PP-2 Block Cipher

Krzysztof Bucholc, Krzysztof Chmiel, Anna Grocholewska-Czurylo, Janusz Stoklosa
Institute of Control and Information Engineering
Poznan University of Technology
Poznan, Poland
{krzysztof.bucholc, krzysztof.chmiel, anna.grocholewska-czurylo, janusz.stoklosa}@put.poznan.pl

***Abstract*—The paper describes the rationale behind PP-2 block cipher and gives the details of the original design of this cipher. PP-2 block cipher is the result of continued development of the PP-1 block cipher, improving on performance, resistance against differential cryptanalysis and the speed of diffusion. Cipher structure is different for encryption and for decryption. Diffusion layer is based on multiple rotations. This paper shows that for block length n = 64 bits a global diffusion is reached after two rounds. Number of rounds is dependent on key size.**

Keywords - block ciphers; S-boxes; differential cryptanalysis; diffusion

I. INTRODUCTION

PP-2 cipher emerged as a continuation of PP-1 cipher development carried out by the Institute of Control and Information Engineering at the Technical University of Poznan. Initially, a 64-bit block length cipher PP-1 was designed, which was aimed at limited resource platforms, especially with very limited internal storage available for storing ciphers components [4]. Next, a scalable version of PP-1 cipher was developed, with block length being a multiple of 64 bits [3][6][8]. A distinctive feature of this cipher is that it constitutes an involutional substitution-permutation network. The same network, and particularly the same single S-box and the same single P-block, are used for both encryption and decryption process. PP-1 cipher is characterized by high resistance to differential and linear cryptanalysis and high performance [5][7].

Motivation for PP-2 block cipher was to develop a cipher which inherits advantages of PP-1 cipher, namely high resistance to cryptanalysis, while improving on performance. Limited resource platforms was no longer a target system for PP-2, as it was for PP-1. One of the aims for both PP-1 and PP-2 ciphers was an efficient implementation in both hardware and software.

Comparing PP-2 cipher to DES, PP-2 is characterized by higher resistance to linear and differential cryptanalysis, its software implementation is faster and it is scalable (variable block and key length). In comparison to Advanced Encryption Standard (AES) [1], PP-2 has roughly the same resistance to linear and differential cryptanalysis, similar software implementation efficiency and is scalable.

Main differences between PP-2 and PP-1 are that PP-2 is no longer involutional (to increase cipher efficiency dictated by higher number of rounds needed to resists Misztal's attack), an affine transform has been added to S-box construction, P-box based on multiple rotations further increases efficiency and the speed of diffusion, reducing the number of round keys and modifications to key generation algorithm increases efficiency even further.

In Section II, a general structure of PP-2 cipher is presented, which is an enhanced version of PP-1 cipher that further improves the performance, increases resistance to differential cryptanalysis [10] and speeds up the diffusion.

Section III describes the S-box designed for PP-2 cipher and gives the details of the method used to generate it.

Section IV describes the round key generation scheme. An important feature of round key generation algorithm is the independence of generated bit sequences forming the round keys, which improves cipher's security, as the cryptanalysis is more difficult for independent keys [2].

As for security considerations, in Section V, upper bounds of effectiveness of the PP-2 nonzero linear and differential approximations are given. Further and more detailed PP-2 security evaluation is planned.

Section VI gives the results of PP-2 cipher speed tests. Two compilers were tested: C++ compiler from MS Visual Studio 2008 package and Intel C++ Compiler Professional 11.1 for Windows. Application extensively utilizes 64-bits operations. For this reason both 32- and 64-bit code was generated for each compiler. Speed of cipher operation is roughly 50% higher for 64-bit version as compared to 32-bit version generated by Intel compiler. For VS2008 compiler, this difference was not that high, at roughly 33%.

II. GENERAL STRUCTURE OF PP-2

A. One round structure

PP-2 cipher is a symmetric scalable block cipher that processes data blocks of n bits in r rounds with the key k of the bit length $|k|$, such that $|k| = d \cdot n$, where $n = t \cdot 64$ and $d, t \in \{1, 2, 3, ...\}$. (Tab. 1).

1) Construction of the round

One round of the cipher is presented in Fig. 1. It is composed of $t = n/64$ parallel processing paths. In each path, a 64-bit nonlinear operation NL is executed. Furthermore, an n-bit permutation P is carried out. In each round, a round key k_i is used.

 ISBN: 978-1-61208-298-1

TABLE I. NUMBER OF ROUNDS DEPENDING ON THE BLOCK SIZE

Key length $\|k\|$ \ Block size n	64	128	192	256	...
64	11				
128	13	22			...
192	15	24	32		...
256	17	26	34	43	...

The nonlinear element *NL* is presented in Fig. 2. A 64-bit subblock is processed as eight 8-bit subblocks by four types of operations: S-boxes *S* of the size 8×8, XOR (⊕) of respective bits, addition (⊞) modulo 256 and subtraction (⊟) modulo 256 of integers represented by respective bytes. Two of these operations (addition and subtraction modulo 256) are local nonlinear mappings.

2) The layer structure of a round

In round *i*, the following 3 layers can be distinguished: the key addition layer, the substitution layer and the permutation layer. Functions of the layers will be denoted by *KL*, *SL* and *P*, respectively.

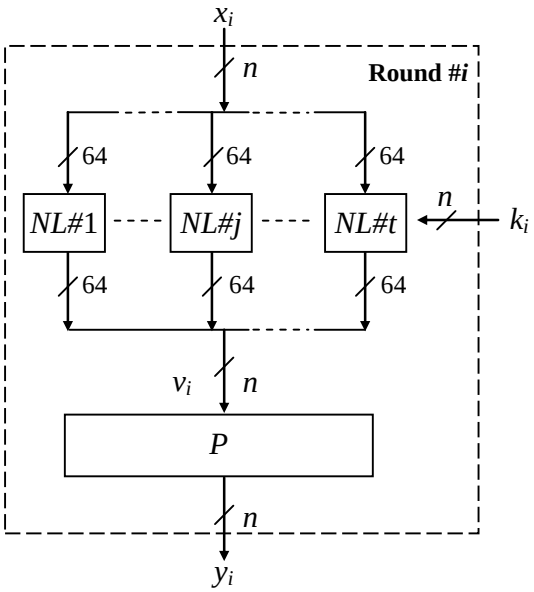

Figure 1. Processing in round #*i* (*i* = 1, 2, ..., *r*)

In Fig. 1, one round of the PP-2 cipher during encryption and decryption is presented. For decryption the order of the layers is inverted, and the appropriate (corresponding) inverse functions are used, i.e., P^{-1}, SL^{-1} and KL^{-1}.

During encryption, the round function *h* is the following composition (product) of the layer functions:

$$h = P \circ SL \circ KL. \quad (1)$$

For decryption, the inverse round function h^{-1} is calculated as follows:

$$h^{-1} = KL^{-1} \circ SL^{-1} \circ P^{-1}. \quad (2)$$

In layer *KL* of elements *NL* (Fig. 2), the following sequence of 8-bit operations is used: (⊞, ⊕, ⊟, ⊕, ⊕, ⊟, ⊕, ⊞), and in layer KL^{-1} – the sequence of their inverse operations, *i.e.* (⊟, ⊕, ⊞, ⊕, ⊕, ⊞, ⊕, ⊟). In layer *SL*, substitution *S* is used, and in layer SL^{-1}, substitution S^{-1}.

3) The cipher structure

In the PP-2 cipher, a different structure for encryption and decryption is used. The function *h* is used in each round *i* = 1, 2,…, *r*–1, while in the round number *r* (the output round) the function $\hat{h} = SL \circ KL$ is different – it is composed of *KL* and *SL*. Thus, the transformation for the encryption is as follows:

$$PP\text{-}2 = \hat{h} \circ h^{r-1} = (SL \circ KL) \circ (P \circ SL \circ KL)^{r-1} \quad (3)$$

Functions h^{-1} in rounds no. *i* = 2, 3,…, *r*, are the same, and function $\hat{h}^{-1} = KL^{-1} \circ SL^{-1}$ of round no. 1 (the input round) does not contain the permutation P^{-1}. The transformation for the decryption has, therefore, the following form:

$$PP\text{-}2^{-1} = (h^{-1})^{r-1} \circ \hat{h}^{-1} = (KL^{-1} \circ SL^{-1} \circ P^{-1})^{r-1} \circ (KL^{-1} \circ SL^{-1}) \quad (4)$$

.

Therefore, in the PP-2 cipher we have:

$$PP\text{-}2 \neq PP\text{-}2^{-1}, \quad (5)$$

and in consequence, a different network is used during encryption and decryption, and the round keys are applied in reverse order.

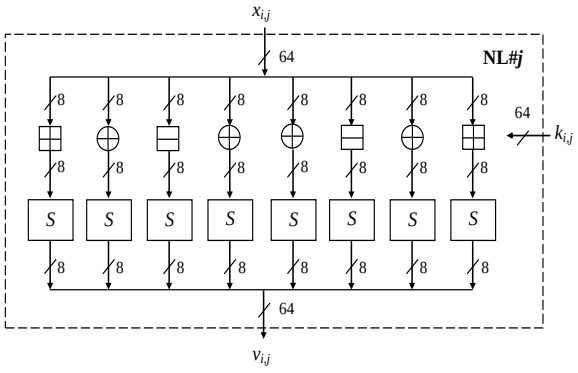

Figure 2. Nonlinear element *NL#j* of PP-2 (*j* = 1, 2, ..., *t*)

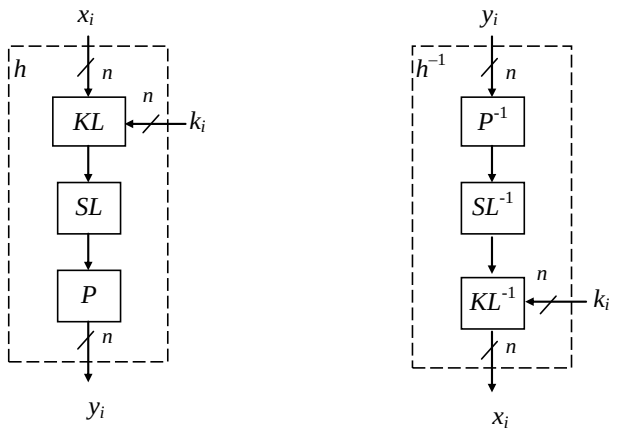

Figure 3. Round function h (encryption) and its inverse function h^{-1} (decryption)

B. Diffusion layer

1) Construction of multiple rotations

Assume that the bits of the input block and the output block are numbered successively, from 1 to n, from the left to the right. Consider permutation P of the PP-2 cipher as the transformation of bit numbers, i.e., as bijection P: {1, 2, ..., n} → {1, 2, ..., n}.

We define the rotation of bit i by b bits to the right as the following mapping:

$$ROR(b, i) = (i + b - 1) \bmod n + 1, \text{ for } i \in \{1, 2, ..., n\}. \quad (6)$$

For the set of indices, $I \subseteq \{1, 2, ..., n\}$ we define a rotation to the right as follows:

$$ROR(b, I) = \begin{cases} ROR(b, i) & \text{for } i \in I \\ 0 & \text{for } i \notin I. \end{cases} \quad (7)$$

For the PP-2 cipher with block length n = 64, the transformation of bit i with use of permutation P, called the multiple rotation to the right, is defined as follows:

$$P(i) = ROR(12, [1]) + ROR(28, [2]) + ROR(44, [3]) + ROR(60, [4]), \quad (8)$$

where $i \in \{1, 2, ..., n\}$ and:

$$[1] = \{1, 5, ..., 61\}, [2] = \{2, 6, ..., 62\}, [3] = \{3, 7, ..., 63\}, [4] = \{4, 8, ..., 64\}. \quad (9)$$

The sets defined by (9) are called the classes of bits. In the general case, *i.e.*, for $n = t \cdot 64$ (t = 1, 2, ...), the permutation P of the PP-2 cipher is defined identically (i.e., by Formula 8), and the classes of bits are defined as follows:

$$[1] = \{1, 5, ..., t\cdot64–3\}, [2] = \{2, 6, ..., t\cdot64–2\}, [3] = \{3, 7, ..., t\cdot64–1\}, [4] = \{4, 8, ..., t\cdot64–0\}. \quad (10)$$

The inverse permutation P^{-1} is defined as the multiple rotations to the left, in the following way:

$$P^{-1}(i) = ROL(12, [1]) + ROL(28, [2]) + ROL(44, [3]) + ROL(60, [4]), \quad (11)$$

where $i \in \{1, 2, ..., n\}$.

2) Diffusion for multiple rotations

In Fig. 4, a diffusion for (and introduced by) permutation P of the PP-2 cipher is presented, in the case of n = 64. For simplicity, the iteration function h of the cipher is restricted to SL and P layers. Bits dependent on bit number 1 after transformations in consecutive layers are denoted by dots. All bits of the output block are dependent on bit number 1 after 3 layers (i.e., after 2 rounds). Similarly, for the remaining bits.

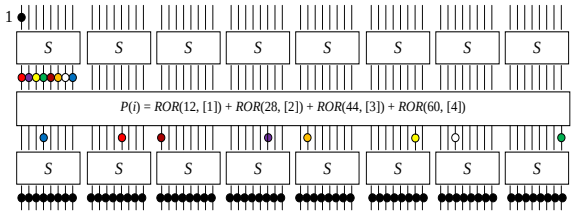

Figure 4. Diffusion for permutation P (n = 64)

Transformation P for bits numbered 1–8, dependent on bit number 1 after substitution S (see Fig. 4), is as follows (bits 1 and 5 are element of class [1], bits 2 and 6 belong to class [2], 3 and 7 – to class [3], while 4 and 8 – to class [4]):

$$P(1) = ROR(12, 1) = 13, P(5) = ROR(12, 5) = 17,$$
$$P(2) = ROR(28, 2) = 30, P(6) = ROR(28, 6) = 34,$$
$$P(3) = ROR(44, 3) = 47, P(7) = ROR(44, 7) = 51,$$
$$P(4) = ROR(60, 4) = 64, P(8) = ROR(60, 8) = 4. \quad (12)$$

3) Implementation of multiple rotations

Permutation P of the PP-2 cipher, which uses the multiple rotations, gives higher diffusion speed than involution P of the PP-1 cipher. Moreover, it is also much faster and easier in the software implementation. Isolation of the classes of bits can be obtained as the result of logical multiplication of the argument (block) by appropriate mask. Thus, calculation of P value, for an argument, requires at most four *AND* operations, four *ROR* operations and three addition operations, performed on n-bit words. Calculation of the inverse permutation P^{-1} during decryption is similar.

III. SUBSTITUTIONS BOXES

The S-box of the PP-2 cipher has been generated using the multiplicative inverse procedure with primitive polynomial \$87 defining the Galois field, so a procedure similar to that used to generate AES cipher S-box as well as that of PP-1 cipher. An important difference to AES cipher S-box is that PP-2 S-box does not have any affine equivalence between its component functions as in AES, which is always the case when an S-box is generated by an unmodified multiplicative inverse procedure.

Nonlinearity of this S-box is 110 and its nonlinear degree is 7. Individual Boolean functions that constitute this S-box S

= (F_0, F_1, F_2, F_3, F_4, F_5, F_6, F_7) have nonlinearities equal to 110 or 112.

Maximum value in the *XOR* profile is 4. For comparison, AES S-box [1] has nonlinearity equal to 112, and all other parameters equal to those of PP-2 cipher S-box.

S-box can be displayed as a 2-dimensional table (Fig. 5). The input is represented as a two digit hexadecimal number where the high order digit is read by giving the row number and the low order digit is read by giving the column number. For example, for an input value 86 the S-box output is FB.

S-box is said to exhibit an affine equivalence is any of its component functions can be mapped to another one using only affine transformations. This is an undesirable property as it theoretically could help to form an algebraic attack on a cipher. Removal of this affine equivalence has been achieved by switching one pair of elements of the S-box table between them. This procedure lowered S-box nonlinearity to 110 (from the initial 112). We consider this a minimal loss of nonlinearity, which in turn allowed the S-box to have that important characteristics missing from AES S-box, that is - no affine equivalence between any component functions.

The S-box of PP-2 cipher is the result of the extensive search of S-boxes generated with multiplicative procedure with randomly selected polynomials. In each of generated S-boxes, a pair of input was randomly searched, switching which would eliminate the affine equivalence present in the S-box. This random search continued until an S-box has been found fulfilling all the required parameters mentioned above.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	9E	81	F7	2F	CC	F1	46	6E	B7	CE	29	76	14	42	E6	BB
1	EC	39	B6	EF	A3	D5	EA	91	3D	37	F0	51	44	C6	0C	BC
2	41	80	AB	1D	6C	D2	C0	ED	00	FE	3B	F9	A4	24	FF	DB
3	4F	7A	4A	B8	A9	4E	79	A8	15	9B	B2	A7	31	52	69	58
4	71	DC	77	99	04	A6	B9	25	E7	92	5E	62	57	89	C1	61
5	D1	66	48	C2	AA	38	2D	8E	65	8C	C3	6A	C8	7B	DA	95
6	90	0E	0A	6B	F4	5B	8D	A1	05	0B	10	03	8B	9D	85	E1
7	BD	19	FA	6D	88	22	E4	4C	AF	49	F8	BE	83	07	FD	D3
8	8F	A5	BF	E8	C4	E5	FB	16	35	3C	64	2C	0D	C5	43	02
9	59	C7	7E	E3	18	CF	06	4B	9C	D0	F3	70	D7	33	87	B1
A	DF	20	E2	EE	F5	32	56	B3	84	74	2B	34	47	36	96	DD
B	63	0F	97	28	D6	5F	7D	9F	53	09	8A	12	5A	AE	1B	3A
C	7F	40	30	A0	D4	27	82	3E	4D	08	7C	1C	17	2E	01	CD
D	B5	CB	54	A2	D9	EB	50	93	F2	3F	1F	9A	13	CA	21	94
E	E9	23	5D	1A	AC	B0	67	86	73	1E	26	6F	45	98	11	BA
F	E0	D8	75	72	AD	55	68	78	F6	2A	B4	5C	C9	60	DE	FC

Figure 5. S-box *S*

PP-2 cipher's S-box is not its own inverse ($S^{-1} \neq S$). To avoid attacks based on algebraic properties of an S-box, after generating the table of inverses, an affine transformation is applied to each entry of the S-box, which introduces diffusion.

This transformation in case of PP-2 cipher is described by the following matrix transformation:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (13)$$

where $[x_7,x_6...,x_0]$ is an entry in the inverse table, $[y_7,y_6...,y_0]$ is S-box output value, and constant c = $[c_7,c_6...,c_0]$ = [10011110] = $9E.

In PP-2 cipher, the constant *c* of the affine transformation has been selected in such a way that the S-box does not have any fixed points (more precisely, the value of *c* has been selected by random search until a suitable value was found that resulted in the absence of fixed points in the S-box). S-box *S* does not have any fixed points when $S(a) \oplus a \neq 00$ and $S(a) \oplus a \neq$ FF for every value of *a*.

IV. KEY SCHEDULE

A. Round key generation

A key schedule of the cipher is an algorithm that, given the key *k*, calculates the subkeys K_i for each of its *r* rounds, and consequently the round key key_i. The length of the block is a multiple of 64, i.e. $n = t \cdot 64$, and the length of the key *k* equals $d \cdot n$, where $d, t \in \{1, 2, 3, ...\}$.

The scheme of one iteration consists of *P* - the same permutation as in the processing path of the cipher (Fig. 1) and *S* - the same S-box as used in nonlinear element *NL* (Fig. 6).

In the *i*-th round of key schedule, the auxiliary key K_i is generated. As the result of each iteration sch_i, we obtain key_i.

Let

$c_0 = RR(0, (E3729424EDBC5389)) \| RR(1, (E3729424EDBC5389)) \| RR(t–1, (E3729424EDBC5389))$, (14)

$c_1 = RR(0, (59F0E217D8AC6B43)) \| RR(1, (59F0E217D8AC6B43)) \| RR(t–1, (59F0E217D8AC6B43))$, (15)

where $RR(b, x)$ is the *b*-bits right rotation of a binary word *x* and || denotes the concatenation.

Furthermore, let for $t = n/64$

$$K_i = K_{i,1} \| K_{i,2} \| \ldots \| K_{i,j} \| \ldots \| K_{i,t} \quad (16)$$

(similarly for $K_{i,j}$ in each block *KS#j*),

$$X_{i-1} = X_{i-1,1} \| X_{i-1,2} \| \ldots \| X_{i-1,j} \| \ldots \| X_{i-1,t} \quad (17)$$

(similarly for $X_{i-1,j}$ in each block *KS#j*),

$$V_{i-1} = V_{i-1,1} \| V_{i-1,2} \| \ldots \| V_{i-1,j} \| \ldots \| V_{i-1,t} \quad (18)$$

(similarly for $V_{i-1,j}$ in each block *KS#j*), where

$$V_{i-1,j} = KS\#j(K_{i,j}, X_{i-1,j}) \text{ for } j = 1,2,\ldots,t, \quad (19)$$

$$Z_{i-1}=P(V_{i-1}), \quad (20)$$

$$key_i=X_{i-1} \oplus Z_{i-1}. \quad (21)$$

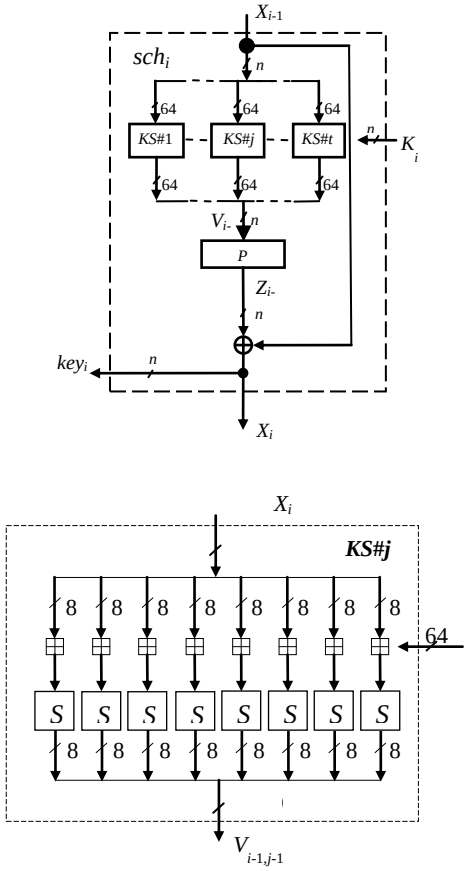

Figure 6. One iteration of the key schedule

B. *Creating auxiliary keys*

For constants c_0 and c_1, and $i = 1, 2, \ldots, \lceil d \rceil \cdot t + r$ we calculate

$$K_i = K_i^* \oplus RR(i-1, c_1),$$

where

$$\left(K_i^*\right)_{i=1}^{\lceil d \rceil t+r} = \left(\kappa_1, \underbrace{0,0,\ldots,0}_{t}, \kappa_2, \underbrace{0,0,\ldots,0}_{t}, \ldots \kappa_{\lceil d \rceil}, \underbrace{0,0,\ldots,0}_{t}, \underbrace{0,0,\ldots,0}_{r-\lceil d \rceil}\right).$$

The round key

$$k_i = \begin{cases} key_{i(t+1)}, & \text{dla } i = 1,2,\ldots,\lceil d \rceil \\ key_{\lceil d \rceil (t+1)+i}, & \text{dla } i = \lceil d \rceil + 1, \lceil d \rceil + 2, \ldots, r \end{cases}$$

The scheme of the generation of round keys for n = 128 and $|k|$ = 256 is presented in Fig. 7.

V. RESISTANCE AGAINST CRYPTANALYSIS

To evaluate the resistance of the PP-2 cipher against the differential and the linear cryptanalysis let us apply the rough method, described in [5] and [7]. The main idea of the rough method is to evaluate the best nonzero approximation of a cipher by a composition of the best nonzero approximation of a single iteration. In the case of the PP-2 cipher, we assume a single active S-box in each round

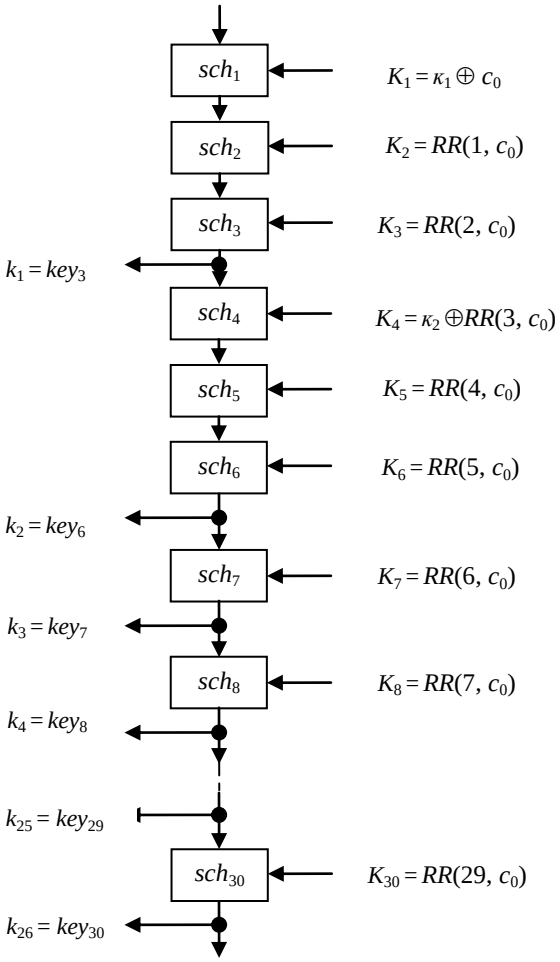

Figure 7. The scheme of the generation of round keys for n = 128 and $|k|$ = 256

The best nonzero linear approximation of the S-box S of PP-2 has the effectiveness $|\Delta p_S^+|$ = 18/256. The effectiveness of the best nonzero differential approximation of the S-box is π_S^+ = 4/256. Assume that for the round function h the same values for effectiveness of the best linear and differential approximation are obtained, i.e. $|\Delta p_h^+|$ = 18/256 and π_h^+ = 4/256. Assume moreover that the best nonzero approximations of PP-2 are composed of r best nonzero approximations of function h. Then the values of effectiveness $|\Delta p_a^+|$ and π_a^+ obtained for the best nonzero approximations of PP-2 are presented in Table II.

TABLE II. UPPER BOUNDS OF EFFECTIVENESS OF THE PP-2 NONZERO LINEAR AND DIFFERENTIAL APPROXIMATIONS

(n, r)	(64, 11)	(128, 22)	(192, 32)	(256, 43)
$\lvert\Delta p_a^+\rvert$	$1.83/2^{33}$	$1.67/2^{64}$	$1.35/2^{92}$	$1.24/2^{123}$
π_a^+	$1/2^{66}$	$1/2^{132}$	$1/2^{192}$	$1/2^{258}$

The best nonzero linear approximation of PP-2 is evidently more effective than the differential one. It does not mean, however, that the linear attack is less complex than the differential one. In the two attacks the number of required texts is of order of $2n$, where n is the block size.

VI. TESTS RESULTS

In this section, we present tentative results of PP-2 performance evaluation. We have measured the encryption speed for three algorithms: PP-1, PP-2, and AES. The reference (unoptimized) implementations of PP-1 and PP-2 were used for this purpose. Therefore, we have chosen partially optimized implementation of AES based on the implementation developed by Brad Conte [9] for comparison. (We plan, in the future, to compare optimized version of PP-2 with best optimized implementations of AES.)

Exemplary implementations use 128-bit key. The data block size is 64-bit for PP-1 and PP-2. All implementations were written in C and compiled using Intel C++ Compiler Professional 11.1 for Windows.

For each algorithm, two versions were generated: 32-bit code and 64-bit code. Experiments were performed using PC computer with AMD Phenom II X4 965 3.4 GHz processor, 8 GB of RAM, Windows 7 operating system (64-bit version).

Tables III and IV present processing speed for data block size 32, 64, 512, and 1024 bytes for 32-bit code and 64-bit code.

TABLE III. THE ENCRYPTION SPEED [MB/S] 32-BIT CODE

	Processed data blocks [bytes]			
	32	64	512	1024
PP-1	3.03	4.08	6.08	6.31
PP-2	12.69	14.23	16.21	16.36
AES	17.80	18.57	19.08	19.29

TABLE IV. THE ENCRYPTION SPEED [MB/S] 64-BIT CODE

	Processed data block [bytes]			
	32	64	512	1024
PP-1	3.38	4.50	6.43	6.60
PP-2	19.35	21.47	25.06	25.16
AES	21.26	22.5	23.31	23.54

As we can see, the PP-2 performs much better than PP-1. For big data blocks encrypted with the same key, the PP-2 outperforms PP-1 by more than 2. For short data blocks, the difference is even greater.

TABLE V. ROUND KEYS GENERATION TIME 32-BIT CODE

	Time [μs]	Processor cycles
PP-1	2.49	8467
PP-2	0.55	1866
AES	0.14	492

There are no big differences in performance between the PP-2 and the AES. For long blocks of data encrypted with the same key AES performs better in 32-bit version, whereas PP-2 slightly outperforms AES in 64-bit version. For short blocks, AES is better in all cases. It means that the round keys generations takes more processor cycles in PP-2 than in AES.

Tables V and VI present measured round keys generation time for PP-1, PP-2, and AES for 32-bit and 64-bit code respectively

TABLE VI. ROUND KEYS GENERATION TIME 64-BIT CODE

	Time [μs]	Processor cycles
PP-1	1.56	5313
PP-2	0.36	1227
AES	0.15	507

As we can see in tables IV and V, the round keys generation requires more time in PP-2 than in AES. On the other hand, from tables III, IV, V and VI, we can deduce that generation of the round keys requires the same amount of time as encryption of about 9.5 bytes of data. It means that performance of PP-2 round keys generation algorithm is fairly good.

VII. CONCLUSION

PP-2 cipher construction details were described in this paper. Cipher structure is different for encryption and for decryption. In particular, a round function construction was presented as well as diffusion layer based on multiple rotations. It was shown, that for block length $n = 64$ bits a global diffusion is reached after two rounds. Number of rounds is dependent on key size.

New S-boxes were designed, S and S^{-1} with very good cryptographic characteristics.

Round key generation algorithm for PP-2 cipher has been considerably simplified in comparison to that of PP-1 cipher. This allowed lowering cipher initialization time. Round keys remained independent, which improves cipher quality when it comes to differential cryptanalysis (such cryptanalysis is more difficult).

ACKNOWLEDGMENT

The paper is a scientific work financed from science research fund in years 2010-2013 as a research project and partially by the grant DS-PB/45-085/12.

REFERENCES

[1] Advanced Encryption Standard (AES), NIST, FIPS Pub. 197, November 26, 2001.

[2] E. Biham and A. Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer-Verlag, New York, 1993.

[3] K. Bucholc at al., Scalable PP-1 block cipher, International Journal of Applied Mathematics and Computer Science, vol. 20, no. 2, 2010, pp. 401−411.

[4] K. Bucholc, K. Chmiel, A. Grocholewska-Czuryło and J. Stokłosa, PP-1 Block Cipher, Polish Journal of Environmental Studies, vol. 16, no. 5B, 2007, pp. 315−320.

[5] Chmiel K., Methods for differnetial and linear cryptanalysis of block ciphers (in Polish), PUT Press (Wydawnictwo Politechniki Poznańskiej), 1–212, Poznań 2010.

 ISBN: 978-1-61208-298-1

[6] K. Chmiel, A. Grocholewska-Czuryło, P. Socha and J. Stokłosa, Involutional scalable block cipher, Metody Informatyki Stosowanej, nr 3/2008 (tom 16), 2008, pp. 65−75.

[7] K. Chmiel, A. Grocholewska-Czuryło and J. Stokłosa, Evaluation of PP-1 Cipher Resistance against Differential and Linear Cryptanalysis in Comparison to a DES-like Cipher, Fundamenta Informaticae, vol. 114(3–4), 2012, pp. 239–269.

[8] K. Chmiel, A. Grocholewska-Czuryło and J. Stokłosa, Involutional block cipher for limited resources, 2008 IEEE Global Telecommunications Conference, Computer and Communications Network Security Symposium, IEEE eXpress Conference Publishing, ISBN 978-1-4244-2324-8 (CD ROM), New Orleans 2008, pp. 1852−1856.

[9] B. Conte, Implementation of AES in C, http://bradconte.com/aes_c, [retrieved: July, 2013]

[10] M. Misztal, Differential Cryptanalysis of PP-1 Cipher, Proceedings of International Cryptology Conference – Recent Advances in Cryptology and National Telecommunication Security Systems, Wojskowa Akademia Techniczna, Warszawa 2011, also in Annales UMCS Informatica AI XI, 2, 2011, pp. 9–24.

 ISBN: 978-1-61208-298-1