DDP-Based Ciphers: Differential Analysis of SPECTR-H64

A.V. Bodrov, A.A. Moldovyan, P.A. Moldovyanu

Abstract

Use of data-dependent (DD) permutations (DDP) appears to be very efficient while designing fast ciphers suitable for cheap hardware implementation, few papers devoted to security analysis of the DDP-based cryptosystems have been published though. This paper presents results of differential cryptanalysis (DCA) of the twelve-round cipher SPECTR-H64 which is one of the first examples of the fast block cryptosystems using DDP as cryptographic primitive. It has been shown that structure of SPECTR-H64 suits well for consideration of the differential characteristics. Experiments have confirmed the theoretic estimations. Performed investigation has shown that SPECTR-H64 is secure against DCA, some elements of this cipher can be improved though. In order to make the hardware implementation faster and cheaper a modified version of this cipher with eight rounds is proposed.

Key words: Fast ciphers, hardware encryption, controlled operations, data-dependent permutations, differential analysis

1 Introduction

Data encryption is widely used to solve different problems of the information security. This defines importance of the design of the ciphers suitable for cheap hardware implementation. Recently the controlled operations (CO) has been proposed as an attractive cryptographic primitive suitable for the design of such ciphers [1, 2]. One of early applications of the CO relates to [1], where the key-dependent substitution boxes are used while designing a block cipher. A class of CO suitable for cryptographic applications is proposed in [2]. Controlled permutations (CP) attract much attention of cryptographers, since they can be implemented in fast and cheap hardware using permutation networks (PN) developed and investigated previously [3-5]. The PN are well suited for cryptographic applications, since they allow one to specify and perform permutations at the same time. A variant of the symmetric cryptosystem based on PN and Boolean functions is presented in [6]. Another cryptographic application of PN is presented by the cipher ICE [7] in which a very simple PN is used to specify a key-dependent permutation. In such applications of CP the permutation on data bit strings is a linear operation. Such use of CP has been shown [8] to be not very effective against differential cryptanalysis (DCA), very large number of different bit permutations can be specified though.

Efficiency of CO as cryptographic primitive crucially increases while using CO as data-dependent (DD) operations (DDO). A particular kind of CP represented by DD rotations (DDR) are successfully used in cryptosystems RC5 [9], RC6 [10], and MARS [11]. In spite of the fact that DDR contain few different realizable modifications they thwart well DCA and linear cryptanalysis (LCA). Use of CP in the form of DD permutations (DDP) appears to be very suitable to design fast ciphers oriented to cheap hardware implementation [12]. The iterative 64-bit block cipher SPECTR-H64 represents an example of DDP-based ciphers [13]. It is interesting that the round transformation of this cipher is not involution, the same algorithm performs encryption and decryption though. Since the DDP-based design is oriented to drastic decrease of the hardware implementation cost, the security estimation of the DDP-based ciphers is of the great importance. If the detailed cryptanalysis show the DDP-based ciphers are secure, then we will have actually a very efficient approach to embed fast encryption algorithm in cheap hardware.

The present paper is one of the first ones devoted to the security analysis of the DDP-based ciphers.

The paper is organized in the following way: In the second section we describe briefly the algorithm SPECTR-H64 paying attention to the design of the operational boxes performing DDP. Section 3 considers differential characteristics of the primitives used in SPECTR-H64 and presents security analysis of this cipher and experimental results confirming our estimations. In section 4 we propose some improvements allowing one to reduce the number of rounds from 12 to 8 that results in the performance increase and hardware cost decrease.

Notation. Let $\{0,1\}^n$ be the set of all binary vectors $U = (u_1, ..., u_n)$, where $\forall i \in \{1, ..., n\}$ $u_i \in \{0,1\}$. Let us denote $U_{\text{lo}} = (x_1, ..., x_{n/2})$ and $X_{\text{hi}} = (x_{n/2+1}, ..., x_n)$, i.e. $X = (X_{\text{lo}}, X_{\text{hi}})$ or $X = (X_{\text{lo}}|X_{\text{hi}})$, where "|" denotes the concatenation operation. Let $e \in \{0,1\}$ denote encryption ($e = 0$) or decryption ($e = 1$).

Let $Y = X^{\ggg k}$ denote cyclic rotation of the word X by k bits, where $Y = (y_1, ..., y_n)$ is the output vector and $\forall i \in \{1, ..., n-k\}$ we have $y_i = x_{i+k}$ and $\forall i \in \{n-k+1, ..., n\}$ we have $y_i = x_{i+k-n}$.

Let XY denote bit-wise AND operation of the two vectors X and Y: $X, Y \in \{0,1\}^n$. Let $\oplus$ denote the XOR operation.

2 Design of the block cipher SPECTR-H64

2.1 General encryption scheme

SPECTR-H64 is a new 12-round block cipher with 64-bit input. The general encryption scheme (Fig. 1) is defined by the following formulas: $C = \mathbf{Encr}(M, K)$ and $M = \mathbf{Decr}(C, K)$, where M is the plaintext, C is the ciphertext ($M, C \in \{0,1\}^{64}$), K is the secret key ($K \in \{0,1\}^{256}$), **Encr** is the encryption function, and **Decr** is the decryption function. In the block cipher SPECTR-H64 the encryption and decryption functions are described by formula

$$Y = \mathbf{F}(X, Q^{(e)}),$$

where $Q^{(e)} = \mathbf{H}(K, e)$ is the extended key, the last being a function of the secret key $K = (K_1, K_2, ..., K_8)$, $K_i \in \{0,1\}^{32}$ for $i = 1, 2, ..., 8$, and of the transformation mode parameter e ($e = 0$ defines encryption, $e = 1$

defines decryption). We have $X = M$, for $e = 0$ and $X = C$ for $e = 1$. Extended key is represented as follows:

$$Q^{(e)} = (Q^{(e)}_{\mathrm{IT}}, Q^{(e)}_1, ..., Q^{(e)}_{12}, Q^{(e)}_{\mathrm{FT}})$$

where $Q^{(e)}_{\mathrm{IT}}, Q^{(e)}_{\mathrm{FT}} \in \{0,1\}^{32}$ and $\forall j = 1, ..., 12 \quad Q^{(e)}_j \in \{0,1\}^{192}$. Each round key $Q^{(e)}_j = (Q^{(1,e)}_j, ..., Q^{(6,e)}_j)$, where $\forall h = 1, ..., 6$ $Q^{(h,e)}_j \in \{0,1\}^{32}$, is represented as concatenation of six subkeys which are selected from the set $\{K_1, K_2, ..., K_8\}$ depending on the number of the current round and the value e. Output value Y is the ciphertext C in the encryption mode or the plaintext M in the decryption mode. The algorithm (function **F**) is designed as sequence of the following procedures: 1) *initial transformation* **IT**, 2) 12 rounds with procedure **Crypt**, and 3) *final transformation* **FT**. For detailed description of the key scheduling and **IT** and **FT** one can see [13]. In our analysis we consider the round keys to be uniformly distributed random values. This makes our security estimate to be valid in the case of the more strong key scheduling. We assume also that **IT** and **FT** do not contribute significantly to security.

Let consider the jth encryption round and denote $Q^{(e)}_j = (A, B, A', B', A'', B'')$. The round transformation of SPECTR-H64 is denoted as procedure **Crypt** shown in Fig. 2. This procedure has the form: $R = \mathbf{Crypt}(R, L, A, B, A', B', A'', B'')$, where $L, R, A, B, A', B', A'', B'' \in \{0,1\}^{32}$. The procedure **Crypt** uses the following operations: cyclic rotation ">>>" by fixed number of bits, XOR operation "$\oplus$", non-linear operation **G**, DDP operations $\mathbf{P}_{32/80}$ and $\mathbf{P}^{-1}_{32/80}$, and extension operation **E**.

2.2 Non-linear operation G

Realization of the operation $Y = \mathbf{G}(X, A, B)$ is defined in the vector form by the following expression:

$$Y = W_0 \oplus W_1 \oplus W_2 A \oplus W_2 W_5 B \oplus W_3 W_5 \oplus W_4 B,$$

where binary vectors W_j for $j = 0, 1, ..., 5$ are expressed as follows: $W_0 = X = (x_1, x_2, ..., x_{32})$, $W_1 = (1, x_1, x_2, ..., x_{31}), ...,$ $W_j =$

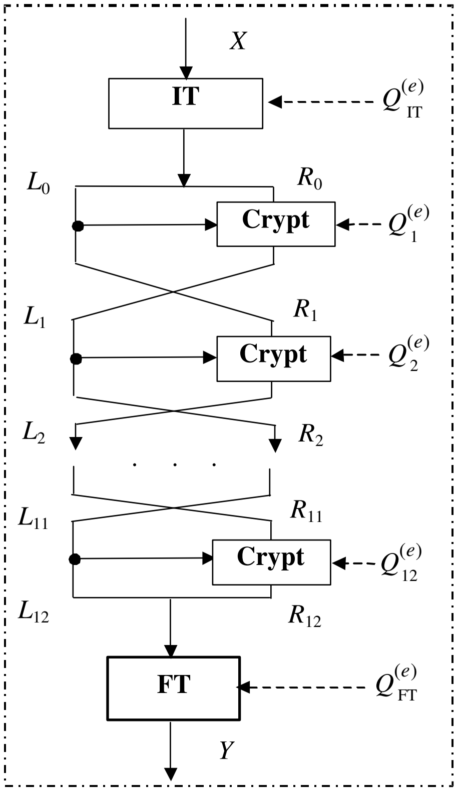

Figure 1. General structure of SPECTR-H64

$(1, ..., 1, x_1, x_2, ..., x_{32-j})$. The ith output bit of the operation $\mathbf{G}$ is the following Boolean function:

$$y_i = x_i \oplus x_{i-1} \oplus x_{i-2}a_i \oplus x_{i-2}x_{i-5}b_i \oplus x_{i-3}x_{i-5} \oplus x_{i-4}b_i,$$

where $x_{-4} = x_{-3} = x_{-2} = x_{-1} = x_0 = 1$.

2.3 CP-boxes $\mathbf{P}_{32/80}$ and $\mathbf{P}^{-1}_{32/80}$

The CP boxes $\mathbf{P}_{32/80}$ and $\mathbf{P}^{-1}_{32/80}$ are built up from switching elements $\mathbf{P}_{2/1}$ in accordance with the layered structure shown in Fig. 3. Each

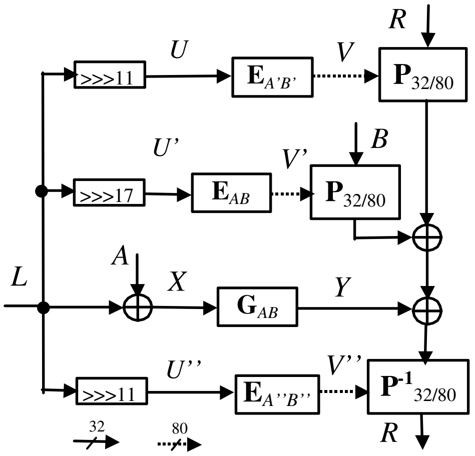

Figure 2. Structure of the procedure **Crypt**

box $\mathbf{P}_{2/1}$ is controlled by one bit v: $y_1 = x_{1+v}$ and $y_2 = x_{2-v}$, where (x_1, x_2) is input and (y_1, y_2) is output. In all figures in this paper the solid lines indicate data movement, while dotted lines indicate the controlling bits. Layered CP boxes we shall denote as $\mathbf{P}_{n/m}$, where n corresponds to the input/output size in bits and m indicates the size of the controlling input that is equal to the number of the used $\mathbf{P}_{2/1}$-boxes. Performing a CP operation can be denoted as $Y = \mathbf{P}_{n/m(V)}(X)$, where X is input vector, Y is output vector, and V is controlling vector. For fixed value V the CP box performs fixed permutation that is called CP modification. Let indexing the elementary $\mathbf{P}_{2/1}$-boxes in a CP box $\mathbf{P}_{n/m}$ from left to right and from top to bottom. The CP-box $\mathbf{P}^{-1}_{n/m}$ is called inverse of $\mathbf{P}_{n/m}$-box, if for all V the corresponding CP modifications $\mathbf{P}_V$ and $\mathbf{P}^{-1}_V$ are mutually inverse [12].

Let given a CP box $\mathbf{P}_{n/m}$. Then it is easy to construct $\mathbf{P}^{-1}_{n/m}$ by changing the direction of the bits moving. We shall enumerate the $\mathbf{P}_{2/1}$-boxes of some $\mathbf{P}^{-1}_{n/m}$-box from left to right and from bottom to top. Thus, in both $\mathbf{P}_{n/m}$ and $\mathbf{P}^{-1}_{n/m}$ boxes the controlling bit v_j controls the jth $\mathbf{P}_{2/1}$-box.

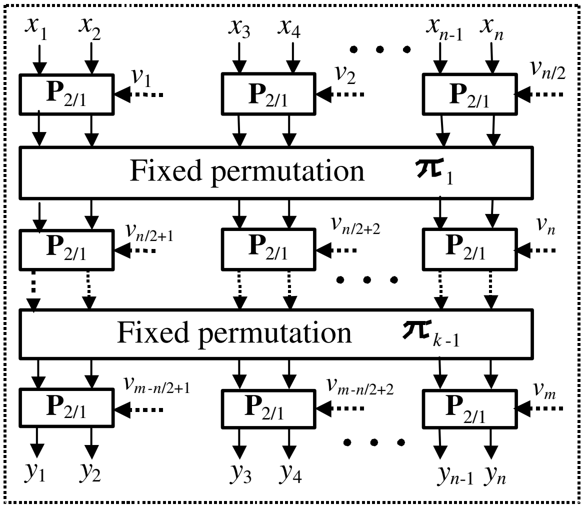

Figure 3. A CP box with layered structure

The structure of the boxes $\mathbf{P}_{32/80}$ and $\mathbf{P}^{-1}_{32/80}$ is explained in Fig. 4-6. The fixed permutational involution between the third and fourth layers of the elementary boxes $\mathbf{P}_{2/1}$ in the $\mathbf{P}_{32/80}$-box is described as follows:

$$\begin{array}{c}(1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)(11,18)\\(12,26)(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32)\end{array}.$$

2.4 Extension box E

The extension box $\mathbf{E}$ is used to form a 80-bit controlling vector, given the 32-bit input vector. The formal representation of the extension transformation is: $V = (V_1|V_2|V_3|V_4|V_5) = \mathbf{E}(U, A', B') = \mathbf{E}_{A',B'}(U)$, where $V \in \{0,1\}^{80}$; $V_1, V_2, V_3, V_4, V_5 \in \{0,1\}^{16}$; $U, A', B' \in \{0,1\}^{32}$. Actually the vectors V_1, V_2, V_3, V_4, V_5 are determined according to the

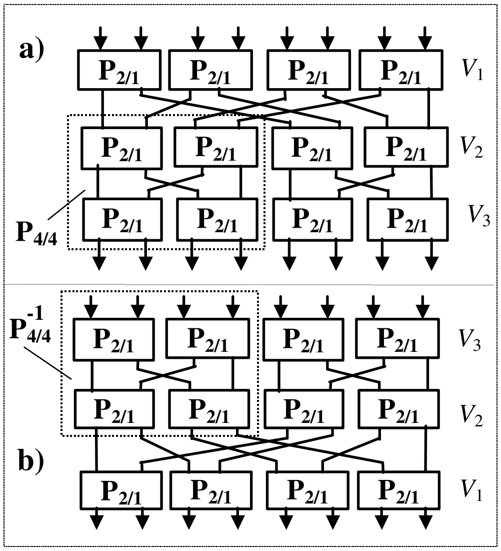

Figure 4. Structure of the boxes $\mathbf{P}_{8/12}$ (a) and $\mathbf{P}^{-1}_{8/12}$ (b)

formulas:

$$V_1 = U_{\mathrm{hi}};\ V_2 = \pi((U \oplus A)_{\mathrm{hi}}); \qquad V_3 = \pi'((U \oplus B')_{\mathrm{hi}});$$

$$V_4 = \pi'((U \oplus B')_{\mathrm{lo}}); \qquad V_5 = \pi((U \oplus A)_{\mathrm{lo}}),$$

where fixed permutations π and π' are the following: $\pi(Z) = Z_{\mathrm{hi}}^{\ggg 1}|Z_{\mathrm{lo}}^{\ggg 1}$ and $\pi'(Z) = Z_{\mathrm{hi}}^{\ggg 5}|Z_{\mathrm{lo}}^{\ggg 5}$. Table 1 specifies which bit $u_i \in U$ controls which $\mathbf{P}_{2/1}$-box in the PN is presenting the $\mathbf{P}_{32/80}$-box. Number i is indicated in the position of the correspondent $\mathbf{P}_{2/1}$-box. In other words this distribution table shows correspondence between bits of the vector U and bits of the vector V. For example, $v_1 = u_{17}$, $v_2 = u_{18}$,..., $v_{16} = u_{32}$, $v_{17} = k'_1 \oplus u_{26}$,..., $v_{32} = k'_{16} \oplus u_{26}$, where k'_1,...,k'_{16} are fixed bits (actually they are some bits of the key). Designed distribution provides that each input bit is affected by five different bits of U for all possible values U, A', and B'.

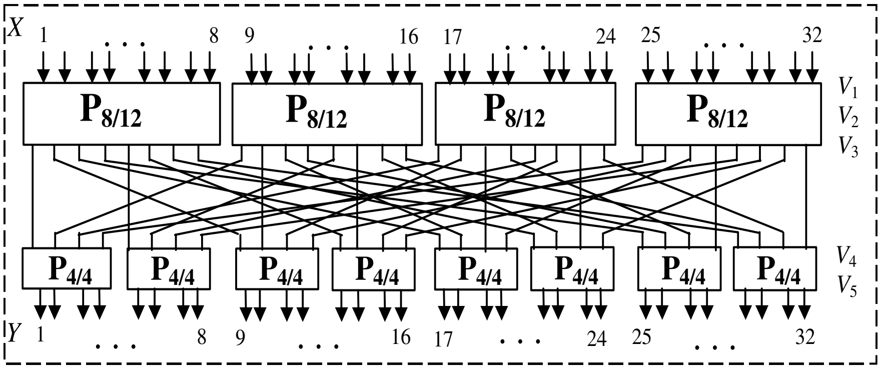

Figure 5. Structure of the box $\mathbf{P}_{32/80}$

Table 1. Distribution of bits of the vector U

V_1	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
V_2	26	27	28	29	30	31	32	25	18	19	20	21	22	23	24	17
V_3	30	31	32	25	26	27	28	29	22	23	24	17	18	19	20	21
V_4	14	15	16	9	10	11	12	13	6	7	8	1	2	3	4	5
V_5	10	11	12	13	14	15	16	9	2	3	4	5	6	7	8	1

3 Differential Analysis of SPECTR-H64

3.1 Some properties of the controlled operations

Let Δ_q^W be the difference with arbitrary q active (non-zero) bits corresponding to the vector W. Let $\Delta_{q|i_1,...,i_q}$ be the difference with q active bits and $i_1,...,i_q$ be the numbers of digits corresponding to active bits. Note that Δ_1 corresponds to one of the differences $\Delta_{1|1}, \Delta_{1|2}, ..., \Delta_{1|32}$. We shall also denote the difference Δ_q at the input or output of the operation $\mathbf{F}$ as $\Delta_q^{\mathbf{F}\downarrow}$ or $\Delta_q^{\mathbf{F}}$, respectively.

Differential properties of the CP boxes with the given structure are defined by properties of the elementary switching element. Using the

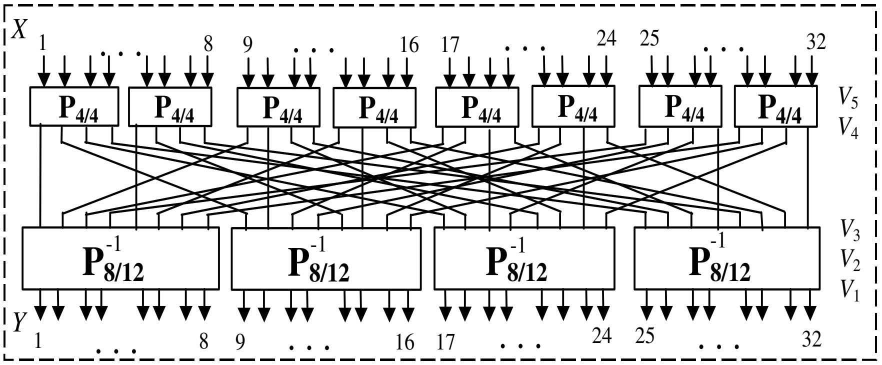

Figure 6. Structure of the box $\mathbf{P}_{32/80}^{-1}$

main properties of the last (see Fig. 7) it is easy to find characteristics of the $\mathbf{P}_{32/80}$-box.

Figure 8 illustrates the case when some difference with one active bit Δ_q^L passes the left branch of the cryptoscheme. The difference Δ_q^L can cause generation or annihilation of $w = 1, 2, 3$ pairs of active bits in the CP box. Let consider the $\mathbf{P}_{32/80}$-box in right branch in the case $q = 1$. The difference $\Delta_{1|i}^L$ is transformed by the extension box into Δ_2^V or Δ_3^V (depending on i) at the controlling input of $\mathbf{P}_{32/80}$, i.e. one active bit in the left subblock influences two or three switching elements $\mathbf{P}_{2/1}$ permuting four or six different bits of the right data subblock. Depending on value of the permuted bits and input difference Δ_q^R of the $\mathbf{P}_{32/80}$-box the output differences Δ'^R_g with different number of active bits can be formed by this CP box.

Avalanche effect corresponding to the operations $\mathbf{G}$ is defined by its structure that provides each input bit influences several output bits (except the 32nd input bit influences only the 32nd output bit). Table 3 presents the formulas describing avalanche caused by inverting the bit l_i. One can see that alteration of the input bit x_i, where $3 \leq i \leq 27$, causes deterministic alteration of two output bits y_i and y_{i+1} and probabilistic alteration of the output bits y_{i+2}, y_{i+3}, y_{i+4}, y_{i+5} which change with probability $p = 0.5$. Note that for $i = 1, 2$ alteration

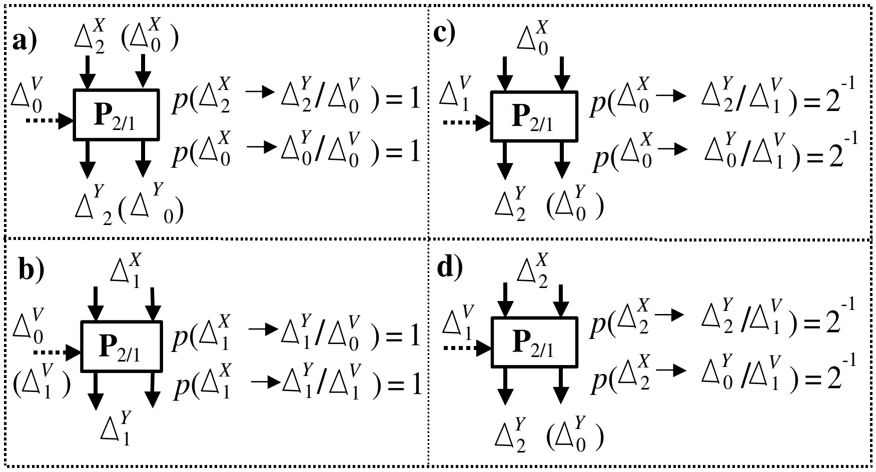
a)
Δ_2^X (Δ_0^X)
Δ_0^V
$\mathbf{P}_{2/1}$
$p(\Delta_2^X \rightarrow \Delta_2^Y/\Delta_0^V) = 1$
$p(\Delta_0^X \rightarrow \Delta_0^Y/\Delta_0^V) = 1$
$\Delta_2^Y (\Delta_0^Y)$
c)
Δ_0^X
Δ_1^V
$\mathbf{P}_{2/1}$
$p(\Delta_0^X \rightarrow \Delta_2^Y/\Delta_1^V) = 2^{-1}$
$p(\Delta_0^X \rightarrow \Delta_0^Y/\Delta_1^V) = 2^{-1}$
Δ_2^Y (Δ_0^Y)
b)
Δ_1^X
Δ_0^V
(Δ_1^V)
$\mathbf{P}_{2/1}$
$p(\Delta_1^X \rightarrow \Delta_1^Y/\Delta_0^V) = 1$
$p(\Delta_1^X \rightarrow \Delta_1^Y/\Delta_1^V) = 1$
Δ_1^Y
d)
Δ_2^X
Δ_1^V
$\mathbf{P}_{2/1}$
$p(\Delta_2^X \rightarrow \Delta_2^Y/\Delta_1^V) = 2^{-1}$
$p(\Delta_2^X \rightarrow \Delta_0^Y/\Delta_1^V) = 2^{-1}$
Δ_2^Y (Δ_0^Y)

Figure 7. Properties of the elementary box $\mathbf{P}_{2/1}$

of x_i causes deterministic alteration of three output bits y_i, y_{i+1}, and y_{i+3}. When passing through the operation $\mathbf{G}$ the difference $\Delta_{1|i}^L$ can be transformed with certain probability in the output differences $\Delta_2^{\mathbf{G}}$, $\Delta_3^{\mathbf{G}}$, ..., $\Delta_6^{\mathbf{G}}$.

3.2 Security of SPECTR-H64

Trying different attacks against SPECTR-H64 we have found that the differential analysis is the most efficient. Our best variant of the DCA corresponds to two-round characteristic with difference (Δ_0^L, Δ_1^R). The difference passes the first round with probability 1 and after swapping subblocks it transforms in (Δ_1^L, Δ_0^R) (see Fig. 9). In the second round the active bit passing through the left branch of cryptoscheme can form at the output of the operation $\mathbf{G}$ the difference $\Delta_g^{\mathbf{G}}$, where $g \in \{1, 2, 3, 4, 5, 6\}$. Only differences with even number of active bits contribute to the probability of the two-round iterative characteristic. The most contributing are the differences $\Delta_{2|i,i+1}^{\mathbf{G}}$. The most contributing mechanisms of the formation of the two-round characteristic belong

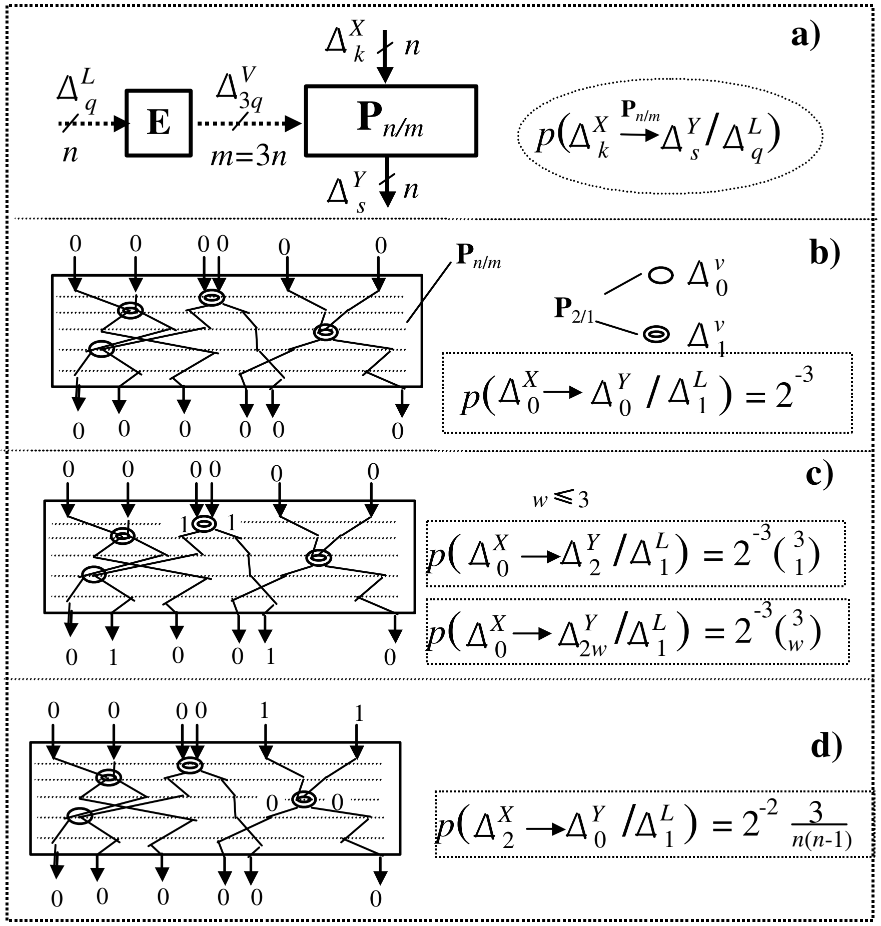
a)
Δ_k^X
n
Δ_q^L
n
E
Δ_{3q}^V
m=3n
$\mathbf{P}_{n/m}$
Δ_s^Y
n
$p(\Delta_k^X \xrightarrow{\mathbf{P}_{n/m}} \Delta_s^Y / \Delta_q^L)$
b)
0 0 0 0 0 0
$\mathbf{P}_{n/m}$
$\mathbf{P}_{2/1}$
Δ_0^v
Δ_1^v
0 0 0 0 0 0
$p(\Delta_0^X \rightarrow \Delta_0^Y / \Delta_1^L) = 2^{-3}$
c)
0 0 0 0 0 0
1 1
$w \leqslant 3$
$p(\Delta_0^X \rightarrow \Delta_2^Y / \Delta_1^L) = 2^{-3}\binom{3}{1}$
$p(\Delta_0^X \rightarrow \Delta_{2w}^Y / \Delta_1^L) = 2^{-3}\binom{3}{w}$
0 1 0 0 1 0
d)
0 0 0 0 1 1
0 0
$p(\Delta_2^X \rightarrow \Delta_0^Y / \Delta_1^L) = 2^{-2}\frac{3}{n(n-1)}$
0 0 0 0 0 0

Figure 8. Some properties of the CP box: a - notation of the general case; b - zero difference passes the CP box; c - formation of two active bits; d - annihilation of two active bits.

Table 2. Change of output bits caused by single bit alteration ($\Delta x_i = 1$) at input of the operation $\mathbf{G}$

Expression	Probability
$\Delta y_i = \Delta x_i$	$p_{\Delta y_i = 1} = 1$
$\Delta y_{i+1} = \Delta x_i$	$p_{\Delta y_{i+1} = 1} = 1$
$\Delta y_{i+2} = \Delta x_i (a_{i+2})$	$p_{\Delta y_{i+2} = 1} = 1/2$
$\Delta y_{i+3} = \Delta x_i (x_{i-2})$	$p_{\Delta y_{i+3} = 1} = 1/2$
$\Delta y_{i+4} = \Delta x_i (b_{i+4})$	$p_{\Delta y_{i+4} = 1} = 1/2$
$\Delta y_{i+5} = \Delta x_i (x_{i+2} \oplus x_{i+3} \oplus b_{i+5})$	$p_{\Delta y_{i+5} = 1} = 1/2$

to Cases 1a, 1b, 1c, 2a, 2b, 3a, 3b, 4a, and 4b, where $i \in \{1, ..., 32\}$, described below.

Case 1a includes the following elementary events:

1) The difference $\Delta_{2|i,i+1}^{\mathbf{G}}$ is formed at the output of the operation $\mathbf{G}$ with probability $p_2^{(i,i+1)} = \Pr\left(\Delta_{2|i,i+1}^{\mathbf{G}} \big/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right)$.

2) The difference $\Delta_{2|i,i+1}^{\mathbf{P}'}$ is formed at the output of the CP box $\mathbf{P}'$ with probability $p_3^{(i,i+1)} = \Pr\left(\Delta_{2|i,i+1}^{\mathbf{P}'} \big/ \Delta_0^{\mathbf{P}'_{\downarrow}}\right)$.

3) The difference $\Delta_0^{\mathbf{P}''}$ is formed at the output of the CP box $\mathbf{P}''$ with probability $p_1 = 2^{-z} = \Pr\left(\Delta_0^{\mathbf{P}''} \big/ \Delta_0^{\mathbf{P}''_{\downarrow}}\right)$, where $z = 2, 3$, depending on i.

4) After XORing differences $\Delta_{2|i,i+1}^{\mathbf{G}}$, $\Delta_{2|i,i+1}^{\mathbf{P}'}$, and $\Delta_0^{\mathbf{P}''}$ we have zero difference $\Delta_0^{\mathbf{P}^*}$ at the input of the $\mathbf{P}^*$-box. It passes this box with probability $p_4 = \Pr\left(\Delta_0^{\mathbf{P}^*} \big/ \Delta_0^{\mathbf{P}^*_{\downarrow}}\right) = 2^{-z}$.

One can denote Case 1a as set of the following events:

$$\left(\Delta_{2|i,i+1}^{\mathbf{G}} \big/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap \left(\Delta_{2|i,i+1}^{\mathbf{P}'} \big/ \Delta_0^{\mathbf{P}'_{\downarrow}}\right) \bigcap$$

$$\bigcap \left(\Delta_0^{\mathbf{P}''} \big/ \Delta_0^{\mathbf{P}''_{\downarrow}}\right) \bigcap \left(\Delta_0^{\mathbf{P}^*} \big/ \Delta_0^{\mathbf{P}^*_{\downarrow}}\right).$$

Using this form of the represention one can describe other cases as follows ($\forall i, t: \quad t \in \{1, 2, ..., 32\}$, $t \neq i$, and $t \neq i+1$):

$$\text{Case 1b:} \qquad \left(\Delta_{2|i,i+1}^{\mathbf{G}} \middle/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap$$

$$\bigcap \left(\Delta_{2|i,i+1}^{\mathbf{P}''} \middle/ \Delta_{0}^{\mathbf{P}''_{\downarrow}}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}'} \middle/ \Delta_{0}^{\mathbf{P}'_{\downarrow}}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}^{*}} \middle/ \Delta_{0}^{\mathbf{P}^{*}_{\downarrow}}\right).$$

$$\text{Case 1c:} \qquad \left(\Delta_{2|i,i+1}^{\mathbf{G}} \middle/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap$$

$$\bigcap \left(\Delta_{0}^{\mathbf{P}'} \middle/ \Delta_{0}^{\mathbf{P}'_{\downarrow}}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}''} \middle/ \Delta_{0}^{\mathbf{P}''_{\downarrow}}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}^{*}} \middle/ \Delta_{2|i,i+1}^{\mathbf{P}^{*}_{\downarrow}}\right).$$

$$\text{Case 2a:} \qquad \left(\Delta_{2|i,i+1}^{\mathbf{G}} \middle/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap$$

$$\bigcap \left(\Delta_{2|i+1,t}^{\mathbf{P}'} \middle/ \Delta_{0}^{\mathbf{P}'_{\downarrow}}\right) \bigcap \left(\Delta_{2|i,t}^{\mathbf{P}''} \middle/ \Delta_{0}^{\mathbf{P}''_{\downarrow}}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}^{*}} \middle/ \Delta_{0}^{\mathbf{P}^{*}_{\downarrow}}\right).$$

$$\text{Case 2b:} \qquad \left(\Delta_{2|i,i+1}^{\mathbf{G}} \middle/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap$$

$$\bigcap \left(\Delta_{2|i,t}^{\mathbf{P}'} \middle/ \Delta_{0}^{\mathbf{P}'_{\downarrow}}\right) \bigcap \left(\Delta_{2|i+1,t}^{\mathbf{P}''} \middle/ \Delta_{0}^{\mathbf{P}''_{\downarrow}}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}^{*}} \middle/ \Delta_{0}^{\mathbf{P}^{*}_{\downarrow}}\right).$$

$$\text{Case 3a:} \qquad \left(\Delta_{2|i,i+1}^{\mathbf{G}} \middle/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap$$

$$\bigcap \left(\Delta_{0}^{\mathbf{P}'} \middle/ \Delta_{0}^{\mathbf{P}'_{\downarrow}}\right) \bigcap \left(\Delta_{2|i+1,t}^{\mathbf{P}''} \middle/ \Delta_{0}^{\mathbf{P}''_{\downarrow}}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}^{*}} \middle/ \Delta_{2|i,t}^{\mathbf{P}^{*}_{\downarrow}}\right).$$

$$\text{Case 3b:} \qquad \left(\Delta_{2|i,i+1}^{\mathbf{G}} \middle/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap$$

$$\bigcap \left(\Delta_{0}^{\mathbf{P}'} \middle/ \Delta_{0}^{\mathbf{P}'_{\downarrow}}\right) \bigcap \left(\Delta_{2|i,t}^{\mathbf{P}''} \middle/ \Delta_{0}^{\mathbf{P}''_{\downarrow}}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}^{*}} \middle/ \Delta_{2|i+1,t}^{\mathbf{P}^{*}_{\downarrow}}\right).$$

$$\text{Case 4a:} \qquad \left(\Delta_{2|i,i+1}^{\mathbf{G}} \middle/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap$$

$$\bigcap \left(\Delta_{0}^{\mathbf{P}''} \middle/ \Delta_{0}^{\mathbf{P}''_{\downarrow}}\right) \bigcap \left(\Delta_{2|i+1,t}^{\mathbf{P}'} \middle/ \Delta_{0}^{\mathbf{P}'\downarrow}\right) \bigcap \left(\Delta_{0}^{\mathbf{P}^{*}} \middle/ \Delta_{2|i,t}^{\mathbf{P}^{*}_{\downarrow}}\right).$$

$$\text{Case 4b:} \qquad \left(\Delta_{2|i,i+1}^{\mathbf{G}} \middle/ \Delta_{1|i}^{\mathbf{G}_{\downarrow}}\right) \bigcap$$

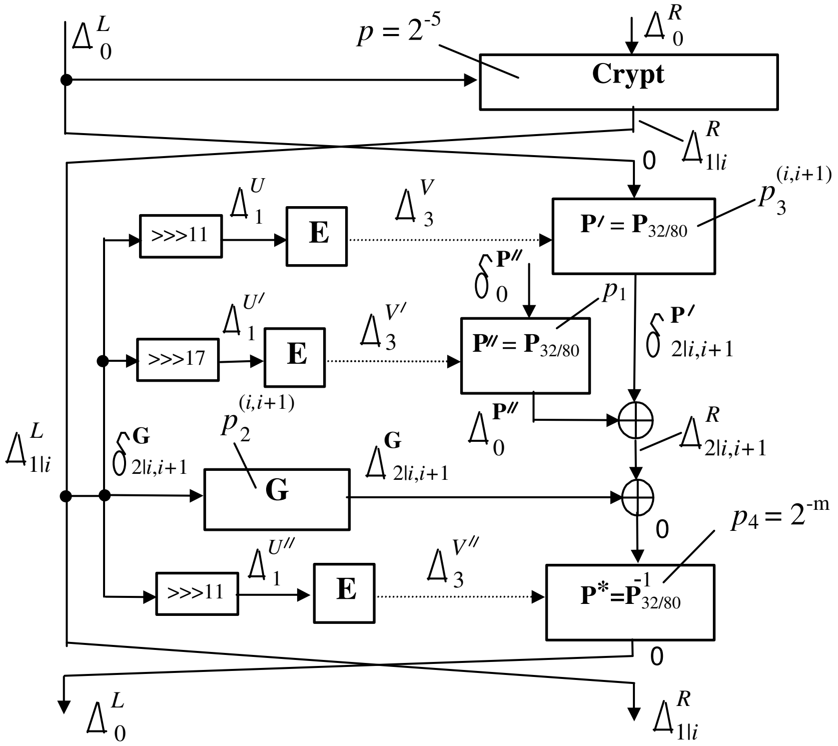

Figure 9. Formation of the two-round characteristic

$$\left(\Delta_0^{\mathbf{P}''}/\Delta_0^{\mathbf{P}''_{\downarrow}}\right)\bigcap\left(\Delta_{2|i,t}^{\mathbf{P}'}/\Delta_0^{\mathbf{P}'\downarrow}\right)\bigcap\left(\Delta_0^{\mathbf{P}^*}/\Delta_{2|i+1,t}^{\mathbf{P}^*_{\downarrow}}\right).$$

There are possible some other mechanisms contributing to the probability of the two-round characteristic and corresponding to generation the differences $\Delta_4^{\mathbf{G}}$ at the output of the operation $\mathbf{G}$. Variants of the formation of the two-round characteristics connected with these mechanisms we shall attribute to the Case 5. Besides, due to the use of the mutually inverse CP boxes $\mathbf{P}_{32/80}$ and $\mathbf{P}_{32/80}^{-1}$ there are possible significantly contributing cases when the box $\mathbf{P}_{32/80}$ generates an additional pair of active bits and the box $\mathbf{P}_{32/80}^{-1}$ annihilates this pair of active bits. Let attribute variants connected with this mechanism to Case 6.

Values $p_1^{(j,t)}$, $p_3^{(j,t)}$, and $p_4^{(j,t)}$, where $j \in \{1, 2, ..., 32\}$, can be easy calculated using the structure of the box $\mathbf{P}_{32/80}$ and distribution of the controlling bits over elementary switching boxes $\mathbf{P}_{2/1}$ (this distribution is defined by Table 1 and the respective bit-rotation operation ("$\ggg$ 11" or "$\ggg$ 17").

For each value $i \in \{1, 2, ..., 32\}$ we have performed the statistic test "1,000 keys and 100,000 pairs of plaintexts" including 10^8 experiments in order to determine the experimental probability $p^{(i)}$ that Δ_0^R passes the right branch of the procedure **Crypt** in the case when in the left data subblock we have the difference $\Delta_{1|i}^L$. Let $s^{(i)}$ be the number of such events. Then we have $\hat{p}^{(i)} = 10^{-8}s^{(i)}$. We have also calculated the probabilities $p^{(i)}$ taking into account the mechanisms of the formation of the two-round characteristic described above. For all i the theoretic values $p^{(i)}$ match sufficiently well the experimental ones $\hat{p}^{(i)}$ (see Table 4) demonstrating that the most important mechanisms of the formation of the two-round differential characteristic correspond to Cases 1-6. The values $p^{(i)*}$ correspond to the modified version of SPECTR-H64+ (see section 3.4).

Probability $P(2)$ of the two-round characteristic can be calculated using the following formula:

$$P(2) = \sum_i p^{(i)}p(i) = 1.15 \cdot 2^{-13},$$

where $p(i) = 2^{-5}$ is the probability that after the first round the active

Table 3. Comparison of the theoretic calculation with experiment

i	1	2	3	4	5	6
#	392	26	117	255	59	50
$\hat{p}^{(i)}$	$1.03{\cdot}2^{-18}$	$1.09{\cdot}2^{-22}$	$1.23{\cdot}2^{-20}$	$1.34{\cdot}2^{-19}$	$1.24{\cdot}2^{-21}$	$1.05{\cdot}2^{-21}$
$p^{(i)}$	$1.08{\cdot}2^{-19}$	$1.43{\cdot}2^{-23}$	$1.35{\cdot}2^{-21}$	$1.9 \cdot 2^{-20}$	$1.13{\cdot}2^{-21}$	$1.39{\cdot}2^{-22}$
$p^{(i)*}$	$1.25{\cdot}2^{-25}$	$1.5 \cdot 2^{-30}$	$0.94{\cdot}2^{-20}$	0	$1.25{\cdot}2^{-21}$	$1.25{\cdot}2^{-21}$

i	7	8	9	10	11	12
#	399	205	679	99	117	388
$\hat{p}^{(i)}$	$1.05{\cdot}2^{-18}$	$1.07{\cdot}2^{-19}$	$1.78{\cdot}2^{-18}$	$1.04{\cdot}2^{-20}$	$1.23{\cdot}2^{-20}$	$1.02{\cdot}2^{-18}$
$p^{(i)}$	$1.13{\cdot}2^{-18}$	$1.6 \cdot 2^{-20}$	$1.13{\cdot}2^{-18}$	$1.31{\cdot}2^{-21}$	$1.36{\cdot}2^{-21}$	$1.56{\cdot}2^{-19}$
$p^{(i)*}$	$0.94{\cdot}2^{-20}$	2^{-21}	$1.3 \cdot 2^{-20}$	$1.5 \cdot 2^{-20}$	$1.5 \cdot 2^{-22}$	2^{-21}

i	13, 14, 15	16	17	18	19,20	21
#	0	467	54054	158196	0	238520
$\hat{p}^{(i)}$	0	$1.22{\cdot}2^{-18}$	$1.11{\cdot}2^{-11}$	$1.62{\cdot}2^{-10}$	0	$1.22 \cdot 2^{-9}$
$p^{(i)}$	0	$1.09{\cdot}2^{-18}$	2^{-11}	$1.5 \cdot 2^{-10}$	0	$1.25 \cdot 2^{-9}$
$p^{(i)*}$	0	0	0	0	0	0

i	22,...,27	28	29	30	31	32
#	0	820	33	141	1901	0
$\hat{p}^{(i)}$	0	$1.07 \cdot 2^{-17}$	$1.38 \cdot 2^{-22}$	$1.48 \cdot 2^{-20}$	$1.25 \cdot 2^{-16}$	0
$p^{(i)}$	0	$1.56 \cdot 2^{-18}$	$1.25 \cdot 2^{-21}$	$1.25 \cdot 2^{-20}$	$1.19 \cdot 2^{-16}$	0
$p^{(i)*}$	0	0	0	0	0	0

bit moves to ith digit. Thus, the performed analysis has shown that the 21st and the 18th digits contribute to $P(2)$ about 88% and the 17th digit contributes to $P(2)$ about 12%. Contribution of other digits is very small. Such strongly non-uniform dependence of $p^{(i)}$ on i is caused by several lacks in the distribution Table 1, nevertheless the ten-round and twelve-round variants of SPECTR-H64 are secure against DCA. Indeed, the difference (Δ_0^L, Δ_1^R) passes ten and twelve rounds with probability $P(10) \approx 2^{-64}$ and $P(12) \approx 1.2 \cdot 2^{-77}$ (for random cipher we have $P = 2^5 \cdot 2^{-64} = 2^{-59} > 2^{-64} > 2^{-77}$).

3.3 Modified version SPECTR-H64+

Differential analysis has shown that the structure of the extension box (i.e. the table describing distribution of the bits of the left data subblock over elementary switching elements of the CP boxes) is a critical part in the design of SPECTR-H64. It is easy to see that small changes in the extension box can cause significant decrease or increase of the probability of two-round characteristic. Taking into account the results of DCA one can easy change positions of the 17th, 18th, and 21st bits and obtain value $P(2) < 2^{-18}$. More accurate modification of the extension box shown in Table 4 gives $P(2) \approx 0.92 \cdot 2^{-22}$. We shall denote SPECTR-like cryptosystem with extension box described in Table 4 as SPECTR-H64+.

For SPECTR-H64+ the most efficient differential characteristic is the three-round one. This characteristic corresponds to the difference $(\Delta_0^L, \Delta_{1|32}^R)$. The peculiarity of the three-round characteristic consists in that the active bit spreads in the second and third rounds through the 32nd digit in the left data subblock (in this case the active bit in the left data subblock generates the single active bit at the output of the operation $\mathbf{G}$ with probability 1). The formation of this characteristic is shown in Fig. 10. This characteristic does not depend on small modifications of the distribution table. The probability of the tree-round characteristic is $P(3) = P_1 P_2 P_3$.

Probability P_1 corresponds to the event that after the first round and swapping the data subblocks we have the difference $(\Delta_{1|32}^L, \Delta_0^R)$.

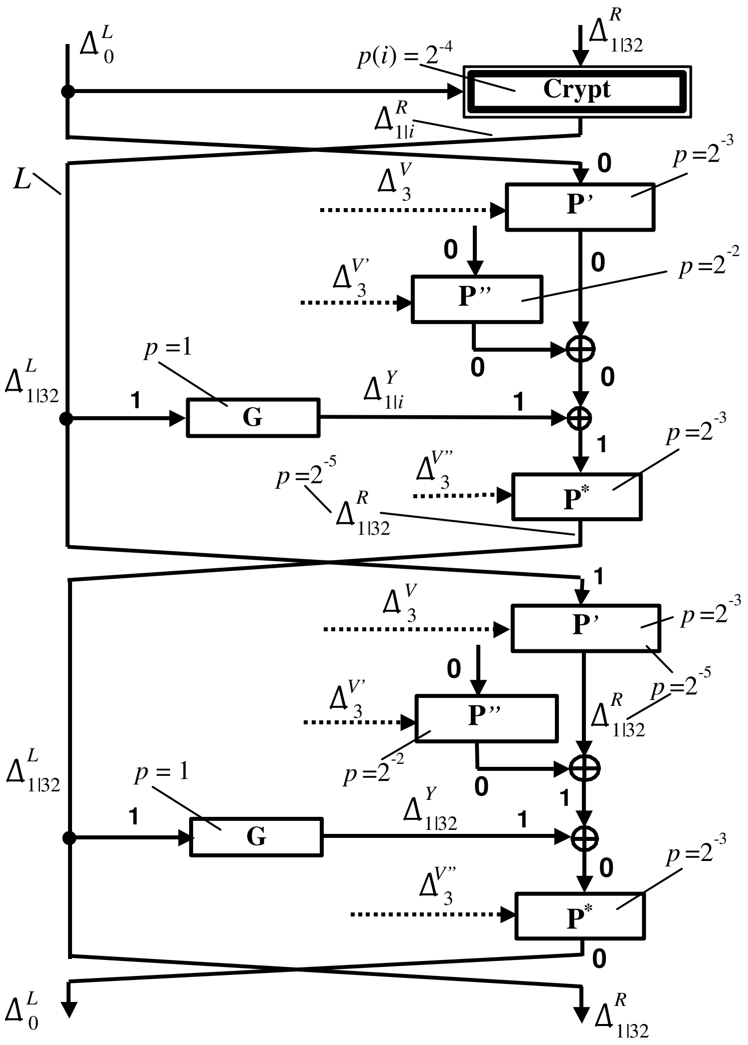

Figure 10. Formation of the three-round characteristic

Table 4. Modified structure of the extension box

V_1	21	26	27	24	28	27	23	24	30	26	32	22	24	30	28	22
V_2	18	19	17	29	25	20	22	25	18	19	31	23	31	21	32	17
V_3	28	20	32	25	26	29	30	29	27	20	21	17	18	19	31	23
V_4	6	7	16	9	10	11	12	13	14	15	8	1	2	3	4	5
V_5	10	11	12	13	14	15	4	9	2	3	16	5	6	7	8	1

Since we consider the case when the difference $\Delta^R_{1|32}$ passes the right branch of the first round, it should be taken into account that the first active layer in $\mathbf{P}_{32/80}$-box and the fifth active layer in $\mathbf{P}^{-1}_{32/80}$-box are controlled with the same controlling vector. Because of the last fact and symmetry of these CP boxes for odd (even) i the difference $\Delta^R_{1|i}$ transforms into $\Delta^R_{1|j}$, where $j \neq i$ and $j \neq i+1$ ($j \neq i-1$), with probability 2^{-5}. The difference $\Delta^R_{1|i}$ transforms into $\Delta^R_{1|i}$ with probability 2^{-4}. It never transforms into $\Delta^R_{1|i+1}$ for odd i and $\Delta^R_{1|i-1}$ for even i. Thus we have $P_1 = 2^{-4}$.

The probability P_2 corresponds to the event that at the output of the second round we have difference $(\Delta^L_{1|32}, \Delta^R_{1|32})$. One can calculate P_2 as $P_2 = 2^{-5}P'P''P^*$, where P', P'', and P^* are the probabilities of the events that CP boxes $\mathbf{P}'$, $\mathbf{P}''$, and $\mathbf{P}^*$, respectively, do not generate active bits. Coefficient 2^{-5} corresponds to the probability that the box $\mathbf{P}^*$ moves the active bit in the 32nd digit. It is easy to see that $P' = P^* = 2^{-3}$, $P'' = 2^{-2}$, and $P_2 = 2^{-13}$.

The probability P_3 corresponds to the event that after the third round and swapping the data subblocks we have the difference (Δ^L_0, $\Delta^R_{1|32}$). One can calculate P_3 as $P_3 = 2^{-5}P'P''P^*$, where 2^{-5} corresponds to probability that at the output of the CP box $\mathbf{P}'$ we have the difference $\Delta^{\mathbf{P}'}_{1|32}$ which annihilates after XOR-ing with output difference of the operation $\mathbf{G}$: $\Delta^{\mathbf{P}'}_{1|32} \oplus \Delta^{\mathbf{G}}_{1|32} = \Delta^R_0$. Using value $P_3 = P_2 = 2^{-13}$ one can calculate $P(3) = 2^{-30}$.

Taking into account the probability of the three-round characteristic one can calculate that six-round SPECTR-H64+ is undistinguishable from a random cipher and conservatively estimate that eight round SPECTR-H64+ is secure against DCA. Thus, due to optimization of the **E**-box structure we have reduced the number of rounds from 12 to 8. This significantly reduces the hardware implementation cost and increases performance.

3.4 Comments on other attacks

Our preliminary study of the security of SPECTR-H64 against LCA has shown that structure of this cipher is also suitable for calculation of the biases of the linear characteristics in the case of few active bits, such characteristics having the largest values of the bias. Our best linear characteristic corresponding to one round has the bias $b(1) \leq 2^{-11}$. A rough estimation of SPECTR-H64 and SPECTR-H64+ for six rounds gives $b(6) \leq 2^5 b^6(1) \approx 2^{-61}$. Thus, these ciphers with six and more rounds are undistinguishable from a random cipher with LCA.

High degree of the algebraic normal form and the complexity of the Boolean functions describing round transformation of SPECTR-H64 prevent the interpolation and high order differential attacks. In spite of the use of very simple key scheduling the described ciphers are secure against slide attack due to non-periodic use of the round subkeys and data-dependent subkey transformation. Truncated differentials attack is prevented, since (1) the data block is transformed as a single unit and (2) each bit of the controlling data subblock influences the selection of the current permutation operation.

3.5 Comments on key scheduling

In the case when encryption and decryption are performed with the same algorithms the direct use of subkeys of the secret key produces the problem of weak and semi-weak keys, the portion of such keys is very low though. It is evident that for SPECTR-H64 the key $K' = (X, X, X, X, X, X, X, X)$ is a weak one (probability to select at random one of such keys is 2^{-192}). One can easy avoid problem of the weak and semi-weak keys introducing minor modification of the initial and final transformations in SPECTR-H64. For example, to implement this one can perform two XOR operations: 1) between subkey A' used in the initial transformation and the parameter $E' \in \{0,1\}^{64}$, where $\forall i \in \{1,2,...,64\}: \ e'_i = e$, and 2) between subkey A'' used in the final transformation and the parameter $E'' \in \{0,1\}^{64}$, where $\forall i \in \{1,2,...,64\}: \ e''_i = e \oplus 1$. After this modification we have e-dependent initial and final transformations: $Y = \mathbf{IT}(X, A' \oplus E')$ and $Y = \mathbf{FT}(X, A'' \oplus E'')$. Another way to prevent weak keys is the use of the rotation operation by j bits, where j is the number of the current round, performed, for example, on the output of the operation $\mathbf{G}$. For majority of the fast implementations this requires no additional hardware resources. Yet another way is the use of some simple key processing, however this requires the use of some additional NAND gates. The use of the simple e-dependent and r-dependent procedures or operations in the ciphers with simple key scheduling appears to be a preferable way to avoid weak and semi-weak keys. The CP boxes suite well to the design of different kinds of the e-dependent permutations.

4 Conclusion

Investigating security of SPECTR-H64 we have shown that security of the DDP-based ciphers depends significantly on the structure of the extension box. The performed analysis of SPECTR-H64 and SPECTR-H64+ has shown that variable bit permutations suite well for calculation of the differential and linear characteristics. Comparative analysis of different attacks against SPECTR-H64 shows that DCA is the most efficient one. Differential analysis presented in this paper allows one to conclude that twelve-round SPECTR-H64 is secure, some optimization of its structure is possible though. Optimized eight-round version SPECTR-H64+ has been proposed. Thus, due to performed security analysis we have found significant reserves in the encryption mechanism of SPECTR-H64 which can be easy used to design new SPECTR-like cryptosystems free of weak keys that will be faster and cheaper in hardware.

References

[1] J.B. Kam and G.I. Davida. *Structured design of substitution-permutation encryption networks*, IEEE Transactions on computers, vol. C-28, no 10 (1979), pp. 747–753.

[2] B.V. Izotov, A.A. Moldovyan, and N.A. Moldovyan, *Controlled operations as a cryptographic primitive*, Proceedings of the International workshop, Methods, Models, and Architectures for Network Security. Lect. Notes Comput. Sci. Berlin: Springer-Verlag, vol. 2052 (2001), pp. 230–241.

[3] V.E. Benes, *Mathematical theory of connecting networks and telephone trafic*, Academic Press, New York, 1965.

[4] A.A.Waksman. *Permutation Network*, Journal of the ACM, vol. 15, no 1 (1968), pp. 159–163.

[5] D.S. Parker. *Notes on shuffle/exchange-type switching networks*, IEEE Transactions on computers, vol. C-29, no 3 (1980), pp. 213–223.

[6] M. Portz, *A generallized description of DES-based and Benes-based permutation generators*, Lect. Notes Comput. Sci. Berlin: Springer-Verlag, vol. 718 (1992), pp. 397–409.

[7] M. Kwan, *The design of the ICE encryption algorithm*, Proceedings of the 4th International Workshop, Fast Software Encryption - FSE '97, Lect. Notes Comput. Sci. Berlin: Springer-Verlag, vol. 1267 (1997), pp. 69–82.

[8] B. Van Rompay, L.R. Knudsen, and V. Rijmen, *Differential cryptanalysis of the ICE encryption algorithm*, Proceedings of the 6th International Workshop, Fast Software Encryption - FSE'98, Lect. Notes Comput. Sci. Berlin: Springer-Verlag, vol. 1372 (1998), pp. 270–283.

[9] R.L. Rivest, *The RC5 Encryption Algorithm*, Proceedings of the 2nd International Workshop, Fast Software Encryption - FSE'94,

Lect. Notes Comput. Sci. Berlin: Springer-Verlag, vol. 1008 (1995), pp. 86–96.

[10] R.L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, *The RC6 Block Cipher*, 1st Advanced Encryption Standard Candidate Conference Proceedings, Venture, California, Aug. 20-22, 1998.

[11] C.Burwick, D.Coppersmith, E.D'Avingnon et al. *MARS - a Candidate Cipher for AES*, 1st AES Candidate Conference Proc., Venture, California, Aug. 20-22, 1998.

[12] A.A. Moldovyan, N.A. Moldovyan, *A cipher based on data-dependent permutations.* Journal of Cryptology. 2002, vol. 15, no. 1, pp.61–72.

[13] N.D. Goots, A.A. Moldovyan, N.A. Moldovyan, *Fast encryption algorithm SPECTR-H64*, Proceedings of the International workshop "Methods, Models, and Architectures for Network Security". LNCS, Springer-Verlag, vol. 2052 (2001) pp. 275–286.

A.V. Bodrov, A.A. Moldovyan, P.A. Moldovyanu,

Received August 1, 2005

Specialized Center of Program Systems "SPECTR",
Kantemirovskaya, 10, St.Petersburg 197342, Russia
ph./fax.7-812-2453743.
E–mail: *mold@cobra.ru*