

It is possible to encrypt/decrypt 16-bit block (2 bytes) of plaintext/cryptogram after executing the statements:

```
> MD := 2^4:
printf("key = %d%s",52*4, " bits");
key = 208 bits
```

Let us observe the key schedule generation and encryption of the message **n!**. Since the IDEA accepts four 16-bit plaintext subblocks, we must convert 2 bytes of the message into four subblocks. After doing it we are ready to encrypt the message **m** and observe the encryption process:

```
> Z := iksgmd(123456789);
idispks(Z);
m := ip24(convert("n!", bytes));#see help page
c:= iddea(m);
```

$$Z := \begin{bmatrix} 8 & 12 & 2 & 9 & 10 & 13 \\ 12 & 7 & 10 & 4 & 15 & 3 \\ 13 & 12 & 4 & 0 & 3 & 15 \\ 4 & 14 & 2 & 4 & 11 & 15 \\ 12 & 5 & 11 & 1 & 3 & 10 \\ 8 & 15 & 7 & 10 & 5 & 14 \\ 10 & 9 & 0 & 10 & 5 & 3 \\ 14 & 0 & 0 & 3 & 8 & 15 \\ 3 & 14 & 9 & 0 & 0 & 0 \end{bmatrix}$$

```
Key schedule:
round No. 1: 8 C 2 9 A D
round No. 2: C 7 A 4 F 3
round No. 3: D C 4 0 3 F
round No. 4: 4 E 2 4 B F
round No. 5: C 5 B 1 3 A
round No. 6: 8 F 7 A 5 E
round No. 7: A 9 0 A 5 3
round No. 8: E 0 0 3 8 F
round No. 9: 3 E 9 0
```

$$m := [6, 14, 2, 1]$$

```
input:              6 E 2 1
after round No. 1: 7 D 2 1
after round No. 2: D 1 3 3
after round No. 3: 3 4 A 9
after round No. 4: 9 9 9 9
after round No. 5: D F F 8
after round No. 6: 0 4 B 9
after round No. 7: 4 8 1 9
after round No. 8: A E 6 4
output:             D 4 7 D
```

$$c := [13, 4, 7, 13]$$

```
> Z := iksd(Z);
mr := iddea(c);
convert(ip42(%),bytes); #see help page
```

$$Z := \begin{bmatrix} 6 & 2 & 7 & 0 & 8 & 15 \\ 11 & 0 & 0 & 6 & 5 & 3 \\ 12 & 0 & 7 & 12 & 5 & 14 \\ 15 & 9 & 1 & 12 & 3 & 10 \\ 10 & 5 & 11 & 1 & 11 & 15 \\ 13 & 14 & 2 & 13 & 3 & 15 \\ 4 & 12 & 4 & 0 & 15 & 3 \\ 10 & 6 & 9 & 13 & 10 & 13 \\ 15 & 4 & 14 & 2 & 0 & 0 \end{bmatrix}$$

```
input:              D 4 7 D
after round No. 1: 5 1 8 A
after round No. 2: 7 B D 5
after round No. 3: 2 6 E C
after round No. 4: 6 4 E 9
after round No. 5: C C 2 2
after round No. 6: 0 7 D E
after round No. 7: 0 C 4 4
after round No. 8: E 4 A 9
output:             6 E 2 1
```

$$mr := [6, 14, 2, 1]$$

"

n!"

The mode of encryption with the key less than 208 bits is left to the reader.

5. Conclusions

It has been shown that by putting to use the package named **topicIDEA** we can encrypt/decrypt with the IDEA more securely than usually. We may also construct many protocols offering the secret sharing and database encryption, by applying the presented here generalized method.

References

[1] C. Koscielny: *The* **topicIDEA** *package,*
http://www.maplesoft.com/applications/app_center_view.aspx?AID=1922&CID=5&SCID=6

[2] C. Koscielny: *The application of DES, IDEA and AES in strong encryption,* Quasigroups and Related Systems (accepted for publication in 2006)