UDC 621.391:519.2:519.7

RELATED-KEY CRYPTANALYSIS OF PERSPECTIVE SYMMETRIC BLOCK CIPHER

ROMAN OLIYNYKOV, DMYTRO KAIDALOV

Symmetric block ciphers are among the most widely used cryptographic primitives. In addition to providing privacy via encryption, block ciphers are used as basic components in the construction of hash functions, message authentication codes, pseudorandom number generators as part of various cryptographic protocols etc. One of the most popular block ciphers nowadays is AES (Advanced Encryption Standard), which is used as a standard of symmetric encryption in many countries of the world. Several years ago a theoretical attack against the AES key-expansion algorithm was suggested, and complexity of this attack turned out to be significantly lower compared to brute force search. This paper considers the method of estimating encryption algorithm security against related-key attacks, and its application to the perspective block cipher, which is a candidate to the block encryption standard in Ukraine.

Keywords: block cipher, cryptanalysis, related key attack.

1. INTRODUCTION

From [5,6,7] it is known that related-key attacks were found for algorithms AES-192 and AES-256 which significantly reduced the theoretical security level of these algorithms. That is why the question about new SPN-based block cipher which would be secure from attacks which is present in AES is arisen.

In this article a new improved SPN-like cipher is presented and its security estimation is made. Authors propose this cipher as a perspective algorithm [4,17] which is based on the AES but has significant improvements such as a new key expansion scheme, resistance against algebraic attacks, higher productivity, etc. The new encryption algorithm was proposed to the public competition of block cipher selection to be a prototype during development of Ukrainian National Standard [4,17]. That is the main reason of researching this cipher.

But the security proof against related-key attacks is highly required. This is a challenging problem to solve since for now there are not so many efficient algorithms for estimating cipher security against related-key attacks. So the main goal of our research was to develop such algorithm. The article contains detailed description of developed algorithm, its complexity and result of applying this algorithm to the described SPN-based cipher.

2. RELATED-KEY ATTACK FOR MODERN BLOCK CIPHERS

2.1 Description of related-key attack for AES

Brief description of this attack is taken from [5]. This attack is based on the differential cryptanalysis [16]. The idea of this attack is to inject a difference into the internal state, causing a disturbance, and then to correct it with the next injections. The resulting difference pattern is spread out due to the message schedule causing more disturbances in other rounds. The goal is to have as few disturbances as possible in order to reduce the complexity of the attack [5].

In the related-key scenario it is allowed to inject difference into the key, and not only into the plaintext as in the pure differential cryptanalysis. However the attacker cannot control the key itself and thus the attack should work for any key pair with a given difference.

Local collisions in AES-256 are best understood on a one-round example (fig. 1).

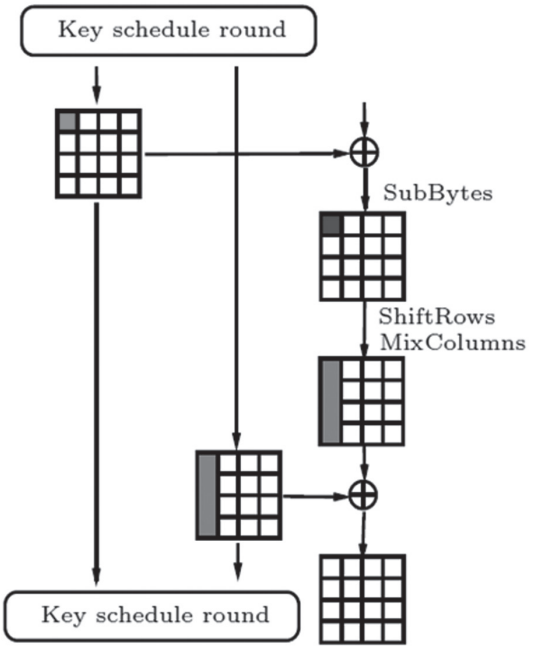

Fig. 1. Local collisions in AES[5]

Here one active S-box is needed (it is a short name for bytes Substitution operation) and five non-zero byte differences in the two subkeys. These five bytes split into two parts: one-byte disturbance and four-byte correction.

Due to the key schedule the differences spread to other rounds. Most of the AES key schedule operations are linear, so a sequence of several consecutive round key values can be viewed as a codeword of a linear code. This is the case, particularly, when a trail does not have active S-boxes in the key schedule.

Let figure out how to build an optimal trail for the key recovery attack. Typically, a trail is better if it has fewer active S-boxes. Disturbance differences

form a codeword, which should have low weight. Simultaneously, correction differences also must form a codeword, and the key schedule codeword is the sum of the disturbance and the correction codewords. In further trails, the correction codeword is constructed from the former one by just shifting four columns to the right and applying the S-box and Mix Columns expansion. Synchronization is simple since the injection is made to the first row, which is not rotated by Shift Rows. Otherwise, the task of synchronizing two codewords would have been much harder and would have lead to high-weight codewords [5].

An example of a good key-schedule pattern for AES-256 is depicted in fig. 2 as a 4.5-round codeword.

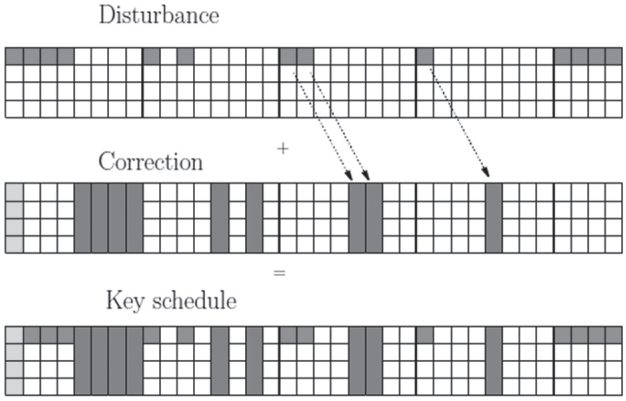

Fig. 2. AES-256 key schedule codeword (4.5 key-schedule rounds)[5]

In the first four key-schedule rounds the disturbance codeword has only 9 active bytes, which is the lower bound. It is needed to avoid active S-boxes in the key schedule as long as possible. Due to a weak diffusion in the AES key schedule the difference affects only one more byte per key schedule round. The correction column should be positioned four columns to the right, and propagates backwards in the same way. The last column in the first round key is active, so all S-boxes of the first round are active as well, which causes unknown difference in the first column. This «alien» difference should be canceled by the plaintext [5].

So such collisions can considerably reduce the complexity of key-recovery attack.

2.2 Current researches on the automatic search for related-key differential characteristics

Automatic search for best differential characteristics was first performed by Matsui [12] for DES. Algorithms for automatic search of differential characteristics for MD4 were presented in [13], and for MD5 in [14]. De Canni_ere and Rechberger in [15] described a method that finds characteristics in SHA-1 in an automatic way.

The first automatic tool for finding related-key differential characteristics was presented by Alex Biryukov and Ivica Nikolic in [11]. They described algorithm which allows to compute differential characteristics for AES, Camellia, Khazad and others. General idea of our searching algorithm is close to the one in [11]. But unlike algorithm from Biryukov and Nikolic which controls each byte of state, in the presented algorithm entire column is controlled for amount of active bytes in it without exact position of these bytes. It is become possible because of the structure of researched encryption algorithm. Such scheme decreases the complexity of computation and reduces influence of high branching mentioned in [11].

3. PERSPECTIVE SPN-BASED BLOCK CIPHER

This section describes all parts of proposed SPN cipher in details because our estimation algorithm is based on it. As was mentioned before this cipher is based on the AES. The main differences compared to AES are the following: pre- and post-whitening using modulo addition, expanded size of MDS matrix for linear transformation, application of several S-boxes optimized with respect to differential, linear and algebraic cryptanalysis, and considerably redesigned key expansion procedure.

3.1 General parameters

This algorithm can be used with different types of input data. The block of input data can be 128, 256 or 512 bits. The length of the key can be also 128, 256 or 512 bits [4, 17].

The state of the cipher can be represented as a matrix. Each element of the matrix is a byte. Matrix consists of N_b columns. Each column consists of 8 bytes. So in total matrix has $8 * N_b$ bytes.

In the next table (table 1) acceptable combinations of different blocks and keys are represented.

Table 1

Acceptable combinations of blocks and keys

Size of block, bits	Supported key size, bits
128 ($N_b = 2$)	128, 256
256 ($N_b = 4$)	256, 512
512 ($N_b = 8$)	512

Amount of encryption rounds depends on the size of block and key. Amount of rounds for different versions of cipher is represented in the table 2.

Table 2

Amount of encryption rounds for different versions of cipher

Size of block, bits	Size of key 128 bits	Size of key 256 bits	Size of key 512 bits
128 ($N_b = 2$)	10	14	-
256 ($N_b = 4$)	-	14	18
512 ($N_b = 8$)	-	-	18

The research deals with the 128-bits version of the cipher, so in the next chapters presented information concerns only this version, even if it is not evidently mentioned.

3.2 Basic transformations

In the presented algorithm four basic transformations are used:

- key addition;
- bytes substitution;

– shift rows;
– mix columns.

These transformations are the base of all high-level structure [4, 17].

3.3 Key expansion scheme

Let K_M be the main key of encryption and $(K_1, K_2, ..., K_m)$ are the round keys which are generated by key expansion scheme.

The cipher has an SPN structure which is based on the AES cipher:

$$Cipher_{K_M} = \prod_{i=1}^{N_r} \theta \circ \gamma \circ \sigma_{K_i}$$

Symbols: σ_{K_i} – round key adding; γ – nonlinear layer (bytes substitution); θ – linear layer (mix columns, shift rows); N_r – amount of rounds in block cipher.

Proposed key-expansion scheme contains two steps:

1. Computing of intermediate value K_t which is based on the master key of encryption K_M and some constant.

2. Computing of round keys $(K_1, K_2, ..., K_m)$ which are based on the master key K_M, intermediate value K_t and some constant.

Intermediate value K_t is computed with the next algorithm (fig. 3):

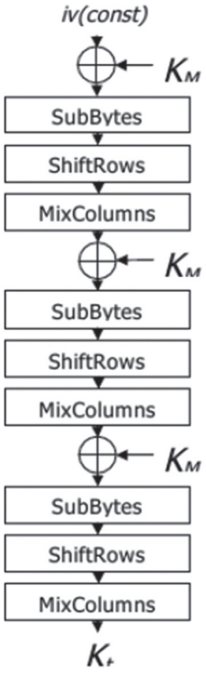

Fig. 3. Computing of intermediate value K_t

As it is shown in figure 3 this algorithm contains all those transformations which are used in the cipher. This reduces implementation complexity because all basic operations are the same.

Computing of intermediate value can be formalized with the following formulas:

$$IM_{K_M} = \prod_{i=1}^{3} \theta \circ \gamma \circ \sigma_{K_M}$$

$$K_t = IM_{K_M}(iv)$$

where iv is some constant which reduces symmetry in the key.

Generation of round keys can be formalized with next expression:

$$RK_{K_t}[K_M] = \sigma_{Kt+tmv_0} \circ \prod_{i=1}^{2} \theta \circ \gamma \circ \sigma_{K_t+tmv_i}$$

where tmv_i are constants which are used for generation round keys. For each round this value should be unique but computation of these constants can be very simple (for example it is a simple shift on a few positions).

General algorithm for round keys generation is represented on the figure 4 [4, 17].

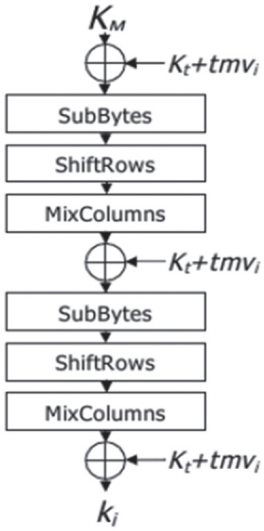

Fig. 4. Computation of round keys

The above scheme (figure 4) is similar to the scheme of computation K_t and also it is similar to the general scheme of the cipher. This issue reduces the complexity of implementation.

According to the [10] such scheme of key expansion has some properties, such as:

1. One-way mapping: having encryption key it is very easy to generate round keys, but having one or more round keys it is computationally very difficult to retrieve encryption key or another round key.

2. Non-linear dependence between each bit of encryption key.

3. Good statistical properties of this key schedule (verified by NIST STS statistical tests).

4. Simple implementation (based on cipher round transformations only), good key agility and possibility to generate round keys in direct and reverse order with the same computational complexity.

3.4 Encryption transformation

Encryption transformation is almost the same as in AES algorithm. General scheme for 128-bits version of cipher is represented in figure 5. Adding keys K_0 and K_{10} are performed by modulo 2^{64}. In other cases simple XOR operation is used [4, 17].

As it has seen this cipher is a classic SPN structure. The same structure is used in key-expansion algorithm so it simplifies implementation.

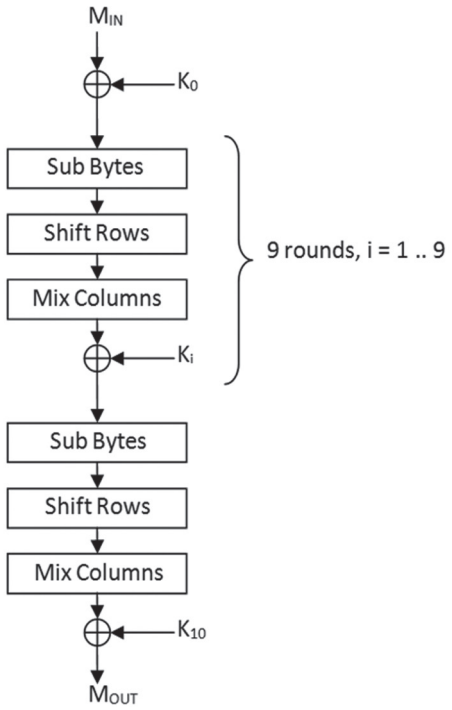

Fig. 5. Main encryption loop for 128 bit block and key cipher

4. THE ALGORITHM FOR ESTIMATION CIPHER SECURITY AGAINST RELATED-KEY ATTACKS

4.1 Proposed method for cipher security estimation

To prove the security of encryption algorithm against such type of attacks it is needed to find the best differential characteristic. The best characteristic is such one that has as few active bytes as possible. So after the best differential characteristic is found the amount of active bytes which were used during its construction should be counted. Active byte is a non-null byte which was passed through the substitution table. As it is known substitution of bytes is a non-linear transformation so the output is undefined and attacker should attack each active substitution transformation (which has complexity of 2^6 operations). If the amount of active bytes exceeds some boundary value the cipher can be considered as safe against key-related attacks because the complexity becomes bigger than the complexity of brute force attack on the cipher. Boundary value depends on the size of cipher block and size of encryption key.

The best differential characteristic can be found by searching between all possible input differences. This can be done automatically with special software. As it is shown later this search can be done in reasonable time for 128-bits version of cipher. To do this a special technique should be used. It is shown in the next chapters.

4.2 The algorithm for counting active bytes

As was mentioned before the best differential characteristic (with the fewest amounts of active bytes) can be found by searching between all possible input differences. For each characteristic the amount of active bytes after key-expansion scheme and after all rounds of encryption should be counted. Proposed algorithm is for 128-bits version of the cipher.

Let examine the next example. It is a differential characteristic for *Kt*-computation. As was mentioned before *Kt* is an intermediate value which is computed in the key-expansion scheme. General scheme for computation of *Kt* is represented in figure 6.

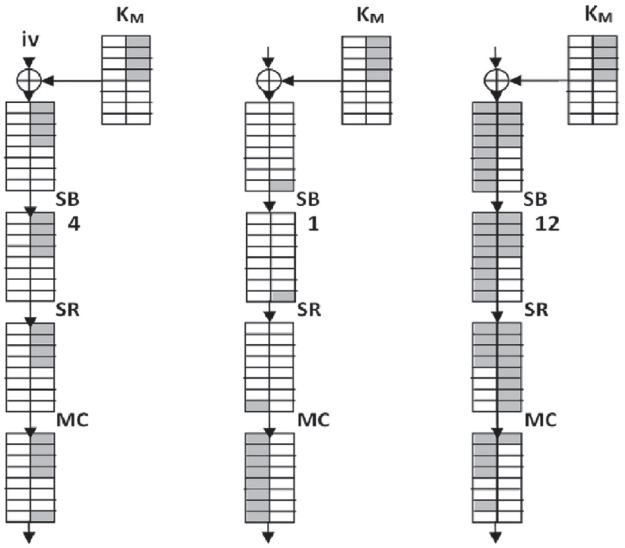

Fig. 6. Differential characteristic for *Kt* computation

Symbols: *iv* – the difference in initial vectors (always equals zero); K_M – master key of encryption (which is needed to be expanded); *SB* – sub-bytes (bytes substitution with substitution tables); *SR* – shift rows; *MC* – mix columns.

In the example (Figure 6), three rounds (from left to right) of *Kt* computation are shown. Input data is a difference of encryption key K_M. Then it is checked how this difference spreads after different transformations. Interested data is the amount of active bytes (such bytes for which difference is not null) on the input of Sub-Bytes transformation. In the example this value equals to 4 for the first round, 1 for the second round and 12 for the third round. In general there are 17 active bytes.

It should be mentioned that in other transformations (which is linear) the decision is based on what is the best for cryptanalyst. For example the difference after Mix Columns has such value that when it is added to the key K_M in the second round collision is occurred.

4.2.1 General description. Simpler algorithm for counting active bytes can be implemented. It this case each column of the state is controlled instead of each byte. It is known that the state of the cipher is represented as columns. Each column consists of 8 bytes. Thus 128-bits version has two columns. In the searching algorithm the amount of active bytes in columns is counted without their exact position. Such algorithm considerably reduces complexity. Accuracy of the results becomes worse because it is possible to find characteristic which does not really exist. Such characteristics are cutting down the acceptable boundary of amount of active bytes.

Thus may be said that a minimal amount of active bytes which is possible to achieve with any differential characteristic was found. This is a very good result.

If the amount of active bytes for investigated algorithm does not exceed theoretical boundary then such cipher is safe against the related-key attacks.

Figure 7 presents an example of differential characteristic for *Kt*-computation scheme. As shown in example the amount of active bytes after each transformation is recalculated. The numbers in figure are amounts of active bytes in columns. This example gives 3 active bytes in each round and 9 active bytes in total.

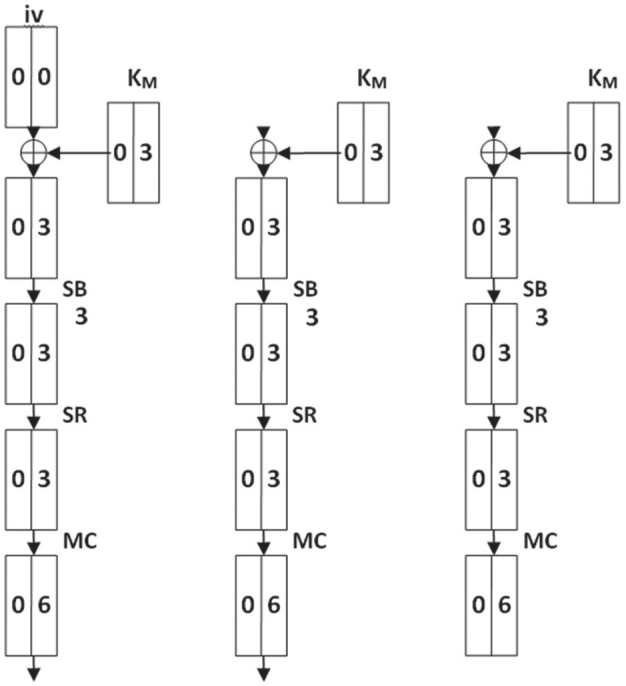

Fig. 7. Computation of active bytes

In order to make it more understandable in the next sections each transformation is described in details.

4.2.2 Key addition. The operation of adding with key is performed with the next rule: the value of one column is substituted from the value of corresponding column from key state. Then an absolute value of this result is taken. Some examples are represented in figure 8.

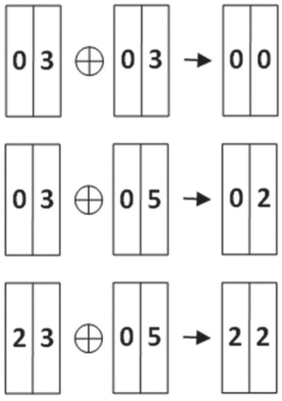

Fig. 8. Key addition

During adding it is assumed that non-zero bytes have such positions and values that collisions is occur. It means that it is the best case for the cryptanalyst. That is why this assumption can lower the resulting amount of active bytes but cannot increase it.

4.2.3 Bytes substitution. During bytes substitution transformation the amount of non-zero bytes does not change. That is why the values in the columns are not changed. But this transformation is very important because the security of the cipher depends on amount of non-zero bytes which are passed through the substitution tables. Such bytes become active bytes. The more active bytes the better cipher.

4.2.4 Shift rows. The operation of shifting bytes in 128-bits version of the cipher is performed as in figure 9:

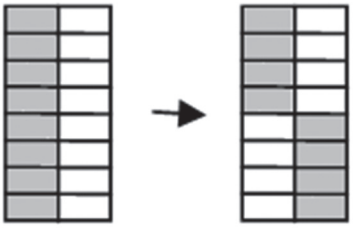

Fig. 9. Shift rows

Figure 9 shows that last 4 bytes of left and right columns change places. In algorithm only amount of non-zero bytes in column is controlled but the actual places of these bytes are undefined. Challenging problem here is how the shift should be performed and what amount of bytes should be shifted from one column to another. This problem was solved in such way: all possible variants of shifting were taking into account so the differential characteristic can get additional branches which are independent between each other. This part is the hardest part in the algorithm. It takes a lot of computation because on each round new branches appear and amount of these independent branches is growing exponentially depends on the amount of rounds.

Example of this transformation is represented in figure 10.

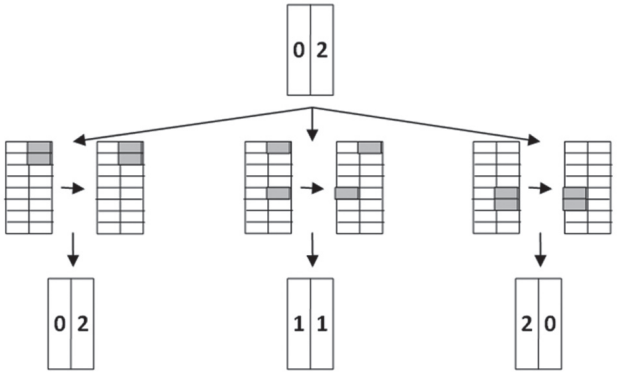

Fig. 10. Shift rows

Above example (Figure 10) shows that the state (0; 2) has three independent branches for continuation and for finding best differential characteristic, all this branches should be computed.

Table 3 represents amount of different alternatives for shifting in the column depends on amount of non-zero bytes.

Table 3

Amount of branches depends on amount of non-zero bytes in column

Amount of non-zero bytes in column	Amount of branches
0	1
1	2
2	3
3	4
4	5
5	4
6	3
7	2
8	1

Total amount of branches after shift rows transformation is equal to the product of possible shifts of the left and right columns. For example if left column has 3 non-zero bytes and right column has 4 non-zero bytes then the amount of branches for this transformation is 4*5 = 20.

4.2.5 Mix columns. Mixing in columns is performed as the product of column and some fixed matrix. The main property of this transformation is a fact that the sum of non-zero bytes on the input and output cannot be less than 9:

$$N_{in} + N_{out} \geq 9.$$

Exception is the situation when column has no active bytes, then on the output there is also no active bytes.

In the developed algorithm it is assumed that the best case for the cryptanalyst is taken. It means that there is always 9:

$$N_{in} + N_{out} = 9.$$

Some examples of how Mix Columns works are represented in figure 11.

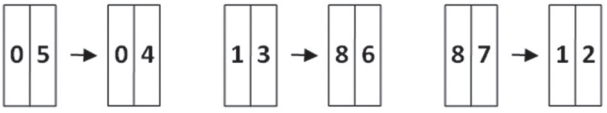

Fig. 11. Mix columns

This transformation does not create new branches.

4.3 Description of results

The minimal safe threshold for estimated cipher is 26 active bytes. This value is derived from the next equation:

$$\left(2^{-5}\right)^{x} < 2^{-128},$$

where 2^{-128} – probability of breaking the cipher with brute force attack; 2^{-5} – probability of breaking one substitution box of perspective cipher with brute force attack.

So if there are 26 active bytes it means that the entire complexity of breaking the cipher with such differential characteristic is 2^{-130} which is more than the complexity of brute force attack. 25 active bytes gives the complexity less than 2^{-128}. So the minimal safe threshold is 26 active bytes.

As a result of experiments the best differential characteristic for 128-bits version of cipher was found. This characteristic has 27 active bytes. The results for each round are represented in the table 4.

Table 4

Description of best differential characteristic

Part of cipher	Round	Amount of active bytes	Accumulated amount of active bytes
Kt computation	1	7	7
	2	4	11
	3	2	13
Key expansion scheme	1	2	15
	2	2	17
Main encryption loop	1	1	18
	2	1	19
	3	1	20
	4	1	21
	5	1	22
	6	1	23
	7	1	24
	8	1	25
	9	1	26
	10	1	27
TOTAL			27

Table 4 shows that the differential characteristic in the main encryption loop is iterative and 1 active byte is added on each round. Detailed characteristic is represented in the next figures.

Symbols: K – round key. For different stages of cipher this value is different; SR – Shift Rows; MC – Mix Columns.

The stage of computing *Kt* is represented in figure 12. In this figure first 3 rounds are represented. As was mentioned before intermediate value *Kt* is needed for key expansion scheme.

Initially the state of transformation is equal to (0;0) and key is equal to (1;6). This key is the master key which is chosen. To be more precisely these values are the differences between keys or states.

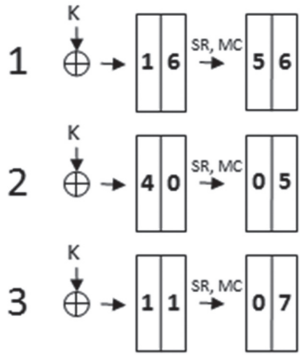

Fig. 12. *Kt* computation

In figure 13 two round of key-expansion scheme are represented. Initially the state of transformation is

equal to (0;7). This value is output value of previous transformation. The difference of the key is equal to (1;6). This is the master key.

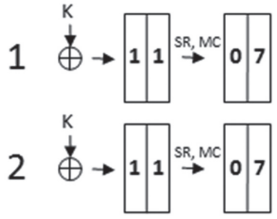

Fig. 13. Round key computation

In figure 14 ten rounds of the main encryption loop are represented. On this stage input states can be chosen so in this differential characteristic the input state (0;6) is taken. Round key is equal to (0;7). This is the output value of previous stage. The difference (0;7) is the same for all round keys because all of them have the same difference. So the values on each round also are equal

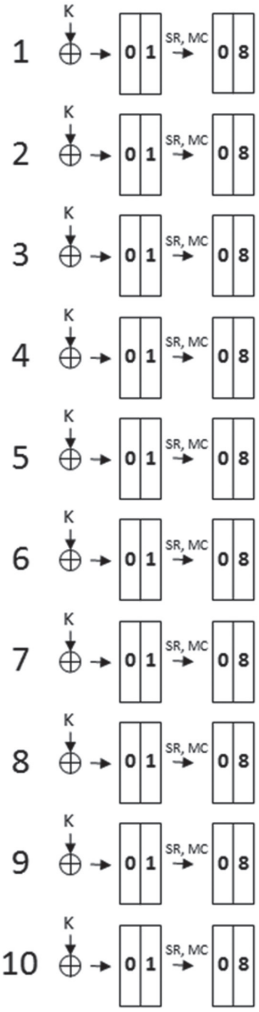

Fig. 14. Main encryption loop

The result is a differential characteristic which has 27 active bytes for 128-bits version of cipher. The theoretical threshold value for this version of cipher is equal to 26 active bytes. But as was mentioned before the obtained value is the minimum amount of active bytes which can be achieved. Practically the amount of active bytes in best differential characteristic can be higher but it cannot be lower. This is the main property of algorithm. This issue is connected with the implementation of algorithm. This implementation allows to find characteristics which are impossible in practice. But still this algorithm can show the security of investigated cipher. For example if the amount of active bytes is lower than the threshold it is possible to find differential characteristic with such amount of active bytes which is potentially can lead to vulnerabilities.

4.4 Complexity of algorithm

The input values of algorithm are the different values of differences of the master key on the input of key-expansion scheme (it is a stage of *Kt* computation). The algorithm was build in such way that it controls 2 columns. Each column can have values between 0 to 9 (amount of active bytes). This means that each column have 9 different variants of input values. 128-bits version of cipher has 2 columns. The values of each column can be picked independently so the total amount of input values is 81 (formula 5).

$$N_{MK} = N_{Col}{}^{c} = 9^2 = 81 \qquad (1)$$

Symbols: N_{MK} – amount of differences for master key; N_{Col} – amount of states for one column; c – amount of columns in the cipher state.

Also different open texts can be chosen on the input of main loop of encryption. Amount of different open texts are the same as amount of keys. It can be calculated with the next formula:

$$N_{PT} = N_{Col}{}^{c} = 9^2 = 81 \qquad (2)$$

N_{PT} – amount of possible differences for open texts.

Also as mentioned before the Shift Rows transformation can create new branches. It means that additional paths for searching differential characteristic appear. It is connected with different possibilities of shifts. Algorithm does not control each byte separately but it controls amount of active bytes in columns. That is why different variants of shifts appear because the positions on which the active bytes are placed are unknown. Amount of such variants depends on amount of active bytes in column. Table 5 represents amount of branches which are created after Shift Rows for one column.

Table 5

Amount of branches depends on the amount of active bytes in column

Amount of active bytes in column	Amount of branches
0	1
1	2
2	3
3	4
4	5
5	4
6	3
7	2
8	1

To find total amount of branches after Shift Rows transformation it is needed to multiply amount of branches for each column.

$$N_{DC} = \prod_{i=0}^{c} N_i \quad (3)$$

N_{DC} – amount of branches after Shift Rows transformation; c – amount of columns in the cipher state; N_i – amount of branches for each column (according to table 3).

For example if current state of cipher is (3;4) then after Shift Rows there are 4*5 = 20 independent branches for searching differential characteristic.

The algorithm is searching among all possible input text differences. Each column on the average makes 3 different branches. It means that there are 9 different branches for the whole transformation for 128-bits version of cipher.

This transformation is the most complex one because it is performed in each round so there is an exponential growth of complexity. Formula 4 demonstrates amount of branches depends on the amount of rounds.

$$N_{DC} = \prod_{i=0}^{r} \prod_{j=0}^{c} N_{ij} \quad (4)$$

N_{DC} – total amount of branches; c – amount of columns in the cipher state; r – amount of rounds; N_{ij} – amount of branches for each column for one round (according to table 5).

The complexity of algorithm can be reduced with filtering bad characteristics. Bad characteristic is a characteristic which exceeds some threshold value of active bytes. So when it is found that differential characteristic reaches some threshold value it is dropped. It considerably simplifies the computation.

General formula for counting is represented below:

$$O = N_{PT} N_{MK} N_{DC} = \\ = N_{Col}{}^{c} N_{Col}{}^{c} \prod_{i=0}^{r} \prod_{j=0}^{c} N_{ij} = (N_{Col}{}^{c})^2 \prod_{i=0}^{r} \prod_{j=0}^{c} N_{ij} \quad (5)$$

Using formula 5 the total complexity for 128-bits version of cipher for all 10 rounds can be computed. It is expected that Shift Rows transformation creates 9 branches on each round.

In general there are 15 rounds (5 rounds of key-expansion scheme and 10 rounds of main loop of encryption).

The calculation for the 128-bits version of cipher is represented below:

$$O_{128} = N_{PT} N_{MK} N_{DC} = 9^2 9^2 9^{15} = 3^{57} \approx 2^{90}.$$

The complexity is very big but this complexity can be considerably reduced with filtering differential characteristic which exceeds some threshold. The minimal value for 128-bits version of cipher is 27 — this is minimal amount of active bytes which is allowed. So there are more than 27 active bytes in differential characteristic it is dropped.

On quad core system with AMD Phenom 3.2 Hz processor this algorithm takes a time about 10 minutes.

CONCLUSIONS

Method for security estimation of perspective SPN-based block cipher against related-key attacks are presented in this paper. The new cipher is based on the AES construction with redesigned key schedule for providing protection against such type of attacks. The cipher was proposed to the public competition of block cipher selection to be a prototype during development of Ukrainian National Standard [4,17].

Proposed method is based on counting amount of active bytes in differential characteristics. But unlike algorithm from Biryukov and Nikolic [11] which controls each byte of state, in the presented algorithm entire column is controlled for amount of active bytes in it without exact position of these bytes. Such scheme decreases the complexity of computation and reduces influence of high branching mentioned in [11].

Developed method can show existence of such differential characteristic which can lead to the key recovery with complexity lower then complexity of brute force attack.

Also the calculation of the complexity for the developed method is presented. This method with direct application has large computational complexity which is about $O = 2^{90}$ for 128-bits version of cipher. For modern computers such complexity is too big so some optimizations were performed. These optimizations are based on the dynamic elimination of differential characteristics which reach some threshold value of active bytes. Such optimization allows considerably reducing complexity. Optimized algorithm which is implemented in C++ language takes about 10 minutes for 128-bits version of cipher.

So proposed method of estimation gives the analytic proof of security to related key attacks. It was shown that 128-bits version of cipher does not have such differential characteristic which can lead to the key recovery with complexity which is low then complexity of brute force attack.

This method can be extended to different variants of block size and key lengths of proposed ciphers as well as to other block ciphers with similar structure.

References

[1] Schneier Bruce «Applied Cryptography: Protocols, Algorithms, and Source», Code in C, Second Edition, John Wiley & Sons, 1994

[2] Stallings William «Cryptography and Network Security: Principles and Practice», Prentice Hall, 2006

[3] Federal Information Processing Standards Publication 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES), November 26, 2001

[4] R.V. Oliynykov, I.D. Gorbenko, V.I. Dolgov, V.I. Ruzhentsev. Perspective symmetric block cipher "Kalina" – basic terms and specification. Applied radioelectronics. Special issue which is devoted to the problems of information security. Kharkiv. Volume 6, № 2, 2007 // in Ukrainian

[5] Alex Biryukov and Dmitry Khovratovich «Related-key Cryptanalysis of the Full AES-192 and AES-256», University of Luxembourg 29 May 2009, http://impic.org/papers/Aes-192-256.pdf / (visited 25.05.2014)

[6] Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, Adi Shamir «Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds», http://eprint.iacr.org/2009/374.pdf / (visited 25.05.2014)

[7] Alex Biryukov, Dmitry Khovratovich, Ivica Nikoli «Distinguisher and Related-Key Attack on the Full AES-256», University of Luxembourg, 10 August 2009, http://www.iacr.org/archive/crypto2009/56770229/56770229.pdf / (visited 25.05.2014)

[8] I.D. Gorbenko «Information security in information and telecommunication systems» / Textbook. Part 1. Cryptographic protection of information – Kharkiv KNURE, 2004. – 368 p. // In Ukrainian

[9] GOST 28147-89: Information processing systems. Cryptographic protection. Cryptographic transformation algorithm. // in Russian / http://protect.gost.ru/document.aspx?control=7&id=139177 / (visited 25.05.2014)

[10] R.V. Oliynykov, V.I. Ruzhentsev. A new approach of key schedule construction for symmetric block ciphers. Proceedings of the SFU. Engineering. "Information Security." – Russia, Taganrog: Publisher TTISFU (Taganrog Technological Institute of Southern Federal University), 2010. № 11 (112), p.156–161.

[11] Alex Biryukov, Ivica Nikoli «Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others»

[12] M. Matsui. On correlation between the order of S-boxes and the strength of DES. In A. D. Santis, editor, EUROCRYPT, volume 950 of Lecture Notes in Computer Science, pages 366{375. Springer, 1994.

[13] P.-A. Fouque, G. Leurent, and P. Nguyen. Automatic search of differential path in MD4. Cryptology ePrint Archive, Report 2007/206.

[14] M. Stevens. Fast collision attack on MD5. Cryptology ePrint Archive, Report 2006/104.

[15] C. D. Canniere and C. Rechberger. Finding SHA-1 characteristics: General results and applications. In X. Lai and K. Chen, editors, ASIACRYPT, volume 4284 of Lecture Notes in Computer Science, pages 1-20. Springer, 2006.

[16] Howard M. Heys. A tutorial on linear and differential cryptanalysis. Cryptologia, Volume 26, Issue 3, July 2002, Pages 189–221.

[17] Oliynykov, R., Gorbenko, I., Dolgov, V., & Ruzhentsev, V. (2010). Results of Ukrainian national public cryptographic competition. Tatra Mountains Mathematical Publications, 47(1), 99–113.

Поступила в редколлегию 10.06.2014

Олейников Роман Васильевич, доктор технических наук, профессор кафедры БИТ ХНУРЭ. Область научных интересов: анализ и синтез симметричных криптографических преобразований, безопасность программного обеспечения.

Кайдалов Дмитрий Сергеевич, аспирант кафедры БИТ ХНУРЭ. Область научных интересов: анализ стойкости блочных симметричных шифров.

УДК 621.391:519.2:519.7

Криптоанализ перспективного симметричного блочного шифра с использованием связанных ключей / Р.В. Олейников, Д.С. Кайдалов // Прикладная радиоэлектроника: научн.-техн. журнал. — 2014. — Том 13. — № 3. — С. 192–200.

Симметричные блочные шифры являются одними из наиболее распространенных криптографических примитивов. Кроме обеспечения конфиденциальности через шифрование, блочные шифры применяются как базовые компоненты при построении хэш-функций, кодов аутентификации сообщений, генераторов псевдослучайных чисел в составе различных криптографических протоколов и др. Одним из распространенных шифров в настоящее время является AES (Advanced Encryption Standard), используемый как стандарт симметричного шифрования во многих странах во всем мире. Несколько лет назад была предложена теоретическая атака против схемы разворачивания ключей AES, и сложность этой атаки оказалась значительно меньше по сравнению с полным перебором ключей. В статье рассмотрен метод оценки стойкости алгоритма шифрования к атаке на связанных ключах, и его применение для перспективного блочного шифра-кандидата на стандарт блочного шифрования в Украине.

Ключевые слова: блочный шифр, криптоанализ, атака на связанных ключах.

Табл.: 5. Ил.: 14. Библиогр.: 17 назв.

УДК 621.391:519.2:519.7

Криптоаналіз перспективного симетричного блокового шифру із застосуванням зв'язаних ключів / Р.В. Олійников, Д.С. Кайдалов // Прикладна радіоелектроніка: наук.-техн. журнал. — 2014. — Том 13. — № 3. — С. 192–200.

Симетричні блокові шифри є одними з найпоширеніших криптографічних примітивів. Крім забезпечення конфіденційності через шифрування, блокові шифри застосовуються як базові компоненти в ході побудови геш-функцій, кодів автентифікації повідомлення, генераторів псевдовипадкових послідовностей, у складі різних криптографічних протоколів та ін. Одним із поширених шифрів зараз є AES (Advanced Encryption Standard), що використовується як стандарт у багатьох країнах світу. Декілька років тому була запропонована теоретична атака проти схеми розгортання ключів AES, і складність цієї атаки виявилася значно меншою порівняно із повним перебором ключів. У статті розглянуто метод оцінки складності алгоритму шифрування до атаки на зв'язаних ключах, і його застосування для перспективного блокового шифру-кандидата на стандарт блокового шифрування в Україні.

Ключові слова: блоковий шифр, криптоаналіз, атака на зв'язаних ключах.

Табл.: 5. Іл.: 14. Бібліогр.: 17 найм.