Computer Science Journal of Moldova, vol.**5**, no.2(14), 1997

# Provably indeterminate 128-bit cipher

N. Moldovyan

**Abstract**

There is described a class of the encryption functions based on data-dependent selection of subkeys. Such functions are proposed for elaboration of the indeterminate ciphers with very large number of different modifications of the cryptalgorithm. A 128-bit undetermined cipher is described. The number of possible modifications is $\approx 10^{20R}$, where $2 \leq R \leq 7$, $R$ is the number of rounds. It is proved that all modifications are unique.

Key words: software encryption, indeterminate algorithm, block cipher, flexible cryptosystem

## 1 Introduction

The majority of known single key cryptosystems are hardware-oriented. At present the problem of creation of fast software encryption ciphers seems to be actual [1]. One of the promising trends is the creation of block software encryption systems is based on data-dependent selection of subkeys [2, 3].

In present paper there is described a class of the encryption functions based on data-dependent subkey selection and an indeterminate 128-bit cipher with provably nonequivalent cryptalgorithm modifications.

## 2 Cryptoscheme with a tuning subsystem

It is assumed that there is used a tuning subsystem (TS) and a resident cryptomodule in the cryptoscheme described below [2]. Tuning procedures are to be executed only once when cryptosystem is switched on

therefore they do not influence the encryption speed. In many applications the run duration of TS (about 0.1–1 s) is not critical [4] and it is possible to include very complicated functions in the tuning algorithm. The tuning consists in executing a one-way conversion of the source key (password) into a comparatively large secondary sequence of subkeys (encryption key) and in "pseudorandom" password-controlled modification of the encryption algorithm of the resident subroutine [5]. The main requirements to TS seem to be the following:

(1) every bit of the password must influence about equiprobably all bits of the key area and

(2) complexity of the calculation of the password from the known secondary key must be very high.

Parameters of the resident cryptomodule determine all critical characteristics of such ciphers. In next sections there is considered a new encryption mechanism with flexible subkey selection. If TS executes modification of the encryption procedures of the resident module, then the cryptalgorithm is not predetermined. In this case we have an indeterminate cipher. Design of such flexible ciphers is connected with the following problems:

(1) all possible modifications of the cryptalgorithm must be secure,

(2) number of nonequivalent modifications is to be very large,

(3) selection of nonequivalent modifications must be approximately equiprobable.

# 3 Encryption functions based on data-dependent subkey selection

## 3.1 Notations

Input text block is represented as concatenation of subblocks:

$$T = B^{(n-1)} \| B^{(n-2)} \| \ldots \| B^{(1)} \| B^{(0)}.$$

$b$ is the length of the subblocks $B^{(i)}$ in bits.

$\#T$ is the cardinal number of the set of all possible data blocks, $\#T = 2^{nb}$.

$\Delta$ is a subsubblock defining the current subkey $Q_\Delta = Q[\Delta]$.

$\delta$ is the length of subsubblock $\Delta$ in bits.

$\{Q_j\} = \{Q[j]\}$ is an encryption key represented as a sequence of $b$-bit subkeys $Q[j]$, $j = 0, 1, 2, \ldots, 2^\delta - 1$. This sequence is to be generated under control of the password during initialization stage.

$d$ is an integer defining the number of encryption subrounds in one round.

$s$ is the number of the current subround, $s = 1, 2, \ldots, b/d$.

$i$ is the number of the current encryption step, $i = 1, 2, \ldots, nb/d$.

$\alpha(g)$ is a permutation of the set of numbers $\{0, 1, 2, \ldots, n-1\}$.

$B^{>x>}$ ($B^{<x<}$) denotes to-right (to-left) rotation of $B$ by $x$ bits.

### 3.2 Encryption mechanism

The one-round encryption function is

$$C = E(T) = e_k(e_{k-1}(\ldots e_2(e_1(T)))), \tag{1}$$

where $k = nb/d$ is the whole number of elementary encryption steps, $C$ is the ciphertext block, $e_1, e_2, \ldots, e_k$ are elementary encryption functions. At the current encryption step $i$ there is executed only the conversion of the subblock $B^{\alpha(g)}$, where $g = i \ mod \ n$:

$$B^{\alpha(g)} := e_i(B^{\alpha(g)}, Q[\Delta(s, g')]), \tag{2}$$

where $\Delta(s, g') = (B^{\alpha(g')})^{>d(s-1)>} \, mod \, 2^\delta$, $g' = i - 1 \, mod \, n$, $e_i$ is an elementary encryption function, which for fixed $B^{\alpha(g)}$ defines a permutation $\gamma(Q_\Delta)$ of the set of numbers $\{0, 1, 2, \ldots, 2^b - 1\}$.

Encryption of some input data block $T$ can be described by the following generalized formula:

$$C = f(T, Q_{h_1}, Q_{h_2}, \ldots, Q_{h_k}), \tag{3}$$

where $\{Q_{h_i}\}$, $i = 1, 2, \ldots, k$, is a set of subkeys used while encrypting. Generation of the current set $\{Q_{h_i}\}$ depends on both the input message

and the key area. One can say that for given input data block $T$ there is generated a set of indices $\{h_i\}$. The set $\{Q_{h_i}\}$ is some virtual key used for encryption of the given block $T$. There are possible only $N_T = 2^{bn}$ different input data blocks, but for given secondary key the number of different sets $\{Q_{h_i}\}$ equals $N_h \approx (2^{\delta nb/d})^r$, where $r$ is the number of encryption rounds. Such ratio allows one to suppose that the probability of the generation of two equal sets $\{Q_{h_i}\}$ for two different input blocks is very low. Below it will be shown that there exists a wide class of encryption functions for which the virtual keys are unique for all input blocks.

The one-round decryption function is

$$T = D(C) = d_k(d_{k-1}(\dots d_2(d_1(C)))), \tag{4}$$

where $d_i$ is the elementary function which is to be executed at the step $i$. It is inverse as regards to $e_{n-i+1}$. Elementary decryption functions have the following structure:

$$B^{\alpha(q)} := d_i(B^{\alpha(q)}, Q[\Delta(s, q')]), \tag{5}$$

where $\Delta(s, q') = (B^{\alpha(q')})^{>b-sd>} \, mod \, 2^{\delta}$, $q = n + 1 - i \, mod \, n$, $q' = n - i \, mod \, n$. It is evident that for given $\{Q_j\}$ at the respective decryption and encryption steps there are used the same subkeys.

Let us consider the case $d = \delta$ and such elementary encryption functions which satisfy the following condition:

**Design Criterion 1.** *Elementary encryption functions define a permutation of the value of the higher subsubblock.*

It is supposed that the current subblock is represented as concatenation of higher and lower subsubblocks $B = H \| L = H \cdot 2^{d(s-1)} + L$, where the size (in bits) of the lower subsubblock is $l = d(s-1)$ and depends on the number of the conversion subround. Design Criterion 1 means that elementary encryption function must have the following structure

$$B := e(B) = \beta(H) \cdot 2^{d(s-1)} + f(L), \tag{6}$$

where $f(L) \leq 2^{d(s-1)} - 1$, $\beta(H)$ is a permutation of the set of numbers $\{0, 1, 2, \ldots, 2^{b-d(s-1)} - 1\}$.

If all elementary functions in (2) satisfy Design Criterion 1, then Statement 1 take place. Criterion 1 is oriented to implementation of this statement. Theorem 1 shows the subkey selection is uniform over the whole set of input data blocks. Furthermore, it is evident that for all input blocks the ordered sets of indices of selected subkeys are unique for arbitrary number of rounds.

**Statement 1** *If an arbitrary key area $\{Q_j\}$ is given and $d = \delta$, then for all input blocks $T$ there are generated unique sets of indices $\{h_i\}$.*

**Theorem 1** *If $d = \delta$, then for arbitrary key $\{Q_j\}$ and arbitrary set of indices $\{h_i\}$, where $i = 1, 2, \ldots, n/d$, there exists unique input data block $T$ for which it is generated the given set of indices.*

**Proof.** The cardinal number of the set of possible combinations of $k$ indices is $\#\{h_i\} = (2^\delta)^{nb/d}$. If $d = \delta$, then $\#\{h_i\} = 2^{nb} = \#T$. Taking into account Statement 1 we obtain the one-to-one correspondence for elements $\{h_i\}$ and $T$. This means that for any set $\{h_i\}$ there exists unique $T$ which generates the given set of indices. $\diamond$

### 3.3 Ciphers with indeterminate cryptalgorithm

Security of the cryptoschemes based on pseudorandom selection of subkeys appears to be not sensitive as regards to particular kind of used elementary functions $e_i$ and their random modification. This allows designing flexible ciphers in which functions $e_i$ are to be modifyed by means of the password-control tuning of some set of the reserved operations. One can assign also initialization of different permutations $\alpha(g)$ for every subround. Such an approach allows developing flexible cryptosystems with $> 10^{80}$ different possible modifications of the encryption algorithm.

It is known that for given encryption key a $b$-bit block encryption function defines a permutation $\epsilon(T)$ of the set of numbers $\{0, 1, 2, \ldots, 2^{nb} - 1\}$. We will use the following

**Definition 1** *Two modifications of encryption function (1) are equivalent if they define the same permutations $\epsilon(T)$ for all encryption keys.*

To evaluate the number of nonequivalent modifications it is necessary to consider particular algorithms implementing an indeterminate cryptoscheme. Below it is presented a 128-bit cipher in which there are no equivalent modifications of the encryption function.

# 4 Algorithm implementation

In algorithms described below the 32-bit words of input data blocks are converted consequently using simple arithmetic operations: module 2 addition ($\oplus$), module $2^{32}$ addition ($+$) and subtraction ($-$), as well as fixed and data-dependent rotations. It is supposed that there is used a 1024-byte encryption key $\{Q_j\} = \{Q_\Delta\}$, where $Q$ is a 32-bit subkey ($\Delta, j = 0, 1, 2, \ldots, 255$). Algorithm 1 represents a 128-bit cipher with fixed encryption algorithm. Algorithm 2 describes an indeterminate 128-bit cipher.

Below there is used the following notation:

$X, Y, Z,$ and $W$ are variables;

$$u = (s-1)\, mod\, 4; \quad t = (4-s)\, mod\, 4\ ; \quad Q(\Delta) = Q_\Delta,$$

where

$\Delta = a_f, b_f, d_f, e_f$ and $f = u, t$;

$$a_u = (X_s)^{>8u>}\, mod\, 2^8; \quad b_u = (Y_s)^{>8u>}\, mod\, 2^8;$$

$$d_u = (Z_s)^{>8u>}\, mod\, 2^8; \quad e_u = (W_{s-1})^{>8u>}\, mod\, 2^8;$$

$$a_t = (X_{s-1})^{>8t>}\, mod\, 2^8; \quad b_t = (Y_{s-1})^{>8t>}\, mod\, 2^8;$$

$$d_t = (Z_{s-1})^{>8t>}\, mod\, 2^8; \quad e_t = (W_s)^{>8t>}\, mod\, 2^8;$$

$V = v_4 \| v_3 \| v_2 \| v_1,$ where $V = X, Y, Z, W$ and $v = x, y, z, w$.

**Algorithm 1: 128-bit cipher** ($d = \delta = 8; b = 32; n = 4$)

**a) Encryption**

**INPUT:** 128-bit data block $T = D\|B\|A\|E$ represented as a concatenation of four 32-bit words.

1. Set: $R = 3$, $s = 1$, $X_0 = A$, $Y_0 = B$, $Z_0 = D$, and $W_0 = E$.
2. Transform $X$: $X_s = X_{s-1} \oplus Q(e_u)$.
3. Transform $Y$: $Y_s = Y_{s-1} \oplus Q(a_u)$.
4. Transform $Z$: $Z_s = Z_{s-1} \oplus Q(b_u)$.
5. Transform $W$: $W_s = W_{s-1} \oplus Q(d_u)$, $(w_4\|w_3)_s := (w_4\|w_3)_s^{>Y_s>}$, and increment $s$.
6. $X_s = (X_{s-1} - W_{s-1} \, mod \, 2^{32}) \oplus Q(e_u)$.
7. $Y_s = [Y_{s-1} + Q(a_u) \, mod \, 2^{32}] \oplus X_s$.
8. $Z_s = (Z_{s-1} \oplus Y_s) - Q(b_u) \, mod \, 2^{32}$
9. $W_s = W_{s-1} - Q(d_u) \, mod \, 2^{32}$, $(w_2\|w_1)_s := (w_2\|w_1)_s^{>Y_s>}$, increment $s$.
10. $X_s = [X_{s-1} - Q(e_u)] \, mod \, 2^{32}$.
11. $Y_s = [Y_{s-1} \oplus Q(a_u)] + W_{s-1} \, mod \, 2^{32}$.
12. $Z_s = [Z_{s-1} \oplus Q(b_u)] - X_s \, mod \, 2^{32}$.
13. $W_s = W_{s-1} \oplus Q(d_u)$, increment $s$.
14. $X_s = X_{s-1} + Q(e_u) \, mod \, 2^{32}$.
15. $Y_s = Y_{s-1} \oplus Q(a_u)$.
16. $Z_s = Z_{s-1} - Q(b_u) \, mod \, 2^{32}$.
17. $W_s = (W_{s-1} \oplus Z_s)^{>Y_s>} + Q(d_u) \, mod \, 2^{32}$.

18. If $s < 4R$, then increment $s$ and jump to step 2, otherwise STOP.

**OUTPUT:** 128-bit ciphertext block $C = Z_{4R}\|Y_{4R}\|X_{4R}\|W_{4R}$.

### b) Decryption

**INPUT:** 128-bit ciphertext block $C = D\|B\|A\|E$.

1. Set: $R = 3$, $s = 1$, $X_0 = A$, $Y_0 = B$, $Z_0 = D$, and $W_0 = E$.
2. $W_s = [W_{s-1} - Q(d_t)\, mod\, 2^{32}]^{<Y_{s-1}<} \oplus Z_{s-1}$,
3. $Z_s = Z_{s-1} + Q(b_t)\, mod\, 2^{32}$,
4. $Y_s = Y_{s-1} \oplus Q(a_t)$.
5. $X_s = X_{s-1} - Q(e_t)\, mod\, 2^{32}$, increment $s$.
6. $W_s = W_{s-1} \oplus Q(d_t)$.
7. $Z_s = (Z_{s-1} + X_{s-1}\, mod\, 2^{32}) \oplus Q(b_t)$.
8. $Y_s = (Y_{s-1} - W_s\, mod\, 2^{32}) \oplus Q(a_t)$.
9. $X_s = X_{s-1} + Q(e_t)\, mod\, 2^{32}$, increment $s$.
10. $(w_2\|w_1)_{s-1} := (w_2\|w_1)_{s-1}^{<Y_{s-1}<}$, $W_s = W_{s-1} + Q(d_t)\, mod\, 2^{32}$.
11. $Z_s = [Z_{s-1} + Q(b_t)\, mod\, 2^{32}] \oplus Y_{s-1}$.
12. $Y_s = (Y_{s-1} \oplus X_{s-1}) - Q(a_t)\, mod\, 2^{32}$.
13. $X_s = [X_{s-1} \oplus Q(e_t)] + W_s\, mod\, 2^{32}$, increment $s$.
14. $(w_4\|w_3)_{s-1} := (w_4\|w_3)_{s-1}^{<Y_{s-1}<}$, $W_s = W_{s-1} \oplus Q(d_t)$.
15. $Z_s = Z_{s-1} \oplus Q(b_t)$.
16. $Y_s = Y_{s-1} \oplus Q(a_t)$.
17. $X_s = X_{s-1} \oplus Q(e_t)$.

18. If $s < 4R$, then increment $s$ and jump to step 2, otherwise STOP.

**OUTPUT:** 128-bit data block $T = Z_{4R}\|Y_{4R}\|X_{4R}\|W_{4R}$.

Note that data-dependent rotations at steps 5a and 9a do not violate Design Criterion 1. In the following algorithm there are reserved binary operations $\lfloor p \rfloor = +, -, \oplus$ and rotation operations $^{>c_l>}$ defining to-right circular shift by $c_l$ bits, where $1 \le c_l \le 31$ ($p$ and $l$ are the numbers of the reserved operations).

### Algorithm 2: Indeterminate 128-bit cipher
$(d = \delta = 8;\ b = 32;\ n = 4)$

**INPUT:** 128-bit data block $T = D\|B\|A\|E$.

1. Set: $R = 3$, $m = 1$, $s = 1$, $X_0 = A$; $Y_0 = B$; $Z_0 = C$; $W_0 = D$.
2. $X_s = [X_{s-1}\ \lfloor 22m - 21 \rfloor\ Q(e_u)]^{>c_{7m-6}>}$;
3. $Y_s = [Y_{s-1}\ \lfloor 22m - 20 \rfloor\ Q(a_u)]^{>c_{7m-5}>}$.
4. $Z_s = [Z_{s-1}\ \lfloor 22m - 19 \rfloor\ Q(b_u)]^{>c_{7m-4}>}$;
5. $W_s = W_{s-1}\ \lfloor 22m - 18 \rfloor\ Q(d_u)$.
6. $Z_s := Z_s\ \lfloor 22m - 17 \rfloor\ X_s$; $Y_s := Y_s\ \lfloor 22m - 16 \rfloor\ W_s$; $s := s + 1$.
7. $(w_4\|w_3)_{s-1} := (w_4\|w_3)_{s-1}^{>c_{7m-3}>}$; $X_s = X_{s-1}\ \lfloor 22m - 15 \rfloor\ Q(e_u)$.
8. $(x_4\|x_3)_s := (x_4\|x_3)_s^{>c_{7m-2}>}$; $Y_s = Y_{s-1}\ \lfloor 22m - 14 \rfloor\ Q(a_u)$.
9. $(y_4\|y_3)_s := (y_4\|y_3)_s^{>c_{7m-1}>}$; $Z_s = Z_{s-1}\ \lfloor 22m - 13 \rfloor\ Q(b_u)$.
10. $(z_4\|z_3)_s := (z_4\|z_3)_s^{>c_{7m}>}$; $W_s = W_{s-1}\ \lfloor 22m - 12 \rfloor\ Q(d_u)$.
11. $Y_s := Y_s\ \lfloor 22m - 11 \rfloor\ X_s$; $Z_s := Z_s\ \lfloor 22m - 10 \rfloor\ W_s$; $s := s + 1$.
12. $X_s = X_{s-1}\ \lfloor 22m - 9 \rfloor\ Q(e_u)$;

13. $Y_s = Y_{s-1} \lfloor 22m-8 \rfloor Q(a_u)$.
14. $Z_s = Z_{s-1} \lfloor 22m-7 \rfloor Q(b_u)$;
15. $W_s = W_{s-1} \lfloor 22m-6 \rfloor Q(d_u)$; $s := s+1$.
16. $X_s = X_{s-1} \lfloor 22m-5 \rfloor Q(e_u)$; $Y_{s-1} := Y_{s-1} \lfloor 22m-4 \rfloor W_{s-1}$.
17. $Y_s = Y_{s-1} \lfloor 22m-3 \rfloor Q(a_u)$; $Z_{s-1} := Z_{s-1} \lfloor 22m-2 \rfloor X_s$.
18. $Z_s = Z_{s-1} \lfloor 22m-1 \rfloor Q(b_u)$.
19. $W_s = W_{s-1} \lfloor 22m \rfloor Q(d_u)$; $s := s+1$.
20. If $s < 4R$, then increment $m$ and jump to step 2, otherwise STOP.

**OUTPUT:** 128-bit ciphertext block $C = Z_{4R} \| Y_{4R} \| X_{4R} \| W_{4R}$.

Decryption algorithm is evident. The subroutines implementing Algorithms 1 and 2 provide conversion speed about $60/R$ Mbit/s (for microprocessor Intel 486/100). The allowable number of encryption rounds is $R \geq 2$. In Algorithm 2 there are reserved $22R$ bynary operations and $7R$ rotation operations ($4R$ positions for rotation by from 1 to 15 bits and $3R$ positions for rotation by from 1 to 31 bits). Thus, there are possible $3^{22R} \cdot 15^{4R} \cdot 31^{3R} \approx 10^{20R}$ different combinations of conversion operations. Due to data-dependent subkey selection one can expect the majority of possible modifications of the encryption function of Algorithm 1 are not equivalent. Let us consider this problem. Desigh Criterion 1 is satisfied for Algorithm 1 and for all possible modifications of Algorithm 2. Thus, Theorem 1 is valid and uniform subkey selection take place for all modifications. Considering consequently all encryption steps for arbitrary value $R$ one can establish the following statement:

**Statement 2** *If two arbitrary modifications $E'$ and $E''$ correspond to sets of the operations differing at least in one reserved position, then there exists an encryption key* $\mathbf{Q}$ *which for arbitrary given $T$ defines different ordered sets of indices $\{h'_i\}$ and $\{h''_i\}$, $i = 1, 2, \ldots, 16R$, for $E'$ and $E''$ respectively.*

**Theorem 2** *For $R \leq 7$ there exist no equivalent modifications of Algorithm 2.*

**Proof.** Let us consider an encryption key $\mathbf{Q}$ for which a given $T$ generates sets of indices $\{h'_i\}$ and $\{h''_i\} \neq \{h'_i\}$, where $i = 1, 2, \ldots, 16R$, corresponding to modifications $E'$ and $E''$. To prove Theorem 2 we shall construct a key for wich these modifications define different permutations $\epsilon(T)$. Since $\{h''_i\} \neq \{h'_i\}$ one can find $i = I$ for wich $h''_I \neq h'_I$. Let us mark indices $h'_1 = h''_1$, $h'_2 = h''_2$,..., $h'_{I-1} = h''_{I-1}$ as used ones. For given $T$ we have $N_{used} \leq 16R$ used indices. Other indices we shall name vacant indices.

*Case 1:* $I = 1$.

Modifying the subkeys $Q_{h'_I}$ and $Q_{h''_I}$ one can define a couple of indices $H'_{I+1}$ and $H''_{I+1} \neq H'_{I+1}$ (the capital letter $H$ denotes index after modification) which are selected from the set of vacant indices, when the block $T$ is encrypted with the modifyed key. Since we do not modify subkeys corresponding to previous selection steps, indices $h'_1$ and $h''_1$ do not change. In this way one can modify consequently $Q_{H'_{I+1}}$ and $Q_{H''_{I+1}}$, $Q_{H'_{I+2}}$ and $Q_{H''_{I+2}}$, ..., $Q_{H'_{8R-1}}$ and $Q_{H''_{8R-1}}$. Now we have indices $H'_{8R}$ and $H''_{8R} \neq H'_{8R}$ which are selected for the first time. Accumulating all such local modifications we obtain a new key $\mathbf{Q}'$. If $E'(T, \mathbf{Q}') \neq E''(T, \mathbf{Q}')$, then we have proved that $E'$ and $E''$ are not equivalent. If $E'(T', \mathbf{Q}') = E''(T', \mathbf{Q}')$, then one can modify $Q_{H''_8}$ obtaining another new key $\mathbf{Q}''$ and $E'(T', \mathbf{Q}'') = E'(T', \mathbf{Q}') = E''(T', \mathbf{Q}') \neq E''(T', \mathbf{Q}'')$. Thus, for the case 1 we have proved Theorem 2.

*Case 2:* $I \geq 2$.

We shall modify the subkey $Q_{h'_{I-1}}$. There is 255 different values $Q_{h'_{I-1}}$ which define 255 different values $H'_I \neq h'_I$ and 255 different values $H''_I \neq h''_I$. In Algorithm 2 there are used such operations for which for about all different $Q_{h'_{I-1}}$ we have $H'_I \neq H''_I$. For $R \leq 7$ we have $256 - 2N_{used} > 32$ different values $Q_{h'_{I-1}}$ for which values $H'_I$ and $H''_I$ correspond to vacant indices and for several cases we have $H'_I \neq H''_I$. Selecting a couple of such indices $H'_I$ and $H''_I \neq H'_I$ and modifying the encryption key as in the first case, we obtain a new

key $\mathbf{Q}''$ satisfying the condition $E'(T,\mathbf{Q}'') \neq E''(T,\mathbf{Q}'')$. Theorem 2 is proved. $\diamond$.

Theorem 2 shows that the proposed encryption algorithm is really an indeterminate cipher.

## 5 Conclusion

Data-dependent selection of subkeys seems to be a perspective encryption mechanism for creation of fast sotware-oriented indeterminate ciphers with provably nonequivalent modifications of the cryptalgorithm. Using the conversion scheme described above one can compose many different 64-, 96-, and 128-bit block indeterminate ciphers. The size of input data block is not limited. It is possible also to use this scheme for creation 512-byte block cryptosystems, however we estimate the approach described in [2] to be preferable for such purpose.

Security of the considered 128-bit indeterminate cipher is based on the following.

1. The most powerful known cryptanalytic attacks, for example differential cryptanalysis [6], are based on the preliminary investigation of statistic properties of conversion procedures. In the case under consideration cryptalgorithm modification is selected randomly from the set of $\approx 10^{20R}$ potentially implementable modifications, all of them being unique.

2. For all modifications there are no two different input data blocks for encryption of wich there is used the same set of subkeys. Selection of subkeys is changed "pseudorandomly" from one block to another. We expect that this fact should help defeat standard differential and linear cryptanalysis even in the case of the known cryptalgorithm modification.

## References

[1] Fast Software Encryption. Third Intern. Workshop. Proceedings. Lecture Notes Computer Science, Springer-Verlag, 1996. Vol.1039. X, 219 p.

[2] Moldovyan A.A., Moldovyan N.A. Fast Software Encryption System Based on Local Pseudorandomness. Comput. Sci. J. of Moldova. 1995. V.3. No.3(9), p.252–262.

[3] Moldovyan A.A., Moldovyan N.A., Andronati N.R., Rogozhin Yu.V. Software-Oriented Approach to Computer Processing Protection. Europ. simul. meet.: Simulation Tools and Applications. Gyor, Hungary, Aug. 28–30, 1995. Proc. p.143–149.

[4] Moldovyan A.A., Moldovyan N.A., Moldovyan P.A. Effective Software-Oriented Cryptosystem in Complex PC Security Software. Comput. Sci. J. of Moldova. 1994. V.2. No.3, p.269–282.

[5] Moldovyan A.A., Moldovyan N.A. Fast Software Encryption Systems for Secure and Private Communication. 12th Intern. Conf. on Comput. Communic., Seoul, Korea, Aug. 21-24, 1995. Proc. p.415–420.

[6] Biham E., Shamir A. Differential Analysis of DES-like Cryptosystems. Journal of Cryptology. 1991. V.4. No.1, p.3–72.

N.A.Moldovyan,
Institute of Modelling and
Intellectualization of Complex Systems,
Prof. Popov Str., 5, St-Petersburg
197376, Russia;
phone: 7(812) 2340415; fax: 7(812)2349093
e-mail: *sovetov@imics.spb.su*

Received Martch 25, 1997