Title: Tabu Cryptanalysis of VMPC Stream Cipher

Author: Iwona Polak, Mariusz Boryczka

Citation style: Polak Iwona, Boryczka Mariusz. (2019). Tabu Cryptanalysis of VMPC Stream Cipher. “Tatra Mountains Mathematical Publications” (73, (2019), s.145–162).

sciendo

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.2478/tmmp-2019-0011
Tatra Mt. Math. Publ. **73** (2019), 145–162

TABU CRYPTANALYSIS OF VMPC STREAM CIPHER

Iwona Polak — Mariusz Boryczka

Institute of Computer Science, University of Silesia, Sosnowiec, POLAND

ABSTRACT. In the era of global informatization, transmitting and storing information in digital form it is very important to ensure an adequate level of security of ciphers used. Cryptanalysis deals with studying the level of security, thus exposing the weakness of theoretical and implemented cryptographic solutions. In this paper cryptanalysis of stream cipher VMPC using Tabu Search is shown. From estimates made on a full version of VMPC cipher we concluded that about 2^{157} possibilities needs to be checked in order to find the proper one, which would be the best attack known so far.

1. Introduction

Every day people communicate with each other. To secure this communication in computer systems a cryptography is widely used. A cryptography is such a transformation of a plaintext message that it becomes unreadable for a third party without a secret key.

Modern cryptography systems can be divided into two branches depending on what unit they transform. If a larger block (e.g., 64-128 bits) is transformed as a whole, this is a block cipher. If a small unit (e.g., 1 bit or 1 byte) is transformed separately one by one, this is a stream cipher. In this paper, a stream cipher called VMPC will be discussed.

In order to make a cryptanalysis of VMPC cipher, we propose a modification of Tabu Search (TS) metaheuristic algorithm [4] and we called it tabu cryptanalysis. Metaheuristics were used so far for cryptanalysis, but mainly for classical and modern block ciphers. E.g., cryptanalysis of substitution cipher with Genetic Algorithm [1]; cryptanalysis of Vigenère cipher with Genetic Algorithm,

2010 Mathematics Subject Classification: 62K05.
Keywords: stream cipher, VMPC, cryptanalysis, state recovery attack, Tabu Search, TS.

Particle Swarm Optimization and Cuckoo Search [2]; cryptanalysis of DES reduced to four and six rounds with Particle Swarm Optimization and Differential Evolution [6]; cryptanalysis of DES reduced to six rounds with Genetic Algorithm and Simulated Annealing [3]. The cryptanalysis of stream ciphers with metaheuristic algorithms has not been tested so well.

Tabu Search was introduced in 1986 by F. Glover [4]. It is an iterated algorithm. In every iteration a neighbourhood of the current solution is checked. A neighbourhood is formed by a predefined move. Solutions from the neighbourhood are evaluated with some fitness function and the best one is chosen as a current solution for the next iteration. The important part of a Tabu Search is tabu list. It is a list of moves done recently. At the same time, these are forbidden moves. The number of iterations in which a move is forbidden is called horizon. The purpose of tabu list is to escape from local optima and go to a new area of potential solutions.

The attack, presented in this work, is a state recovery attack. That means that our aim is to find the internal state of the cipher right after key scheduling algorithm and before pseudo-random generation phase. This would be equivalent to revealing the key. The attack requires $|S|$ bytes of keystream, where $|S|$ is the size of permutation array in bytes. In [9] the same kind of attack was presented for the VMPC cipher. However, further research led to conclusion that parameters used then could be insufficient in order to find internal state of the cipher. In current research four types of fitness function, four types of neighbourhood, six types of horizon and two types of aspiration criteria were used. Experiments on the reduced version of VMPC cipher showed that a neighbourhood type used in research presented in [9] is ineffective.

Research carried out on reduced to 10 and 16 bytes versions of VMPC cipher showed implicitly that tabu cryptanalysis is possible. They were also a basis to select the values of parameters for tabu cryptanalysis of the full version of VMPC cipher.

In the following Subsection 1.1 the notation used in our work is introduced. In Section 2 the VMPC cipher is described. The attacks known so far are also presented there. Section 3 contains the description of tabu cryptanalysis and considered values of parameters. In Section 4 the experiments and their results are presented. Section 5 concludes the work.

1.1. Notation

In this paper we use the following notation:

- avg – arithmetic mean,
- B – compatibility function,
- f_{fit} – fitness function,

- i, j – 8-bit integer indices,
- κ – keystream,
- κ' – keystream from the current solution,
- K – key,
- l – iteration number,
- max – maximum value,
- min – minimum value,
- P – plaintext (in bytes),
- r – current solution (permutation),
- $R(r)$ – neighbourhood of the current solution,
- R^2 – coefficient of determination,
- $rBest$ – the best permutation found in the whole algorithm run,
- S – a byte array containing the permutation of natural numbers $\{0, 1, \ldots, S-1\}$,
- s – standard deviation,
- TABU – tabu list,
- $\oplus$ – exclusive or (XOR),
- $+$ – addition mod 256,
- $|\cdot|$ – cardinality of a set or the length of stream in bytes (depending on context).

The abbreviated permutation $(a_0, a_1, a_2, a_3, \ldots, a_{|S|-1})$ will mean the mapping of 0 in a_0, element 1 in a_1 and so on:

$$\big(a_0, a_1, a_2, a_3, \ldots, a_{|S|-1}\big) = \begin{pmatrix} 0 & 1 & 2 & 3 & \ldots & |S|-1 \\ a_0 & a_1 & a_2 & a_3 & \ldots & a_{|S|-1} \end{pmatrix}. \tag{1}$$

2. VMPC

Stream cipher VMPC is Polish cipher introduced in 2004 by B. Ż ó ł t a k [13]. Its author claims that it is used by Polish firms, institutions and individuals. Although, in our opinion, security of VMPC cipher has not been verified sufficiently.

VMPC cipher consists of a 256-byte array S, which contains a permutation of integer numbers from 0 to 255, and two 8-bit variables i and j. In every clock this cipher produces one byte of a keystream by function:

$$g(x) = \text{VMPC}\big(f(x)\big) = f\big(f\big(f(x)\big) + 1\big) \bmod 256. \tag{2}$$

Internal state of VMPC cipher changes with every clock, because of a 2-element swap.

The key K in the VMPC cipher is not used directly, which is common in stream ciphers. The secret key and public initialization vector (IV) are the input for the key scheduling algorithm (KSA). In this phase they are being diffused in the internal state of the VMPC cipher. After this phase is completed, the pseudo-random generation algorithm (PRGA) follows. This is the phase that generates the keystream κ for encryption. In the PRGA phase the key is not used any more. The PRGA phase of the VMPC is shown in Algorithm 1. The KSA phase is not included in the article as it is irrelevant for presented research.

Algorithm 1: The PRGA phase of the VMPC cipher.

```
1 i ← 0
2 for k ← 1 to |P| do
3 |   j ← S[ j + S[i] ]
4 |   output ← S[S[S[j]]+1]
5 |   swap (S[i], S[j])
6 |   i ← i + 1
```

Keystream κ byte by byte is added to subsequent bytes of plaintext stream P using exclusive or function (XOR). This way a ciphertext stream C is obtained:

$$P_m \oplus \kappa_m = C_m, \tag{3}$$

where:

P_m**:** m-th byte of plaintext,

κ_m**:** m-th byte of keystream,

C_m**:** m-th byte of ciphertext.

2.1. Known attacks

The author of VMPC cipher claims that inverting the base function requires an estimated average of 2^{260} computational operations [13]. In literature, a few distinguishing attacks can be found [7, 8, 11, 12]. The best of them was proposed by S. S a r k a r [11] and requires 2^{24} bytes of a keystream. Although the existence of distinguishing attacks is considered as security threat for a stream cipher, the distinguishing attack does not reveal directly any information about the key or the plaintext. This kind of attack only finds statistical deviations from truly random stream in order to distinguish if the generated stream is random or produced by VMPC cipher. Therefore, it seems that the security of VMPC cipher needs to be better investigated.

3. Tabu cryptanalysis

Cryptanalytic attack presented in the paper is based on metaheuristic technique called Tabu Search (TS) [4]. TS was derived from how people solve tasks: they avoid repeating recently made moves and search new types of solutions somewhere else. We have adopted the technique to cryptanalysis and we named our approach a tabu cryptanalysis (Algorithm 2).

Algorithm 2: TABU CRYPTANALYSIS.

```
1  r ← random permutation
2  rBest ← r
3  TABU ← {}
4  l ← 0
5  while termination criterion not satisfied do
6  |   R(r) ← {permutations arising from possible 2-element swap on r}
7  |   r = arg max_{n∈R(r)} (f_fit(n))
8  |   if f_fit(r) > f_fit(rBest) then
9  |   |   rBest ← r
10 |   update TABU
11 |   l ← l + 1
12 return rBest
```

At the start we pick some random solution (random permutation). Next we define a neighbouring permutation as a permutation that has two elements swapped with each other. Every permutation is rated by fitness function f_{fit}. From those neighbouring permutations the best one is chosen as current solution. But at the same time this swap is put on tabu list. This will prevent from using the same swap in some nearest iterations. The number of iterations in which a recently done move is forbidden is called horizon and can be parameterized. At the end, the best solution found during the whole run is returned.

The termination criterion (Algorithm 2, line 5) can be different, e.g., a predefined number of iterations, fixed amount of time, reaching predefined value of fitness function or no improvement during fixed number of subsequent iterations. Termination criterion in Section 4 is defined separately for every experiment and it is included in the conditions of the experiment.

The $rBest$ is updated as global best solution found so far. Current and next solution are not compared with each other. The current solution r is changed in every iteration as the best from the legitimate neighbourhood (Algorithm 2, line 7).

3.1. Fitness function

In all types of fitness function compatibility function of two bytes was used. This compatibility function was defined as follows:

$$B(\kappa'_m, \kappa_m) = \begin{cases} 0, & \kappa'_m \neq \kappa_m, \\ 1, & \kappa'_m = \kappa_m. \end{cases} \tag{4}$$

Four types of fitness function were tested:

a) byte fitness:

$$f_{fit}(\kappa') = \frac{\sum_{m=1}^{|\kappa|} B(\kappa'_m, \kappa_m)}{|\kappa|} \cdot 100\,\%, \tag{5}$$

b) adjacent byte fitness:

$$f_{fit}(\kappa') = \frac{2 \cdot \sum_{m=1}^{|\kappa|} B(\kappa'_m, \kappa_m) + \sum_{m=1}^{|\kappa|-1} B(\kappa'_{m+1}, \kappa_m) + \sum_{m=2}^{|\kappa|} B(\kappa'_{m-1}, \kappa_m)}{2 \cdot |\kappa|} \cdot 100\,\%, \tag{6}$$

c) weighted fitness:

$$f_{fit}(\kappa') = \frac{\sum_{m=1}^{|\kappa|} (|\kappa| - m + 1) \cdot B(\kappa'_m, \kappa_m)}{\frac{|\kappa|+1}{2} \cdot |\kappa|} \cdot 100\,\%, \tag{7}$$

d) weighted reversely fitness:

$$f_{fit}(\kappa') = \frac{\sum_{m=1}^{|\kappa|} m \cdot B(\kappa'_m, \kappa_m)}{\frac{|\kappa|+1}{2} \cdot |\kappa|} \cdot 100\,\%. \tag{8}$$

3.2. Neighbourhood

In the problem presented in the article, an acceptable solution is any permutation of natural numbers from 0 to a given range, depending on the size of the analysed cipher ($|S|$). For the VMPC cipher, this permutation will determine the internal state at the beginning of communication, i.e., right after the end of the KSA phase, and before starting the PRGA phase.

The solution from the permutation neighbourhood is also a permutation, almost identical to that considered. Permutations differ from each other by any two elements that swapped their places. Having the given permutation: $(a_0, a_1, \ldots, a_x, \ldots, a_y, \ldots, a_{|S|-1})$ neighbouring permutation is the permutation $(a_0, a_1, \ldots, a_y, \ldots, a_x, \ldots, a_{|S|-1})$ for each pair $x, y \in \langle 0, |S| - 1\rangle$ and $x \neq y$. Each swapping of such pairs will generate a different permutation.

Four types of neighbourhood were tested:

a) all pairs – all possible pairs of swaps were tested in each iteration,

b) half of pairs – the random half of possible pairs of swaps were tested in each iteration,

c) elements random – all elements were tested with one randomly generated swap in each iteration,

d) elements adjacent – all swaps of adjacent elements were tested (including a swap of the last element with the first one) in each iteration.

3.3. Horizon

A horizon is the number of iterations in which a move is forbidden. Six types of horizon were tested:

a) $\ln |R(r)|$,

b) $\log_2 |R(r)|$,

c) $\sqrt{|R(r)|}$,

d) $|R(r)|/2$,

e) $3|S|$,

f) random from the range $\langle \log_2 |R(r)|, 3|S| \rangle$ (different for every iteration).

3.4. Aspiration criterion

Two types of aspiration criterion were tested:

a) none – the prohibition resulting from the tabu list is strict,

b) best so far – the move from tabu list is accepted if it produces solution of higher value of fitness function than the best found so far.

4. Experiments and results

All experiments were conducted with the usage of an application written in C# in Visual Studio 2015. The experiments were carried out using a computer equipped with an Intel Core i7 processor (3.30 GHz) and 16 GB of DDR 4 RAM and with a 64-bit operating system Windows 7 Pro installed. Presented in the article tabu cryptanalysis (Section 3) is probabilistic algorithm and every run can give different course. Because of that fact a test for every set of parameters was repeated 32 times. Conclusions were drawn on the basis of an average of the returned values from those 32 tests.

The attack is known plaintext attack. In this approach a cryptanalyst has access to a plaintext and corresponding ciphertext. From those two in stream ciphers one can calculate keystream. The aim of the research was to find state of the permutation $|S|$ of VMPCcipher, right after KSA. No initial bytes were drawn. Revealing the initial internal state would allow to decrypt the rest of the secret message, knowledge of the key is in this case unnecessary.

4.1. Selection of parameters' values for the tabu cryptanalysis

Conditions of experiments:

- cryptanalysis of: VMPC_10,
- analysis of 10 B of keystream,
- 10 different keys (Table 1),
- termination criterion: checking 3 628 800 permutations or $f_{fit} = 100\,\%$,
- random starting permutation,
- 4 types of fitness function (Subsection 3.1),
- 4 types of neighbourhood (Subsection 3.2),
- 6 types of horizon (Subsection 3.3),
- 2 types of aspiration criterion (Subsection 3.4),
- number of independent tests for one set of parameters: 32.

The size of the search space is equal $10! = 3\,628\,800$.

VMPC_10 cipher works the same as full version of VMPC cipher, the only difference is that the base permutation S contains 10 elements from 0 to 9. Also all operations from Algorithm 1 are made mod 10.

TABLE 1. Keys used in cryptanalysis of the VMPC_10 cipher.

Key	IV
0x0007050003010609	0x0502070002030309
0x0000000000000000	0x0502070002030309
0x0000000000000000	0x0505050505050505
0x0000000000000000	0x0000000000000000
0x0008000000000000	0x0000000000000800
0x0909090909090909	0x0000000000000000
0x0001020304050607	0x0000000000000000
0x0003060306030609	0x0000000000000000
0x0000000000000000	0x0000000600000000
0x0000000000000000	0x0003060306030609

Different neighbourhoods have different sizes. In experiment with VMPC_10 we wanted to compare the same number of solutions checks, because we think that such a comparison is fair. Comparing the same number of iterations would not be authoritative, because a test for "all pairs" would last much longer than for "elements random". Because of the different sizes of neighbourhoods it is not possible to give one universal number of iterations used. Moreover if the maximum number of checked solutions was exhausted, the iteration was stopped in the process.

The selection of the parameter values of the tabu cryptanalysis was made according to the following scheme: the values of the determined parameter were changed in a given range, at the set values of the other parameters. The value of the parameter, which made it possible to obtain the best results, was each time determined as valid at the later stage of the experiment. This scheme was performed successively for all of the mentioned parameters. The following initial values of parameters were assumed:

- fitness function: byte (eq. (5)),
- neighbourhood: half of pairs,
- horizon: $\sqrt{|R(r)|}$,
- aspiration criterion: none.

First tests were performed for the fitness function (Table 2). Then the neighbourhood (Table 3) and the horizon (Table 4) were tested in turn. Finally, the aspiration criterion was checked (Table 5).

TABLE 2. The results for the fitness function for VMPC_10 cipher.

f_{fit}	Number of successes	
byte	**302**	**94.4 %**
adjacent byte	287	89.7 %
weighted	219	68.4 %
weighted reversely	206	64.8 %

The best results (94.4 %) were achieved for simple byte fitness. A little worse results (89.7 %) were achieved for more complicated adjacent byte fitness. Both weighted fitness resulted definitely worse (64.8-68.4 %) (Table 2).

TABLE 3. The results for the neighbourhood for VMPC_10 cipher.

Neighbourhood	Number of successes	
all pairs	0	0.0 %
half of pairs	**302**	**94.4 %**
elements random	299	93.4 %
elements adjacent	0	0.0 %

In each iteration, swaps are randomly selected, which further reduces the chances of falling into the cycle. Even if the same solution is analysed again, in its neighbourhood there may not be a solution obtained by the swap that was done previously.

The best results (94.4 %) were achieved for "half of pairs" neighbourhood. A slightly worse results (93.4 %) were achieved for "elements random" neighbourhood. The other two types of neighbourhood resulted in 0 % success rate (Table 3).

In the case of "elements random" analysis, the exploration of the solution space in relation to the "all pairs" neighbourhood is significantly increased (Table 3). In "all pairs" neighbourhood there is the largest neighbourhood size from those under consideration. This resulted in the smallest number of actual swaps. It seems that this is the reason of poor results in this case. In "elements adjacent" neighbourhood swaps can be made only between adjacent elements. Probably this caused the algorithm to stuck in local optimum.

TABLE 4. The results for the horizon for VMPC_10 cipher.

Horizon	Number of successes	
$\ln \lvert R(r)\rvert$	302	94.4 %
$\log_2 \lvert R(r)\rvert$	295	92.2 %
$\sqrt{\lvert R(r)\rvert}$	302	94.4 %
$\lvert R(r)\rvert/2$	292	91.3 %
$3\lvert S\rvert$	**305**	**95.3 %**
random	297	92.8 %

The results for different horizon functions are more or less similar. This shows that the algorithm is not quite sensitive for this parameter. The best results (95.3 %) were obtained for $3|S|$ (for VMPC_10 horizon $3|S| = 30$) (Table 4).

TABLE 5. The results for the aspiration criterion for VMPC_10 cipher.

Aspiration	Number of successes	
none	**305**	**95.3 %**
best so far	299	92.2 %

Better results (95.3 %) were achieved without aspiration criterion (Table 5). The reason for this is not quite obvious.

On the basis of this experiment on the VMPC_10 cipher, the following set of parameters was chosen for further experiments:

- fitness function: byte (eq. (5)),
- neighbourhood: half of pairs,
- horizon: $3|S|$,
- aspiration criterion: none.

For these parameter values, the proper permutation was found in 305 runs from 320 (95.3 %) after checking an average of 858 708 permutations, which is equivalent to searching 23.7 % of the entire solution space. A graph of average value of fitness function in subsequent iterations is presented in Fig. 1. The graph includes only tests that ended with success.

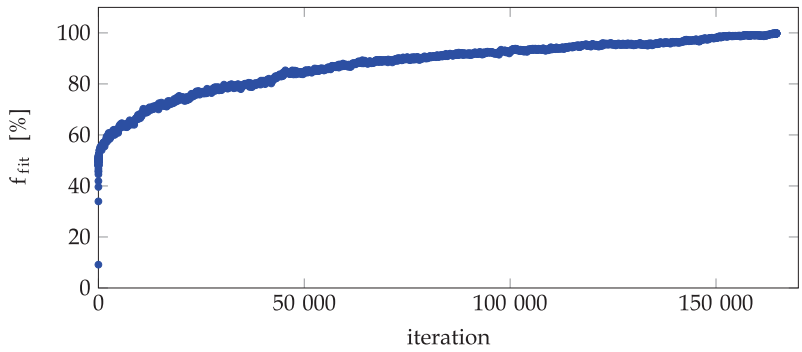

FIGURE 1. A graph of fitness function in subsequent iterations of tabu cryptanalysis of VMPC_10 cipher.

4.2. Cryptanalysis of VMPC_16 cipher

Conditions of experiments:

- cryptanalysis of: VMPC_16,
- analysis of 16 B of keystream,
- 5 different keys (Table 6),
- termination criterion: checking 1 600 000 000 permutations or $f_{fit} = 100\,\%$,
- random starting permutation,
- parameters:
  - byte fitness (eq. 5),
  - neighbourhood: half of pairs,
  - horizon: $3|S|$,
  - aspiration criterion: none,
- number of independent tests for one set of parameters: 32.

The size of the search space is equal $16! = 20\,922\,789\,888\,000 \approx 2^{44}$.

TABLE 6. Keys used in cryptanalysis of the VMPC_16 cipher.

Key	IV
0x0601010A07070809 0B060C01070D0607	0x0B0C0F000E07030507 08020F0D020105
0x0A0A0A0A0A0A0A0A 0A0A0A0A0A0A0A0A	0x0505050505050505 0505050505050505
0x0000000000000000 0000000000000000	0x080F060D040B0209 0070F0205080B0E
0x0100000000000000 0000000000000000	0x0000000000000000 0000000000000000
0x0000000000000000 0000000000000000	0x0000000000000000 0000000000010000

TABLE 7. Keys used in cryptanalysis of the VMPC cipher (the one with ID 0 is the official test vector [13]).

ID	Key	IV
0	0x9661410ab797d8a9eb767c21172df6c7	0x4b5c2f003e67f39557a8d26f3da2b155
1	0xaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	0x4b5c2f003e67f39557a8d26f3da2b155
2	0xaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	0x55555555555555555555555555555555
3	0x00000000000000000000000000000000	0x00000000000000000000000000000000
4	0x80000000000000000000000000000000	0x00000000000000000000000000000000
5	0x09090909090909090909090909090909	0x00000000000000000000000000000000
6	0x000102030405060708090a0b0c0d0e0f	0x00000000000000000000000000000000
7	0x288ff65dc42b92f960c70f62b5085bae	0x00000000000000000000000000000000
8	0x00000000000000000000000000000000	0x00000010000000000000000000000000
9	0x00000000000000000000000000000000	0x288ff65dc42b92f960c70f62b5085bae

VMPC_16 cipher works the same as the full version of VMPC cipher, the only difference is that the base permutation S contains 16 elements from 0 to 15. Also all operations from Algorithm 1 are made mod 16.

Of the 160 tests carried out, 129 (80.6 %) ended in finding the proper permutation in a given number of verified permutations. Fig. 2 presents in the form of a box chart the number of permutations checked before finding the proper permutation. The graph includes only those tests that were successful, i.e. the proper permutation was found. On average, it was necessary to check 278 033 351 permutations before finding the proper permutation. This is 0.0013 % of the entire solution space.

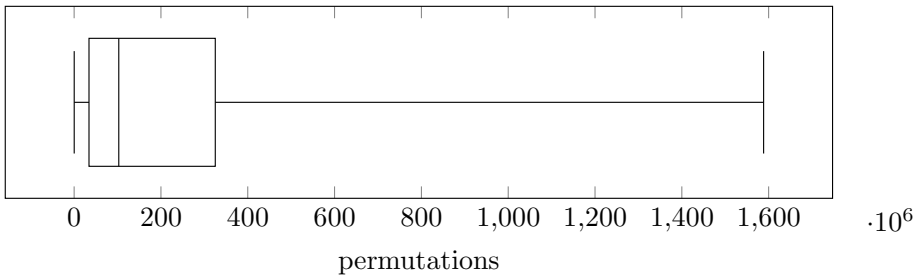

FIGURE 2. The number of permutations needed to find the correct permutation of the VMPC_16 cipher.

4.3. Cryptanalysis of VMPC cipher

Conditions of experiments:

- cryptanalysis of: VMPC,
- analysis of 256 B of keystream,
- 10 different keys (Table 7),
- termination criterion: 306 iterations (with 16 320 permutations checked in each iteration),
- random starting permutation,
- parameters:
  - byte fitness (eq. 5),
  - neighbourhood: half of pairs,
  - horizon: $3|S|$,
  - aspiration criterion: none,
- number of independent tests for one set of parameters: 32.

The size of the search space is equal $256! \approx 2^{1684}$.

For comparison, a 5 000 000 random permutations were also generated and evaluated using the same fitness function. This is approximately equal to the number of permutations evaluated during one tabu cryptanalysis run.

The results are summarized in the Table 8. The value of fitness function was 9.8 % on average for tabu cryptanalysis. At the same time random permutation check gave compatibility of 0.4 % at average and 3.9 % at maximum.

The approximation of the fitness function was verified by regression analysis. Four types of regression were tested: linear, logarithmic, exponential and power. Logarithmic regression gives the smallest error value R^2, so in the further analysis the results will be presented using it. The average course of the fitness function in subsequent iterations is shown in Fig. 3 in blue. The graph also shows the logarithmic regression line corresponding to the presented course of the fitness function in red. The logarithmic regression equation for this plot can be described by the formula:

$$f(x) = 0.0096496251 \ln(x) + 0.0447960899 \qquad (R^2 = 0.903). \tag{9}$$

Using extrapolation, it can be calculated from the eq. (9) that full compatibility of the keystream will be achieved after about 2^{143} iterations. In each iteration, $16\,320 \approx 2^{14}$ solutions are checked. So in order to find the proper permutation, it is necessary to check about 2^{157} possibilities.

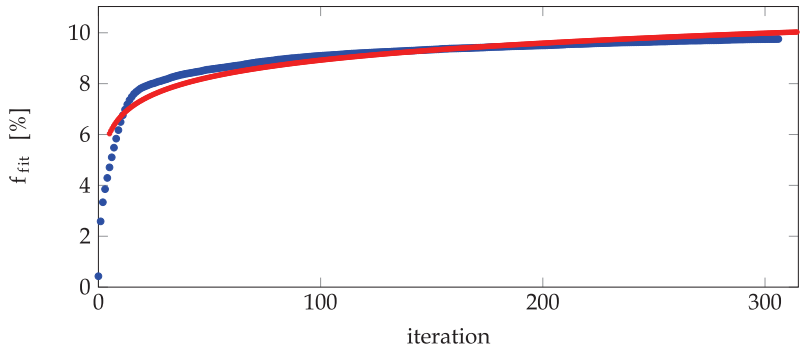

FIGURE 3. A graph of fitness function in subsequent iterations of tabu cryptanalysis of VMPC cipher — blue dots; logarithmic regression graph is marked in red.

The role of tabu list is to prevent falling local optimum. On the reduced versions of the VMPC cipher, it was implicitly shown that the method works. Assuming — with tabu list and effectiveness of Tabu Search for other big problems — the algorithm will not fall into local optimum and if the upward trend is maintained, the regression curve will be maintained. This is valuation with high probability.

TABLE 8. Results obtained for the VMPC cipher; in the case of a tabu cryptanalysis, for each key the value of the fitness function was given and (below) the iteration number in which this value was obtained. For the permutations generated randomly, for each key the obtained value of the fitness function was given in the given number of runs.

Fitness function	byte							
Algorithm	tabu cryptanalysis				random			
Key ID	*min*	*avg*	*max*	*s*	*min*	*avg*	*max*	*s*
0	7.4 %	9.5 %	11.7 %	1.1 %	0.0 %	0.4 %	3.9 %	0.4 %
	39	159	292	81	–	–	–	–
1	6.6 %	9.7 %	11.7 %	1.3 %	0.0 %	0.4 %	3.5 %	0.4 %
	21	152	285	86	–	–	–	–
2	6.6 %	9.7 %	12.5 %	1.4 %	0.0 %	0.4 %	3.5 %	0.4 %
	21	148	282	80	–	–	–	–
3	6.6 %	9.7 %	12.5 %	1.2 %	0.0 %	0.4 %	3.9 %	0.4 %
	23	156	306	84	–	–	–	–
4	7.0 %	9.8 %	12.1 %	1.4 %	0.0 %	0.4 %	3.5 %	0.4 %
	31	156	291	66	–	–	–	–
5	7.8 %	9.8 %	13.7 %	1.3 %	0.0 %	0.4 %	3.5 %	0.4 %
	24	187	305	90	–	–	–	–
6	6.3 %	9.7 %	14.1 %	1.9 %	0.0 %	0.4 %	3.5 %	0.4 %
	23	168	306	82	–	–	–	–
7	8.2 %	10.2 %	12.1 %	1.1 %	0.0 %	0.4 %	3.5 %	0.4 %
	47	174	303	87	–	–	–	–
8	6.6 %	9.4 %	12.5 %	1.6 %	0.0 %	0.4 %	3.5 %	0.4 %
	47	141	292	71	–	–	–	–
9	5.9 %	10.1 %	13.3 %	1.4 %	0.0 %	0.4 %	3.5 %	0.4 %
	25	154	306	82	–	–	–	–
altogether	5.9 %	9.8 %	14.1 %	1.4 %	0.0 %	0.4 %	3.9 %	0.4 %
	18	160	306	80	–	–	–	–

Although this number is beyond the perspective of practical breaking of the cipher, it is a much better result than the best of known attacks (presented in the Subsection 2.1) requiring on average 2^{260} computing operations. The rest of the attacks presented in the Subsection 2.1 are distinguishing attacks and, as mentioned earlier, attacks of this type neither allow direct deduction of anything about the key, nor about the internal state nor about the plaintext.

5. Conclusions and future work

In the paper, tabu cryptanalysis was presented — a state recovery attack based on Tabu Search metaheuristic. On reduced version of VMPC cipher it was shown that this kind of attack is possible. It is also worth pointing out that all successful tests for VMPC_10 and VMPC_16 ciphers returned the proper permutation.

From estimates made on a full version of VMPC cipher we concluded that about 2^{157} permutations needs to be checked in order to find the proper one. This is still a big number, but it is much better than 2^{260} computational operations needed so far [13]. Other attacks are distinguishing ones and they cannot directly lead to revealing the key or the plaintext. Further, research on the proposed cryptanalytic attack can lead to finding real-time attack on VMPC cipher. The same attack could be also possible for other permutation-based stream ciphers, e.g., RC4 [5] or Spritz [10].

REFERENCES

[1] ANTAL, E.—ELIÁŠ, M.: *Evolutionary computation in cryptanalysis of classical ciphers*, Tatra Mt. Math. Publ. **70** (2017), 179–197.

[2] BHATEJA, A. K.—BHATEJA, A.—CHAUDHURY, S.—SAXENA, P. K.: *Cryptanalysis of Vigenère cipher using Cuckoo search*, Appl. Soft Comput. **26** (2015), 315–324.

[3] DWORAK, K.—BORYCZKA, U.: *Genetic algorithm as optimization tool for differential cryptanalysis of DES6*. In: *Computational Collective Intelligence: 9th International Conference, ICCCI 2017, Nicosia, Cyprus, September 27-29, 2017, Proceedings, Part II*. Springer International Publishing, 2017, pp.107–116.

[4] GLOVER, F.: *Tabu search — Part I*, ORSA Journal on Computing, **1** (1989), no. 3, 190–206.

[5] HARRIS, B.: *Improved Arcfour Modes for the Secure Shell (SSH) Transport Layer Protocol*, Networking Working Group—Request for Comments: 4345, 2006.

[6] LASKARI, E. C.—MELETIOU, G. C.—STAMATIOU, Y. C.—VRAHATIS, M. N.: *Applying evolutionary computation methods for the cryptanalysis of Feistel ciphers*, Appl. Math. Comput. **184**, (2007), no. 1, 63–72.

[7] LI, S.—HU, Y.—ZHAO, Y.—WANG, Y.: *Improved cryptanalysis of the VMPC stream cipher*, J. Comput. Inform. Syst. **8** (2012), no. 2, 831–838.

[8] MAXIMOV, A.: *Two linear distinguishing attacks on VMPC and RC4A and weakness of RC4 family of stream ciphers*. In: *Lecture Notes in Comput. Sci. Vol. 3557*, Springer-Verlag, 2005, pp.342–358,

[9] POLAK, I.—BORYCZKA, M.: *Tabu search against permutation based stream ciphers*, International Journal of Electronics and Telecommunications, **64** (2018), no. 2, 137–145.

[10] RIVEST, R. L.—SCHULDT, J. C. N.: *Spritz—A spongy RC4-like stream cipher and hash function*. In: Presented at Charles River Crypto Day, 2014;
`https://people.csail.mit.edu/rivest/pubs/RS14.pdf`

[11] SARKAR, S.: *Further non-randomness in RC4, RC4A and VMPC*, Cryptogr. Commun. **7**(2015), no. 3, 317–330.

[12] TSUNOO, Y.—SAITO, T.—KUBO, H.—SHIGERI, M.—SUZAKI, T.—KAWABATA, T.: *The Most Efficient Distinguishing Attack on VMPC and RC4A*, 2005;
`https://pdfs.semanticscholar.org/86a6/d5bdce46c112ece81982eb189d598e4b0414.pdf`

[13] ŻÓŁTAK, B. *VMPC One-Way Function and Stream Cipher*. In: *Fast Software Encryption*. In: *Lecture Notes in Comput. Sci. Vol. 3017*, 2004, pp. 210–225.

Received August 31, 2018

Institute of Computer Science
University of Silesia
Będzińska 39
41-200 Sosnowiec
POLAND

E-mail: iwona.polak@us.edu.pl
mariusz.boryczka@us.edu.pl