How secure is the Advanced Encryption Standard with random ShiftRows against Fault Analysis

Adrián Alfonso Peñate, Pablo Freyre Arrozarena

***Abstract*— With the approve of the cryptographic algorithm Rijndael as the AES (Advanced Encryption Standard) and a collection of works exists with the purpose that one or several of its internal functions depend on the selected key. In this work we will study a fault analysis model against the algorithm AES, and then we will analyze for dynamic (in the key-dependency sense) cryptographic algorithms based on AES, in the which ones the internal function ShiftRows is randomly selected in every round, how strong is this attack.**

***Tóm tắt*— Với việc thuật toán mã hóa Rijndael được chấp nhận là Tiêu chuẩn mã hóa nâng cao (Advanced Encryption Standard - AES), và một loạt các công trình nghiên cứu về một hoặc một số chức năng bên trong của thuật toán phụ thuộc vào khóa đã được công bố. Bài báo này trình bày về nghiên cứu mô hình phân tích lỗi dựa trên thuật toán AES và phân tích các thuật toán mã hóa động (theo nghĩa phụ thuộc khóa) dựa trên AES. Trong đó, các hàm nội bộ ShiftRows được chọn ngẫu nhiên trong mỗi vòng, để đánh giá mức độ nguy hiểm của cuộc tấn công.**

***Keywords*— ShiftRows; Rijndael; Fault Analysis.**

***Từ khóa*— ShiftRows; thuật toán Rijndael; tấn công phân tích lỗi.**

I. INTRODUCTION

Rijndael is a cryptographic algorithm submitted to the AES competition, launched in 1997 by the National Institute of Standards and Technology for select one block cipher algorithm that processes sensible but not classified information, and announced as the winner in 2001 [13]. This algorithm has a simple mathematical structure and guarantee resistance against differential and linear cryptanalysis [8]. At the same time, Rijndael resists other classic attacks arising to exploit its algebraic structure. There is not yet a practical way to cause a real damage to their security by such attacks [19], and for this reason is that the European Union Agency for Network and Information Security recommends the algorithm Rijndael to be used in the future [12].

With some specifications (128 bits of input/output block and 128, 192, 256 bits of secret key) Rijndael is adopted as the Advanced Encryption Standard and is known as the algorithm AES. It has become in a very popular algorithm and one of the current trends in cryptography has been make it dynamic, in the sense that one or several of its internal functions depend on the secret key. According to [24] the classic attacks are more difficult to apply over this kind of ciphers if the S−boxes are selected at random and key-dependent. One cryptographic algorithm should be more secure if the relationships between the plain text and the cipher text are unknown, examples of dynamic block cipher algorithms like Rijndael can be found in the literature.

On the other hand, it has been possible to appreciate that the so-called side channel attacks can break the strength of AES, and in this respect, solutions are sought. In [5] the authors express that to replace the constants in the algorithm Rijndael can be a solution against fault analysis and power analysis, and the same idea is defended in other works. Our work is directed to answer the question: How secure is the Advanced Encryption Standard with random ShiftRows against Fault Analysis? We will assume the replace of the transformation ShiftRows in every round of the algorithm AES and we will see if this change is or not a

This manuscript is received on July 18, 2018. It is commented on November 20, 2018 and is accepted on November 30, 2018 by the first reviewer. It is commented on December 16, 2018 and is accepted on December 26, 2018 by the second reviewer.

solution against the Differential Fault Analysis described in [4], proposed for the authors like the optimal model for this attack.

In section 2 we will give a short description of AES and its internal function ShiftRows. In section 3 we will comment a brief state of the art of the dynamic variants of AES with random ShiftRows, and we will explain how the transformation ShiftRows can be replaced in all the possible ways. In section 4 we will analyze the Differential Fault Analysis of AES in the fault model of [4], and we will give a new complexity of this attack for any dynamic variant of AES with random ShiftRows in every round of the encryption process.

II. SHORT DESCRIPTION OF AES

AES is a block cipher algorithm for encrypt input/output block of 128 bits, arranged as a matrix of bytes calling state, with 4 rows and 4 columns through the bijective application

$$t : M_{4\times4}(\mathrm{GF}(2^8)) \rightarrow \mathrm{GF}(2^8)^{4\times4}$$

such that for every $1 \leq i, j \leq 4$ the next relationship is hold

$$t(state)_{4(j-1)+i} = state_{i,j}$$

Here $\mathrm{GF}(2^8)$ is the finite Galois field of 256 elements and $M_{4\times4}(\mathrm{GF}(2^8))$ is the set of all the matrices of 4 rows and 4 columns. In next we will use the fact, that for every natural numbers k and n the map

$$t : M_{k\times n}(\mathrm{GF}(2^8)) \rightarrow \mathrm{GF}(2^8)^{k\times n}$$

is a bijective application.

The secret key is another block of length $4N_k$ bytes (N_k = 4,6,8), equally arranged as a matrix with 4 rows and N_k columns. The state is transformed through $N_r = 6+N_k$ rounds providing a good security margin and the key is expanded to $N_r + 1$ round keys of the same length that the state matrix, by one algorithm usually called key schedule. During the encryption process of AES the four transformations:

SubBytes: Apply to every byte of the state a bijective S−box;

ShiftRows: Apply to every row of the state a cyclic displacement;

MixColumns: Apply to the state a MDS matrix multiplication;

AddRoundKey: Apply to every byte of the state a XOR with the key

are applied on the state in the order of the following algorithm. Since MixColumns is not considered in the last round the decryption process is performance similarly keeping in mind the inverses of the previous transformations.

Input: input block P and secret key K

1. begin
2. $state = t^{-1}(P)$
3. $state = AddRoundKey(state, RoundKey[0])$
4. for i from 1 to $N_r - 1$ do
5. $state = SubBytes(state)$
6. $state = ShiftRows(state)$
7. $state = MixColumns(state)$
8. $state = AddRoundKey(state, RoundKey[i])$
9. end for
10. $state = SubBytes(state)$
11. $state = ShiftRows(state)$
12. $state = AddRoundKey(state, RoundKey[N_r])$
13. $C = t(state)$
14. end

Output: output block C

The transformation ShiftRows

The transformation ShiftRows is applied on the state displacing the row i cyclically $i - 1$ positions to left, $1 \leq i \leq 4$. It is possible to see that

$$\mathrm{ShiftRows(state)} = t^{-1}(\Pi(t(\mathrm{state}))) \quad (1)$$

where Π is a permutation of 16 elements defined for every $1 \leq i, j \leq 4$ as

$$\Pi[4(j-1)+i] = 4(j - i \bmod N_b) + i$$

ShiftRows offer high dispersion of the bytes in the state, in the sense that, in each column of the state after ShiftRows all the bytes belong to different columns of the state before ShiftRows. The matrix of the index position of the bytes in the state are transformed by ShiftRows as following:

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

→

1	5	9	13
6	10	14	2
11	15	3	7
16	4	8	12

and then we are able to see the permutation Π of the equation 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	6	11	16	5	10	15	4	9	14	3	8	13	2	7	12

As we know, the transformation MixColumns is applied on the state post-multiplying every column of the same one for a fixed MDS matrix, and offer local full diffusion, in the sense that, in each column of the state after MixColumns all the bits depends of every bit of the same column of the state before MixColumns. The round function

AddRoundKey(MixColumns(ShiftRows(SubBytes(state))))

provide full diffusion in 2 rounds of AES, although the transformations SubBytes and AddRoundKey do not provide diffusion at all.

III. DYNAMIC AES WITH RANDOM SHIFTROWS

In this section we are considering that the algorithm AES is only modified replacing the transformation ShiftRows by another function in random way. There are in the literature some propositions for make practical this change [17, 16, 1, 21], but the most common criteria is to realize random displacements of the rows in the state. If these displacements are designed to be different in every row, like in AES, then there are only 24 ways to replace ShiftRows, otherwise the full diffusion that AES provide is never obtained and the amount of all the possibilities is still too small.

One permutation Φ that act over the state through the equation 1 has been defined in [8, Definition 9.4.1] like a diffusion optimal permutation, if all bytes in each column are distributed over all columns under the action of Φ. If one permutation Φ is diffusion optimal, then ShiftRows can be replaced by a similar function substituting Π for Φ in the equation 1. For this reason the amount of all the possibles variations of AES, changing only the transformation ShiftRows, is the amount of all the diffusion optimal permutations of size 16.

The shiftrows Construction

In a previous work [2] the amount of all the diffusion optimal permutations that acts over one state of k rows and k columns is calculated, and for the case of AES this number is 24^8. The practical way to transform the state, such that all bytes in each column are distributed over all columns, is presented next

1. Let $M_1 = t^{-1}(\Pi_1(t(state)))$ be the matrix that result of change the positions of the bytes inside every column of the state using a random permutation. The state is a matrix of k rows and k columns.
2. Let $M_2 = t^{-1}(\mathrm{Tr}(t(M_1)))$ be the matrix that result of transpose M_1, then it is easy to see that $M_2 = t^{-1}(\Pi_1(\mathrm{Tr}(t(state))))$.
3. Let $M_3 = t^{-1}(\Pi_1(t(M_2)))$ be the matrix that result of change the positions of the bytes inside every column of M_2 using a random permutation, then it is easy to see that $M_3 = t^{-1}(\Pi_1(\mathrm{Tr}(\Pi_2(t(state)))))$ and the permutation $\Pi = \Pi_2 o \mathrm{Tr} o \Pi_1$ is diffusion optimal.

Demonstration 1 *Let* $\varphi : M_{k\times k}(\mathrm{GF}(2^8)) \times S_k^{2k} \rightarrow M_{k\times k}(\mathrm{GF}(2^8))$ *be the function defined for all* $\tau = (\tau_1,...,\tau_{2k})$ *and for all state* $\in M_{k\times k}$ *as* $\varphi(state,\tau) = state_\tau$ *where* τ_i *is a random permutation of* S_k *for all* $1 \le i \le 2k$ *and* $state_\tau$ *is the resultant matrix of the proposed construction, using* $\tau_1,...,\tau_k$ *to form* M_1 *and* $\tau_{k+1},...,\tau_{2k}$ *to form* M_3*. We must see that for every fixed state the function* φ *already defined is an injective application.*

Let us consider τ *and* γ *two elements of* S_k^{2k} *such that* $\tau \neq \gamma$*, then exists at less* $1 \le j \le 2k$ *for the which one* $\tau_j[i] \neq \gamma_j[i]$ *for some* $1 \le i \le k$*. Note that only two things can happen in the way that the state is transformed.*

If $1 \le j \le k$ *the element i of the column j of the state is located in a different column of* $state_\tau$ *with respect to* $state_\gamma$*, and therefore both matrices are different. On the other hand if* $\tau_1 = \gamma_1,...,\tau_k = \gamma_k$ *happens, the element i of the column j of* $state_\tau$ *will be different from the element i of the column j of* $state_\gamma$*, and therefore both matrices are different.*

The algorithm to construct diffusion optimal permutations, as result of the proposed construction and for replace ShiftRows in the equation 1 is presented next.

Input: random permutations $\tau_1,\tau_2,...,\tau_{2k} \in S_k$

1. begin
2. for $j = 1 \cdots k$ do
3. for $i = 1 \cdots k$ do
4. $\Pi_1[k(j-1)+i] = k(j-1) + \tau_j[i]$
5. $\mathrm{Tr}[k(j-1)+i] = k(i-1) + j$
6. $\Pi_2[k(j-1)+i] = k(j-1) + \tau_{k+j}[i]$
7. end for
8. end for
9. for $j = 1 \cdots k$ do
10. for $i = 1 \cdots k$ do
11. $\Pi[k(j-1)+i] = \Pi_1[\mathrm{Tr}[\Pi_2[k(j-1)+i]]]$
12. end for
13. end for
14. end

Output: random diffusion optimal permutation Π

In [3] a dynamic variant of AES is proposed, in which the function ShiftRows is replaced for another transformation that acts randomly over the state in every round of the encryption process. For this variant the permutation Π of the previous algorithm is generated in terms of three round keys, and it is applied over the estate by means of the equation 1. A similar construction is proposed in [25] and the objective in this work is to protect AES against Differential Power Analysis. The authors express that the diffusion optimality can be reached if we transpose the state, then we apply random permutations in every row and later we apply random permutations in every column.

IV. DIFFERENTIAL FAULT ANALYSIS AGAINST AES

The cryptanalytic attacks based in fault injections exploit the leak of one computational error induced into the cryptographic device to get some information about the secret key. The fault attacks constitute a class of side channel attacks and the same ones can be applied to symmetric cryptosystems. Biham and Shamir introduce Differential Fault Analysis (abbreviated DFA) of secret key cryptosystems in [6]. In this work the authors expresses that both, the secret key and the $S-$box, can be recovered in ciphers such as Khufu, Khafre and Blowfish where the $S-$box is computed in terms of the secret key.

It is known that the algorithm AES is susceptible against DFA (see [18]) even when the attacker need access to the encrypting device. The fault injections are today a powerful tool against AES, and the study of new variants of fault attacks is strongly evidenced in the most recent literature [9, 10, 14].

In [4] the DFA is applied against the algorithm AES, and it is assumed that an attacker has the ability to induce a single byte random fault in any chosen point during the computation of AES. In these work, the model used corresponds with the fault injected in the round eight of the encryption process and has also been analyzed in [22, 20, 23]. This scenario is shown as optimal for the recovery of the secret key, on the contrary of [7, 11, 15] where the injection of the fault is done in the round nine of the encryption process.

We only explain this attack when the fault is induced in the state for AES-128. The basic idea behind this model is to obtain a pair of cipher text with error and cipher text without error corresponding to the same plain text and the same secret key, and then following the differential characteristic, some equations related to the input-output of the $S-$box are formed and the key space is reduced.

A. The Explanation of the Attack

As we said in subsection II.1 the full diffusion of AES is reached in two rounds, i.e. every bit of the state depends of all the bits of the state two rounds ago. From this property the complete cipher text depends of one error induced into the state at round eight of AES-128, since MixColumns do not operate in round ten.

The next we keep in mind is that one difference β at the input of the $S-$box is

translated to one difference α at the output of the $S-$box through the differential equation

$$\alpha = S(X \oplus \beta) \oplus S(X)$$

and find the relationship between α and β, or the same thing X, is not a trivial task. Differential Fault Analysis, and the particular attack proposed in [4], is directed to solve these differential equations. For MixColumns and ShiftRows is easy to see that $\alpha = MixColumns(\beta)$ and $\alpha = ShiftRows(\beta)$ respectively, and for the function AddRoundKey is easy to see that $\alpha = \beta$.

To illustrate this attack we assume that one byte difference or error, without loss generality the first,

ϵ			

is induced in the state before the operation MixColumns of the round eight in the encryption process, then we will see how it spreads until the cipher text. After the operation MixColumns of the round eight we have the difference

02ϵ			
ϵ			
ϵ			
03ϵ			

the which one, by the complexity of the difference across the $S-$box, after the operation SubBytes of the round nine is turned in the new difference

ϵ_1			
ϵ_2			
ϵ_3			
ϵ_4			

where the values ϵ_1, ϵ_2, ϵ_3 and ϵ_4 are unknown. Then after the operation ShiftRows of the round nine we have the difference

ϵ_1			
			ϵ_2
		ϵ_3	
	ϵ_4		

the which one after the operation MixColumns of the round nine is seen as

$02\epsilon_1$	ϵ_4	ϵ_3	$03\epsilon_2$
ϵ_1	ϵ_4	$03\epsilon_3$	$02\epsilon_2$
ϵ_1	$03\epsilon_4$	$02\epsilon_3$	ϵ_2
$03\epsilon_1$	$02\epsilon_4$	ϵ_3	ϵ_2

and for this reason after the operation SubBytes of the round ten it is turned in the new difference

ϵ'_1	ϵ'_5	ϵ'_9	ϵ'_{13}
ϵ'_2	ϵ'_6	ϵ'_{10}	ϵ'_{14}
ϵ'_3	ϵ'_7	ϵ'_{11}	ϵ'_{15}
ϵ'_4	ϵ'_8	ϵ'_{12}	ϵ'_{16}

the which one only differs of the cipher text difference in the positions of the values ϵ'_i, for all $1 \leq i \leq 16$ after the operation ShiftRows of the round ten, so, the respective cipher text with error and cipher text without error differs only in the difference

ϵ'_1	ϵ'_5	ϵ'_9	ϵ'_{13}
ϵ'_6	ϵ'_{10}	ϵ'_{14}	ϵ'_2
ϵ'_{11}	ϵ'_{15}	ϵ'_3	ϵ'_7
ϵ'_{16}	ϵ'_4	ϵ'_8	ϵ'_{12}

If C and C' are the cipher text without error and the cipher text with error respectively, and X is the input in the round ten of the encryption process, then is obvious that

$$C' = C \oplus ShiftRows(\epsilon') \qquad (2)$$

where ϵ' is the difference obtained after the operation SubBytes of the round ten. Starting from this point, since an attacker knows C and C', it is possible to assume known the bytes differences ϵ'_i, for $1 \leq i \leq 16$, and then solve the following four equations systems

$$\begin{cases} \epsilon'_1 = SubBytes(X_1) \oplus SubBytes(X_1 \oplus 02\epsilon_1) \\ \epsilon'_2 = SubBytes(X_2) \oplus SubBytes(X_2 \oplus \epsilon_1) \\ \epsilon'_3 = SubBytes(X_3) \oplus SubBytes(X_3 \oplus \epsilon_1) \\ \epsilon'_4 = SubBytes(X_4) \oplus SubBytes(X_4 \oplus 03\epsilon_1) \end{cases}$$

$$\begin{cases} \epsilon'_5 = SubBytes(X_5) \oplus SubBytes(X_5 \oplus \epsilon_4) \\ \epsilon'_6 = SubBytes(X_6) \oplus SubBytes(X_6 \oplus \epsilon_4) \\ \epsilon'_7 = SubBytes(X_7) \oplus SubBytes(X_7 \oplus 03\epsilon_4) \\ \epsilon'_8 = SubBytes(X_8) \oplus SubBytes(X_8 \oplus 02\epsilon_4) \end{cases}$$

$$\begin{cases} \epsilon'_9 = SubBytes(X_9) \oplus SubBytes(X_9 \oplus \epsilon_3) \\ \epsilon'_{10} = SubBytes(X_{10}) \oplus SubBytes(X_{10} \oplus 03\epsilon_3) \\ \epsilon'_{11} = SubBytes(X_{11}) \oplus SubBytes(X_{11} \oplus 02\epsilon_3) \\ \epsilon'_{12} = SubBytes(X_{12}) \oplus SubBytes(X_{12} \oplus \epsilon_3) \end{cases}$$

$$\begin{cases} \epsilon'_{13} = SubBytes(X_{13}) \oplus SubBytes(X_{13} \oplus 03\epsilon_2) \\ \epsilon'_{14} = SubBytes(X_{14}) \oplus SubBytes(X_{14} \oplus 02\epsilon_2) \\ \epsilon'_{15} = SubBytes(X_{15}) \oplus SubBytes(X_{15} \oplus 02\epsilon_2) \\ \epsilon'_{16} = SubBytes(X_{16}) \oplus SubBytes(X_{16} \oplus \epsilon_2) \end{cases}$$

From the set of all the possibilities for X, it is possible reach the round key *RoundKey*[10] through the equation

$$C = RoundKey[10] \oplus ShiftRows(SubBytes(X)) \quad (3)$$

and then using the key schedule properties the attacker can reach the round key *RoundKey*[9] and therefore the secret key. Keeping in mind the properties of diffusion of the algorithm AES this is the optimal model for a single fault injection [4], since the best propagation of the error is obtained and the respective equations have the least number of variables as possible. The amount of solutions estimated for every equations system is 2^8, then, the key space is reduced to 2^{32}. The next algorithm describe the complete Differential Fault Analysis described above.

Input: cipher text without error C and cipher text with error C'

begin

1. Solve the four equations systems independently

2. (this give 2^{32} candidates for*RoundKey*[10])

3. for *each candidate of RoundKey*[10] do

4. Get *RoundKey*[9] from *RoundKey*[10] using AES key schedule

5. Get unique choices of 14 bytes of *RoundKey*[10]

6. (except *RoundKey*[10](1,2))

7. Test specific equations

8. if *satisfied* then

9. for *each candidate of RoundKey*[10](1,2) do

10. Test another specific equations

11. if *satisfied* then

12. Save the candidate of *RoundKey*[10]

13. end if

14. end for

15. end if

16. end for

17. end

Output: round key *RoundKey*[10]

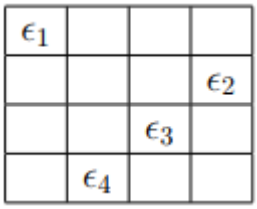
B. The Analysis with Random ShiftRows

The known of the operation ShiftRows over the state make possible in the Differential Fault Analysis of AES, that the appropriate equations systems are chosen and latter solved. Evidently the respective values ϵ'_i, $1 \leq i \leq 16$, that intervene in the systems depends of the positions of the real errors determined from the equation 2. There are 24^8 ways to consider the positions of the real errors produced during the encryption process $ShiftRows^{-1}(C \oplus C')$ in the state, gives by the construction proposed in [2], and this is the maximal number of ways as is possible.

On the other hand, the respective errors ϵ_1, ϵ_2, ϵ_3 and ϵ_4 that we can appreciate at the output of the operation MixColumns, intervene in the equations systems as another variables and they allow to choose every set of equations. If we remind that the matrix

$02\epsilon_1$	ϵ_4	ϵ_3	$03\epsilon_2$
ϵ_1	ϵ_4	$03\epsilon_3$	$02\epsilon_2$
ϵ_1	$03\epsilon_4$	$02\epsilon_3$	ϵ_2
$03\epsilon_1$	$02\epsilon_4$	ϵ_3	ϵ_2

is obtained from the matrix



although the index of the bytes differences do not be important, since they are variables in the equations, the order of the same ones in the columns is our concern and there are 24^4 ways to give this order from the output of the operation ShiftRows in the round nine, considering that in every round ShiftRows is taken at random like in [3]. These conclusions assure that the complexity of the Differential Fault Analysis of AES, with random ShiftRows in every round increase to $2^{87.0196}$.

V. CONCLUSION

In [4] is analyzed one model of Differential Fault Analysis of the algorithm AES-128, in which a single byte difference is induced in the input of the transformation MixColumns of the round eight of the encryption process. This scenario is shown as optimal for a single fault injection and is also analyzed in other works. The complexity of this attack is 2^{32} and is equally possible for AES-192 since the properties of the key schedule. For AES-256 the attack is performed similarly.

The complexity of this attack increase to $2^{87.0196}$ when the function ShiftRows is replaced in every round, and it is selected at random in the set of all their possibilities. As always, the strength of these dynamic variants against fault attack should be considered in practical applications.

REFERENCE

[1]. A. Al-Wattar and et al. “A New DNA Based Approach of Generating Key-Dependent Shift Rows Transformation”. International Journal of Network Security and Its Applications, 7(1), 2015.

[2]. A. Alfonso. “Generaci´on Aleatoria de Permutaciones con Optima Di-´ fusio´n”. Memorias del III Seminario Cient´ıfico Nacional de Criptograf˜ıa, Instituto de Criptograf˜ıa de la Universidad de la Habana, La Habana, Cuba, 2016.

[3]. A. Alfonso and P. Freyre, “AES Modificado con ShiftRows Aleatorio”. Memorias del XV Congreso Internacional de Matem´atica y Computaci´on, Sociedad Cubana de Matem´atica y Computaci´on, La Habana, Cuba, 2017.

[4]. S. Ali, D. Mukhopadhyay, and M. Tunstall. “Differential Fault Analysis of AES: Towards Reaching its Limits”. Journal of Cryptographic Engineering, 3(2):pp. 73–97, 2012.

[5]. E. Barkan and E. Biham. “In How Many Ways Can You Write Rijndael”. LNCS 2501, pp. 160–175, 2002.

[6]. E. Biham and A. Shamir. “Differential Fault Analysis of Secret Key Cryptosystems”. LNCS 1294, pp. 513–525, 1997.

[7]. J. Blomer and J. Seifert. “Fault Based Cryptanalysis of the Advanced Encryption Standard (AES)”. IACR Cryptology ePrint Archive, 075, 2002.

[8]. J. Daemen and V. Rijmen. “The Design of Rijndael: AES - The Advanced Encryption Standard”. Springer-Verlag, 2002.

[9]. C. Dobraunig and et al. “Exploiting Ineffective Fault Inductions on Symmetric Cryptography”. IACR Cryptology ePrint Archive., 071, 2018.

[10]. C. Dobraunig and et al. “Statistical Ineffective Fault Attacks on Masked AES with Fault Countermeasures”. IACR Cryptology ePrint Archive., 357, 2018.

[11]. P. Dusart, G. Letourneox, and O. Vivolo. “Differential Fault Analysis on AES”. IACR Cryptology ePrint Archive, 010, 2003.

[12]. “European Union Agency for Network and Information Security”. Algorithms, Key Size and Parameters Report. 2014.

[13]. “Federal Information Processing Standard. Announcing the Advanced Encryption Standard (AES)”. FIPS Publication 197, 2001.

[14]. A. Ghoshal, S. Patranabis, and D. Mukhopadhyay. “Template-based Fault Injection Analysis of Block Ciphers”. IACR Cryptology ePrint Archive., 072, 2018.

[15]. G. Giraud. “DFA on AES”. LNCS 3373, pp. 27–41, 2005.

[16]. N. Hussein and et al. “A Byte-Oriented Multi Keys ShiftRows Encryption and Decryption Cipher Processes in Modified AES”. International Journal of Scientific and Engineering Research, 5, 2014.

[17]. I. Ismail and et al. “Performance Examination of AES Encryption Algorithm with Constant and Dynamic Rotation”. International Journal of Reviews in Computing, 12, 2012.

[18]. M. Joye and M. Tunstall. “Fault Analysis in Cryptography”. SpringerVerlag, 2012.

[19]. L. Knudsen and M. Robshaw. "The Block Cipher Companion". SpringerVerlag Berlin Heidelberg, 2011.

[20]. D. Mukhopadhyay. "An Improved Fault Based Attack of the Advanced Encryption Standard". LNCS 5580, pp. 421–434, 2009.

[21]. P. Nidhinraj and J. George. "DNA-based Approach of AES with Key Dependent ShiftRows". International Journal of Control Theory and Applications, 9(43), 2016.

[22]. G. Piret and J. Quisquater. "A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD". LNCS 2779, pp. 77–88, 2003.

[23]. D. Saha, D. Mukhopadhyay, and D. RoyChowdhury. "A Diagonal Fault Attack on the Advanced Encryption Standard". IACR Cryptology ePrint Archive, 581, 2009.

[24]. B. Schneier. Applied Cryptography: "Protocols, Algorithms, and Source Code in C". John Wiley & Sons, 1996.

[25]. M. Spain and M. Varia. "Diversity Within the Rijndael Design Principles for Resistance to Differential Power Analysis". *LNCS 10052*, pp. 71–87, 2017.

ABOUT THE AUTHORS

MSc. Adrián Alfonso Peñate

Workplace: Institute of Cryptography. University of Havana.

Email: pfreyre@matcom.uh.cu

The education process: Graduated of Mathematics in 2014; receive Master's degree in 2018.

Research today: He has been involved in various research projects and actually works in the design and analysis of block ciphers.

PhD. Pablo Freyre Arrozarena

Workplace: Institute of Cryptography. University of Havana.

Email: pfreyre@matcom.uh.cu

Research today: Graduated of Mathematics in 1988; receive Doctor's degree in 1998.

Research today: He has been directed various research projects and has formed another researchers under his tutelage. Actually works in the design and analysis of block ciphers.