Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

Variable Bit Permutations: Linear Characteristics and Pure VBP-Based Cipher

N.A. Moldovyan, A.A. Moldovyan, N.D. Goots

Abstract

This paper describes linear characteristics of the variable bit permutations (VBP) that are used in the form of the data-dependent permutations. This primitive suites well to the design of fast cheap-hardware-oriented ciphers. Because of the existence of one characteristic with bias 1/2 we discuss possibility to design a pure VBP-based block ciphers that are indistinguishable from a random transformation. We present design of the cipher which is based only on VBP, fixed permutations, and XOR operations. Performed analysis has shown that the designed pure VBP-based block cipher is secure against differential and linear attacks confirming the efficiency of the VBP as cryptographic primitive.

Key words: variable bit permutations, data-dependent permutations, linear analysis, fast block cipher

1 Introduction

Permutation networks (PNs) have been widely studied in the field of parallel processing and telephone switching systems [1] and they are very interesting to be used as cryptographic primitives. The PNs 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 the key-controlled PNs and Boolean functions is presented in [2]. Another cryptographic application of PNs is presented by the cipher ICE [3] in which a very simple PN is used to specify a key-dependent fixed permutation. Such use of PNs has been shown [4] to be not very effective against differential cryptanalysis. A more attractive approach is the use of PNs to perform variable bit permutations (VBP) implemented as data-dependent permutations (DDP) [5]. Efficiency of the use of data-dependent operations has been demonstrated by examples of ciphers RC5 [6], RC6 [7] and MARS [8], which are based on data-dependent rotations with 32 different modifications. The PNs can be used as controlled permutation (CP) boxes to perform DDP. It is easy to design CP boxes (CPBs) giving possibility to specify 2^{64} and more different modifications of the VBP performed on data subblocks [5] and subkeys [9].

This paper considers the linear characteristics of VBP, design of the pure VBP-based cipher oriented to cheap hardware implementation, and its security against differential and linear attacks. In section 2 we consider general design of the CP boxes. We also construct mutually inverse CP boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$ of the order $h = 2$, (see Definition 3) both of them having the same topology. In section 3 we consider algebraic and probabilistic properties of CP. Linear characteristics of the CP boxes are estimated in general case. In section 4 a pure VBP-based cipher DDP-64 using simple key scheduling is described. In section 5 the linear and differential analysis of DDP-64 is considered. We also propose to use switchable operations to avoid weak keys and homogeneity of the encryption in the case of simple key scheduling.

Notation

$\diamond$ Let $\{0,1\}^s$ denote the set of all binary vectors $U = (u_1, ..., u_s)$, where $\forall i \in \{1, ..., s\}$ $u_i \in \{0,1\}$.

$\diamond$ The Hamming weight $\varphi(U)$ of U be defined as the number of nonzero components of U and $\varphi'(U)$ denote the parity of $\varphi(U)$, i.e. $\varphi(U) \overset{def}{=} \sum_{i=1}^{s} u_i$, where $\varphi(U) \in \{0, 1, ..., s\}$ and $\varphi'(U) \overset{def}{=} \varphi(U) \bmod 2$.

$\diamond$ Let us fix i, $i \in \{1, ..., s\}$, and write E_i, for which $\varphi(E_i) = 1$ and $e_i = 1$.

$\diamond$ Let $E_0 = (0, ..., 0)$ and $D_0 = (1, ..., 1)$, i.e. $\varphi(E_0) = 0$ and $\varphi(D_0) = s$.

$\diamond$ Let $X \oplus Y$ denote the bit-wise XOR (EXCLUSIVE-OR) operation of the two vectors X and $Y :$ $X, Y \in \{0,1\}^s$.

$\diamond$ Let $X \otimes Y$ denote bit-wise AND operation of the two vectors

X and Y: $X, Y \in \{0,1\}^s$. For $c \in \{0,1\}$ and $X \in \{0,1\}^s$ we define $Y = c \cdot X$, where $y_i = c \cdot x_i$ $\forall i \in \{1, ..., s\}$.

$\diamond$ $\overline{U}$ denotes bit-wise complement of U, i.e. $\overline{U} \overset{def}{=} U \oplus D_0$ $\forall U \in \{0,1\}^s$.

$\diamond$ Let $\bullet$ denote the binary scalar product: $c = A \bullet X = \varphi'(A \otimes X)$ $(c \in \{0,1\})$.

$\diamond$ Let $Y = X^{\ggg k}$ denote rotation of the word X by k bits, where $\forall i \in \{1, ..., n-k\}$ we have $y_i = x_{i+k}$ and $\forall i \in \{n-k+1, ..., n\}$ we have $y_i = x_{i+k-n}$.

2 Design of fast CP boxes

Let $Y = \mathbf{F}(X, V)$ be the two-variable function $\mathbf{F}: \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n$.

Definition 1. *Function $\mathbf{F}(X, V)$ is called a CP box (or $\mathbf{P}_{n/m}$-box), if for each fixed V the function $\mathbf{F}(X, V)$ is a bijective mapping defined as bit permutation.*

For fixed V we have fixed bit permutation operation called CP modification or modification of VBP operation. We shall denote modifications as $\mathbf{F}_V$ or $\mathbf{P}_V$. We shall also use notation $\mathbf{P}_{n/m}^{(V)}$ for CPB with n-bit input, n-bit output, and m-bit control input . Thus, the notation $Y = \mathbf{P}_{n/m}(X, V) = \mathbf{P}_{n/m}^{(V)}(X)$ means $Y = \mathbf{P}_V(X)$.

In section 3.2 we use the following statements:

1. $\mathbf{F}_V(A) = B \Rightarrow \mathbf{F}_V(\overline{A}) = \overline{B}$.

2. $\mathbf{F}_V(A \otimes B) = \mathbf{F}_V(A) \otimes \mathbf{F}_V(B)$.

3. $\mathbf{F}_V(A) = B \Rightarrow \varphi(A) = \varphi(B)$ and $\varphi(A) \neq \varphi(B) \Rightarrow \mathbf{F}_V(A) \neq B$.

4. $\varphi(A \oplus B) = \varphi(A) + \varphi(B) - \varphi(A \otimes B)$. If $A \otimes B = E_0$, then $\varphi(A \oplus B) = \varphi(A) + \varphi(B)$.

While constructing CPBs it is preferable to use the layered topology of PNs, since it permits to design very fast CPBs. A layered CPB $\mathbf{P}_{n/m}$ (Fig. 1) can be represented as superposition of $s = 2m/n$ active layers separated with $s-1$ fixed permutations that are implemented in hardware as simple connections. Each active layer (Fig. 1b) in a CPB with n-bit input is represented by the set of $n/2$ elementary boxes $\mathbf{P}_{2/1}$ controlled with one bit v: $y_1 = x_{1+v}$ and $y_2 = x_{2-v}$ (see Fig. 1a). General structure of the layered CPB is shown in Fig. 1c.

In all figures in this paper the solid lines indicate data movement, while dotted lines indicate the controlling bits. We assume that in a layered CP box all elementary switching elements are consecutively numbered from left to right and from top to bottom and the ith bit of vector V controls the ith switching element $\mathbf{P}_{2/1}$. In accordance with the number of layers the vector V can be represented as concatenation of s vectors $V_1, V_2, ..., V_s \in \{0,1\}^{n/2}$, i.e. $V = (V_1, V_2, ..., V_s) = V_1|V_2|...|V_s$.

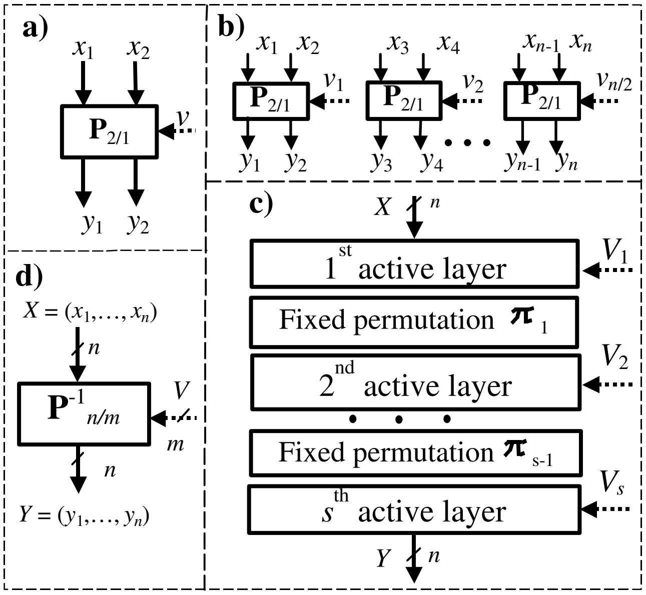

Figure 1. Notation of the $\mathbf{P}_{2/1}$- (a) and $\mathbf{P}^{-1}_{n/m}$-boxes (d), structure of one active layer (b) and general structure of the layered CP boxes (c)

The following two definitions we use according to [5].

Definition 2. *The CP boxes $\mathbf{P}_{n/m}$ and $\mathbf{P}^{-1}_{n/m}$ are mutual inverses, if for all possible values of the vector V the corresponding CP modifications $\mathbf{P}_V$ and $\mathbf{P}^{-1}_V$ are mutual inverses.*

Definition 3. *Suppose for arbitrary $h \le n$ input bits $x_{\alpha_1}, x_{\alpha_2}, ..., x_{\alpha_h}$ and arbitrary h output bits $y_{\beta_1}, y_{\beta_2}, ..., y_{\beta_h}$ there is at least one value V which specifies a permutation $\mathbf{P}_V$ moving x_{α_i} to y_{β_i} for all $i = 1, 2, ..., h$. Such a $\mathbf{P}_{n/m}$-box is called a CP box of the order h.*

One active layer can be considered as the single-layer CPB $\mathbf{S}_n$. It is evidently that $\mathbf{P}_{2/1} = \mathbf{P}_{2/1}^{-1}$, therefore $\mathbf{S}_n = \mathbf{S}_n^{-1}$. A layered CPB $\mathbf{P}_{n/m}$ can be represented as superposition of the bit permutation operations: $\mathbf{P}_{n/m} = \mathbf{S}_n^{(V_1)} \circ \pi_1 \circ \mathbf{S}_n^{(V_2)} \circ \pi_2 \circ ... \circ \pi_{S-1} \circ \mathbf{S}_n^{(V_s)}$. The respective box $\mathbf{P}_{n/m}^{-1}$ has the following structure $\mathbf{P}_{n/m}^{-1} = \mathbf{S}_n^{(V_s)} \circ \pi_{s-1}^{-1} \circ \mathbf{S}_n^{(V_{s-1})} \circ \pi_{s-2}^{-1} \circ ... \circ \pi_1^{-1} \circ \mathbf{S}_n^{(V_1)}$. Thus, to construct inverse of the CP box $\mathbf{P}_{n/m}$ it is sufficient to number the boxes $\mathbf{P}_{2/1}$ from left to right and *from bottom to top* and to replace π_i by π_{s-i}^{-1}. We shall assume that in the boxes $\mathbf{P}_{n/m}^{-1}$ the switching elements $\mathbf{P}_{2/1}$ are consecutively numbered from left to right and from bottom to top, i.e. in the both $\mathbf{P}_{n/m}$ and $\mathbf{P}_{n/m}^{-1}$ the ith bit of V controls the ith elementary box $\mathbf{P}_{2/1}$. Note that the vector V_j corresponding to the jth active layer in the box $\mathbf{P}_{n/m}$ controls the $(s-j+1)$th active layer in $\mathbf{P}_{n/m}^{-1}$.

In the VBP-based ciphers described below there are used $\mathbf{P}_{32/96}$- and $\mathbf{P}_{32/96}^{-1}$-boxes. These boxes have the same structure comprising two subsequent cascades. The upper one consists of four boxes $\mathbf{P}_{8/12}$ (Fig. 2a) and the lower one consists of four parallel boxes $\mathbf{P}_{8/12}^{-1}$ (Fig. 2b). The cascades are separated with fixed permutational involution described as follows:

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

The structure of the boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}_{32/96}^{-1}$ is presented in Fig. 2c and 2d. One can show that both of these CP boxes have the second order and each of two superpositions $\left(\mathbf{P}_{32/96}^{(V)}\right)^{-1} \circ \mathbf{P}_{32/96}^{(V')}$ and $\mathbf{P}_{32/96}^{(V)} \circ \left(\mathbf{P}_{32/96}^{(V')}\right)^{-1}$ represent a twelve-layer CPB of the order $h = 32$.

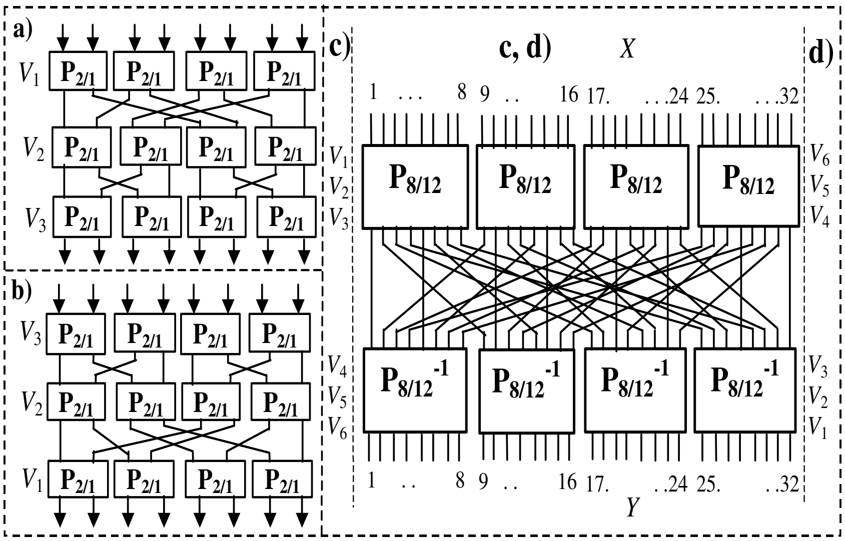

Figure 2. The first-order boxes $\mathbf{P}_{8/12}$ (a) and $\mathbf{P}^{-1}_{8/12}$ (b) and the second-order boxes $\mathbf{P}_{32/96}$ (c) and $\mathbf{P}^{-1}_{32/96}$ (d)

3 Properties of the controlled permutations

3.1 Terms of the linear cryptanalysis

Let $\mathbf{F} : \{0,1\}^r \to \{0,1\}^n, (r \geq n)$ be given. The resistance of the function against linear cryptanalysis (LCA) [10,11] is determined by the maximal value $|p|$, where

$$p = p_F = p_F(\Gamma u, \Gamma y) \overset{def}{=} \Pr_U(U \bullet \Gamma u \oplus Y \bullet \Gamma y = 0) - \frac{1}{2}, \qquad (1)$$

$U, \Gamma u \in \{0,1\}^r$, $Y \in \{0,1\}^n$, $\Gamma y \in \{0,1\}^n \setminus E_0$, and Γu, Γy are fixed vectors, that we called masks, and the value p is called the deviation (or bias). Similar to the notation used in [11] we describe linear characteristic (LC) of the function $\mathbf{F}$ as the combination $(\Gamma u, \Gamma y, p_F)$.

Let $Y = \mathbf{F}(X, V)$ be the two-variable function $\mathbf{F}$: $\{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$. Then for $U = X|V$, and $\Gamma u = \Gamma x|\Gamma v$ (1) is transformed in

$$p_F(\Gamma x, \Gamma y, \Gamma v) =$$
$$= p_F(\Gamma x|\Gamma v, \Gamma y) \overset{def}{=} \Pr_{X|V}(X \bullet \Gamma x \oplus V \bullet \Gamma v \oplus Y \bullet \Gamma y = 0) - \frac{1}{2}. \quad (2)$$

In particular, the value V corresponds to a subkey. For fixed value V the deviation has the form

$$p_{F_V}(\Gamma x, \Gamma y, \Gamma v) =$$
$$= p_{F_V}(\Gamma x|\Gamma v, \Gamma y) \overset{def}{=} \Pr_{X}(X \bullet \Gamma x \oplus V \bullet \Gamma v \oplus Y \bullet \Gamma y = 0) - \frac{1}{2}. \quad (3)$$

Below we shall also use the LC with the value p_{F_V}. According to different papers that dealt with the case of the uniformly distributed independent variables X and V the resistance of the function $\mathbf{F}$ against LCA can be estimated with the help of formulas using deviations p_{F_V}. For example, in [11] it has been derived the following formula:

$$LP^F_{max} \overset{def}{=}$$
$$\overset{def}{=} \max_{\Gamma x, \Gamma y \neq 0} LP^F(\Gamma x \to \Gamma y) \overset{def}{=} \max_{\Gamma x, \Gamma y \neq 0} \frac{1}{2^m} \sum_{V \in \{0,1\}^m} LP^{F_V}(\Gamma x \to \Gamma y),$$

where $\qquad LP^{F_V}(\Gamma x \to \Gamma y) \overset{def}{=} (2\frac{\#\{X \in \{0,1\}^n : X \bullet \Gamma x = F_V(X) \bullet \Gamma y\}}{2^n} - 1)^2.$

Actually $$LP^{F_V}(\Gamma x \to \Gamma y) = (2p_{F_V}(\Gamma x, \Gamma y))^2, \quad (4)$$

where $$p_{F_V}(\Gamma x, \Gamma y) = \Pr_X(X \bullet \Gamma x \oplus F_V(X) \bullet \Gamma y = 0) - \frac{1}{2}. \quad (5)$$

The next section of the present paper considers the case of arbitrary distribution of the variable V including the case $V = const$. In this calculation of the deviation p it is necessary to use the total probability formula:

$$p_F(\Gamma x, \Gamma y, \Gamma v) = \sum_V p_{F_V}(\Gamma x, \Gamma y, \Gamma v) \cdot P_V, \quad (6)$$

where P_V is the probability of the given value V.

3.2 Linear characteristics

Analyzing LC of CP boxes we assume, that X and V are independent variables, and $P_X = \frac{1}{|\{X\}|} = \frac{1}{2^n}$, secondly, for masks Γx, Γy, Γv we will use identifications A, B, C, in accordance with [11]. Let

$$\lambda_F(A,B) \overset{def}{=} \sum_V \theta_{F_V}(A,B) \cdot P_V, \tag{7}$$

where $\quad \theta_{F_V}(A,B) \overset{def}{=} \begin{cases} 1, & \text{if } \mathbf{F}_V(A) = B; \\ 0, & \text{otherwise.} \end{cases}$

Note that $\lambda_F(A,B)$ is the probability that $\mathbf{F}(A) = B$, i.e. $P_F(A \to B) \overset{def}{=} \lambda_F(A,B)$. Accordingly, let

$$\lambda_F(A,B,C) \overset{def}{=} \sum_V \theta_{F_V}(A,B,C) \cdot P_V, \tag{8}$$

where $\quad \theta_{F_V}(A,B,C) \overset{def}{=} \begin{cases} 1, & \text{if } \mathbf{F}_V(A) = B, \ C \bullet V = 0; \\ 0, & \text{otherwise.} \end{cases}$

Below we use the following statements:

1. $\varphi(A) \neq \varphi(B) \Rightarrow \mathbf{F}_V(A) \neq B \ \forall V \Rightarrow \lambda_F(A,B) = 0$.
2. $\varphi(A) = \varphi(B) \Rightarrow \lambda_F(\overline{A}, \overline{B}) = \lambda_F(A,B)$.
3. $\lambda_F(A,B) = \lambda_F(A,B,E_0)$.
4. $\forall C \ \lambda_F(A,B,C) \leq \lambda_F(A,B)$.

These statements are quite evident and can be easy derived using general properties of permutations. Let us consider the function

$$\Psi_V(X) \overset{def}{=} \begin{cases} 1, & \text{if } A \bullet X \oplus B \bullet \mathbf{F}_V(X) \oplus C \bullet V = 0 \ ; \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to see, that if $P_X = \frac{1}{|\{X\}|}$, then

$$p_{F_V}(A,B,C) = \frac{1}{|\{X\}|} \cdot \sum_X \Psi_V(X) - \frac{1}{2}. \tag{9}$$

Lemma. *Let X and V be independent variables and $P_X = \frac{1}{|\{X\}|}$. Then*

$$\frac{1}{|\{X\}|} \cdot \sum_X \Psi_V(X) = \begin{cases} \theta_{F_V}(A,B,C), & \textit{if } \ \mathbf{F}_V(A) = B; \\ 1/2, & \textit{if } \ \mathbf{F}_V(A) \neq B. \end{cases} \tag{10}$$

Proof. Let us consider two variants: $\mathbf{F}_V(A) = B$ and $\mathbf{F}_V(A) \neq B$.

Case 1. $\mathbf{F}_V(A) = B \Rightarrow \forall X: \ A \bullet X \oplus B \bullet \mathbf{F}_V(X) =$
$= A \bullet X \oplus \mathbf{F}_V(A) \bullet \mathbf{F}_V(X) = \varphi'(A \otimes X) \oplus \varphi'(\mathbf{F}_V(A) \otimes \mathbf{F}_V(X)) =$
$= \varphi'(A \otimes X) \oplus \varphi'(\mathbf{F}_V(A \otimes X)) = \varphi'(U) \oplus \varphi'(\mathbf{F}_V(U)) \equiv 0.$
Hence, $\forall X \ \Psi_V(X) = C \bullet V \oplus 1$ and

$$\frac{1}{|\{X\}|} \cdot \sum_{X, \mathbf{F}_V(A) = B} \Psi_V(X) = \begin{cases} 1, & \text{if } C \bullet V = 0; \\ 0, & \text{if } C \bullet V \neq 0. \end{cases} = \theta_{F_V}(A, B, C).$$

Case 2. $B \neq \mathbf{F}_V(A)$. Let A' : $B = \mathbf{F}_V(A')$. It is obvious, that $A' \neq A$. Let $A^{(1)} = \overline{A'} \otimes A$, $A^{(12)} = A \otimes A'$ and $A^{(2)} = \overline{A} \otimes A'$. Since $A' = A^{(12)} \oplus A^{(2)}$ and $^{(12)} \otimes A^{(2)} = E_0$ are held, then $\varphi(A' \otimes X) = \varphi(A^{(12)} \otimes X \oplus A^{(2)} \otimes X) = \varphi(A^{(12)} \otimes X) + \varphi(A^{(2)} \otimes X)$ and $\varphi(A \otimes X) = \varphi(A^{(1)} \otimes X) + \varphi(A^{(12)} \otimes X)$.
According to Case 1, $\forall X$ the equation $A' \bullet X \oplus B \bullet \mathbf{F}_V(X) = 0$ is held, i.e. $B \bullet \mathbf{F}_V(X) = A' \bullet X$. Then $A \bullet X \oplus B \bullet \mathbf{F}_V(X) = A \bullet X \oplus A' \bullet X =$
$= \varphi'(A \otimes X) \oplus \varphi'(A' \otimes X) =$
$= \varphi'(A^{(1)} \otimes X) \oplus \varphi'(A^{(12)} \otimes X) \oplus \varphi'(A^{(12)} \otimes X) \oplus \varphi'(A^{(2)} \otimes X) =$
$= \varphi'(A^{(1)} \otimes X) \oplus \varphi'(A^{(2)} \otimes X) =$
$= \left(\sum_{i|a_i^{(1)}=1} x_i\right) \bmod 2 \oplus \left(\sum_{i|a_i^{(2)}=1} x_i\right) \bmod 2 \Rightarrow$

$$\Psi_V(X) = \left(\sum_{i|a_i^{(1)}=1} x_i\right) \bmod 2 \oplus \left(\sum_{i|a_i^{(2)}=1} x_i\right) \bmod 2 \oplus C \bullet V \oplus 1.$$

Let $t = \varphi(A^{(2)})$. The whole set of binary vectors $\{X\}$ ($|\{X\}| = 2^n$) can be represented as an association of 2^{n-t} disjoint subsets, each of which contains 2^t vectors differing only in the digits corresponding to the active (non-zero) bits of the mask $A^{(2)}$. Note that for each such subset $\left(\sum_{i|a_i^{(1)}=1} x_i\right) \bmod 2$ and $C \bullet V \oplus 1$ are constants and $\left(\sum_{i|a_i^{(2)}=1} x_i\right) \bmod 2$ is even in exactly one-half cases, i.e. for $2^{n-t} \cdot 2^{t-1} = 2^{n-1}$ values X $\Psi_V(X) = 0$ and for remaining 2^{n-1} values X $\Psi_V(X) = 1$ are held. Therefore, for each permutations $\mathbf{F}_V$: $\mathbf{F}_V(A) \neq$

B we have

$$\frac{1}{|\{X\}|} \cdot \sum_{X, \mathbf{F}_V(A) \neq B} \Psi_V(X) = \frac{2^{n-1}}{2^n} = \frac{1}{2}. \qquad \square$$

Theorem 1. (About deviations of controlled permutations). *Let X and V be independent variables and $P_X = \frac{1}{|\{X\}|}$. Then*

$$2p_F(A,B,C) = 2\lambda_F(A,B,C) - \lambda_F(A,B) \tag{11}$$

Proof. In accordance with (6),(9) and (10) we have

$$2p_F(A,B,C) = 2\sum_V p_{F_V}(A,B,C) \cdot P_V =$$

$$= 2 \sum_{V, \mathbf{F}_V(A)=B} P_V \cdot p_{F_V}(A,B,C) + 2 \sum_{V, \mathbf{F}_V(A) \neq B} P_V \cdot p_{F_V}(A,B,C) =$$

$$= 2 \sum_{V, \mathbf{F}_V(A)=B} P_V \cdot \left(\theta_{F_V}(A,B,C) - \frac{1}{2}\right) + 2 \sum_{V, \mathbf{F}_V(A) \neq B} P_V \cdot \left(\frac{1}{2} - \frac{1}{2}\right) =$$

$$= 2\sum_V P_V \cdot \theta_{F_V}(A,B,C) - \sum_V P_V \cdot \theta_{F_V}(A,B) = 2\lambda_{F_V}(A,B,C) - \lambda_{F_V}(A,B).$$

□

Corollary 1. $2p_F(A,B) = 2p_F(A,B,E_0) = P_F(A \to B)$.

Indeed,
$2p_F(A,B) = 2p_F(A,B,E_0) = 2\lambda_{F_V}(A,B,E_0) - \lambda_{F_V}(A,B) =$
$= 2\lambda_{F_V}(A,B) - \lambda_{F_V}(A,B) = \lambda_{F_V}(A,B) = p_F(A \to B)$. □

Corollary 2. $\forall A, B \;\; 0 \leq 2p_F(A,B) \leq 1$.

Corollary 3. $\varphi(A) \neq \varphi(B) \Rightarrow p_F(A,B) = 0$.

Indeed,
$\varphi(A) \neq \varphi(B) \Rightarrow B \neq \mathbf{F}_V(A) \; \forall V \;\; \Rightarrow \lambda_F(A,B) = 0 \Rightarrow p_F(A,B) = 0$. □

Corollary 4. $\forall C \; |p_F(A,B,C)| \leq p_F(A,B)$.

Indeed, $0 \leq \lambda_F(A,B,C) \leq \lambda_F(A,B) \Rightarrow |2p_F(A,B,C)| =$
$= |2\lambda_F(A,B,C) - \lambda_F(A,B)| \leq \lambda_F(A,B) = 2p_F(A,B) \Rightarrow$
$\Rightarrow |p_F(A,B,C)| \leq p_F(A,B)$. □

Corollary 5. $\forall A,B \ \sum_{B:\varphi(B)=\varphi(A)} 2p_F(A,B) = 1$,
$\sum_{A:\varphi(A)=\varphi(B)} 2p_F(A,B) = 1$

Indeed, $\forall A,V \quad \exists! B : B = \mathbf{F}_V(A) \Rightarrow \sum_B \lambda_F(A,B) = 1$. Since $\sum_{B:\varphi(B)\neq\varphi(A)} \lambda_F(A,B) = 0$, then $\sum_B \lambda_F(A,B) =$
$= \sum_{B:\varphi(B)=\varphi(A)} \lambda_F(A,B) = 1 \Rightarrow \ \sum_{B:\varphi(B)=\varphi(A)} 2p_F(A,B) =$
$= \sum_{B:\varphi(B)=\varphi(A)} \lambda_F(A,B) = 1$. The second formula can be similarly derived. □

Corollary 6. $\forall A,B \ p_F(\overline{A},\overline{B}) = p_F(A,B)$. This is obvious, since $\lambda_F(\overline{A},\overline{B}) = \lambda_F(A,B)$. □

Corollary 7. Let $A = B = E_0$ or $A = B = D_0$. Then $2p_F(A,B) = 1$.

Corollary 8. Let $\varphi(A) = 1$. If $P_F(A \to B) = const \ \forall B \in \{0,1\}^n : \ \varphi(B) = 1$, then $2p_F(A,B) = P_F(A \to B) = \frac{1}{n}$, i.e. $p_F(A,B) = \frac{1}{2}(P_F(A \to B) = \frac{1}{2n}$.

Indeed, there exist exactly n of vectors $B \in \{0,1\}^n : \ \varphi(B) = 1$, therefore $P_F(A \to B) = const = 1/n$. This result was obtained in [12] for data-dependent rotations which are a particular case of the CP operations. □

Conclusion *The absolute value of the deviation of LC with a non-zero mask of the controlling vector does not exceed the value of the deviation of LC with a zero mask of V, the last being equal to half of the probability that a given input mask transforms into a given output mask.*

Since the theorem and its corollaries include the doubled value of deviation, it is reasonable to use the parameter $p' = |2p|$ which for zero mask of the controlling vector coincide with the probability $P_F(A \to B)$, i.e. $p' = |2p| = 2p_F(A,B) = P_F(A \to B)$.

The practical significance of the derived theorem and its corollaries lies in simplification of the calculation of LC:

1. To estimate LC of the CP boxes one can analyze only linear characteristics with zero mask of the controlling vector V.

2. The calculation of the deviation $p_F(A,B)$ is equivalent to the calculation of the probability $P_F(A \to B)$.

3. It is sufficient to analyze only deviations of LC for which $\varphi(A) \leq n/2$.

4. While designing CP boxes, the condition $\forall A, B \notin \{E_0, D_0\}$ $P_F(A \to B) \leq 1/n$ is to be satisfied.

Let us consider item 4. Let $\xi(t) \overset{def}{=} \max_{A,B:\varphi(A)=t} P_F(A \to B)$ be the function of the maximum value $P_F(A \to B)$ for given weight t. It is easy to see, that $\xi(n-t) = \xi(t)$, therefore it is enough to analyze this function only for $t = \{1, ..., n/2\}$. Since the number of different vectors B with weight $\varphi(B) = 1$ equals n, for $\varphi(A) = 1$ from corollary 5 one can obtain $\xi(t) \geq 1/n$. If $\forall A, B \in 0, 1^n : \varphi(A) = \varphi(B) = 1$ we have $P_F(A \to B) = const$, then $\xi(t) = 1/n$ is held. This is the case of the uniform CP boxes of the first order.

For arbitrary $t \leq n$ the number of different vectors B with weight t equals $\binom{n}{t}$. Thus, there are premises of the construction of CP boxes with monotonically descending function $\xi(t)$ for $t = \{1, ..., n/2\}$. For CP boxes of the hth order we have $P_F(A \to B) > 0$ $\forall A, B : \varphi(A) = \varphi(B) = h$. The approximately uniform CP boxes are characterized by the condition $P_F(A \to B) \approx const$ [5], hence for them we have $P_F(A \to B) \approx 1/q$, where $q = \binom{n}{h}$. A necessary condition for construction of such CP boxes is the inequality $2^m \geq \binom{n}{h}$, where m is the length of the controlling vector. However for weight 1 it is impossible to design a CP box with $p_F < \frac{1}{2n}$. While designing ciphers the VBP operations should be combined with other operations in order to thwart linear attacks using the LC with masks $A = B = (1, ..., 1)$ (see, for example, the use of the special nonlinear operation **G** in the cipher SPECTR-H64 [13]).

4 The block cipher DDP-64

While designing the single key cryptosystem DDP-64 our strategy was oriented to the extensive use of the controlled operations in the form of the CP box operations. This cryptographic primitive is fast and inexpensive while implementing in hardware. Our design criteria were the following:

1. The cryptosystem should be an iterated 64-bit cipher.

2. The cryptalgorithm should be able to perform encryption and decryption with simple and fast change of the sequence of the used subkeys.

3. The cipher should be fast in the case of frequent change of keys. For this reason we do not use precomputations.

4. Round transformation of data subblocks should be characterized by high parallelism.

5. The cipher should use only DDP as basic cryptographic primitive (therefore it is called DDP-64).

When designing a cipher based only on XOR and bit permutations (fixed and data-dependent ones) one of important problems is that combination of these operations usually gives ciphers which have the following property: "the parity of the plaintext + the parity of the key = the parity of the ciphertext". Such ciphers are not pseudorandom. To avoid this problem we have constructed a special operational box **F** that is based on fixed and data-dependent permutations. Structure of this box provides the arbitrary change of the oddness of the output. General structure of the encryption round proposed in [14] and implemented in the cipher SPECTR-H64 suites very well to satisfy our design criteria, therefore we have used SPECTR-H64 as a prototype when developing the pure VBP-based cipher DDP-64 having 64-bit input. The general encryption scheme of DDP-64 is described by the following formulas:

$$C = \mathbf{T}^{(e=0)}(M,K) \quad \text{and} \quad M = \mathbf{T}^{(e=1)}(C,K),$$

where M is the plaintext, C is the ciphertext ($M, C \in \{0,1\}^{64}$), K is the secrete key ($K \in \{0,1\}^{128}$), $\mathbf{T}$ is the transformation function, and

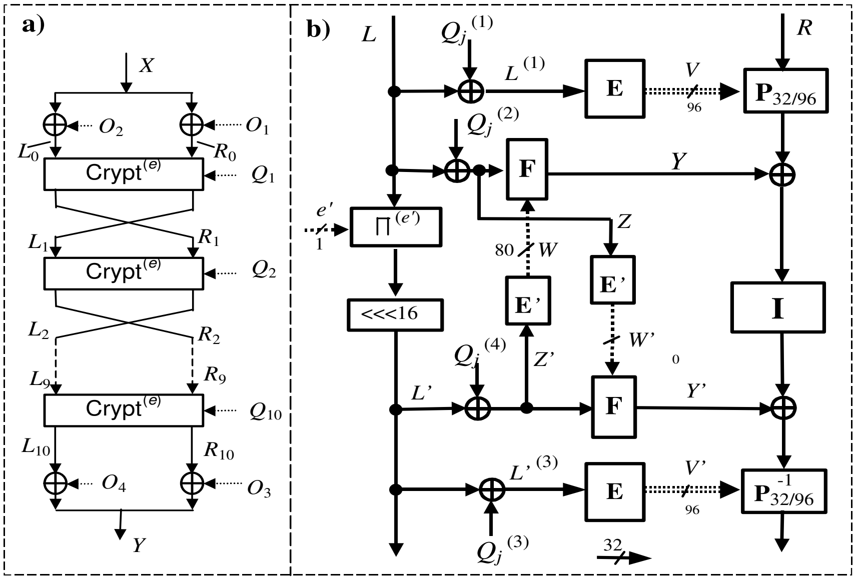

Figure 3. General structure of DDP-64 (a) and procedure **Crypt**$^{(e)}$ (b)

$e \in \{0, 1\}$ is a parameter defining encryption ($e = 0$) or decryption ($e = 1$) mode. The secrete key is considered as concatenation of four 32-bit subkeys K_i, $i = 1, 2, 3, 4$: $K = (K_1, K_2, K_3, K_4,)$. DDP-64 uses no preprocessing to transform subkeys. Iterative structure of DDP-64 is shown in Fig. 3a and can be described as follows. First data block X is divided into two 32-bit subblocks L and R and initial transformation is performed as XORing subblocks with corresponding subkeys. Then 10 rounds with procedure **Crypt**$^{(e)}$ followed by final transformation are performed. The structure of the procedure **Crypt**$^{(e)}$ is shown in Fig. 3b.

4.1 Formation of the round keys

Each round key $Q_j = (Q_j^{(1)}, Q_j^{(2)}, Q_j^{(3)}, Q_j^{(4)}) \in \{0, 1\}^{32}$ is some e-dependent transposition of the subkeys K_1, K_2, K_3, K_4. Figure 4 and Table 1 specify round subkeys and their correspondence to the secret key. Subkeys K_i $(i = 1, ..., 4)$ are used directly in each round avoiding any processing them. The transposing subkeys K_1, K_2, K_3, K_4 is performed with two boxes $\mathbf{P}^{(e)}_{2\times 32/1}$. The box $\mathbf{P}^{(e)}_{2\times 32/1}$ is some single-layer CPB in which all elementary switching elements are controlled with the same bit e. The pairs (K_1, K_3) and (K_2, K_4) are inputs of the corresponding boxes $\mathbf{P}^{(e)}_{2\times 32/1}$. Four 32-bit outputs of two boxes $\mathbf{P}^{(e)}_{2\times 32/1}$ are the e-dependent subkeys O_i $(i = 1, 2, 3, 4)$. Thus, we have $O_i = K_i$, if $e = 0$, and $O_1 = K_3$, $O_2 = K_4$, $O_3 = K_1$, $O_4 = K_2$, if $e = 1$. Being free of any precomputing subkeys and using the same algorithm to perform encryption and decryption the cipher DDP-64 suites well to cheap hardware implementation.

The left data subblock combined with subkeys $Q^{(1)}_j$ and $Q^{(3)}_j$ is used to form the controlling vectors V and V' which specify the current modifications of the VBP performed on the right data subblock with boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$, respectively. The left data subblock combined with subkeys $Q^{(2)}_j$ and $Q^{(4)}_j$ is also transformed with two $\mathbf{F}$-boxes implementing special variant of VBP.

4.2 Switchable fixed permutations

Change of the ciphering mode is defined by swapping subkeys K_i with two single-layer boxes $\mathbf{P}^{(e)}_{2\times 32/1}$ (see Fig. 4a) and by switching the e'-dependent fixed permutation $\Pi^{(e')}$, where $e' \in \{0, 1\}$ and e' depends on e and on the round number j. The e'-dependent fixed permutation in the left branch of the cryptoscheme is used to prevent homogeneity of the encryption procedure in the case of the key having structure $K = (X, X, X, X)$. For this reason the schedule of the switching bit e' is non-periodic (see Table 1). The structure of the switchable operations $\Pi^{(e')}$ is shown in Fig. 4b. It is easy to see that we have $\Pi^{(0)} = \Pi$, $\Pi^{(1)} = \Pi^{-1}$, and $\Pi^{(e'\oplus 1)}(Y) = X$, if $Y = \Pi^{(e')}(X)$. The permutation Π is specified as follows:

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

Table 1. Specification of the round subkeys and switching bit e'

$j =$	1	2	3	4	5	6	7	8	9	10
$Q_j^{(1)} =$	O_3	O_2	O_1	O_4	O_3	O_3	O_4	O_1	O_2	O_3
$Q_j^{(2)} =$	O_4	O_3	O_2	O_1	O_2	O_2	O_1	O_2	O_3	O_4
$Q_j^{(3)} =$	O_1	O_4	O_3	O_2	O_1	O_1	O_2	O_3	O_4	O_1
$Q_j^{(4)} =$	O_2	O_1	O_4	O_3	O_4	O_4	O_3	O_4	O_1	O_2
$e_j'^{(e=0)} =$	1	0	1	1	0	1	1	1	0	1
$e_j'^{(e=1)} =$	0	1	0	0	0	1	0	0	1	0

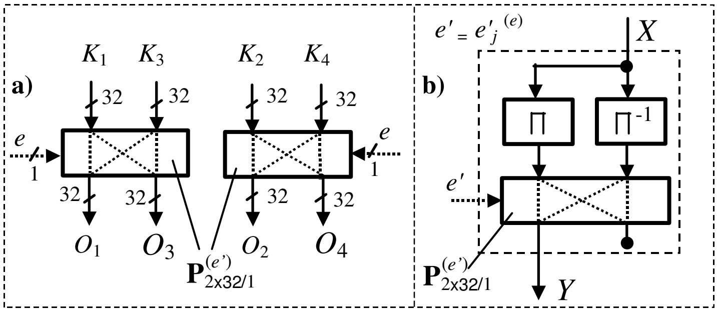

Figure 4. Swapping subkeys (a) and structure of the switchable fixed permutation (b)

4.3 Variable permutations

Variable permutations are performed with the CP boxes of the second order $\mathbf{P}^{(V)}_{32/96}$ and $\left(\mathbf{P}^{(V')}_{32/96}\right)^{-1}$ (see section 2) and **F**-boxes. The **F**-boxes represent special type of VBP. Construction of the **F**-boxes provides arbitrary change of the output vector weight. Indeed, depending on L, $Q_j^{(2)}$, and $Q_j^{(4)}$ eight of 32 input bits are replaced by bits of the constant $C = (10101010)$ while performing the operation **F**. Structure of the **F**-box is presented in Fig. 5. The **F**-box comprises two three-layer CP boxes $\mathbf{P}_{32/48}$ and $\mathbf{P}^{-1}_{32/48}$ separated with fixed permutation Π' which is described as follows:

(1,33)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,34,29,40)(10,35)(11,18)
(12,26)(14)(15,36,22,38)(16,30)(19,37)(20,27)(23)(24,31)(28,39)(32).

The 80-bit controlling vector $W = (W_1, W_2, W_3, W_4, W_5)$, where $W_i \in \{0,1\}^{16}$, of the $\mathbf{F}$-box is divided into 48-bit controlling vector (W_1, W_2, W_3) of the CP box $\mathbf{P}_{32/48}$ and 32-bit part (W_4, W_5) of the controlling vector (W_6, W_4, W_5) of the $\mathbf{P}^{-1}_{32/48}$-box. The 16-bit vector W_6 is formed with the extension box "Ext" (Fig. 5a) using eight of the most significant bits of the output $H = (H_1, H_2, H_3, H_4, H_5)$, where $H_i \in \{0,1\}^8$, of the permutation Π': $W_6 = (H_5, H_5)$. The 80-bit controlling vector W is formed with the extension box $\mathbf{E}'$ input of which is the vector Z'. Relation between Z' and W is the following: $W_1 = Z'_l$, $W_2 = Z'^{\ggg 5}_l$, $W_3 = Z'^{\ggg 10}_l$, $W_4 = Z'_h$, $W_5 = Z'^{\ggg 5}_h$.

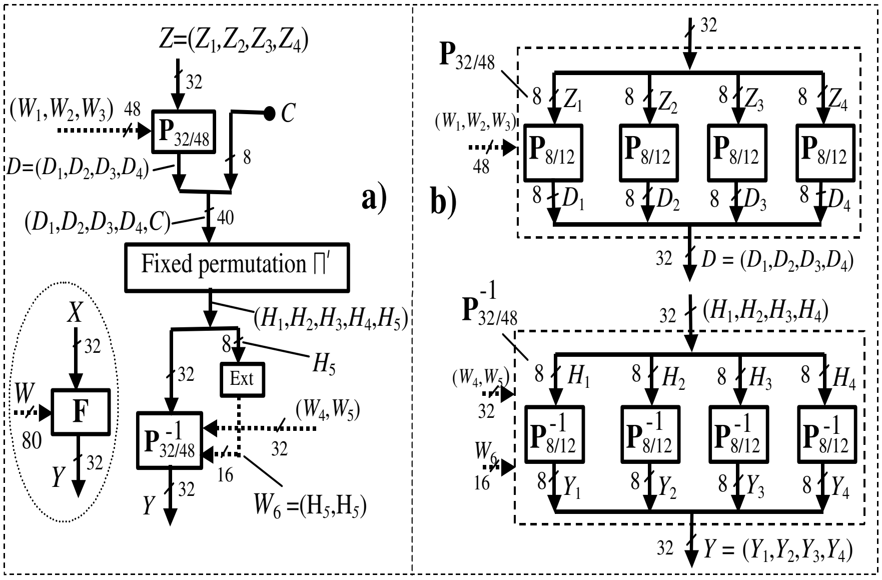

Figure 5. Structure of the $\mathbf{F}$-box (a) and of the CP boxes $\mathbf{P}_{32/48}$ and $\mathbf{P}^{-1}_{32/48}$ (b)

The vectors W_1, W_2, and W_3 control the 1st, 2nd, and 3d active layers of the $\mathbf{P}_{32/48}$-box and the vectors W_4, W_5, and W_6 control the 1st, 2nd, and 3d active layers of the $\mathbf{P}^{-1}_{32/48}$-box, correspondingly. The vector D that is the output of $\mathbf{P}_{32/48}$ is concatenated with constant C forming the vector (D_1, D_2, D_3, D_4, C) at input of the fixed permutation Π'. At output of Π' the vector $(H_1, H_2, H_3, H_4, H_5)$, where $H_5 = (d_1, d_8, d_{10}, d_{15}, d_{19}, d_{22}, d_{28}, d_{29})$, is formed. Taking into account the structure of the $\mathbf{P}_{32/48}$-box one can see that superposition $\mathbf{P}_{32/80} \circ \Pi'$ moves arbitrary two bits of each byte Z_i of the vector $Z = (Z_1, Z_2, Z_3, Z_4)$ to H_5 with the same probability. Arbitrary single bit of each byte Z_i moves to H_5 with probability 2^{-2}. Thus, the vector H_5 is composed of eight bits of $Z = L \oplus Q^{(4)}_j$ which are replaced by 8 bits of C at the output of the $\mathbf{F}$-box. Depending on W different bits of Z are replaced, therefore the oddness of the output vector of the $\mathbf{F}$-box changes arbitrarily.

4.4 Permutational involutions

Rotation operation "$\ggg$ 16" performed on the left data subblock is used as permutational involution saving the "symmetric" use of the most significant (L_h) and least significant (L_l) halfs of L while performing two $\mathbf{F}$-box operations. The fixed permutation $\Pi^{(e')}$ has been selected to provide condition $(\Pi^{(e')}(L))^{\ggg 16} = \Pi^{(e')}(L^{\ggg 16})$ for $e' \in \{0, 1\}$ which is necessary for correct decryption. Permutational involution $\mathbf{I}$ in the right branch provides each bit at the input of the box $\mathbf{P}_{32/96}$ influences 31 bits at the output of the box $\mathbf{P}^{-1}_{32/96}$ even in the case $V = V'$ (without $\mathbf{I}$ in the case $V = V'$ each input bit of $\mathbf{P}_{32/96}$ influences only one output bit of $\mathbf{P}^{-1}_{32/96}$). The involution $\mathbf{I}$ is described with two rotations by eight bits: $Y = \mathbf{I}(X_1, X_2) = (X_1^{\ggg 8}, X_2^{\ggg 8})$, where $X_1, X_2 \in \{0, 1\}^{16}$. This permutation improves the resultant VBP corresponding to subsequently performed operations $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$. Indeed, even in the case $V = V'$ the superposition $\mathbf{P}^{(V)}_{32/96} \circ \mathbf{I} \circ \left(\mathbf{P}^{(V')}_{32/96}\right)^{-1}$ forms an effective CP box permutation all modifications of which are different permutational involutions. In general case we have $V \neq V'$, since the data are combined with different subkeys while forming the controlling vectors corresponding to the operations $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$. Investigating the role of the fixed permutation between two mutually inverse CP box operations we have performed many statistic experiments which have shown that the use of such permutation significantly improves the properties of the transformation performed with two mutually inverse CP boxes.

4.5 Formation of the controlling vectors V and V'

Controlling vectors corresponding to the boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$ are formed using the same extension box $\mathbf{E}$ implemented with simple connections. The inputs of the $\mathbf{E}$-boxes corresponding to the boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$ are $L^{(1)} = L \oplus Q^{(1)}_j$ and $L'^{(3)} = (L'^{\ggg 16}) \oplus Q^{(3)}_j$ (see Fig. 3b), where $L' = \left(\Pi^{(e')}(L)\right)^{\ggg 16}$, respectively. Let the 96-bit vectors $V = (V_1, V_2, V_3, V_4, V_5, V_6)$ and $V' = (V'_1, V'_2, V'_3, V'_4, V'_5, V'_6)$ be the outputs of the respective $\mathbf{E}$-boxes. The extension box provides the following relations:

$$V_1 = L^{(1)}_l, \quad V_2 = (L^{(1)}_l)^{\ggg 6}, \quad V_3 = (L^{(1)}_l)^{\ggg 12}, V_4 = L^{(1)}_h,$$
$$V_5 = (L^{(1)}_h)^{\ggg 6}, \quad V_6 = (L^{(1)}_h)^{\ggg 12},$$
$$V'_1 = L^{(3)}_l, \quad V'_2 = (L^{(3)}_l)^{\ggg 6}, \quad V'_3 = (L^{(3)}_l)^{\ggg 12}, V'_4 = L^{(3)}_h,$$
$$V'_5 = (L^{(3)}_h)^{\ggg 6}, \quad V'_6 = (L^{(3)}_h)^{\ggg 12}.$$

The extension box provides each bit of L influences three elementary boxes $\mathbf{P}_{2/1}$ in the CP box $\mathbf{P}_{32/96}$ and three $\mathbf{P}_{2/1}$-boxes in $\mathbf{P}^{-1}_{32/96}$. While designing the box $\mathbf{E}$ we have used the following criterion: *For all values of the controlling vector the permutation of each input bit of CPB must be defined by six different bits of L.* Due to realization of this criterion each bit of L influences exactly six bits of R while performing the CPB operation. It is easy to see that such distribution of the controlling bits provides that arbitrary input bit of the boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$ moves to each output position with the same probability if L is a uniformly distributed random variable.

5 Discussion

Cipher DDP-64 presents an example of the pure VBP-based ciphers. The CP are extensively used in three different ways: (1) as VBP that are the basic cryptographic primitive, (2) as e-dependent swapping subkeys to change ciphering mode, and (3) in the switching permutation $\Pi^{(e')}$. Analogously to the VBP-based cipher SPECTR-H64 [13] the cryptosystem DDP-64 is fast in the case of frequent change of keys, since it is free of the key preprocessing. Avalanche effect spreads mostly when the changed bits are used as controlling ones, but not when they are transformed with the CP box operations (some avalanche connected with **F**-boxes is defined by the use of eight input bits as an internal controlling vector denoted earlier as W_6). In comparison with SPECTR-H64 the cipher DDP-64 has the following features:

1. It uses all secrete key in each round.

2. The DDP-64 is free of any additional nonlinear primirives (for example, the operation **G** in SPECTR-H64) and uses two **F**-boxes executed in parallel with the CP box operation $\mathbf{P}_{32/96}$. Each of two **F**-boxes is a special CP box generating at output the binary vector with arbitrary weight.

3. Round transformation includes special permutational involutions performed on the left and right data subblock and a switchable fixed permutation.

5.1 Some properties of VBP

For operations **F**, $\mathbf{P}_{32/96}$, and $\mathbf{P}^{-1}_{32/96}$ it is quite easy to calculate differential characteristics (DC) corresponding to differences with few number of active bits. Let Δ^U_h be the difference with arbitrary h active (non-zero) bits corresponding to some vector U. Let $\Delta_{h|i_1,...,i_n}$ be the difference with active bits corresponding to digits $i_1, ..., i_h$.

Avalanche effect corresponding to the operations $\mathbf{P}_{32/96}$, and $\mathbf{P}^{-1}_{32/96}$ is caused by the use of the data subblock L to define the values V and V'. Each bit of the left data subblock influences three bits of each of these controlling vectors. Each controlling bit influences two bits of the right data subblock. Thus, due to VBP performed on the right data subblock R with boxes $\mathbf{P}_{32/96}$, and $\mathbf{P}^{-1}_{32/96}$ one bit of L influences statistically about 12 bits of R. In the case when some difference with one active bit $\Delta^L_{1/i}$ passes the left branch of the cryptoscheme it influences three elementary switching elements permuting six different bits of the right data subblock. For example, if the input difference of the CP box $\mathbf{P}_{32/96}$ has no active bits (the case of zero difference), then the difference $\Delta^L_{1/i}$ can cause the generation at output of the $\mathbf{P}_{32/96}$-box the following differences: (1) Δ'_0 with probability 2^{-3}; (2) Δ'_2 with probability $3 \cdot 2^{-3}$; (3) Δ'_4 with probability $3 \cdot 2^{-3}$; (4) Δ'_6 with probability 2^{-3}. (Some other DC of the boxes $\mathbf{P}_{32/96}$ are presented in [15].)

Avalanche effect corresponding to the operations $\mathbf{F}$ relates to the use of the left data subblock to specify controlling vectors W and W'. Besides, avalanche spreads due to dependence of the output of "Ext"-box on L. Let consider the vector $L = (L_l, L_h)$ before the operation "$>\gg 16$". Each bit l_i, where $1 \leq i \leq 16$, of L_l influences three elementary boxes $\mathbf{P}_{2/1}$ of the $\mathbf{P}_{32/48}$-box in the lower $\mathbf{F}$-box and two boxes $\mathbf{P}_{2/1}$ of the $\mathbf{P}^{-1}_{32/48}$-box in the upper $\mathbf{F}$-box. Besides, with probability 2^{-2} (this probability corresponds to the event that l_i is moved to H_5) the bit l_i influences two boxes $\mathbf{P}_{2/1}$ of the $\mathbf{P}^{-1}_{32/48}$-box in the upper $\mathbf{F}$-box and with the same probability l_i influences two boxes $\mathbf{P}_{2/1}$ of the $\mathbf{P}^{-1}_{32/48}$-box in the lower $\mathbf{F}$-box. Analogous properties have all bits of L_h, since after the operation "$>\!\!\gg 16$" we have $(L_l, L_h)^{>\!\gg 16} = (L_h, L_l)$.

5.2 Security estimation

We have considered different types of attacks against DDP-64. Our results show that the differential cryptanalysis (DCA) is the most powerful attack. The iterative two-round DCs with differences (Δ^L_1, Δ^R_0) and (Δ^L_0, Δ^R_1) have the highest probability: $P(2) \approx P = 1.37 \cdot 2^{-17}$. The difference (Δ^L_1, Δ^R_0) passes eight and ten rounds of DDP-64 with probabilities $P(8) = P^4(2) \approx 1.79 \cdot 2^{-67}$ and $P(10) = P^5(2) \approx 1.23 \cdot 2^{-83}$. For some random cipher we have $P\left((\Delta^L_1, \Delta^R_0) \to (\Delta^L_1, \Delta^R_0)'\right) = 2^{-64} \cdot 2^5 = 2^{-59} > P(8) > P(10)$. Thus, DDP-64 with eight and ten rounds is undistinguishable from random cipher with differential attack using the most efficient two-round iterative characteristic.

Linear cryptanalysis (LCA) seems to be less efficient to attack DDP-64 as compared with DCA. Let denote the input mask as $A = (A^L, A^R)$ and the output mask as $B = (B^L, B^R)$. Linear attacks using masks $A = B = (1, 1, ..., 1)$ are prevented because of the use of two operations $\mathbf{F}$ changing arbitrary the oddness of their output. Using results of section 3 it is easy to find that the bias (deviation) of the linear characteristics (LC) with $z \leq 31$ active bits has value $b \leq 2^{-6}$ for each of the boxes $\mathbf{P}_{32/96}$, $\mathbf{P}^{-1}_{32/96}$, and $\mathbf{F}$, the maximal value corresponding to $z = 1$. Our linear analysis of DDP-64 has shown that among masks A^L, A^R, B^L, and B^R corresponding to individual subblocks and having weight less than 31 the masks with weight 1 have the maximal bias.

Analogously to consideration of the LC of the CP boxes, the LCA of the DDP-64 can be performed investigating the movement of the active bits through one or several encryption rounds. For LC with input mask $A = (A^L_{1|i}, A^R_{1|j})$ and output mask $B = (B^L_{1|i'}, B^R_{1|g})$, where indices indicate that we consider 32-bit masks with one active bit corresponding to ith and i'th (jth and gth) digits in the left (right) data subblocks at input and output respectively. It is easy to show that for arbitrary digits i, j, and g (digit i' is defined by digit i) the bias $b(1)$ of the one-round iterative LC (A, B, $b(1)$) is $b(1) = 0.56 \cdot 2^{-16}$. The last value is derived from the probability $p = 0.56 \cdot 2^{-15}$ that the jth bit of R is XORed two times with ith bit of L and then is moved to the gth digit at the output of the operation $\mathbf{P}^{-1}_{32/96}$. For r-round LC (A, B, $b(r)$) one can obtain $b(r) < 2^{-15r-1}$. For the random cipher LCs have bias $b \approx 2^{-32} > b(r) \gg 2^{-46} > b(3)$, therefore we can conclude that three-round DDP-64 is secure against LCA.

In spite of the simplicity of the key schedule the ”symmetric” keys $K' = (X, Y, Y, X)$ and $K'' = (X, X, X, X)$ are not weak or semi-weak, since decryption requires switching the fixed permutation in the left branch of the cryptoscheme of DDP-64 (from Fig. 3 it is easy to see that $\mathbf{T}^{(e=0)}(C, K'') \neq M$, where $C = \mathbf{T}^{(e=0)}(M, K'')$). It seems to be difficult to calculate a semi-weak key-pair for DDP-64, if it is still possible. Slide attacks in the case of ”symmetric” keys are also inefficient, since the encryption with DDP-64 is free of homogeneity (in the sense of [16]) due to the non-periodic schedule of the switching bit e' specifying the fixed permutation $\Pi^{(e')}$ performed on the left data subblock. This shows that the switchable operations can play sufficiently important role in the block ciphers which are free of the key preprocessing. For comparison one can remark that SPECTR-H64 which uses no switchable operations has weak keys (for all X its 256-bit key $K = (X, X, ..., X)$ is a weak one) and in the case of the weak key it seems vulnarable to slide attack.

Use of some strong key scheduling is a standard way to prevent weak keys and homogeneity in DDP-64, however this significantly encreases the hardware implementation cost.

5.3 Conclusion

Theoretic analysis of LC of the VBP operations conserving the weight of the transformed vector has shown that the principal problem in the design of VBP-based ciphers is to prevent LCA using masks $A = B = (1, 1, ..., 1)$. Examples of such attacks are proposed in [17,18]. In the known VBP-based ciphers SPECTR-H64 [13], SPECTR-128 [19], and Cobra-H64 [15] additional non-linear operations are used to thwart such variants of LCA. When developing a pure VBP-based cipher we have proposed a new type of the VBP operations (**F**-box operations) the use of which allows one to solve the mentioned problem without using additional non-linear operations.

The **F**-box operations have been used to design the cipher DDP-64. Presented analysis of DDP-64 illustrates efficiency of the use of VBP in the design of the block ciphers. The VBP thwarts well differential, linear, and other attacks allowing one to use comparatively small number of the encryption rounds. The efficiency of hardware implementation of the VBP-based ciphers is defined by the following factors: (1) VBP are efficient as cryptographic primitive, (2) property of the controllability of this primitive allows designing new advanced cryptoschemes, (3) VBP are fast and cheap in hardware [20]. Design of DDP-64 can be characterized as a design at bit level that defines low hardware implementation cost and high performance including the case of frequent change of keys.

Structure of DDP-64 suites well for detailed estimating DCs and LCs corresponding to differences and masks with few active bits. To attack DDP-64 the DCA is significantly more efficient than LCA. The DCA defines the minimum number of rounds for secure encryption with DDP-64.

We have also shown that in the case of the simple key scheduling the weak keys and homogeneity of the encryption can be prevented using switchable operations. Development of the simple and efficient switchable (e-dependent) operations is a new interesting item in the design of the ciphers that are free of precomputing the round keys.

References

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

[2] M. Portz, *A generallized description of DES-based and Benes-based permutation generators.* LNCS, vol. 718 (1992), pp. 397–409.

[3] M. Kwan, *The design of the ICE encryption algorithm.* The 4th International Workshop, Fast Software Encryption - FSE '97 Proc. LNCS, vol. 1267 (1997), pp. 69–82.

[4] B. Van Rompay, L.R. Knudsen, V. Rijmen, *Differential cryptanalysis of the ICE encryption algorithm.* The 6th International Workshop, Fast Software Encryption - FSE'98 Proc. LNCS, vol. 1372 (1998), pp. 270–283.

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

[6] R.L. Rivest, *The RC5 Encryption Algorithm.* The 2nd International Workshop, Fast Software Encryption - FSE'94 Proc. LNCS, vol. 1008 (1995), pp. 86–96.

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

[8] C. Burwick, D.Coppersmith, E.D'Avingnon, R.Gennaro, Sh.Halevi, Ch.Jutla, Jr.S.M.Matyas, L.O'Connor, M.Peyravian, D.Safford, N.Zunic, *MARS - a Candidate Cipher for AES.* 1st Advanced Encryption Standard Candidate Conference Proceedings, Venture, California, Aug. 20–22, 1998.

[9] V.M. Maslovsky, A.A. Moldovyan, N.A. Moldovyan, *A method of the block encryption of discrete data.* Russian patent # 2140710. Bull. no 30 (1999).

[10] M. Matsui, *Linear Cryptanalysis Method for DES Cipher.* LNCS, vol. 765 (1994), pp. 386–397.

[11] M. Matsui, *New structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis.* LNCS, vol. 1267 (1996), pp. 205–218.

[12] B.S. Kaliski, Y.L. Yin, *On differential and linear cryptanalysis of the RC5 encryption algorithm.* The International conference, Advances in Cryptology - CRYPTO'95 Proc. LNCS, vol. 963 (1995), pp. 171-184.

[13] N.D. Goots, A.A. Moldovyan, N.A. Moldovyan, *Fast encryption algorithm SPECTR-H64.* International workshop, Methods, Models, and Architectures for Network Security - MMM-ANCS'01 Proc. LNCS, vol. 2052 (2001), pp. 275–286.

[14] L.E. Alekseev, T.G. Belkin, A.A. Moldovyan, N.A. Moldovyan, *A method of the iterated encryption of data blocks.* Russian patent # 2140714. Bull. no 30 (1999).

[15] N.A. Moldovyan, *Fast DDP-Based Ciphers: Design and Differential Analysis of Cobra-H64.* Computer Science Journal of Moldova. 2003, no. 2.

[16] A. Biryukov, D. Wagner, *Advanced Slide Attacks.* Advances in Cryptology - Eurocrypt'2000 Proc. LNCS, vol. 1807 (2000), pp. 589–606.

[17] Ch. Lee, D.Hong, Sun. Lee, San. Lee, S. Yang, J. Lim, *A chosen plaintext linear attack on block cipher CIKS-1.* LNCS, vol. 2513, pp. 456-468.

[18] Y.Ko, D.Hong, S.Hong, S.Lee, J.Lim, *Linear Cryptanalysis on SPECTR-H64 with Higher Order Differential Property.* International Workshop, Methods, Models, and Architectures for Network Security Proc. LNCS, vol. 2776 (2003), pp. 298–307.

[19] N.D. Goots, B.V.Izotov, A.A.Moldovyan, N.A.Moldovyan. *Modern cryptography: Protect Your Data with Fast Block Ciphers,* Wayne, A-LIST Publishing, 2003.- 400 p. (www.alistpublishing.com).

[20] N. Sklavos, A. A. Moldovyan, O. Koufopavlou *Encryption and Data Dependent Permutations: Implementation Cost and Performance Evaluation.* International workshop, Methods, Models,and Architectures for Network Security - MMM-ANCS'03 Proc. LNCS, vol. 2776 (2003), pp. 337–348.

N.A. Moldovyan, A.A. Moldovyan, N.D. Goots,

Received July 22, 2003

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