J. Cryptology (2002) 15: 61–72
DOI: 10.1007/s00145-001-0012-9

Journal of
CRYPTOLOGY

A Cipher Based on Data-Dependent Permutations*

A. A. Moldovyan and N. A. Moldovyan
Specialized Center of Program Systems "SPECTR",
Kantemirovskaya 10, St. Petersburg 197342, Russia
spectr@vicom.ru

Communicated by Ernie Brickell

Received March 2000 and revised May 2001
Online publication 29 August 2001

Abstract. Data-dependent permutations (DDP) are introduced as basic cryptographic primitives to construct fast hardware-oriented ciphers. Some variants of the DDP operations and their application in the cipher CIKS-1 are considered. A feature of CIKS-1 is the use of both the data-dependent transformation of round subkeys and the key-dependent DDP operations.

Key words. Internal key scheduling, Data-dependent permutations, Controlled permutations, Fast block cipher.

1. Introduction

Data-dependent rotations (DDR) appear to be a very interesting cryptographic primitive. DDR were used for the first time by Becker in IBM [1] and later by Madryga [8]. DDR gained recognition recently after they were used extensively by Rivest in RC5 [11]. DDR are the only nonlinear building blocks in RC5. Since the strength of RC5 depends mainly on the properties of DDR, in recent years this cryptographic primitive has attracted the attention of cryptographers. It has been shown [5] that mixed use of DDR with some other simple operations is an effective way to thwart linear cryptanalysis.

Efficiency of the use of DDR is connected with the variable amount of the rotations while enciphering different plaintexts. Specifying a rotation amount for an n-bit binary vector takes only $\log_2 n$ bits of some controlling bit string. Therefore in RC5 the execution of DDR on one n-bit data subblock is independent of $n - \log_2 n$ bits of the other subblock. This is used in some theoretical differential attacks [2], [7]. To obtain good resistance to differential cryptanalysis in new ciphers RC6 [12] and MARS [3] DDR are combined with integer multiplication which is used to compute rotation amounts, with the output becoming dependent on all bits of the controlling data subblock.

* This work was carried out as part of the AFRL Funded Project #1994P which supported the authors.

DDR can be interpreted as a particular case of controlled permutations (CP) [10]. A class of the CP operations $P_{n/m}$ can be characterized by an ordered set $\{\Pi_0, \Pi_1, \ldots, \Pi_{2^m-1}\}$, where all Π_i, $i = 0, 1, \ldots, 2^m - 1$, are fixed permutations of some set of n bits (or simply CP-modifications). Let two operands $X = (x_0, x_1, \ldots, x_{n-1})$ and $V = (v_0, v_1, \ldots, v_{m-1})$ be given. The execution of the CP operation $P_{n/m(V)}(X)$ consists in performing the fixed permutation Π_V on X. The value $\Pi_V(X)$ is taken as the n-bit output Y of the CP operation: $Y = P_{n/m(V)}(X) = \Pi_V(X)$. It is assumed that the controlling vector V is dependent on all bits of some n-bit data subblock. Therefore it is desirable to have at least 2^n different CP-modifications. This will allow one to assign directly dependence of the selection of the current modification on all bits of the controlling data subblock.

CP operations can be performed very quickly in hardware, with permutation networks (PN) developed previously [13], and with the hardware implementation cost being inexpensive. At present we consider the CP operations as a cryptographic primitive which is useful for designing fast hardware-oriented encryption algorithms. However, chip makers can easily support encryption techniques based on CP by adding a CP instruction to the CPU.

In the second section some variants of CP-box operations are considered. Certain CP-boxes have a structure analogous to that of layered PN [13] except that all layers of the CP-boxes contain an equal number of the elementary building blocks which perform controlled transposition of two bits. It is shown that CP operations can be used as main building blocks contributing greatly to the security against linear (LA) and differential (DA) analysis.

In the third section we present a new block cipher CIKS-1 based on CP-boxes performing data-dependent permutations (DDP). The cipher is free of precomputations. Therefore it is fast under the condition of frequent key change. A peculiarity of the cipher is use of the data-dependent transformation of the round subkeys which can be interpreted as an internal key scheduling.

2. Design of the CP-Box Permutations

An operational box $P_{n/m}$ executing the CP permutations (CP-box $P_{n/m}$ or simply $P_{n/m}$-box) is presented in Fig. 1(a). In all the figures in this paper solid lines indicate data movement, while dotted lines indicate the control bits. In the general case the value V is assumed to be dependent on encrypted data and/or key. CP-boxes can be constructed as a superposition of the standard elementary $P_{2/1}$-boxes shown in Fig. 1(b). Such elementary building blocks are used, for example, in the ICE encryption algorithm [6]. A $P_{2/1}$-box is controlled by one bit v. It swaps two input bits, if $v = 0$, otherwise ($v = 1$) the bits are not swapped.

Definitions.

1. Let a $P_{n/m}$-box be given. Suppose for arbitrary $h \leq n$ input bits $x_1, x_2, \ldots, x_h$ and arbitrary h output bits $y_1, y_2, \ldots, y_h$ there is at least one CP-modification moving x_i to y_i for all $i = 1, 2, \ldots, h$. Such a $P_{n/m}$-box is called a CP-box of order h.
2. A strict CP-box is a $P_{n/m}$-box in which all CP-modifications are pairwise unequal.

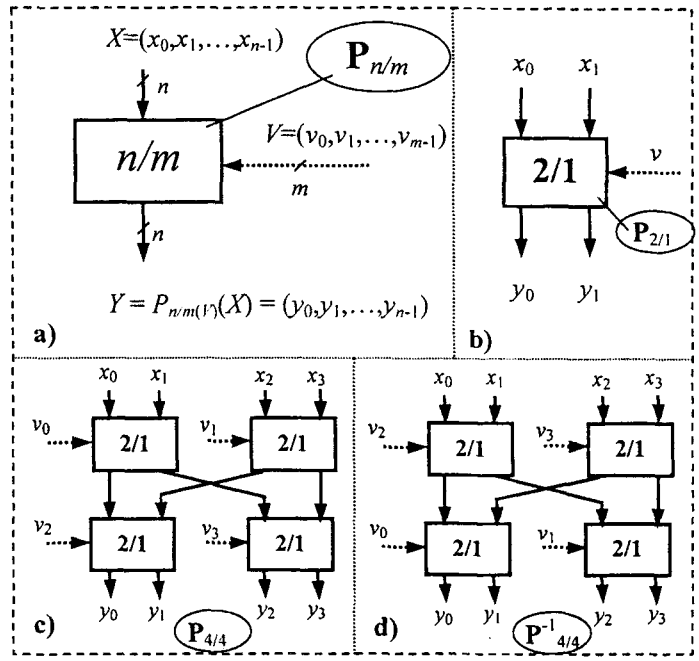

Fig. 1. CP-boxes: (a) $P_{n/m}$, (b) $P_{2/1}$, (c) $P_{4/4}$, and (d) $P_{4/4}^{-1}$.

3. CP-boxes $P_{n/m}$ and $P_{n/m}^{-1}$ are mutual inverses, if for all possible values of the vector V the corresponding CP-modifications Π_V and Π_V^{-1} are mutual inverses.

It is quite evident that the box $P_{n/m}^{-1}$ is of the order h, if the box $P_{n/m}$ is of the order h. If a CP box $P_{n/m}$ is strict, then the box $P_{n/m}^{-1}$ is strict. The maximal order of the CP box $P_{n/m}$ equals n. The set of CP-modifications of the $P_{n/m}$-box of maximal order contains all of $n!$ possible fixed permutations. Different structures of the $P_{n/m}$-boxes of maximal order are described in [4] and [13].

For cryptographic applications it is reasonable to try to compose such CP-boxes of order $h \leq n$ such that arbitrary h input bits $x_1, x_2, \ldots, x_h$ move with equal probability $p \neq 0$ to arbitrary h output bits $y_1, y_2, \ldots, y_h$, if V is a uniformly distributed random variable. Such CP-boxes can be called uniform. Unfortunately only for $h = 1$ is it easy to design uniform CP-boxes. For $h > 1$ CP-boxes can be only approximately uniform. If a CP box $P_{n/m}$ is uniform (or approximately uniform), then the box $P_{n/m}^{-1}$ is uniform (or approximately uniform).

Several variants of the CP-boxes are presented in Figs. 1–4 and Table 1. Each layer contains $n/2$ parallel $P_{2/1}$-boxes. The CP-box consists of $2m/n$ $P_{2/1}$-layers. Controlling bits corresponding to the lth layer we call the lth controlling substring $V(l)$. If each value $V(l)$ is set before the data bits cross the lth layer, then the time delay of the CP-box is defined by the number of layers only. A time delay of one $P_{2/1}$-layer is approximately equal to $t_{\oplus}$, where $t_{\oplus}$ is the time delay of the bit-wise exclusive-or (XOR) operation ($\oplus$).

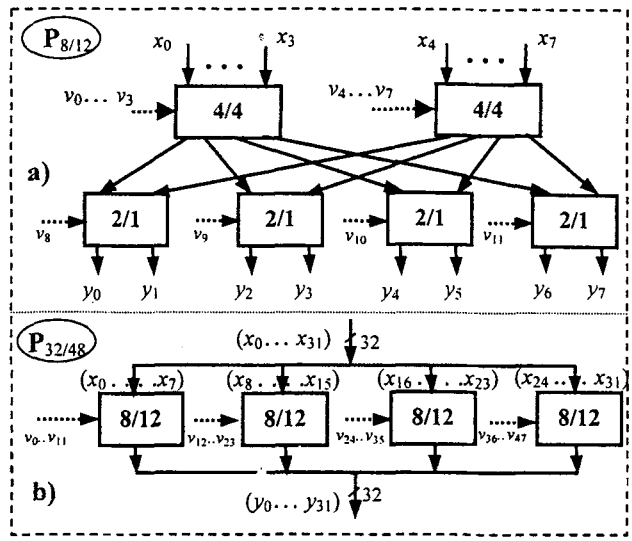

Fig. 2. Structure of the CP-box permutations: (a) $P_{8/12}$ and (b) $P_{32/48}$.

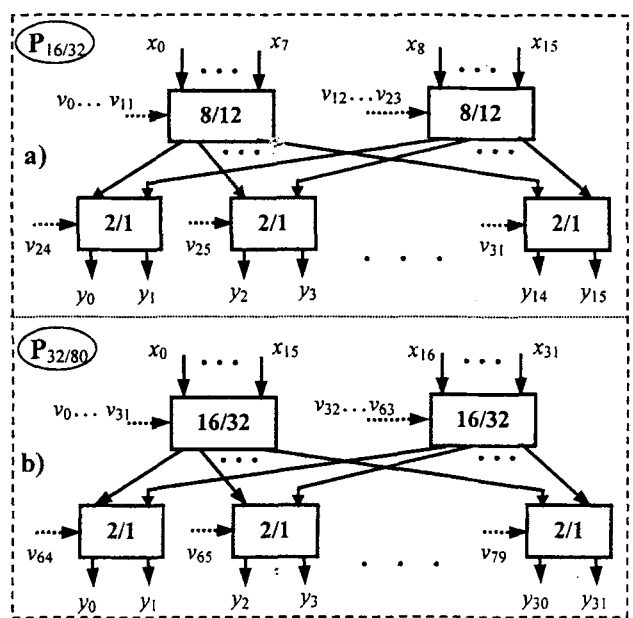

Fig. 3. Structure of the CP-box permutations: (a) $P_{16/32}$ and (b) $P_{32/80}$.

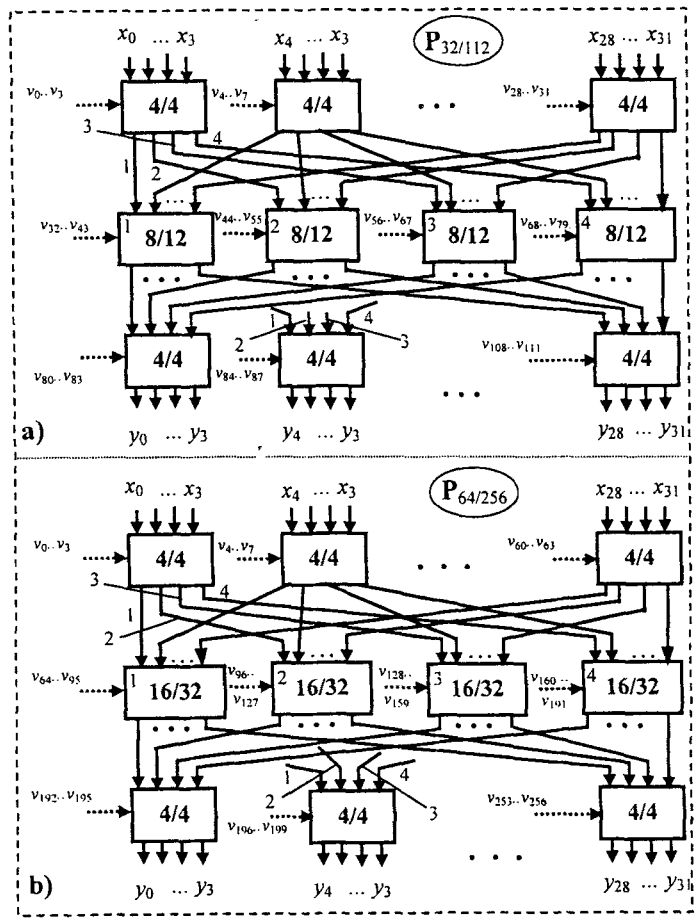

Fig. 4. Structure of the CP-box permutations: (a) $P_{32/112}$ and (b) $P_{64/256}$.

Time delay (T) of the $P_{n/m}$-box can be estimated as $T \approx 2mt_{\oplus}/n$. Since the number of layers is not large (≤ 8), the proposed CP-boxes are fast. The hardware implementation cost of the $P_{n/m}$-box is $4m$ nand gates.

By swapping the input and output of an arbitrary given $P_{n/m}$-box one can easily construct a respective inverse $P_{n/m}^{-1}$-box, with the same controlling substring $V_{(l)}$ corresponding to the lth layer of the $P_{n/m}$-box and to the $(2m/n - l + 1)$th layer of the $P_{n/m}^{-1}$-box. For example, Fig. 1(d) represents the CP-box operation $P_{4/4}^{-1}$ which is the inverse of the CP-box operation $P_{4/4}$ (Fig. 1(c)).

An evaluation of the differential characteristics and linear approximations of the CP-boxes can be obtained by presupposing the uniform distribution of the random variable V. (A justification of this model is that the controlling vector is supposed to be dependent on data and key.) It is quite evident that linear approximations with different numbers of input active bits (z_{in}) and output active bits (z_{out}) have zero bias p_{LA}. In the case

Table 1

CP-box	Figure	Uniform	Strict	h	$p_{LA}(1)$	$p_{DA}(1)$	$T \approx$
$P_{2/1}$	1(b)	+	+	1	2^{-2}	2^{-1}	$t_{\oplus}$
$P_{4/4}$	1(c)	+	+	1	2^{-3}	2^{-2}	$2t_{\oplus}$
$P_{8/12}$	2(a)	+	+	1	2^{-4}	2^{-3}	$3t_{\oplus}$
$P_{32/48}$	2(b)	−	+	0	2^{-4}	2^{-3}	$3t_{\oplus}$
$P_{16/32}$	3(a)	+	+	1	2^{-5}	2^{-4}	$4t_{\oplus}$
$P_{32/80}$	3(b)	+	+	1	2^{-6}	2^{-5}	$5t_{\oplus}$
$P_{32/112}$	4(a)	− (+)	−	4 (1)	2^{-6}	2^{-5}	$7t_{\oplus}$
$P_{64/256}$	4(b)	− (+)	−	4 (1)	2^{-7}	2^{-6}	$8t_{\oplus}$

$0 < z_{in} = z_{out} = z \leq h$ for the approximately uniform $P_{n/m}$-box of the order h $(1 \leq h \leq n-1)$ the bias is given by the following formula

$$p_{LA}(z) \approx \frac{1}{2}\binom{n}{z}^{-1}.$$

Differential characteristics of the approximately uniform CP-boxes with $z \leq h$ active bits have probability

$$p_{DA}(z) \approx \binom{n}{z}^{-1}.$$

If the $P_{n/m}$-box is uniform as a CP-box of the first order, then we have $p_{LA}(1) = (2n)^{-1}$ and $p_{DA}(1) = n^{-1}$. Maximal values of $p_{LA}(z)$ and $p_{DA}(z)$ corresponds to $z = 1$ and are given in Table 1 for the described CP-boxes. For CP-boxes $P_{32/80}$ used in the CIKS-1 cipher described in the next section we have

$$p_{LA}(1) = 2^{-6}; \qquad 2^{-10} < p_{LA}(2) < 2^{-9};$$
$$p_{DA}(1) = 2^{-5}; \qquad 2^{-9} < p_{DA}(2) < 2^{-8}.$$

The rough estimation for $z = 2$ reflects the fact that the $P_{32/80}$-box is not a CP-box of the second order.

3. The Block Cipher CIKS-1

Designing the secret-key cryptosystem CIKS-1 our strategy was oriented to extensive use of the CP-box permutations which are fast and inexpensive in hardware implementation. As cryptographic primitives they have the following peculiarities:

1. It is easy to construct CP-box permutations representing a single nonlinear operation on the whole data block. One of the data subblocks does not change, but it takes part in the transformation since it influences the controlling binary vector V.
2. CP-boxes realize relatively large numbers of different CP-modifications (for example, 2^{80} for $P_{32/80}$-box).
3. The avalanche effect spreads mainly via the use of the data bits as controlling ones while performing the CP-box permutations. If a CP-box permutation is strict and

each bit of the controlling data subblock influences m/n bits of V, then it influences statistically $2m/n$ bits of the permuted subblock.

4. Round subkeys can be combined with data via using them as a part of the controlling vector V. This is a way to define key-dependent fixed permutations and key-dependent DDP-permutations.
5. CP-box permutations can be efficiently used in the design of some internal key scheduling (IKS) which consists in **data-dependent transformation of the round subkeys** [9]. Operations corresponding to IKS can be performed simultaneously with some other operations corresponding to data ciphering. In this case IKS introduces no time delay. It is attractive to use IKS instead of ordinary key scheduling, since ciphers with IKS can provide higher average values of the encryption speed in the case of frequent change of keys.

Cipher CIKS-1 is represented in Figs. 5 and 6. In addition to CP-box permutations, one encryption (decryption) round of CIKS-1 uses fixed permutations (Π_1, Π_2, and rotations by 7 bits), one XOR operation, and 16 parallel modulo 2^2 additions (subtractions) denoted as a single operation "$+\cdots+$" ("$-\cdots-$"). Operation XOR combines the right data subblock with current round subkey after the latter is permuted in dependence on

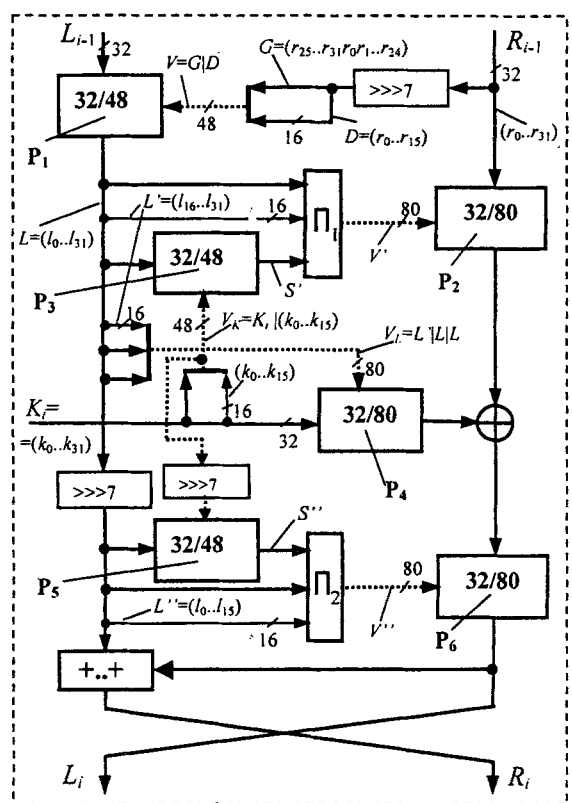

Fig. 5. CIKS-1: a cipher with internal key scheduling.

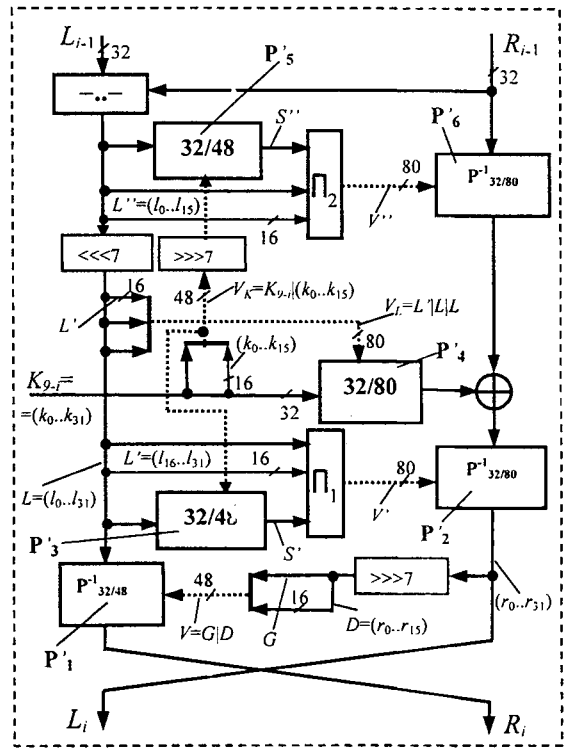

Fig. 6. Decryption round of CIKS-1.

the left subblock. Sixteen parallel modulo 2^2 additions (subtractions) are used to mix together values of the left and right subblocks instead of one modulo 2^{32} addition (subtraction), since they are significantly faster due to the restriction of the carry-spreading. To execute operation "$+\cdots+$" (or "$-\cdots-$") each of two 32-bit operands is divided into 16 2-bit operands and then 16 modulo 2^2 additions (subtractions) are performed simultaneously on respective pairs of the 2-bit operands. The time delay of the operation "$+\cdots+$" (or "$-\cdots-$") can be estimated as $\approx 3t_{\oplus}$.

Fixed permutations improve data diffusion, are obviously cheap in hardware, and introduce no time delay. Permutation Π_1 is described by the following formula:

$$V' = (v'_0, \ldots, v'_{79}) = \Pi_1(L|L'|S') = \Pi_1(l_0, \ldots, l_{31}, l_{16}, \ldots, l_{31}, s'_0, \ldots, s'_{31})$$
$$= (l_8, \ldots, l_{31}, s'_0, \ldots, s'_7, l_{16}, \ldots, l_{31}, l_0, \ldots, l_7, s'_8, \ldots, s'_{31}),$$

where $S' = (s'_0, \ldots, s'_{31})$ is the value of the left subblock permuted in dependence on the current round subkey (see Fig. 5). Since V' depends on both the round subkey and the left subblock, the operation P_2 is actually a key-dependent DDP-permutation. After permutation Π_1 the bits $s'_0, \ldots, s'_{31}$ compose the controlling substrings $V'(4)$ and $V'(5)$, therefore the $P_{32/48}$-box operation P_3 containing three $P_{2/1}$-layers can be executed simultaneously with the permutation corresponding to the first, second, and third

$P_{2/1}$-layers of the $P_{32/80}$-box P_2. The $P_{32/80}$-box operation P_4 on the current subkey is also performed simultaneously with $P_{32/80}$-box permutation P_2 (or with the $P^{-1}_{32/80}$-box permutation P'_6 in the case of decryption). Permutation Π_2 is described by the formula

$$\begin{aligned} V'' &= (v''_0, \ldots, v''_{79}) = \Pi_2(S''|L|L'') = \Pi_2(s''_0, \ldots, s''_{31}, l_0, \ldots, l_{31}, l_0, \ldots, l_{15}) \\ &= (s''_{16}, \ldots, s''_{23}, l_0, \ldots, l_3, s''_{24}, \ldots, s''_{31}, l_4, \ldots, l_{15}, s''_0, \\ &\quad \ldots, s''_7, l_{16}, \ldots, l_{19}, s''_8, \ldots, s''_{15}, l_{20}, \ldots, l_{31}, l_0, \ldots, l_{15}), \end{aligned}$$

where $S'' = (s''_0, \ldots, s''_{31}) = P_{32/48(V_K>>>7)}(L)$. After permutation Π_2 the bits $s''_0, \ldots, s''_{31}$ compose the controlling substrings $V''(1)$ and $V''(2)$ of the $P_{32/80}$-box P_6. Since substrings $V''(1)$ and $V''(2)$ correspond to the fourth and fifth $P_{2/1}$-layers of the $P^{-1}_{32/80}$-box, the $P_{32/48}$-box permutation P'_5 can be executed simultaneously with the permutation corresponding to the first, second, and third $P_{2/1}$-layers of the $P^{-1}_{32/80}$-box P'_6 while deciphering (see Fig. 6).

CIKS-1 is an eight-round iterated cipher with a 256-bit secret key represented as a set of 32-bit subkeys $K_1, K_2, \ldots, K_8$. CIKS-1 can be described by the following algorithm (operation "$\leftrightarrow$" denotes swapping subblocks):

Input: 64-bit plaintext $L|R$ represented as concatenation of two 32-bit subblocks L and R.

1. **For** $i = 1$ **to** 7 **do** $\{L|R := \mathbf{Crypt}(L|R);\ L \leftrightarrow R\}$.
2. $L|R := \mathbf{Crypt}(L|R)$.

Output: ciphertext $L|R$.

Table 2 specifies the procedure **Crypt** and presents an approximate estimation of the time delay T corresponding to the respective ciphering steps.

Table 2

Number	Encryption	$T \approx$
1	$V := (r_{25} \cdots r_{31}, r_0 \cdots r_{24}, r_0 \cdots r_{15});\quad L := P_{32/48(V)}(L)$	$3t_\oplus$
2	$V_L := L'\|L\|L;\quad V_K := k_0 \cdots k_{31}, k_0 \cdots k_{15};\quad S' := P_{32/48(V_K)}(L);$ $K := P_{32/80(V_L)}(K_i);\quad V' := \Pi_1(L\|L'\|S');\quad R := P_{32/80(V')}(R);$ $L := L >>> 7;\quad S'' := P_{32/48(V_K>>>7)}(L)$	$5t_\oplus$
3	$R := R \oplus K;\quad V'' := \Pi_2(S''\|L\|L'')$	$t_\oplus$
4	$R := P_{32/80(V'')}(R)$	$5t_\oplus$
5	$L := L + \cdots + R$	$3t_\oplus$
Number	**Decryption**	$T \approx$
1	$L := L - \cdots - R;\quad V_K := k_0 \cdots k_{31}, k_0 \cdots k_{15}$	$3t_\oplus$
2	$S'' := P_{32/48(V_K>>>7)}(L);\quad V'' := \Pi_2(S''\|L\|L'');$ $L := L <<< 7;\quad V_L := L'\|L\|L;\quad K := P_{32/80(V_L)}(K_{9-i});$ $S' := P_{32/48(V_K)}(L);\quad R := P^{-1}_{32/80(V'')}(R)$	$5t_\oplus$
3	$V' := \Pi_1(L\|L'\|S');\quad R := R \oplus K$	$t_\oplus$
4	$R := P^{-1}_{32/80(V')}(R)$	$5t_\oplus$
5	$V := (r_{25} \cdots r_{31}, r_0 \cdots r_{24}, r_0 \cdots r_{15});\quad L := P^{-1}_{32/48(V)}(L)$	$3t_\oplus$

Table 3

Operation	$\oplus$	$+\cdots+$ $(-\cdots-)$	$P_{32/48}$ $(P_{32/48}^{-1})$	$P_{32/80}$ $(P_{32/80}^{-1})$	One round	Eight rounds
Cell count (number of nand gates)	< 100	< 200	< 200	< 350	< 2000	< 16000
Time delay	$t_{\oplus}$	$\approx 3t_{\oplus}$	$\approx 3t_{\oplus}$	$\approx 5t_{\oplus}$	$\approx 17t_{\oplus}$	$\approx 136t_{\oplus}$

4. Discussion

CIKS-1 is well suited to hardware implementation. Fixed permutations are implemented with simple connections. The cell count for the implementation of other operations is approximately estimated in Table 3. With cheap technology CIKS-1 can be implemented as an iterated circuit using about 4000 nand gates. The full eight-round CIKS-1 can be implemented using about 32,000 nand gates, one 256-bit register, and two 64-bit registers. Our estimations of the encryption speed shows that the value 2 Gbit/s can be easy achieved.

Taking into account that one bit of controlling vector influences statistically two output bits, and that two transposed bits are equal with probability 0.5, it is easy to estimate numerically the avalanche effect. In one round single input bit of the left (right) subblock influences on the average $G_{LR} \approx 13$ ($G_{RR} \approx 17$) bits of the right half of the output and $G_{LL} \approx 8$ ($G_{RL} = 11$) bits of the left half of the output. For the round subkey one can consider analogous coefficients $G_{KL} \approx 6$ and $G_{KR} \approx 8$. After two rounds every input bit influences statistically all output bits. CIKS-1 has been implemented in software and tested. Experimental statistic examination have shown that three rounds sufficed to get uniform correlation between input and output bits.

To obtain some rough estimations of the minimal number of the plaintexts required for LA (N_{LA}) and pairs of the plaintexts required for DA (N_{DA}) one can neglect the contribution of both the IKS and the operation P_1 to the security of CIKS-1 and take into account only two $P_{32/80}$-box permutations performed on the right data subblock. A hypothetical nonzero difference δ passes two $P_{32/80}$-boxes of the right branch of one ciphering round ($f = 1$) with the maximal probability $p_{\mathrm{DA}(f=1)} = [p'_{\mathrm{DA(max)}}]^2$ (where $p'_{\mathrm{DA(max)}}$ is the probability corresponding to one $P_{32/80}$-box), if a zero difference goes through the left branch of the considered round.

Differential characteristics of the PC-boxes P_2 and P_6 with the maximal probability $p'_{\mathrm{DA(max)}}$ correspond to differences with one active bit. Therefore one can take $p'_{\mathrm{DA(max)}} = 2p_{\mathrm{DA}}(1) = 2^{-4}$, where the coefficient 2 takes into account the nonuniform distribution of the random variables V' and V''. Since for the full eight-round CIKS-1 the difference δ goes through the right branch in four rounds ($f = 4$) the maximal resultant probability is $p_{\mathrm{DA}(f=4)} = (p_{\mathrm{DA}(f=1)})^4 = [2p_{\mathrm{DA}}(1)]^8 = 2^{-32}$. From this value one can obtain the following estimation: $N_{\mathrm{DA}} \approx (p_{\mathrm{DA}(f=4)})^{-2} = 2^{64}$.

Analogous evaluation of the value N_{LA} gives $p_{\mathrm{LA}(f=1)} = 2[p'_{\mathrm{LA}}(1)]^2 = 2[2p_{\mathrm{LA}}(1)]^2 = 2^{-9}$; $p_{\mathrm{LA}(f=4)} = 2^{f-1}[p_{\mathrm{LA}(f=1)}]^f = 2^3(2^{-9})^4 = 2^{-33}$; $N_{\mathrm{LA}} \approx (p_{\mathrm{LA}(f=4)})^{-2} = 2^{66}$. Since the values $N_{\mathrm{DA}} \approx 2^{64}$ and $N_{\mathrm{LA}} \approx 2^{66}$ have been obtained with significant assumptions in the attacker's favour, CIKS-1 appears to be secure against both LA and

DA. For example, in the case of DA in several rounds the hypothetical difference δ goes through the left branch of the round encryption scheme causing a new difference in the right branch. The difference δ passes through the left branch without appearance of a new difference in the right branch with essentially low probability $p_{L\to R} < 2^{-6}$.

Preliminary analysis shows that CIKS-1 is secure against differential and linear attacks, although more detailed cryptanalysis is to be done to obtain precise numerical values of the work effort. The aim of this paper is mainly to focus on the CP-box permutations as a source of cryptographic strength and rapidity.

The avalanche effect of one round of CIKS-1 can be strengthened by using $P_{32/112}$-boxes instead of $P_{32/80}$-boxes, the additional 32 controlling bits are assumed to be dependent also on the left data subblock. Such modification evidently strengthens the cipher against DA giving the value $p_{L\to R} < 2^{-9}$. It is easy to construct a 128-bit CIKS-like cryptosystem using the CP-boxes $P_{64/256}$ instead of $P_{32/80}$-boxes and replacing three CP-box permutations P_1, P_3, and P_5 by the XOR operations. In further design of the ciphers based on CP-box permutations one can concentrate on the following items:

- Elaboration of the fast strict CP-boxes $P_{32/m\geq 32}$ and $P_{64/m\geq 64}$ realizing only one-cycle permutations.
- Elaboration of the mathematical techniques for accurate estimations of the linear approximations and differential characteristics for CP-boxes of different types and cryptoschemes on their basis.
- Construction of the fast round functions for the Feistel-like ciphers.
- Design of new variants of the key-dependent CP-box permutations.
- Design of new variants of the internal key scheduling.
- Design of fast CP-box instructions for possible application in mass microprocessors and in smart cards.

Acknowledgement

We are very grateful to an anonymous referee for his very helpful comments and suggestions.

References

[1] W. Becker, Method and system for machine enciphering and deciphering, U.S. patent # 4157454 (1979).

[2] A. Biryukov and E. Kushilevitz, Improved cryptanalysis of RC5, *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT* '98, Lecture Notes in Computer Science, vol. 1403, Springer-Verlag, Berlin, 1998, pp. 85–99.

[3] C. Burwick, D. Coppersmith, E. D'Avingnon, R. Gennaro, Sh. Halevi, Ch. Jutla, Jr., S.M. Matyas, L. O'Connor, M. Peyravian, D. Safford, and N. Zunic, MARS—a candidate cipher for AES, *Proceedings of the 1st Advanced Encryption Standard Candidate Conference*, Venture, California, Aug. 20–22, 1998 (see also http://www.nist.gov/aes).

[4] A.S. Kalendarev, A.A. Moldovyan, N.A. Moldovyan, and N.B. Savlukov, Encryption box, Russian patent # 2127024, Bull. no. 6 (1999).

[5] B.S. Kaliski and Y.L. Yin, On differential and linear cryptanalysis of the RC5 encryption algorithm, *Proceedings of the 15th Annual International Cryptology Conference, Advances in Cryptology - CRYPTO* '95, Lecture Notes in Computer Science, vol. 963, Springer-Verlag, Berlin, 1995, pp. 171–184.

[6] M. Kwan, The design of the ICE encryption algorithm, *Proceedings of the 4th International Workshop, Fast Software Encryption - FSE '97*, Lecture Notes in Computer Science, vol. 1267, Springer-Verlag, Berlin, 1997, pp. 69–82.

[7] R.L. Knudsen and W. Meier, Improved differential attacks on RC5, *Proceedings of the 16th Annual International Cryptology Conference, Advances in Cryptology - CRYPTO '96*, Lecture Notes in Computer Science, vol. 1109, Springer-Verlag, Berlin, 1996, pp. 216–228.

[8] W.E. Madryga, A high performance encryption algorithm, *Computer Security: a Global Challenge*, Elsevier, Amsterdam, 1984, pp. 557–570.

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

[10] A.A. Moldovyan and N.A. Moldovyan, A method of the cryptographical transformation of binary data blocks, Russian patent # 2141729. Bull. no. 32 (1999).

[11] R.L. Rivest, The RC5 encryption algorithm, *Proceedings of the 2nd International Workshop, Fast Software Encryption - FSE '94*, Lecture Notes in Computer Science, vol. 1008, Springer-Verlag, Berlin, 1995, pp. 86–96.

[12] R.L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, The RC6 block cipher, *Proceedings of the 1st Advanced Encryption Standard Candidate Conference*, Venture, California, Aug. 20–22, 1998 (see also http://www.nist.gov/aes).

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