DE GRUYTER
OPEN

TATRA
MOUNTAINS
Mathematical Publications

DOI: 10.1515/tmmp-2017-0023
Tatra Mt. Math. Publ. **70** (2017), 139–149

IMPROVEMENT ON BIT DIFFUSION ANALYSIS OF π-CIPHER

Fatih Sulak — Beyza Bozdemir — Betül A. Özdemir –
– Neşe Koçak — Onur Koçak

ABSTRACT. π-Cipher, designed by Gligoroski et al., is a second round candidate of the CAESAR competition. The designers analyzed the bit diffusion of the cipher by examining the $*$ operation and 1 round π-function. We improve this analysis by applying Strict Avalanche Criterion (SAC) test to $*$ operation and reduced round versions of π-function for π 16-Cipher. We found out that $*$ operation fails SAC test whereas all versions of π-function for π 16-Cipher pass the test.

1. Introduction

The authenticated encryption is a cryptographic tool that provides the privacy and authenticity simultaneously. The need for such tool emerged from special purposes. In the recent years, successfully finished competitions like AES [1], eSTREAM [2], SHA-3 [3] have been organized to answer the demands of the industry and interests in the research community. Similarly, the CAESAR competition [4] was initiated in 2014 in order to boost the design for the authenticated encryption tool which provides the privacy and authenticity together. The CAESAR competition is different from the previous competitions AES [1] and SHA-3 [3] done to determine the standard algorithm since the winner is going to be determined by the competition committee consisting of prominent academicians [5], not by the US National Institute of Standards and Technology (NIST) [6]. In addition, the competition allows the designers to tweak their algorithms. These features are similar to the competition eSTREAM [2].

2010 Mathematics Subject Classification: 00A69, 94A60, 94A62, 60K35, 49J55.
Keywords: The CAESAR competition, π-Cipher, bit diffusion analysis, Strict Avalanche Criterion (SSAC) test.
This work is supported by TÜBİTAK (The Scientific and Technological Research Council of Turkey) under project no 114F130.

There are 57 submissions for the first round of the CAESAR competition. However, some have been withdrawn. Moreover, 19 of them have been eliminated from the competition at the end of the first round. Therefore, there are 29 algorithms left for the second round of the CAESAR competition [7]. The third round started in July 2016. The final portfolio will be announced in the near future. [8].

The construction of Cryptographic algorithms is based on the generation of quantities which are not easily predictable to provide the security of the algorithms. Since an adversary should not be able to observe a leakage of the system or even to break the system, the generated quantities must have an adequate length and size, or have randomness property, etc. Although the randomness property is needed to generate a key for asymmetric or symmetric systems, the keys generated from a deterministic source may cause the system to be broken if they show nonrandom properties. In this paper, we apply Strict Avalanche Criterion (SAC) test to π-Cipher algorithm [17] which is one of the algorithms which passed into the second round of the CAESAR competition. SAC test is one of the randomness tests proposed in the recent test package designed by D o ğ a n a k s o y et al. [10]. SAC test determines the number of rounds of an algorithm, for which it behaves like a random mapping, by analyzing the relation between inputs and outputs. The aim of this method is to get a single p-value related with the data set under consideration, through a large set of p-values produced by SAC test.

This work is organized as follows. In Section 2, the details of randomness tests and SAC test are given. In Section 3, we give the brief description of the π-Cipher algorithm. In the Section 4, we present the application of SAC test and the results of the test. In Section 5, we conclude the paper by giving the results and future work.

2. Randomness tests

Random numbers are generated by using random binary sequences in which each element is either 0 or 1 with an equal probability of $\frac{1}{2}$. Although random numbers play an important role in the cryptographic systems, the generation of random numbers is difficult. To generate a true random number, we can use the true random number generators (TRNGs). Although these sources are nondeterministic, the generation with these sources, the storage and transfer of random numbers are problematic. The solution to this problem is to use the deterministic algorithms that are pseudorandom number generators (PRNGs). PRNGs take a random binary sequence of length k and produce a periodic *random looking* binary sequence of length $l >> k$ [9]. The outputs of these sources are pseudorandom. Because of pseudorandomness, the outputs must be checked whether

they have some non-random properties. They are subjected to the statistical tests which are designed to detect the characteristics expected from a random sequence. With that aim, NIST published a suite of randomness tests [12] which are used to evaluate the numbers and to compare them to truly random numbers via their probability; that is, the output of the generator should not be distinguished from the random numbers, that is, it should be *random looking*. Soto et al. applied NIST randomness tests suite to the candidate and finalist algorithms in AES competition [13], [14]. Therefore, cryptographic algorithms should have the randomness property.

There exist two types of randomness testing: the first one is statistical randomness testing and the second one is cryptographic randomness testing which analyzes the property of cryptographic randomness of the algorithms. In this work, we use a cryptographic testing, namely SAC test that is available in the package of cryptographic randomness testing [10].

2.1. Strict avalanche criterion (SAC) test

In addition to the randomness property, there exist desirable cryptographic properties for block ciphers and hash functions. For example, confusion and diffusion are principal properties for block ciphers while collision resistance is an essential design criterion for hash functions. If cryptographic algorithms do not possess these properties with a significant degree, then they are considered to have poor randomization. In fact, this situation is sufficient to break the algorithms. Hence, cryptographic randomness testing is crucial for the algorithms to determine their security levels [11].

Recently, a package has been designed by Doğanaksoy et. al [10] to evaluate block ciphers and hash functions via cryptographic randomness tests. This package consists of 4 tests:

— SAC Test,

— Linear Span Test,

— Collision Test and

— Coverage Test.

SAC Test is primarily recommended for S-boxes by Webster and Tavares [15]. Furthermore, it is located in a test package designed by Doğanaksoy et. al [10]. SAC test measures whether one input bit change affects any output bit changes with probability $\frac{1}{2}$ or not. To test SAC property, SAC Matrix is formed using 2^{20} different random inputs and corresponding outputs.

SAC test is done as follows:

(1) Set the $n \times n$ SAC Matrix entries to 0.

(2) Get a random plaintext and compute the corresponding ciphertext (original output).

(3) For each $1 \leq i \leq n$:
- Flip the ith bit of the input and get the corresponding output.
- XOR the original output with the corresponding output.
- Write the result of XOR of the original output with the corresponding output to the ith row of SAC Matrix.

(4) Repeat this process for 2^{20} different random inputs.

We use 2^{20} different random inputs and corresponding output sets which are obtained from $*$ operation for $w = 16, 32, 64$ and the π-function of one, two and three rounds for π16-Cipher. We give the details of our method to get output sets in Section 4.

After this process, SAC Matrix is obtained. Using χ^2 *Goodness of Fit Test*, SAC Matrix is evaluated and p-value is obtained. Afterwards, we obtain the number of rounds for which the $*$ operation and the π-function of one, two and three rounds for π16-Cipher behave like random mappings, according to the corresponding p-value, which helps us to estimate the security level of the algorithms [16].

3. π-Cipher

π-Cipher is a sponge-based algorithm with 4 different types named:

— π16-Cipher096,

— π32-Cipher128,

— π64-Cipher128 and

— π64-Cipher256

which was designed by Gligoroski et al.; also, it consists of three rounds [17]. As shown in Figure 1, the encryption scheme is divided into four parts: initialization, associated data processing, secret message number processing, plaintext processing with tag generation. Moreover, the encryption/authentication and decryption/verification of the algorithm have a new construction namely *triplex component* which is related to the dublex sponge.

The triplex component takes the internal state, counter and input string as inputs, and then it always outputs the authentication tag.

The 4 parts use a permutation which is called the π-function. π-function is both an ARX based permutation and the core part of the algorithm. It consists of three rounds. Each round has two consecutive transformations called E_1 and E_2. These transformations are based on $*$ operation given in Figure 2. In other words, $Z = X * Y \equiv \sigma(\mu(X) \boxplus_4 \nu(Y))$, where $\boxplus_4$ is the component-wise addition of two vectors of dimension 4 in $\mathbb{Z}_{2^w}^4$, $w = 16, 32, 64$, and X, Y and Z in $\mathbb{Z}_{2^w}^4$ have

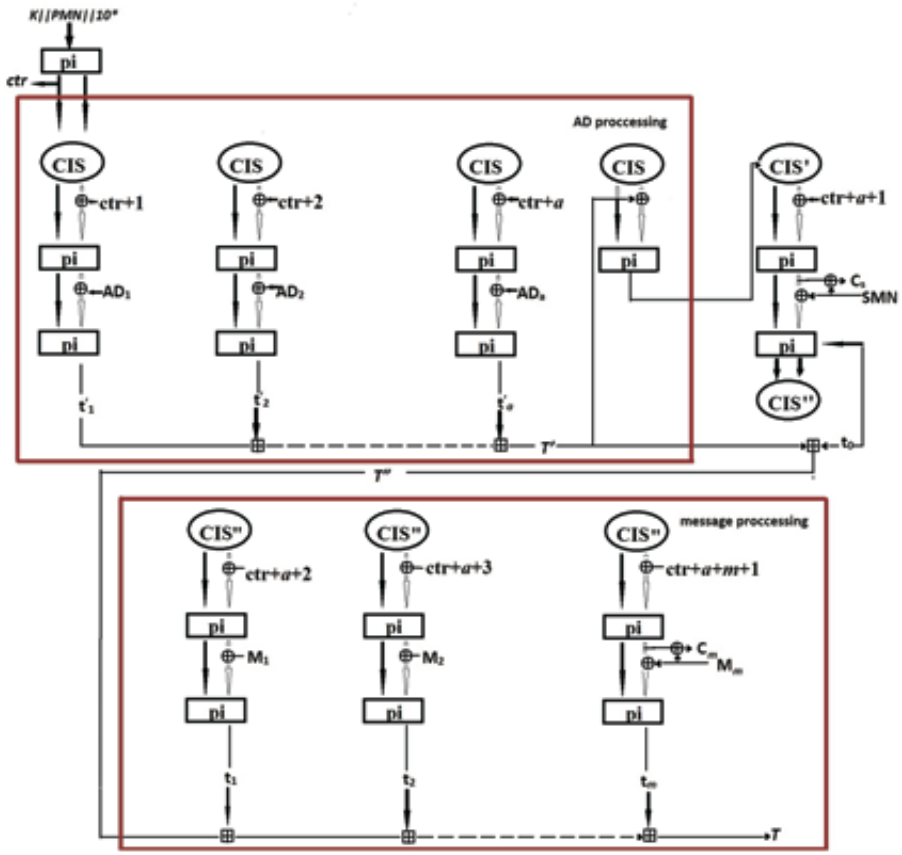

FIGURE 1. π-Cipher process of initialization, secret message number, associated data and plaintext; also, the generation of a tag [17].

different word sizes for types of π-Cipher [17]. Details of the transformations σ, μ, ν can be found in [17].

The π-function has two consecutive transformations E_1 and E_2 for one round. In Figure 3, the definition of E_1 is $E_1 : \mathbb{Z}_{2^w}^{N+1} \to \mathbb{Z}_{2^w}^{N}$ such that

$$E_1(C_1, I_1, \ldots, I_N) = (J_1', \ldots, J_N'),$$

where $J_1 = C_1 * I_1$, $J_i = J_{i-1} * I_i$ for $i = 2, \ldots, N$ and C_1 is a 4-tuple of w-bit constant defined in [17] while the definition of E_2 is $E_2 : \mathbb{Z}_{2^w}^{N+1} \to \mathbb{Z}_{2^w}^{N}$ such that

$$E_2(C_2, J_1', \ldots, J_N') = (J_1, \ldots, J_N),$$

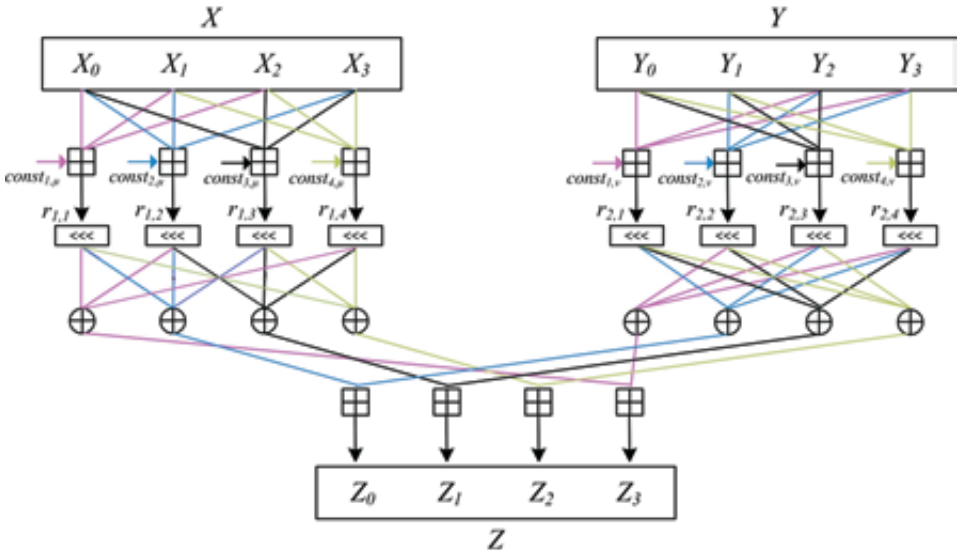

FIGURE 2. Graphical representation of $*$ operation.

where $J_N = J'_N * C_2, J_{N-i} = J'_{N-i} * J_{N-i+1}$ for $i = 1, \ldots, N-1$ and C_2 is a 4-tuple of w-bit constant defined in [17].

π-function is defined as follows for one round:

$$\pi(I_1, \ldots, I_N) = E_2\big(C_2, E_1(C_1, I_1, \ldots, I_N)\big),$$

where N is taken as the value 4 [17]. In version 2, the round number is reduced from 4 to 3. Thus, in this light, the π-function with 3 rounds is defined as

$$\pi(I_1, \ldots, I_N) = E_2\left(C_6, E_1\left(C_5, E_2\left(C_4, E_1\left(C_3, E_2\big(C_2, E_1(C_1, I_1, \ldots, I_N)\big)\right)\right)\right)\right),$$

where C_i's are 4-tuple of w-bit constants for $i = 1, 2, 3, 4, 5, 6$ defined in [17].

3.1. Bit diffusion analysis of π-Cipher

The designers of π-Cipher presented the bit diffusion analysis of $*$ operation of $w = 16, 32, 64$ and one round π-function for π 16-Cipher, π 32-Cipher and π 64--Cipher. They constructed two experimental settings to evaluate the bit diffusion in the analysis. The first setting for $*$ operation was based on 10000 randomly generated right and left inputs of $*$ operation, and then the designers analyzed the propagation of one bit difference for 10000 inputs as follows:

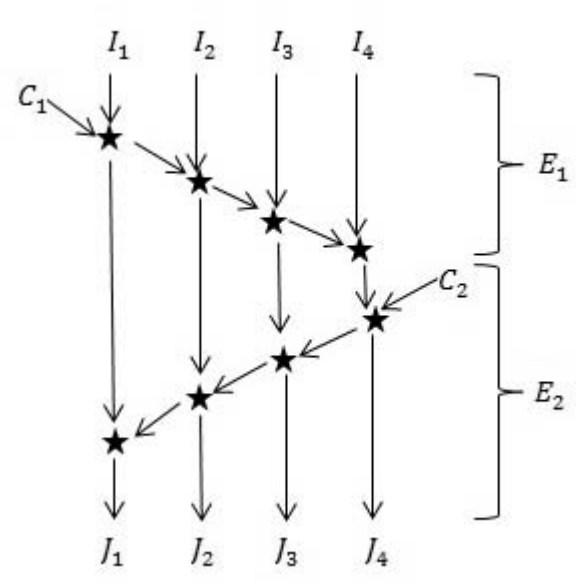

FIGURE 3. Graphical representation of E_1 and E_2 transformations.

(1) Compute $Z = X * Y$, where X and Y are inputs and Z is output of $*$ operation.

(2) Evaluate $Z' = X' * Y$, where X' is an input of $*$ operation such that

$$HammingDist\,(X, X') = 1.$$

(3) Measure the Hamming distance between Z and Z'.

They repeated the same process for Y. Afterwards, they represented the results in figures for X and Y in the values of $w = 16, 32, 64$ without any conclusion [17]. The second setting for one round π-function was based on 1000 randomly generated inputs for IS of π-function, and then the bit difference propagation for 1000 inputs was examined as follows:

(1) Compute the output of one round π-function of IS.

(2) Evaluate the output of one round π-function of IS', where is an input of one bit change in IS .

(3) Measure the Hamming distance between $\pi\,(IS)$ and $\pi\,(IS')$.

This was done for all of π 16-Cipher, π 32-Cipher and π 64-Cipher. Then the designers presented the results in figures in terms of minimum, average and maximum avalanche effect of one bit difference of π 16-Cipher, π 32-Cipher and π 64-Cipher without any conclusion [17].

3.2. The Cryptanalysis of π-Cipher

There are some cryptanalysis papers of π-Cipher [18], [19], [20], [21], [22].

4. SAC test application and results

In this work, we apply SAC test to $*$ operation of π-Cipher for $w = 16, 32, 64$ and reduced versions of π-function for $w = 16$.

We apply SAC test on $*$ operation and π-function in two ways. First, the diffusion property of the $*$ operation is analyzed. For this analysis, steps presented below are followed.

(1) Choose a random Y and fix this value.

(2) Choose a random X and compute $Z = X * Y$. For each i, where
$$i = 0, \ldots, n-1,$$

(3) Generate X_i by flipping the ith bit of X and compute $Z_i = X_i * Y$.

(4) Increment the $(i, j)th$ entry of the $n \times n$ SAC matrix if jth bit of $Z \oplus Z_i$ is 1.

(5) Repeat the steps 2 to 4.

The procedure is carried out for 2^{20} different values of X. Also, the same steps are repeated for Y with fixed X as well.

We apply a χ^2 Goodness of Fit Test to the SAC matrix with the subinterval probabilities stated in [10], flag the entries which deviate from the mean significantly and repeat the test once more. If a previously flagged entry deviates significantly again, we conclude that the algorithm fails from the SAC test and there is a strong evidence of correlation in the flagged input-output bit pair [10].

We observe that the applications of SAC test on $*$ operation for $w = 16, 32, 64$ for both inputs X and Y give the values of $p < 0.01$ and each entry of each SAC matrix is flagged twice. Therefore, $*$ operation is *non-random* for $w = 16, 32, 64$.

The only nonlinear part of the $*$ operation is the modular addition: the diffusion layer consists of two simple permutations and one rotation. Experimental result indicates that a single bit difference in the input affects 18 bits on average instead of the expected 32. Furthermore, we know that the difference in the leftmost bits of each modular addition results in fewer number of bit changes in the output due to the differential characteristics of modular addition. The results for these bits are presented in Table 1 and Table 2.

The results indicate that if there is a difference in the $2nd$ bit of X, then 15.26 output bits change on average and this may be the starting point of a cryptanalysis.

TABLE 1. Results of SAC Test for X.

Position of the flipped bit	Avg. number of output bit changes
2	15.26
50	16.24
17	16.35
18	16.40

TABLE 2. Results of SAC Test for Y.

Position of the flipped bit	Avg. number of output bit changes
33	15.86
34	15.86
49	16.18
50	16.37

The second analysis method of the cipher using SAC Test is the statistical examination of π-function. Similar to the first method, take an input $I \in (\mathbb{Z}^4_{2^{16}})^N$ such that

$$\pi\left(I_0, \ldots, I_{b-1}\right) = J, \quad \text{where} \quad b = N \times 4 \times 16 \quad \text{and}$$

change the input bit I_i for $i = 0, \ldots, b-1$ and compute the corresponding output, i.e.,

$$\pi(\neg I_0, \ldots, I_{b-1}) = J^1, \ldots, \pi\left(I_0, \ldots, \neg I_{b-1}\right) = J^b.$$

Then, XOR the output corresponding to the one-bit change of input and the original output, i.e., $J^i \oplus J$ for $i = 1, \ldots, b$. Finally, write the result of XOR on the ith row of SAC Matrix. The procedure is performed for only one input, so we repeat it for 2^{20} different inputs.

TABLE 3. SAC Test for π-function.

SAC test results for π 16-Cipher			
	1 round	2 rounds	3 rounds
p value	0.969954	0.429349	0.774130

In this work, we apply SAC test to the reduced round versions of π function for π 16-Cipher. We observe that the p values obtained from the test are 0.969954, 0.429349 and 0.77413 for 1, 2 and 3 rounds of π-function, respectively.

Since all p values are greater than 0.01 we conclude that all versions of π-function for π 16-Cipher are *random*.

5. Conclusion

In this work, we apply the Strict Avalanche Criterion (SAC) Test to π-Cipher algorithm of the CAESAR competition. We improve the evaluation of the diffusion of $*$ operation and π-function given by the designers of the algorithm. According to the corresponding test results given in the Table 1, 2, 3 in Section 4, we determine that the algorithm behaves randomly for one, two or three rounds, and explain the diffusion of $*$ operation and π-function in further details. As a future work, we can apply other randomness tests to the algorithm. Furthermore, we plan to apply SAC test to other algorithms that have the potential to remain in the CAESAR competition.

REFERENCES

[1] AES Competition, http://csrc.nist.gov/archive/aes/
[2] eSTREAM Competition, http://competitions.cr.yp.to/estream.html
[3] SHA-3 Competition, http://csrc.nist.gov/groups/ST/hash/index.html
[4] BERSTEIN, D.: The CAESAR competition, http://competitions.cr.yp.to/caesar.html
[5] The CAESAR committee, http://competitions.cr.yp.to/caesar-committee.html
[6] US National Institute of Standards and Technology, http://www.nist.gov/
[7] BERSTEIN, D.: (Google Groups), Cryptographic competitions, https://groups.google.com/forum/#!forum/crypto-competitions
[8] BERSTEIN. D.: The CAESAR competition, http://competitions.cr.yp.to/caesar.html
[9] MENEZES, A. J.—VAN OORSCHOT, P. C.—VANSTONE, S. A. : *Handbook of Applied Cryptography*. CRC Press, 2001.
[10] DOĞANAKSOY, A.—EGE, B. —KOÇAK, O.—SULAK, F.: *Cryptographic Randomness Testing of Block Ciphers and Hash Functions*. In: IACR Cryptology ePrint Archive Vol. 564, 2010.
[11] SULAK, F.: *Statistical Analysis of Block Ciphers and Hash Functions*, PhD Thesis, Middle East Technical University, February 2011.
[12] RUKHIN, A.—SOTO, J.— NECHVATAL, J.—SMID, M.—BARKER, E.— LEIGH, S.—LEVENSON, M.—VANGEL, M.—BANKS, D.—HECKERT, A—DRAY, J.—VO, S.: *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. NIST Special Publication 800-22. Revision 1a. April 2010.
[13] SOTO, J.: *Randomness testing of the advanced encryption standard candidate algorithms*, NISTIR 6390, September 1999. https://csrc.nist.gov/csrc/media/publications/nistir/.../ir6390.pdf
[14] SOTO, J.—BASSHAM, L.: *Randomness testing of the advanced encryption standard finalist candidates*, NISTIR 6483, March 28, 2000.

[15] WEBSTER, A. F.— TAVARES, S. E.: *On the design of S-boxes*. In: Conference on Advances in Cryptology-CRYPTO 85, Lecture Notes in Comput. Sci. Vol. 218, Springer--Verlag, New York, Inc., New York, NY, USA, 1986, pp. 523–534.

[16] ÇALIK, Ç.—DOĞANAKSOY, A.—TURAN, M. S.—SARAN, N. B.: *New distinguishers based on random mappings against stream ciphers*. In: S. W. Golomb, M. G. Parker, A. Pott, and A. Winterhof, eds. SETA, Lecture Notes in Comput. Sci. Vol. 5203, Springer--Verlag, New York, 2008, pp. 30–41.

[17] GLIGOROSKI, D.—MIHAJLOSKA, H.—SAMARDJISKA, S.—JACOBSEN, H. — EL-HADEDY, M.—JENSEN, R. E.—OTTE, D.: *π-Cipher v2.0, Submission to The CAESAR Competition*, August 29, 2015.

[18] LEURENT, G.: *Tag second-peimage attack against π-cipher,* 2014. ($< hal - 00966794v2 >$)

[19] FUHR, T.—LEURENT, G.: *Observation on π-Cipher*, submission to CAESAR competition mailing list, November, 2014.

[20] ALLEY, J.—PIEPRZYK, J.: *State recovery attacks against π-Cipher*. In: Proceedings of the Australasian Computer Science Week Multiconference, ACSW '16, February 1, 2016.

[21] MIHAJLOSKA, H.—MENNINK, B.—GLIGOROSKI, D.: *π-cipher with ntermediate tags*, pi-cipher.org, May 13, 2016.

[22] BOURA, C.—CHAKRABORTI, A.—LEURENT, G.—PAUL, G.—SAHA, D.— SOLEIMANY, H.—SUDER, V. : *Key recovery attack against 2.5-round π-cipher*. In: FSE '16, May 23, 2016, IACR, report Vol. 502, 2016 (accepted paper).

Received September 28, 2016

Fatih Sulak
Department of Mathematics
Atılım University
TURKEY
E-mail: fatih.sulak@atilim.edu.tr

Beyza Bozdemir
Institute of Applied Mathematics
Middle East Technical University
TURKEY
E-mail: e171943@metu.edu.tr

Betül A. Özdemir
Department of Mathematics
Atılım University
TURKEY
E-mail: askinbetul@gmail.com

Neşe Koçak
ASELSAN Inc.
Tübitak Bilgem
TURKEY
E-mail: neseoztop@gmail.com

Onur Koçak
Tübitak Bilgem
TURKEY
E-mail: onur.kocak@metu.edu.tr