CTCRYPT 2023
The 12th Workshop on Current Trends in Cryptology CTCrypt 2023

The PRF pCollapserARX optimal cryptographic characteristic automated search by CASCADA

S.V. Polikarpov
Candidate of Sciences, Associate Professor, Institute of Computer Technologies and Information Security, Southern Federal University

K.E. Rumyantsev
Doctor of Sciences, Head of the department, Institute of Computer Technologies and Information Security, Southern Federal University

V.A. Prudnikov
Postgraduate student, Assistant Professor, Institute of Computer Technologies and Information Security, Southern Federal University

Institute of Computer Technologies and Information Security,
Southern Federal University

Introduction

The structure of the PRF "Collapser" (a black hole) was first presented at CTCrypt'2015.

A high-performance PRF pCollapserARX256-32x2 was presented at the RusCrypto'2022 conference – software-oriented PRF based on PD-sbox functions, where each PD-sbox consist from 4 sboxARX functions (nonbijective functions that using only modulo Addition, Rotate and Xor operations).

The main idea behind the PRF pCollapserARX256-32x2 is the use of *PD-sbox* (pseudo-dynamic substitution box) as a non-linear element a special function that allows to radically change the properties of a group of nested functions.

Initially, it was supposed to use fixed bijective sboxes as nested functions. But the research conducted by the authors showed that the use of ARX-functions with initially weak cryptographic properties is well suited as nested functions.

At RusCrypto'2022, the properties of individual PD-sboxes were presented, but a question arose about the stability of the entire structure as a whole.

We presents the first results of the search for optimal cryptographic characteristics for pCollapserARX32-4x2, pCollapserARX64-8x2 and pCollapserARX128-16x2 PRF belonging to the same family with the pCollapserARX256-32x2 PRF.

The obtained results confirm the correctness of the PRF structure "pCollapserARX" and the ideas embedded in it.

In addition, source codes for the automatic construction of SAT models and the search for cryptographic characteristics using CASCADA are proposed.

Short description of the PRF "pCollapserARX"

The pCollapserARX32-4x2, pCollapserARX64-8x2 and pCollapserARX128-16x2 PRFs under consideration are built on an identical principle:

- Using 4 rounds of transformation.
- Each round uses 16 ARX-functions in parallel.
- All ARX-functions are grouped into four PD-sboxes.
- To ensure the dynamic mode of PD-sboxes operation, input/output control state values are used and formed.
- The master key initializes key for first round, round keys are not used in rounds 3 and 4 (similar to stream ciphers).
- The first round is preparatory, in it *PD-sboxes* work in static mode, but forms control values for the next round.

During the researching of cryptographic characteristics, the following changes were made to the pCollapserARX structure:

1. Internal control state size increased to $L_{control\_state} = N_{rows} \times N_{words} \times L_{word}$, bit.

2. Changed the formation/updating of the control state.

3. Added "extended key" generation. The "extended key" includes the round key 1 and additional control state for the second round. There are no round keys for rounds 3-4, instead an updatable control state is used.

Table 1: Parameters of the PRF pCollapserARX family

pCollapserARX:				
params	**32-4x2**	**64-8x2**	**128-16x2**	**256-32x2**
L_{block} , bit	32	64	128	256
L_{word} , bit	8	16	32	64
N_{rounds}	4	4	4	4
N_{words}	4	4	4	4
N_{rows}	4	4	4	4
L_{key} , bit	32	64	128	256
$L_{control\_state}$, bit	128	256	512	1024

Table 2: Parameters of the ARX-functions for the PRF pCollapserARX128-16x2

	t0	t1	t2	t3	t4	t5	t6	t7
funcARX0:	4	8	8	4	4	8	0	0
funcARX1:	4	8	4	8	8	4	4	4
funcARX2:	8	4	4	8	4	8	8	8
funcARX3:	8	4	8	4	8	4	12	12

Figure 1: Structure of the used ARX-functions

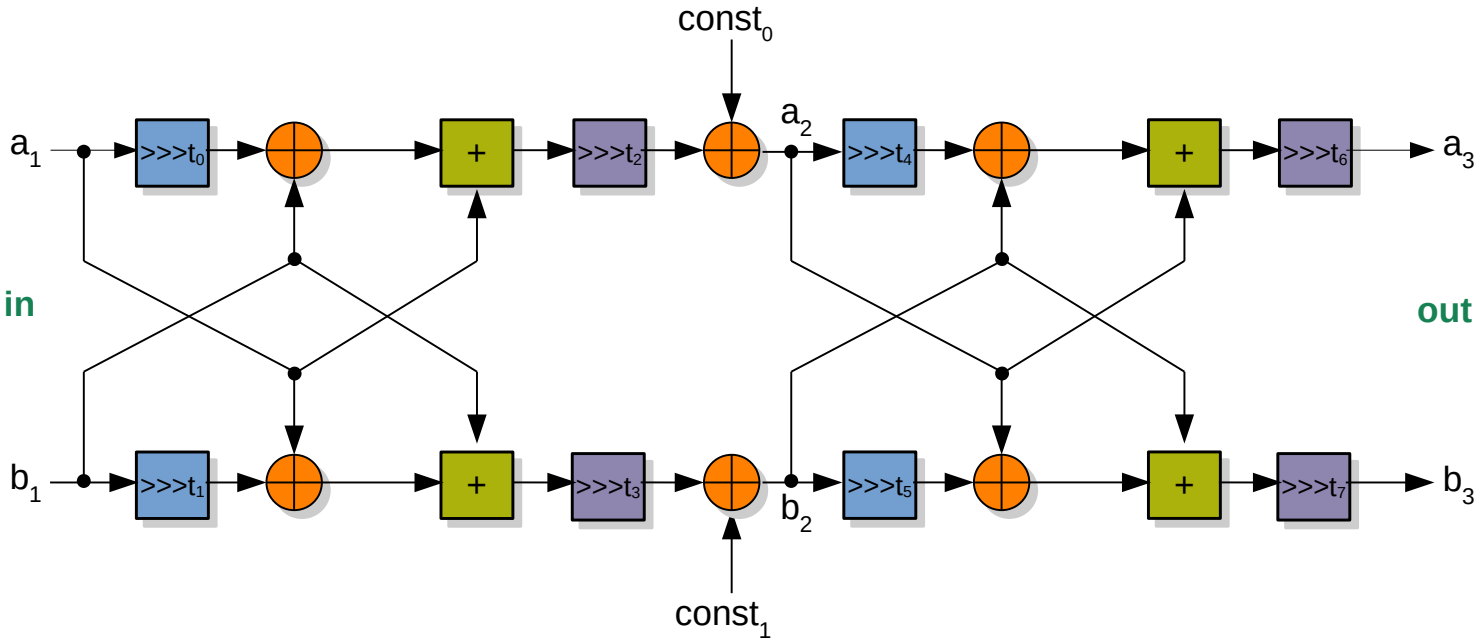
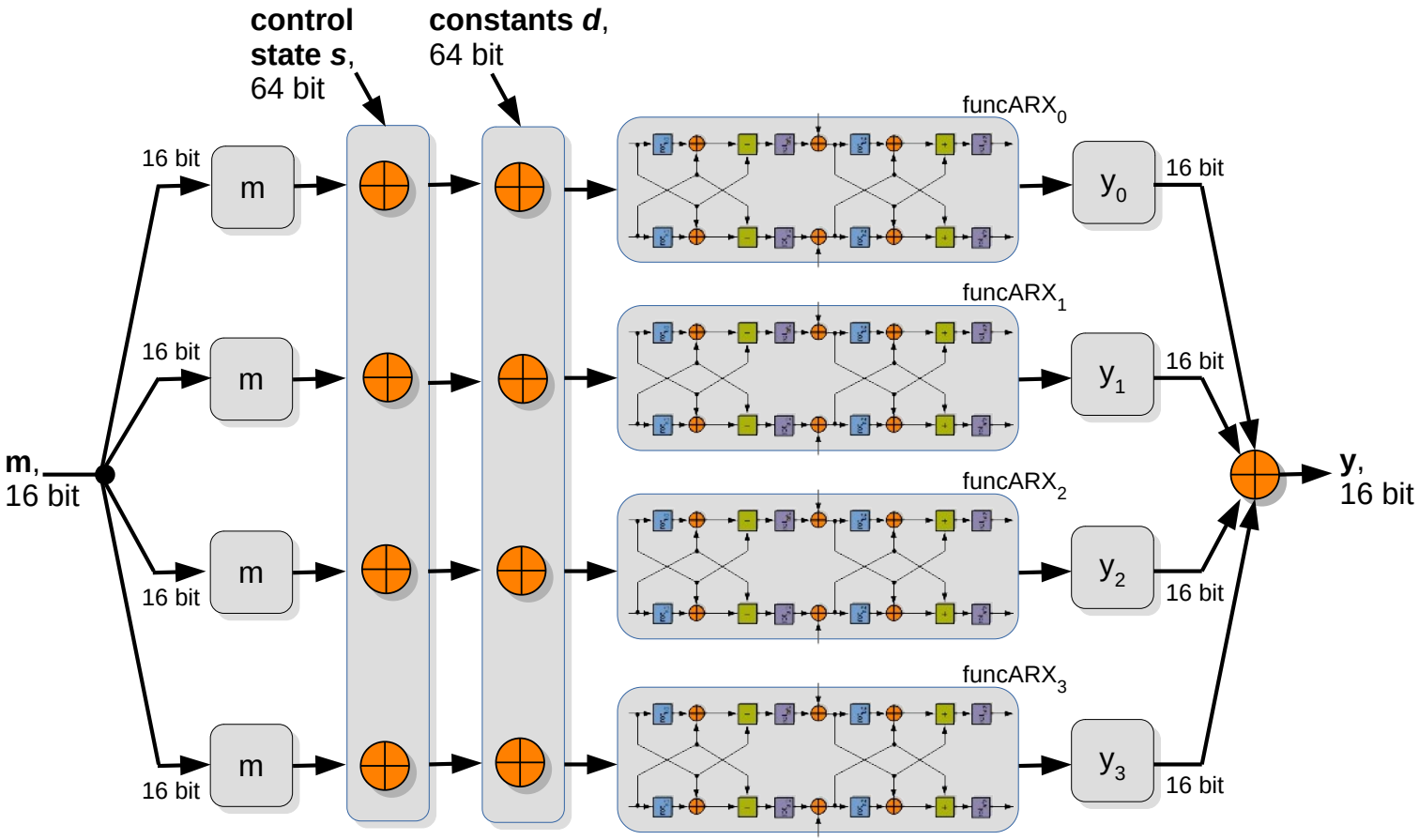


Figure 2: The Pseudo-dynamic substitution box function (pCollapserARX64-8x2 case)



It was shown in www.ruscrypto.ru/resource/archive/rc2022/files/02_polikarpov_rumyantsev_prudnikov.pdf that combining 4 weak ARX functions in PD-sbox allows one to obtain properties of equivalent sboxes close to those of randomly generated sboxes of the same dimension.

Pseudo-dynamic substitution boxes function

Expression for base PD-sbox out:

$$c_i = \bigoplus_{j=0}^{3} funcARX_j(m_i \oplus s_j^i)$$

where: *i* — index for n-bit word from input/output vector and, thereafter, index of PD-sbox; *j* — index of PD-sbox component; m_i — n-bit words from input vector; c_i — n-bit words from output vector; funcARX — ARX-function (components of PD-sbox); s^i_j — n-bit words from control state input vector (individual for each PD-sbox).

Expressions for PD-sbox local (individual) control states output:

$$g_n^i = c_i \oplus funcARX_j(m_i \oplus s_j^i) = \bigoplus_{n=0,\, n\neq i}^{3} funcARX_j(m_i \oplus s_j^i)$$

Figure 3: Structure of one round PRF pCollapserARX

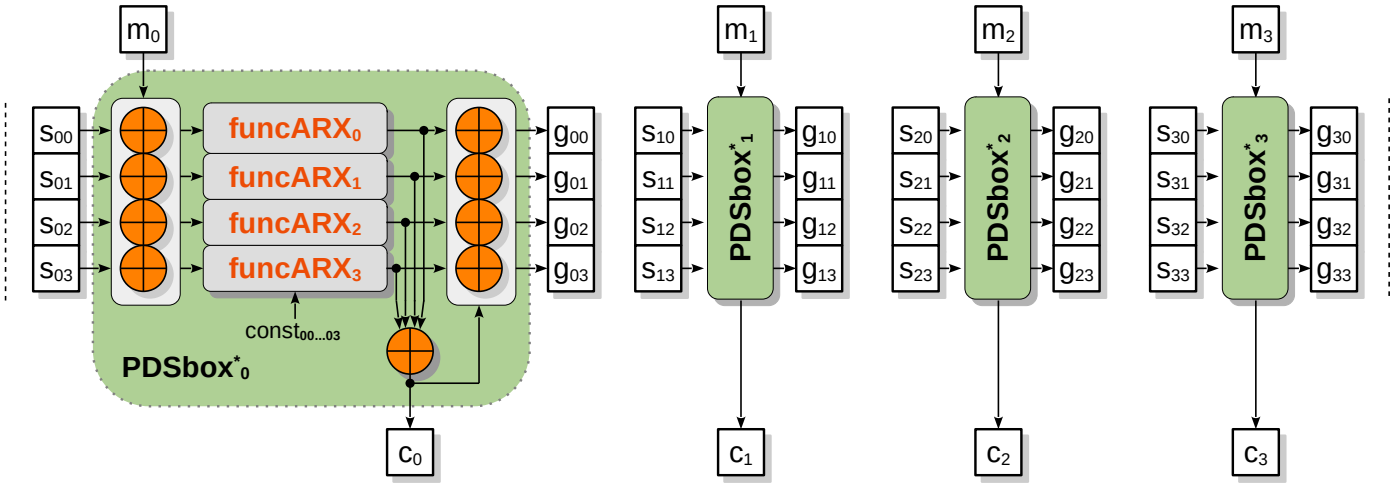


Figure 4: Formation of the output control state. Step 1

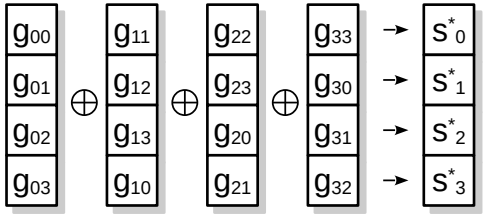


Figure 5: Formation of the output control state. Step 2

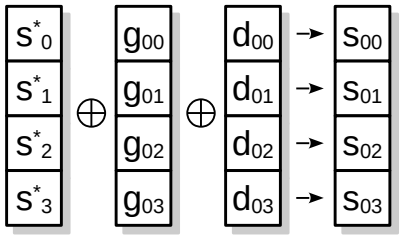
Used params:

$N_{words} = 4$ — number of input words in the block;
$N_{rows} = 4$ — number of ARX-functions in one PD-sbox;
$L_{block} = L_{word} \times 4$, bit - block size in bits;
$L_{key} = (L_{block})$; $(2 \times L_{block})$ *or* $(4 \times L_{block})$, bit — master-key size in bits.

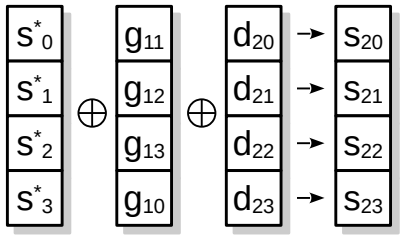
Used vectors:

$m = \{m_0, m_1, m_2, m_3\}$ — vector of the input message;
$c = \{c_0, c_1, c_2, c_3\}$ — cipher-text vector;
$s = \{s_{00}, s_{01}, \ldots, s_{33}\}$ — control state vector;
$d = \{d_{00}, d_{01}, \ldots, d_{33}\}$ — round constants.

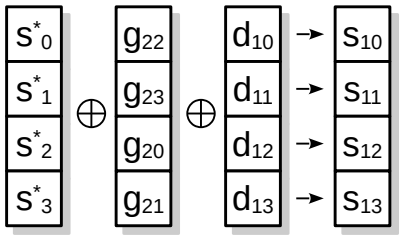
for PDSbox$_0$:



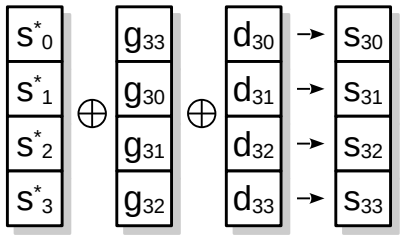
for PDSbox$_2$:



for PDSbox$_1$:



for PDSbox$_3$:



Expression for function, that create new control state output vector (Figures 4 and 5):

Step 1:

$$g^i = (g_0^i, g_1^i, g_2^i, g_3^i) = g^i <<< (i \cdot Lword)$$

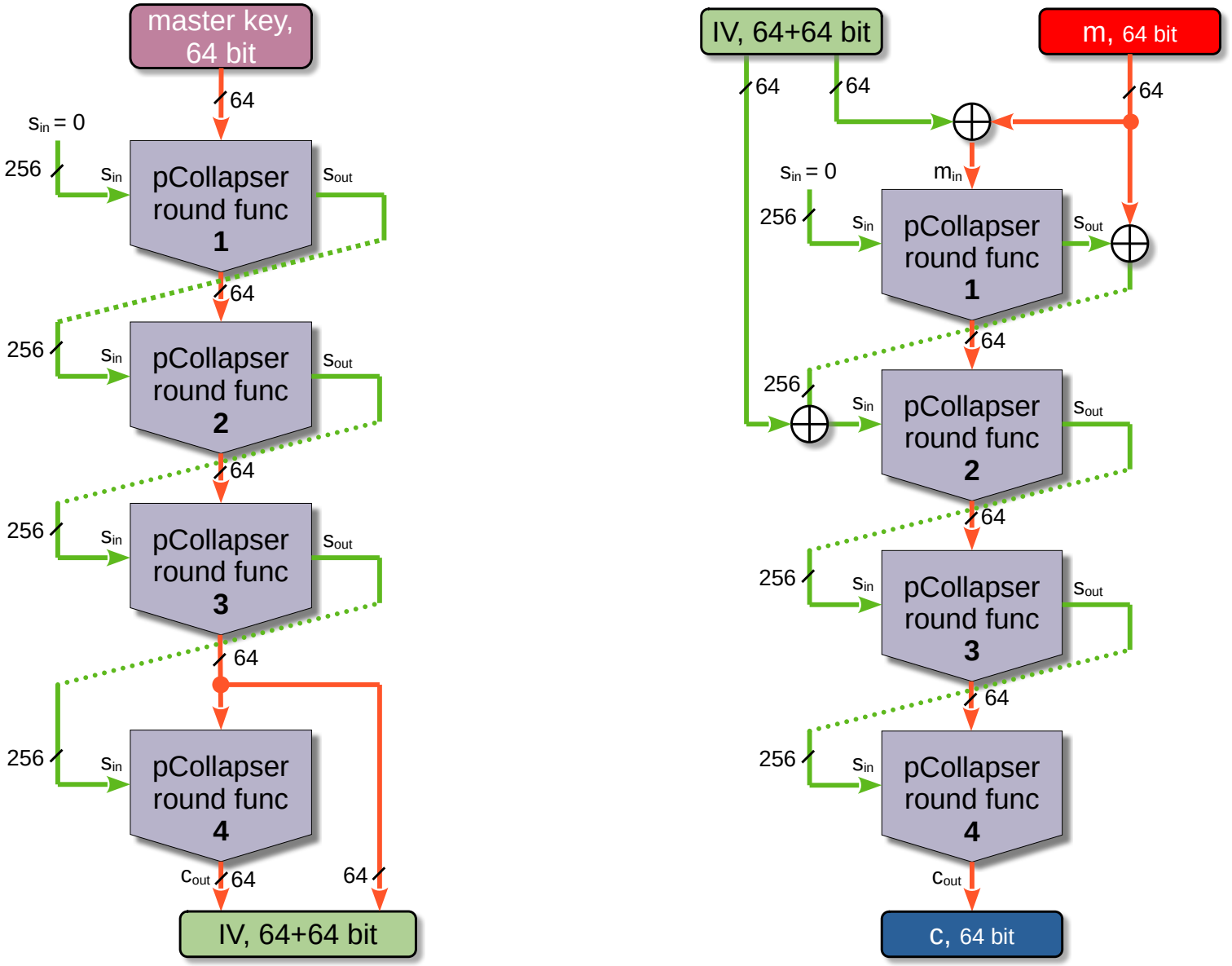
$$s^* = (s_0^*, s_1^*, s_2^*, s_3^*) = \bigoplus_{i=0}^{3} g^i$$

Step 2:

$$s_j^i = s_j^* \oplus d_j^i \oplus g_j^i$$

where: $g^i = (g_0^i, g_1^i, g_2^i, g_3^i)$ – out control state values from each i-th PDsbox; $a <<< b$ – cyclic shift bits in a vector a by b elements in a left direction; d_j^i – n-bit words from constant/decollision vector (individual for each PDsbox)

Figure 6: "Expanded-key" generation (left) and main rounds (right) for the "pCollapser-ARX64"



Characteristics search tool

Ranea A. and Rijmen V. in [1] proposed a powerful tool called CASCADA (Characteristic Automated Search of Cryptographic Algorithms for Distinguishing Attacks) is an open-source Python 3 library to evaluate the security of cryptographic primitives, specially block ciphers, against distinguishing attacks with bit-vector SMT solvers.

To search for characteristics using CASCADA, we created the file "pCollapserARX_full.py" containing the implementation of the PRF pCollapserARX and the file "test_collapser.py" that allows you to start searching for various cryptographic characteristics for the PRF pCollapserARX (available at https://github.com/pruvad/CTCrypt2023).

The result of the CASCADA work is the weights of the found optimal characteristics:

$$w = -log_2 P(.)$$

where *P* (.) is the probability of occurrence of the input/output difference for the tested function (for differential analysis) or the correlation value (for linear analysis).

The following computer and software was used to search for characteristics: AMD Ryzen 5 2600, 32GB RAM, OS Ubuntu 22.04, Python version 3.10, CASCADA version - snapshot from **github.com/ranea/CASCADA** downloaded 24 febrary 2023.

Results of the search for optimal characteristics

Table 3: Weights of found characteristics for pCollapserARX32

$w = -log_2 P$ (.)				
Nrounds	**Correlation**	**XorDiff**	**Related-Key XorDiff**	**RXDiff**
1	8	16	16	1276
2	26	32	64	$2510 < w$
3	37	32	64	$3060 < w$
4	50	32	64	–
bounds	16	32	32	32

For cases when the search for characteristics was not performed, the symbol " – " is put in the table.

The last line of each table shows the applicability limit for the corresponding cryptanalysis method, taking into account the dimensions of the inputs/outputs.

Table 4: Weights of found characteristics for pCollapserARX64

$w = -log_2 P$ (.)			
Nrounds	**Correlation**	**XorDiff**	**RXDiff**
1	11	19	$1508 < w < 1603$
2	42	88	–
3	$55 < w < 88$	88	–
4	$61 < w$	88	–
bounds	32	64	64

Table 5: Weights of found characteristics for pCollapserARX128

$w = -log_2 P$ (.)			
Nrounds	**Correlation**	**XorDiff**	**RXDiff**
1	11	19	$1136 < w$
2	$49 < w < 63$	$110 < w < 180$	–
3	–	–	–
bounds	64	128	128

Found differential characteristics with conventional approach (pCollapserARX32):

1 : Ch(*w* = 16, id=fa 00 00 00, od=8c 00 00 00)

2 : Ch(*w* = 32, id=00 00 00 fa, od=00 00 00 48)

3 : Ch(*w* = 32, id=00 00 00 af, od=00 00 00 00)

4 : Ch(*w* = 32, id=00 00 00 af, od=00 00 00 00)

However, an experimental study of pCollapserARX miniversions showed that in reality the probability of a collision at the output of the second and subsequent rounds corresponds to the probability of a collision at the output of a random function (of the same dimension).

Due to the presence of an actively updated internal state, which significantly exceeds the size of the input / output, the presence of a collision at the output of the second round will not lead to collide output values in subsequent rounds.

The discrepancy between the found characteristics and the real ones is explained by the fact that the differential model for the SAT solver is built taking into account the hypothesis of stochastic equivalence (also known as the Markov-cipher assumption) [2]. This allows to split and analyze iterative functions in parts, thereby dramatically reducing the complexity of finding characteristics.

Research results showed, that the hypothesis of stochastic equivalence does not hold for the pCollapserARX (the intermediate propagations of properties within the characteristic are not independent – due to the nature of the dynamic operation of *PD-sboxes*) and the absence of round keys (updatable internal control state is used).

The problem also affects the search for characteristics for related-key, impossible-differential and related-key impossible-differential.

For this case we can use experimental script, created by Ranea and based on ideas from [3, 4].

Note that this script is experimental, and it has not been fully tested. Script available at (github.com/ranea/CASCADA/blob/master/cascada/experimental/diffvalchsearch.py)

Table 6: Weights of found characteristics by *diffvalchsearch.py* script

$w = -log_2 P$ (.)				
Nrounds	**pCollapserARX16**	**pCollapserARX32**	**pCollapserARX64**	**pCollapserARX128**
1	13	18	21	19
2	48	$73 < w$	$100 < w$	$100 < w$
3	75	–	–	–
bounds	16	32	64	128

Found differential characteristics with conventional approach (for mini-version pCollapserARX16):

1 : Ch(*w* = 11, id=0 0 f 0, od=0 0 6 0)

2 : Ch(*w* = 23, id=f 0 0 0, od=7 0 0 0)

3 : Ch(*w* = 24, id=0 0 0 f, od=0 0 0 0)

4 : Ch(*w* = 24, id=0 0 0 f, od=0 0 0 0)

Found valid differential characteristics (for pCollapserARX16):

1 : Ch(*w* = 13, id=0 d 0 0, od=0 8 0 0)

2 : Ch(*w* = 48, id=0 0 8 0, od=3 8 a f)

3 : Ch(*w* = 75, id=0 0 8 0, od=f 0 b 0)

As you can see by the example of the mini version of pCollapserARX16, the obtained valid characteristics do not collide (the output difference is non zero) and the weight value increases with each round. Last results showed, that the hypothesis of stochastic equivalence does not hold for the pCollapserARX.

Conclusions:

For the first time, a publicly available description of the PRF pCollapserARX for the CASCADA framework has been proposed, which allows to generate SAT-models and search for optimal cryptographic characteristics.

The results of the search for optimal cryptographic characteristics using CASCADA showed the resistance of the first 2 rounds (out of 4) of the PRF pCollapserARX to building distinguishers for attacks such as linear, differential, Related-Key differential, RX-differential and Related-Key RX-differential cryptanalysis.

Considering that the 1st round does not work in dynamic mode, we believe that the main contribution to resistance comes from the 2nd round, which works in dynamic mode. This confirms the correctness of both the use of pseudo-dynamic substitutions as a nonlinear element and the use of weak ARX-functions in their composition.

The presence of additional 3 and 4 rounds gives a significant security margin.

It is shown that, unlike most known symmetric cryptoalgorithms, the usual approach based on the Markov-cipher assumption is not suitable for searching for valid differential characteristics of PRF pCollapserARX, and a special model is required to search them, while the real weights of optimal characteristics are much higher.

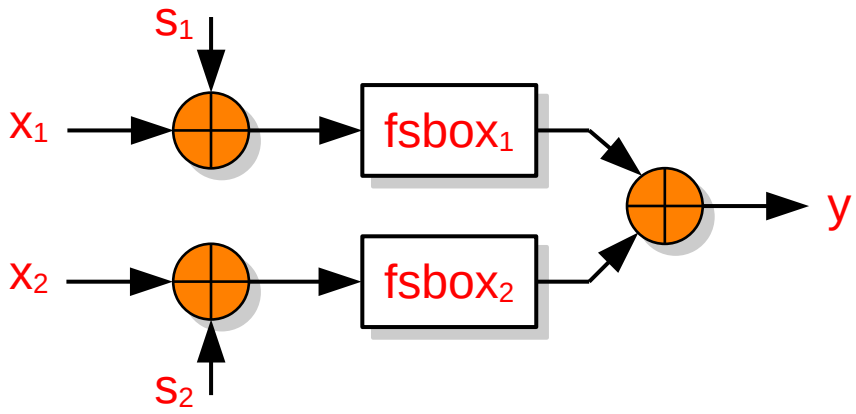
V.A. Prudnikov
Postgraduate student, Assistant Professor, Institute of Computer Technologies and Information Security, Southern Federal University
prudnikov@sfedu.ru

Appendix A. References

[1] Ranea, A. Rijmen, V. Characteristic Automated Search of Cryptographic Algorithms for Distinguishing Attacks (CASCADA). (Cryptology ePrint Archive, Paper 2022/513),2022 https://eprint.iacr.org/2022/513.
[2] Zheng, X. Yongqiang, L. Lin, J. Mingsheng, W. Willi, M. Do NOT Misuse the Markov Cipher Assumption Automatic Search for Differential and Impossible Differential Characteristics in ARX Ciphers,2022 https://eprint.iacr.org/2022/135.pdf.
[3] Sadegh S. Rijmen V. Bagheri N. Proposing an MILP-based Method for the Experimental Verification of Difference Trails. (Cryptology ePrint Archive, Paper 2020/632), 2020 https://eprint.iacr.org/2020/632.
[4] Ananth P. Rai A. Jain A. A New Approach to Round-Optimal Secure Multiparty Computation. (Cryptology ePrint Archive, Paper 2017/402), 2017 https://eprint.iacr.org/2017/402.

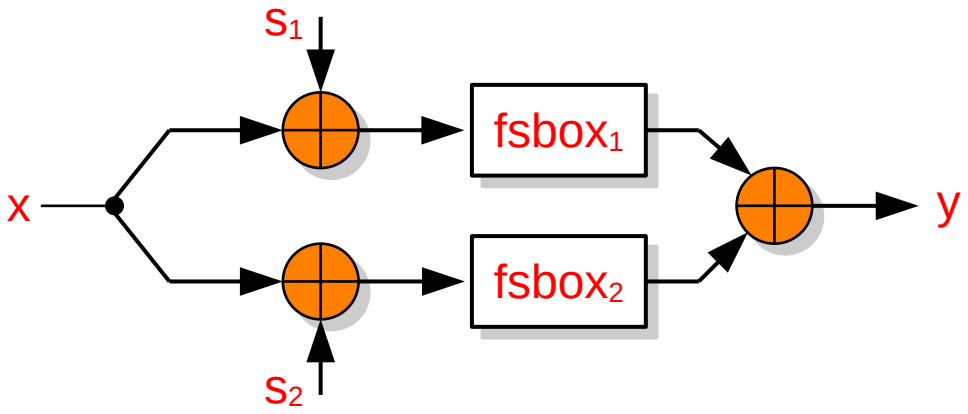
Appendix B. Examples typical application of fixed substitutions and pseudo-dynamic substitution

An example of a typical application of fixed substitutions:



where:
x_1, x_2 – 6-bit inputs;
y – 6-bit output;
s_1, s_2 – 6-state bit values.

PDSbox_2x6x6 pseudo-dynamic substitution example:



where:
x – 6-bit inputs;
y – 6-bit output;
s1, s2 – 6-state bit values.

Appendix C. Differential properties of the presented structures:

Typical inclusion of substitutions:

For the input difference Δx = (Δx1 | Δx2), Δx1 = 7, Δx2 = 8:

1) state values s1 = 1, s2 = 1:
Δx/Δy: [228, 88, 156, 12, 76, 16, 72, 40, 32, 52, 24, 24, 88, 60, 92, 48, 128 ...]

2) state values s1 = 1, s2 = 2:
Δx/Δy: [228, 88, 156, 12, 76, 16, 72, 40, 32, 52, 24, 24, 88, 60, 92, 48, 128 ...]

PDSbox_2x6x6:

For the input difference Δx = 7

1) state values s1 = 1, s2 = 1:
Δx/Δy: [2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 4, 0, 2, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 2, 0 ...]

2) state values s1 = 1, s2 = 2:
Δx/Δy: [0, 0, 2, 0, 4, 2, 2, 4, 0, 0, 0, 4, 0, 8, 2, 2, 0, 0, 0, 2, 0, 0, 2, 0, 2, 0, 2, 0, ...]