CUSTOMIZATION OF SPN BLOCK CIPHER CRYPTOGRAPHIC ALGORITHMS

Srećko Jovanović[1,2]*,
[0009-0008-2267-183X]

Mladen Veinović[2],
[0000-0001-6136-1895]

Tomislav Unkašević[1],
[0000-0002-6456-9250]

Zoran Banjac[3],
[0000-0001-8195-8576]

Tijana Aleksić[1]
[0009-0006-8875-203X]

[1]Institute VLATACOM,
Belgrade, Serbia

[2]Singidunum University,
Belgrade, Serbia

Abstract:

Substitution-Permutation Network (SPN) block cyphers are among the most important structures in modern symmetric cryptography. They provide strong security through repeated nonlinear substitution and linear diffusion layers and are widely used in secure communication, embedded systems, and lightweight cryptography. It is possible that, due to application in specific environments, it is necessary to modify the cryptographic algorithm; therefore, in this paper, we analyzed the structure of SPN cryptographic algorithms and identified the parameters that enable their modification. Modifying a Substitution-Permutation Network (SPN) algorithm is typically driven by the need to optimize the balance between cryptographic strength and computational efficiency. Researchers often implement customizations to enhance resistance to modern linear and differential cryptanalysis while tailoring the algorithm to specific hardware constraints, such as those of IoT devices or high-speed networks. Ultimately, these modifications aim to produce a more robust, specialized cypher that maintains high statistical randomness without sacrificing performance. Potential failures in the modification process were identified, and recommendations were provided to avoid them.

To demonstrate the feasibility of the proposed approach, the modification of the AES algorithm is shown by constructing a non-standard substitution permutation and a modified key expansion algorithm. |For the S-box selection a novel method of random walk over the set of all permutations was applied. A short security analysis of the obtained algorithm is presented for the given modification. The results show that customized SPN algorithms can achieve high security and performance, making them suitable for IoT devices, embedded systems, and hardware security modules.

Keywords:

SPN Block Cipher, Substitution-Permutation Network, Cryptographic Customization, Block Cipher Design, Differential Cryptanalysis, Linear Cryptanalysis.

INTRODUCTION

In the age of digitalization of business and life processes and the transition to a cyber world, data security plays an essential role that makes this transition possible and applicable. Through information protection mechanisms, it is possible for life processes to be implemented in a way that will not jeopardize the interests and rights of users or the credibility of the activities carried out. In this context, almost all security mechanisms in place rely on proven and secure cryptographic algorithms.

Correspondence:
Srećko Jovanović

e-mail:
srecko.jovanovic@vlatacom.com

DOI: 10.15308/Sinteza-2026-307-314

Ensuring security in a heterogeneous environment such as cyberspace requires the use of mechanisms to ensure interoperability, and this is achieved most simply by using standardized cryptographic algorithms such as AES [1], RSA [2], ECC [3] and others. A cryptographic algorithm is considered reliable if its impenetrability can be formally proved by the methods of Information Theory in terms of absolute security or by methods of computational complexity in terms of practical security. In both cases, the starting point is that all the algorithm's parameters and functionality are known to the person analyzing it, except for the cryptographic key's value, Kerkhof's principle [4].

The design and analysis of a new, purpose-built cryptographic algorithm is a very serious research task that requires significant technical and human resources. Therefore, modifying existing secure cryptographic algorithms by introducing new transformations or modifying other parameters is considered a less demanding operation. However, ignorance of how to transform a cryptographic algorithm is an additional aggravating factor that a potential attacker encounters, greatly increasing the resources required for a successful attack. Therefore, users who have the necessary knowledge and technological capabilities create their own dedicated solutions for use in internal environments and, by law, obligate participants in their communication networks to implement them. The GHOST standard in Russia [5] and SM2/SM3/SM4 in China [6] are examples of purpose-built standard solutions. In addition to the above, within the security spheres of the state apparatus, it is necessary to use cryptographic algorithms that are adapted to comply with the specified legal levels of protection. Those kinds of algorithms and their adaptations are most often not publicly available, for example, in the United States, NSA Suit A set of cryptographic algorithms [7]. Although ignorance of a cryptographic algorithm's description and functionality is not an element on which the algorithm's quality or achieved protection is based, it significantly increases the resources an attacker needs to try to compromise the algorithm's usability and functionality. Also, the need to adapt cryptographic algorithms has a technological background, reflected in the perspective of the emergence of quantum computers, which necessitates increasing cryptographic key lengths by a factor of two for symmetric algorithms to maintain the same level of resistance to attacks. Increasing the length of a cryptographic key is not always a simple operation and carries additional consequences, which we will discuss later.

2. STRUCTURE OF BLOCK CRYPTOGRAPHIC ALGORITHMS OF THE SPN TYPE

Block cryptographic algorithms of the SPN type consist of a fixed number of rounds, each performing a set of operations on input data represented as a series of bytes of a fixed length, called a block of data. A single round of transformation usually consists of substituting bytes of a block of processing by applying a substitution transform (S-box) and then applying a specific permutation P (P-box) over a block of data. At the end of each round, a coupling is made with the key of the round. Round keys are derived from the key of a cryptographic algorithm, according to the cryptographic transformation's key expansion definition. Graphical presentation of the SPN block cypher algorithm is shown in Figure 1.

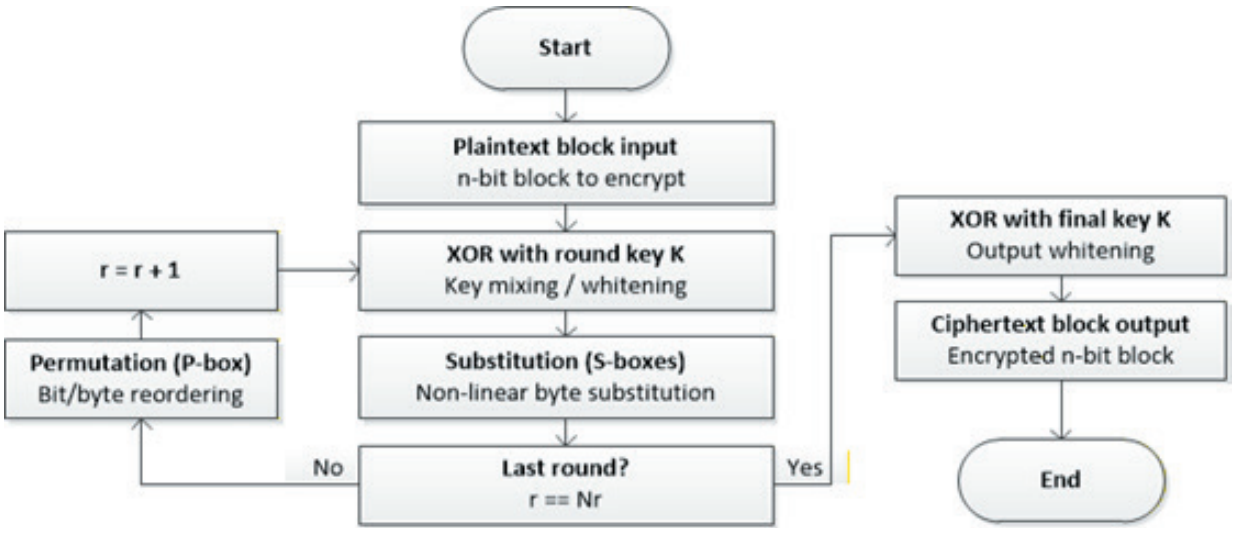

Figure 1. Structure of the SPN type cryptographic algorithm

A well-designed SP network with several alternating rounds of S-box and P-box transformations satisfies Shannon's principles of confusion and diffusion [8]. In concrete terms, this means that a change in a single bit, either in a block of plaintext (diffusion) or in the cryptographic key used (confusion), results in a change in half of the bit in the block of cipher.

From the previous description of the structure of cryptographic algorithms of the SPN type, it follows that the adjustment of this type of cryptographic algorithms can be realized in relation to the S-box and P-box transformations, the length of the cryptographic key and the algorithm for its expansion, as well as the size of the block to be processed.

2.1. SUBSTITUTION TRANSFORMATION – S-BOX

A substitution transform is a bijective nonlinear Boolean function with constraints $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ that $S(x) \neq x$ and $S(\bar{x}) \neq x$ is true. From a cryptographic point of view, important properties of this function are the differential characteristic and the degree of nonlinearity. The differential characteristic is defined as follows:

$$\Delta_{\alpha,\beta} = \{(x_1, x_2) \in \mathbb{F}_2^n \times \mathbb{F}_2^n : x_1 \oplus x_2 = \alpha \text{ and } S(x_1) \oplus S(x_2) = \beta\}$$

a set whose elements are ordered pairs whose Heming distance is equal to α and the Heming distance of values of the function S at those points is equal to the β. The probability of differential is defined as`

$$P_{\alpha,\beta} = \frac{|\Delta_{\alpha,\beta}|}{2^n}$$

where the $|\Delta_{\alpha,\beta}|$ denotes the number of elements in the set $\Delta_{\alpha,\beta}$. The goal is to keep the maximum value of a given function as small as possible, and this represents a measure of resistance to differential cryptanalysis attacks.

The degree of nonlinearity of a function S is defined as follows:

Let be a given function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and let A_n be the set of all affine Boolean functions of n arguments. The nonlinearity of the function f is equal to

$$nl(f) = \min_{g \in A_n} d(f, g)$$

where the $d(f,g)$ is Heming distance of functions f,g.

The nonlinearity of the Boolean function is a measure of resistance to attack by linear cryptanalysis.

It is desirable that for the selected substitution permutation, the differential characteristic be as small as possible and the nonlinear characteristic as large as possible.

For each SPN cryptographic algorithm, when analyzing its strength, the properties that the specific applied transformation must satisfy are derived, with emphasis on the following: $P_{\alpha,\beta}$ and $nl(f)$.

Various methods of generating transformation candidates are present in the literature, usually using algebraic or stochastic methods.

2.2. PERMUTATION TRANSFORMATION

The purpose of the permutation transform is to diffuse the influence of each input bit across all bits of the output block within the round. This transformation is usually implemented as a series of matrix operations over the block, and the choice of specific operations depends on the set security goals.

2.3. THE LENGTH OF THE CRYPTOGRAPHIC KEY AND THE NUMBER OF ROUNDS

A cryptographic key is an essential component of a cryptographic algorithm and is a series of random bits of appropriate length. The method for generating the cryptographic key must provide maximum entropy, and its length must be sufficient to prevent brute-force attacks by checking all possible values. If the length of the cryptographic key is modified, then it must be ensured that its use is such that

- Each bit of the cryptographic key affects each bit of the cipher. This means that as the length of the key increases, the number of rounds must also increase.
- The number of rounds aligns with the way round keys are generated, ensuring the algorithm's security is not compromised, for example, by repeating the round keys.

2.4. POSSIBLE RISKS AND HOW TO AVOID THEM

Designing cryptographic algorithms is a serious research task, but adapting existing good solutions is not entirely harmless. The most common situation is when new constructions are introduced into existing solutions, without sufficient knowledge and experience to formally substantiate the safety characteristics of the proposed/implemented modification. In real life, it is also not uncommon for the values of parameters to be deliberately changed to weaken the protection provided by a cryptographic algorithm, as in the case of a class of export cryptographic algorithms introduced in the United States in 1990.

It may also be shown that selecting poor S-boxes, for example, with low nonlinearity or a large differential characteristic, can completely undermine the efforts put into the design of the AES algorithm.

The dangers of customization can be significantly reduced if the following principles are taken into account in the adaptation process:

- When possible, use standardized, customizable solutions with instructions for selecting customization parameters (KMAC, HKDF, etc.).
- Any customization should be clearly documented, including the need and reasons, the detailed technical specification and the impact on the safety of the original solution or the resulting solution.
- Conduct an expert analysis of the obtained solution and analyze the formal mathematical security analysis of the original algorithm and the compatibility of the customized solution with it.

3. MODIFICATION OF THE AES CRYPTOGRAPHIC ALGORITHM

The AES cryptographic algorithm was selected by NIST as the version of the Rijndael algorithm following a public competition that ran from 1997 to 2000. AES is a symmetric block cryptographic algorithm of the SPN type. The size of the data block to be transformed is 128 bits, and the key size can be 128, 192, or 256 bits. The state of the algorithm consists of 16 bytes (128 bits) written as a 4x4 matrix over GF(28). The state is transformed by the number of rounds, determined by the key length (10, 12, or 14).

For each round, it contains four operations

- With the SubBytes operation, the value of each byte of the current state of the algorithm is replaced by the contents of a predefined table, the S-box, so that the value of each byte of the state represents the address of the content in the S-box table with which that byte will be replaced.
- With the ShiftRows operation, each type of 4x4 matrix is shifted to the left cyclically by 0,1,2,3 places.
- The MixColumns operation represents the multiplication of a state matrix with a fixed 4x4 matrix over the field GF(2^8)
- The AddRoundKey operation consists of XOR summing the bytes of the state and the bytes of the key of the current round.

A detailed description of the AES algorithm is given in [1]. A full description of its operation and the operations it involves. The block diagram of the AES algorithm is shown in Figure 2, and it can be seen that the basic parameters of the algorithm are the key length, the S-box, the number of rounds, and the block size to be transformed. Each of these parameters can be subject to algorithm modification, but, as we will see, their influences are intertwined and affect the safety of the resulting algorithm.

In this example, we will modify the algorithm by replacing the standard S-box transform with a purpose-generated one. In the literature, several methods for the substitution transformation construction have been proposed [4], [9], [10], [11].

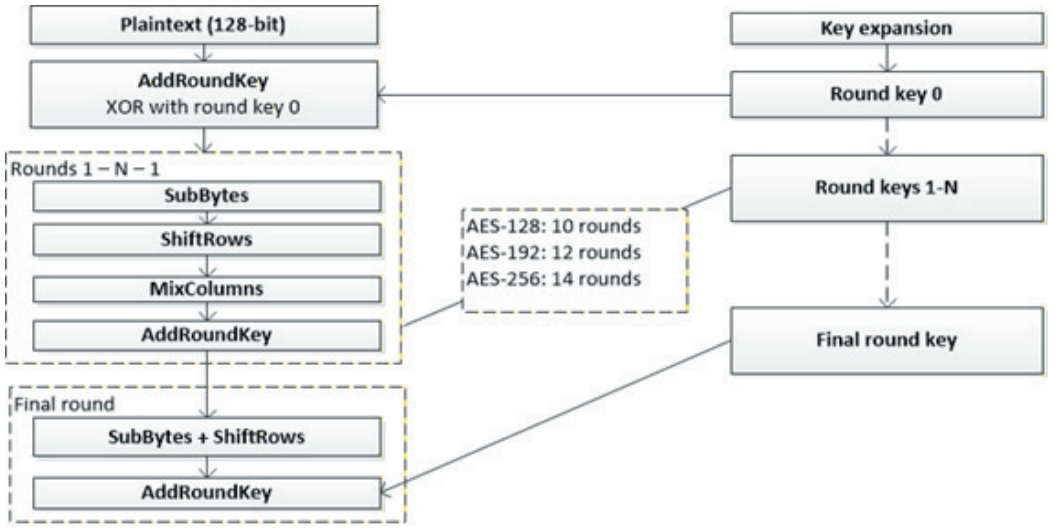

Figure 2. Block diagram of the AES algorithm

The choice of substitution transformation in our case is realized by searching the set of all permutations, and the method the method for generating candidate transformations is described in [12]. In brief, it is given by the following system of equations

$$\mathrm{S}_0 = \mathrm{S}_{\mathrm{AES}}$$
$$\mathrm{S}_{\mathrm{n+1}} = \mathrm{S}_{\mathrm{n}} \circ \mathrm{T}_{\mathrm{i_n,j_n}}$$

Where T_{i_n}, j_n permutation is the transposition of positions i_n, j_n whose values are obtained as output from the vTRNG random number generator, designed and implemented at the VLATAKOM Institute. By Theorem 1 in [12] It has been shown that if Π is an arbitrary permutation of a set of 256 elements, then

$$\mathrm{P}(\mathrm{S_n} = \Pi) = \frac{1}{256!}$$

for n sufficiently large. This means that the process represents an absolutely random walk over the set of all permutations of 256 elements. For each resulting candidate transformation, it is checked whether there are stationary points and complementary stationary points; if either condition is met, the transformation is discarded. If the transformation is not rejected in the previous step, its linear and differential characteristics are calculated and, and based on the values defined by the security policy, it is accepted or rejected as a candidate for customization of the AES algorithm.

For the standard AES S-box, the differential characteristic is $p = 0.015625$ and nonlinearity $nl = 112$.

During the generation of candidates for customization of substitution transforms to adapt the AES algorithm, transformations with different values of differential and nonlinear characteristics were obtained. Although we get examples with the same values as in the standard AES algorithm in Table 1, the transformations shown have one characteristic in common with a standard AES transformation and another that is worse. For this modification, we chose the transformation from Table 1

The second adjustment we have made compared to the standard AES is in the round key generation section. The customization consists of applying the SHA-256 hash algorithm to each round key generated by the key expansion algorithm, and the new round key is obtained as the XOR value of the first and second 128 bits of the result.

Naturally, this raises the question of how such customization impacts security. At a minimum, one must evaluate the algorithm's resistance to linear and differential cryptanalysis, as well as its overall statistical properties. By applying Theorems 13.2.1, 13.2.2. from [13], resistance to linear and differential cryptanalysis can be confirmed.

Table 1. Constructed S-box transformations

(a)

$P_{\alpha,\beta} = 0.015625$, $nl = 110$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	7E	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	9C	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

(b)

$P_{\alpha,\beta} = 0.0234375$, $nl = 112$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C9	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	63	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

For the expected probability of the occurrence of a differential (α,β) after two rounds of the AES algorithm with a substitution transform *S*,

$$\mathrm{EDP}_{\alpha,\beta} \leq (\mathrm{MDP})^{\mathrm{n_t}}$$

MDP is the maximal differential characteristic of the transformation S.

The expected correlation with an arbitrary linear combination of input and output values (ELP) may be shown to be less than or equal to the value $(MLP)^{n_t}$ where MLP is the maximum correlation with the linear functions .

And in both cases, the size of the block is n_t=16 bytes.

For selected transformation

$$\mathrm{MDP} = 0.0234375 = \frac{6}{256}$$

$$\mathrm{MLP} = 0{,}015625 = \frac{1}{64}$$

accordingly, it is

$$\mathrm{EDP}_{\alpha,\beta} \leq \left(\frac{6}{256}\right)^{16} \leq \left(\frac{8}{256}\right)^{16} \leq 2^{-80}$$

$$\mathrm{ELP} \leq \left(\frac{1}{64}\right)^{16} \leq 2^{-96}$$

This shows that the probability of success of an attack on this algorithm by linear or differential cryptanalysis is negligible.

Statistical analysis was performed on the output array of the modified algorithm in counter mode, with a size of 500 MB. The NIST software package [14]. Partial results of the test are presented in Table 2 due to the volume of data. The final conclusion of the testing is that the results obtained do not contradict the assumption that the tested sample originated from a random sequence.

For the proposed modification, we have shown that the obtained algorithm is resistant to linear and differential cryptanalysis in accordance with the methodology and results of formal mathematical analysis published in [13], [15]. We've also shown that the obtained output sample of the customized algorithm is not in contradiction with the hypothesis that the sample stems from a truly random source.

Table 2. Results of NIST statistical testing

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
20	19	15	16	24	8	18	24	18	18	0.307407	0.9833	Frequency
25	19	20	19	10	19	12	24	19	13	0.207118	0.9944	BlockFrequency
22	13	19	19	19	23	12	17	16	20	0.706149	1.0000	CumulativeSums
16	15	20	24	17	17	21	11	24	15	0.458035	0.9722	Runs
17	17	24	21	17	18	13	25	12	16	0.437274	1.0000	LongestRun
24	11	16	18	16	25	16	21	21	12	0.268170	0.9889	Rank
17	17	14	22	20	18	21	22	11	18	0.717488	0.9944	FFT
16	23	13	24	16	18	27	10	17	16	0.139036	0.9889	NonOverlappingTemplate
16	20	16	21	20	19	17	13	21	17	0.944046	0.9944	OverlappingTemplate
25	16	15	25	12	17	19	14	16	21	0.359555	0.9778	Universal
16	23	17	16	17	12	18	18	17	26	0.579479	0.9889	ApproximateEntropy
10	10	9	17	14	12	13	7	10	13	0.637119	0.9913	RandomExcursions
11	11	11	9	12	11	14	15	7	14	0.834308	0.9826	RandomExcursionsVariant
21	10	12	23	22	16	19	21	16	20	0.387648	0.9778	Series
18	15	14	19	22	18	22	19	18	15	0.925420	0.9944	Series
18	18	16	14	22	19	30	11	15	17	0.148094	1.0000	LinearComplexity

4. CONCLUSION

Driven by the demand for specialized cryptographic solutions, this paper explores the modification of established, secure SPN-based algorithms to create purpose-defined alternatives. The structure of cryptographic SPN algorithms is analyzed, and segments that can be modified, key length, processing block size, number of rounds, S-box and P-box transformations were detected based on it. Each of these parameters should be aware that this process, by its very nature, does not mean that the obtained solution is automatically better or worse than the initial cryptographic algorithm. Possible failures in that process were identified, which could degrade the security of the obtained solution compared to the initial solution. In principle, each of the listed parameters can be used for modification, but it should be borne in mind that changes to the length of the cryptographic key or the block size require a corresponding adjustment in the number of algorithm rounds. From a practical standpoint, the simplest change is replacing the S-box transformation, provided it satisfies the cryptographic criteria we discussed. In this context, one possible modification of the AES algorithm with a suitable substitution transformation is presented. Using the methodology developed and presented in [13], [15]. For formal security analysis, it has been shown that the modified algorithm has satisfactory resistance to differential and linear cryptanalysis. Using the NIST software package for statistical analysis demonstrates that the generated sample has good statistical properties.

The results show that customized SPN algorithms can achieve high security and performance, making them suitable for IoT devices, embedded systems, and hardware security modules.

5. ACKNOWLEDGEMENT

We gratefully acknowledge the support of the Vlatacom Institute of High Technologies in Belgrade, as well as our colleagues and associates at the Institute, for their assistance and valuable suggestions during the realization of this project.

REFERENCES

[1] Advanced Encryption Standard (AES), 2001.

[2] K. Moriarty, B. Kaliski, J. Jonsson and A. Rusch, *PKCS #1: RSA Cryptography Specifications Version* 2.2, RFC Editor, 2016.

[3] S. Turner and D. R. L. Brown, *Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)*, RFC Editor, 2010.

[4] C. Boura and M. Naya-Plasencia, Symmetric Cryptography 1: Design and Security Proofs, Wiley, 2023.

[5] A. Gridnev, "Russian cryptography standards," *Medium*, 19 June 2019.

[6] C. I. N. I. Center, "Standards of encryption algorithms," 7 April 2026.

[7] Wikipedia, "NSA Suite A Cryptography," 12 September 2025.

[8] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, p. 379–423, July 1948.

[9] C. Carlet, Boolean Functions for Cryptography and Coding Theory, Cambridge University Press, 2020.

[10] A. S. El Batouty, T. A. Hamdalla, H. A. Fayed and M. H. Aly, "Bit-Independent Criteria Evaluation of Custom S-Boxes: Enhanced AES Encryption Security," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 14, p. e32318, November 2025.

[11] K. Sakiyama, Y. Sasaki and Y. Li, Security of Block Ciphers: From Algorithm Design to Hardware Implementation, Wiley, 2015.

[12] T. Unkašević, Z. Banjac and M. Milosavljević, "A Generic Model of the Pseudo-Random Generator Based on Permutations Suitable for Security Solutions in Computationally-Constrained Environments," *Sensors*, vol. 19, p. 5322, December 2019.

[13] J. Daemen, The Design of Rijndael, 2nd ed. 2020. ed., V. Rijmen, Ed., Berlin, Heidelberg: Imprint: Springer, 2020.

[14] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray and S. Vo, A statistical test suite for random and pseudorandom number generators for cryptographic applications, 2010.

[15] C. Boura and M. Naya-Plasencia, Symmetric Cryptography 2: Cryptanalysis and Future Directions, Wiley, 2023.