ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 10, No. 1, 37-47 (2025)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

Hardware and Secure Implementation of Enhanced ZUC Steam Cipher Based on Chaotic Dynamic S-Box

Mahdi Madani*[,1], El-Bay Bourennane[1], Safwan El Assad[2]

[1]*Laboratoire ImViA (EA 7535) , Université Bourgogne Europe, 21000 Dijon, France*

[2]*IETR , University of Nantes/Polytech Nantes, France*

ARTICLE INFO

Article history:
Received: 14 October, 2024
Revised: 05 January, 2025
Accepted: 06 January, 2025
Online: 04 February, 2025

Keywords:
Dynamic S-box
FPGA design
Hardware metrics
ZUC stream cipher
Mobile security
Cryptography
Cryptanalysis

ABSTRACT

Despite the development of the Internet and wired networks such as fiber optics, mobile networks remain the most used thanks to the mobility they offer to the user. However, data protection in these networks is more complex because of the radio channels they use for transmission. Hence, there is a need to find more sophisticated data protection means to face any attack. But, this is not an easy task, especially with the emergence of AI-based attacks. In this context, we proposed in this work a solution that can significantly improve data protection in a new-generation mobile network. Therefore, the main objective of this study is to improve and implement an enhanced version of the standard ZUC algorithm designed by the Data Assurance and Communication Security Research Center of the Chinese Academy of Sciences and standardized by the 3GPP (3rd Generation Partnership Project) organization to ensure the LTE (Long Term Evolution of radio networks) security. The proposed design is principally based on replacing the static S-boxes of the original algorithm (S_0 and S_1) with a chaos-based dynamic S-boxes thus allowing to generate a different key-stream for any change on the secret key and with the best randomness and robustness properties. The two new dynamic S-boxes are initialized with 256 initialization values each (x"00"), then filled in parallel using two chaotic maps that use the ZUC algorithm registers, the CK (Cipher Key), and the IV (Initial Vector) to form two different initial values for each chaotic map. To reach the hardware performance, we implemented the system on a Xilinx XC7Z020 PYNQ-Z2 FPGA platform. The designed architecture occupies low logic resources (1135 Slice LUTs, 762 Slice Registers, and 8 DSP48E1) on the used FPGA device and can reach a throughput of 2515.84 Mbps with a running frequency of 78.62 Mhz by consuming only 0.188 W. To evaluate the resistance of the proposed cryptosystem, we used many security tests (keystream distribution, keystream randomness, key sensitivity, plaintext sensitivity, keyspace, and NIST statistical tests). The experimental results and comparison with other S-boxes based algorithms prove on one hand that using the dynamic S-box technique has enforced considerable data protection against cryptanalysis attacks, and on another that the hardware metrics (used logic resources, achieved throughput, and efficiency) are suitable for real-time applications such as mobile security transmission.

1. Introduction

Despite the development of high throughput Internet based fiber optics, mobile and connected objects networks remain the most used thanks to the mobility and ease they offer to the user. The main component they use is the smartphone which facilitates access to most of our daily services such as video calls, social network messaging, e-payment, smart-home, smart-city, etc.

However, data protection in these networks is more complex due to the physically unprotected radio channels they use for communications.

Hence, it is necessary to find more sophisticated means of data protection to deal with any attacker trying to illegally access data by going directly to the storage location (mainly servers or cloud) or by capturing encrypted data and trying to decrypt it by cracking the encryption algorithm used or looking for the secret key.

Therefore, protecting personal and sensitive information (naturally circulates on the physically unprotected radio transmission

*Corresponding Author: Mahdi Madani, Laboratoire ImViA, Université Bourgogne Europe, 21000 Dijon, France & Mahdi.Madani@u-bourgogne.fr

channel) is not an easy task, especially with the emergence of AI-based (Artificial Intelligence) attacks. This is why a cryptographic algorithm with the best robustness and resistance against computer attacks is needed to success this task. Since many decades, different cryptosystems have been designed as: block ciphers: DES (Data Encryption Standard), AES (Advanced Encryption Standard), KASUMI; stream ciphers: RC4 (Rivest Cipher 4), SNOW-3G, ZUC; hashing functions: DSA (Digital Signature Algorithm), Secure Hash Algorithms SHA-0, SHA-1, SHA-2, SHA-3; chaotic systems: Lorenz, Chen, logistic map, skew tent map, and other methods.

In this study, we evaluate the security performance of the ZUC stream cipher which is designed by the Data Assurance and Communication Security Research Center of the Chinese Academy of Sciences and standardized by the 3GPP (3rd Generation Partnership Project) organization to ensure the LTE (Long Term Evolution of radio networks) security. ZUC algorithm forms also the kernel of the confidentiality (128-EEA3) and integrity (128-EIA3) functions used in the LTE networks security [1, 2]. In addition, we propose an enhanced version that can significantly improve data protection in a new-generation of mobile network.

We started by analyzing the internal architecture of the original ZUC algorithm is based on three main layers, LFSR (Linear Feedback Shift Register), the BR (Bit Reorganization), and the NLF (Non Linear Function) [3]. The state of the art has proven that the ZUC architecture has certain weaknesses that require immediate improvements [4, 5, 6, 7, 8].

To remedy the identified problems, we improved the non linearity part of the standard algorithm by replacing its two static S-boxes (S_0 and S_1) with a chaos-based dynamic S-boxes [9, 10, 11, 12]. The new version allows the generation of different key-streams for any change on the secret key with the best randomness and robustness properties that increase the complexity of cryptanalysis attacks.

This paper is principally based on extending our work initially presented at the ICEET23 conference and improving the performance of the implemented architecture. Therefore, we can summarize the two main contributions of this extended version on: Firstly, we used a chaotic map to generate dynamically the internal two S-boxes (S_0 and S_1) of the ZUC algorithm. The experimental analysis show the respect of the generated S-boxes to the non-linearity recommendations. Secondly, we designed an optimized FPGA implementation with the best hardware metrics. A comparative study with the literature is given to confirm our results on the two listed steps. The proposed architecture consists of using two parallel chaotic maps to generate two dynamic S-boxes SD_0 and SD_1. They are dynamic because the chaotic maps are initialized using control parameters derived from the combination of the CK, the IV, and the internal registers (S_{15}, S_{14}, S_5, and S_7) of the LFSR layer. This technique ensures that any change (one bit is enough) in these three parameters will result in a different S-boxes. Many examples of generated S-boxes are examined using known theoretical tests for similar analysis, and the results are conclusive, unlike the original work which failed some tests.

To reach the hardware implementation, the proposed architecture is coded using a VHSIC Hardware Description Language (VHDL) and implemented on Field-Programmable Gate Array (FPGA) technology [13, 14] to explore its offered parallel calculations capabilities and the low power consumption.

The implementation on a Xilinx XC7Z020 PYNQ-Z2 FPGA hardware platform achieves a throughput of 767.52Mbps at an operating frequency of 94.34 Mhz. The robustness of the proposed architecture is evaluated using the keystream performance: analyzing the uniformity (histogram, chi square), randomness, key sensitivity, plaintext sensitivity, examining the key space complexity, and investigating the NIST (National Institute of Standards and Technology) statistical tests [15].

The experimental results prove on one hand that using the dynamic S-boxes technique has enforced considerable data protection against cryptanalysis attacks, and on another that the hardware metrics (used logic resources, achieved throughput, and efficiency) are suitable for real-time applications such as mobile security transmission.

The reminder of this paper is organized as follows. Section 2 summarize the internal architecture and the processing steps of the regular ZUC stream cipher in its two operating modes. Section 3 describes the proposed architecture including the chaos-based dynamic S-boxes designed to enhance the security and enforce the resistance of the standard algorithm face to attacks. Section 4 presents the FPGA implementation results in terms of the occupied hardware metrics (logic resources, FFs, BRAMs) and the achieved timing metrics (throughput, frequency, efficiency). It also shows the behavioral simulation results under Vivado tools to prove the best functionality of our design. Section 5 investigates cryptanalytic analysis and stream cipher performance allowing to prove the robustness of the proposed scheme. Finally, section 6 summarizes the whole article and gives directions for our perspectives in the future.

2. Original ZUC stream cipher overview

As we already discussed, ZUC algorithm is a word-oriented stream cipher designed by the Data Assurance and Communication Security Research Center of the Chinese Academy of Sciences and standardized by the 3GPP organization to ensure the LTE and 5G (the fifth generation of cellular network technology) security. In this section, we present briefly its internal architecture, its processing steps in the two operating modes, and some attacks from the literature that subjected the standardized version.

ZUC is a word-oriented algorithm that generates a 32-bits word key-stream under the control of a 128-bits CK and 128-bits IV [3, 16, 17]. Its internal architecture is formed by three main interacting layers corresponding to the LFSR, the BR, and the NLF layers, respectively. The LFSR layer is formed by 16 stages of 31-bits registers ($S_0, S_1, \ldots, S_{15}$). The BR layer is composed of 4 stages of 32-bits registers (X_0, X_1, X_2, X_3) filled from the LFSR layer ($S_{15}, S_{14}, S_{11}, S_9, S_7, S_5, S_2, S_0$). The NLF layer is made up of 2 S-boxes (S_O, S_1) and 2 intermediate 32-bits registers (R_1, R_2) sequentially updated based on the output of the BR layer.

The ZUC stream cipher runs in two operating modes to generate a valid output, initialization and key-stream, as described below.

- Initialization mode: consist of loading the control parameters (CK and IV) to initiate the internal states of the LFSR registers according to the following Formula.

$$\begin{cases} S_0 = CK(127:120)||1000100110101111||IV(127:120) \\ S_1 = CK(119:112)||0100110101111100||IV(119:112) \\ S_2 = CK(111:104)||1100010011010111||IV(111:104) \\ S_3 = CK(103:96)||0010011010111110||IV(103:96) \\ S_4 = CK(95:88)||1010111100010011||IV(95:88) \\ S_5 = CK(87:80)||0110101111000101||IV(87:80) \\ S_6 = CK(79:72)||1110001001101011||IV(79:72) \\ S_7 = CK(71:64)||0001001101011111||IV(71:64) \\ S_8 = CK(63:56)||1001101011110000||IV(63:56) \\ S_9 = CK(55:48)||0101111000100111||IV(55:48) \\ S_{10} = CK(47:40)||1101011110001001||IV(47:40) \\ S_{11} = CK(39:32)||0011010111100011||IV(39:32) \\ S_{12} = CK(31:24)||1011110001001101||IV(31:24) \\ S_{13} = CK(23:16)||0111100010011011||IV(23:16) \\ S_{14} = CK(15:8)||1111000100110101||IV(15:8) \\ S_{15} = CK(7:0)||1000111101011001||IV(7:0) \end{cases}$$

Then, combining the output of the NLF layer (W), a primitive polynomial over the Galois Gield $GF(2^{31}-1)$, and a modulo operations [3] to updates the register S_{15} according to Equation 1.

$$\begin{cases} u = W >> 1 \\ v = 2^{15}S_{15} + 2^{17}S_{13} + 2^{21}S_{10} + 2^{20}S_4 \\ \quad +(1+2^8)S_0 \ \ mod\ (2^{31}-1) \\ Fb = (v+u) \ \ mod\ (2^{31}-1) \end{cases} \quad (1)$$

In addition, the remainder registers are right shifted to update the LFSR layer, as follows.

$$\begin{cases} S_{15} = Fb \\ S_{14} = S_{15} \\ S_{13} = S_{14} \\ \cdots \\ S_0 = S_1 \end{cases}$$

This mode is executed for 32 clock cycles without generating output sequence Z, as illustrated in Figure 1.

- Key-stream mode: consist of using the outputs of the NLF (W) and the BR (X_3) layers to generate a 32-bits output key-stream word (Z) at each clock cycle according to Equation 2.

$$Z = W \oplus X_3 \quad (2)$$

The processing of this operating mode is illustrated in Figure 2.

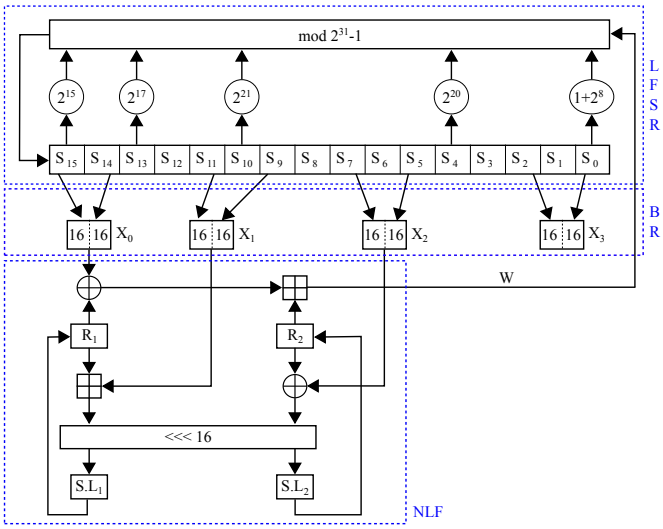


Figure 1: ZUC stream cipher initialization mode.

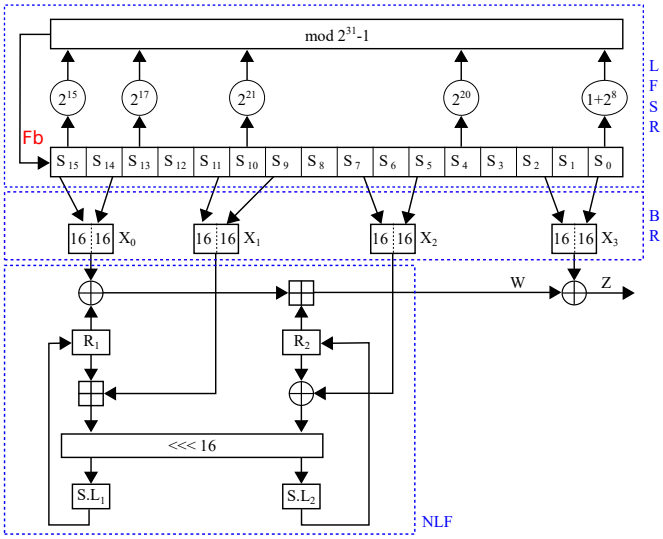


Figure 2: ZUC stream cipher key-stream mode.

Since its inception, the robustness of the ZUC stream cipher has been analyzed and the algorithm has suffered numerous attacks. In the literature several works have identified some drawbacks. Among them, we cite the alternative algebraic analysis [4], differential attacks[5], satisfiability solvers based analysis [6], and NIST statistical analysis [8, 7, 18, 19].

To overcome these weaknesses, we proposed a solution in this work based on enhancing the nonlinear part of the algorithm to resist cryptanalysis attacks. In the following sections, we will detail the technique used and give the results proving the improvement.

3. Proposed chaos-based architecture

In this section, we present the proposed architecture focused on improving security resistance against cryptographic attacks. The

adopted technique is based on the use of a chaos-based dynamic S-boxes by the NLF layer, unlike the original one using two static S-boxes S_0 and S_1.

3.1. Chaotic dynamic S-box implementation

S-boxes known as lookup tables are a non linear functions widely used by cryptographic algorithms. They are defined to ensure no repetition and to generate a non-linear output value [9, 10, 11]. However, if the CK used is cracked, the entire security of the algorithm will be compromised and the encrypted data will be exposed. To overcome this issue, we designed a dynamic chaotic S-box that ensures that even if the internal architecture of the algorithm and the CK are compromised, only modifying the CK will provide good data protection in the future. This will be guaranteed by the chaos map control parameters which change with every small modification in the CK and IV mobile client parameters ensuring a good confusion properties of the NLF layer and generating a random output key-stream.

Basically, ZUC stream cipher runs on two operating modes (initialization and key-stream modes) using two static S-boxes S_0 and S_1. However, the proposed design uses two chaos-based dynamic S-box.

We began initializing the internal LFSR, BR, and NLF layers, like the standard algorithm. The only difference is the replacement of the standard S-boxes (S_0 and S_1) by two new dynamic S-boxes (SD_0 and SD_1) of the same length (16×16) but with internal values initialized to zero at this step. In parallel, we set two logistic chaotic maps using different initial conditions extracted from the BR layer (driven from the control parameters CK and IV) according the Formulas 3 and 4.

$$DK_1 = X0(31\ downto\ 16) \parallel X2(15\ downto\ 0) \tag{3}$$

$$DK_2 = X0(15\ downto\ 0) \parallel X2(31\ downto\ 16) \tag{4}$$

Note that DK_0 and DK_1 form the dynamic keys of the chaotic system, $X0$ and $X2$ are registers from the BR layer, and $\parallel$ is the concatenation operator.

After initializing the chaotic system based on two Logistic maps (non-linear chaotic discrete function), it produces two 32-bits random sequences. The first sequence will be used to complete the dynamic S-box SD_0 and the second sequence to complete the S-box SD_1. The mathematical model of the discrete logistic map is defined by Equation 5.

$$X_{n+1} = \begin{cases} \dfrac{X_n \times (2^N - X_n)}{2^{N-2}} & if\ X_n \neq [3 \times 2^{N-2}, 2^N] \\ 2^N - 1 & if\ X_n = [3 \times 2^{N-2}, 2^N] \end{cases} \tag{5}$$

Where X_{n+1} is the new value calculated from the previous one X_n, N is the output size of the discrete logistic map (N=32-bit).

To fill one S-box, we run the chaotic system that generates 32-bit key-stream words at each clock cycle. We take 8-bits to fill one of the 256 available cells. To avoid repetition, we used a control vector of the same size ($16 \times 16 = 256$) based on a repetition flag set to zero. The principle consists of saving the value (x) on the S-box (at position [i, j], i, j = 0 to 15) and setting the corresponding flag to one (flag [x] = 1). This ensures that If the value is generated again during the filling process, it will be ignored. The block diagram of the algorithm is shown in Figure 3.

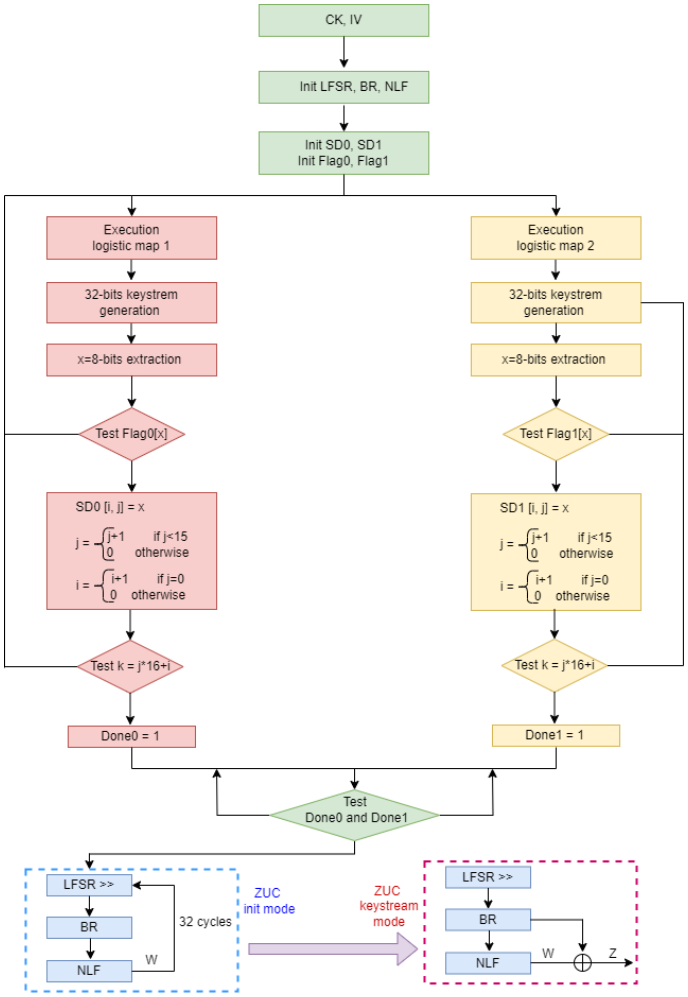


Figure 3: S-boxes generation procedure.

To complete the second S-box in parallel, we used the same principle. Initializing the chaotic maps with different keys ($DK_1 \neq DK_2$) guarantees the generation of two distinct S-boxes. This technique allows the completion of the two dynamic S-boxes without repetition by respecting the principle of creating lookup tables.

After completing both the S-boxes, the proposed ZUC stream cipher will be executed similarly to the original one, with 32 cycles running the initialization mode and then the key-stream mode for the remainder but using the proposed dynamic S-boxes, as explained above. For more clarity, we illustrate the architecture of the proposed design in Figure 4.

3.2. S-box analysis

The security of algorithms using on S-box is principally based on this non-linear component. Therefore, any weakness or problem in its construction will significantly affect the whole security of the algorithm and weaken its resistance to attacks such as linear and

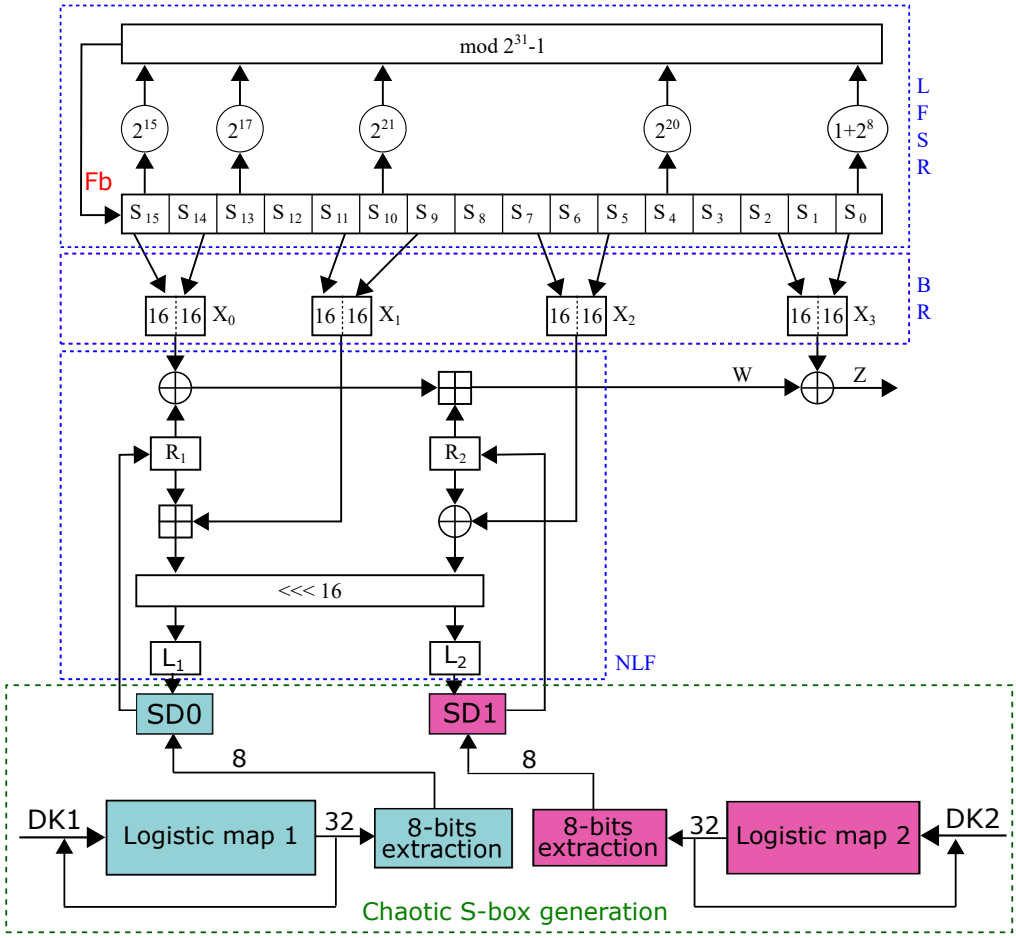


Figure 4: Proposed ZUC with dynamic S-boxes architecture.

differential cryptanalysis. For this, we evaluate the main performances of the proposed dynamic S-box to avoid any unpleasant surprises. To prove the expected high level S-box, we analyzed its satisfactory to the following criteria: bijection, strict avalanche criterion (SAC), non-linearity, output bits independence criterion (BIC), equiprobable input/output XOR distribution, differential approximation probability (DP), and maximum expected linear probability (LP).

To facilitate understanding the analysis, we present in Table 1 a example of a generated S-box using the proposed technique. So, the analysis study in this section will be based on this sample. In Table 2 we give a comparative study with the literature works based on the mentioned criteria.

3.2.1. Bijectivity and non-linearity

The bijective property of an $N \times N$ constructed S-box is respected if there is no repetition of its values in the interval $[0, 2^N - 1]$. Therefore, as we can see from Table 1, our S-box satisfy this criteria because all its values $[0, 255]$ are different.

According to the S-box non-linearity definition given by [23, 34], our S-box highly non-linear because the minimum non-linearity indicator of 100 when $n = 8$. It is better than all the results presented in Table 2.

3.2.2. SAC criterion

As defined by [35], the SAC criteria is satisfied if changing a single input bit will conduct to change a half of the output bits. To evaluate this parameter in our S-box we used the dependence matrix (see [23]). As we can see from Table 2, the mean (0.4976), value is closed to the optimal value (0.5) and the offset value (0.0156) is closed to zero confirming the satisfaction of the SAC criteria.

3.2.3. Output bits independence criterion

Similarly to SAC, the authors in [35] defined BIC indicating the pair-wise independent for a given vector and its corresponding avalanche (complementing 1 bit). Applying this test to our S-box, we obtained a minimum value of BIC non-linearity (100) and a maximum value of DP (7) ([23, 32]) indicating the satisfaction of the BIC criteria.

3.2.4. Equiprobable input/output XOR distribution

For the analyzed S-box, the high value from the maximum expected differential probability matrix is 12 indicating a few imbalance between the input and output XOR distribution on the S-box.

Table 1: A generated S-box using the proposed technique.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	57	97	54	116	40	213	98	165	231	50	7	73	37	47	46	216
1	19	136	238	83	71	207	84	95	48	86	100	225	151	162	255	240
2	112	24	27	128	67	227	94	169	13	79	138	203	201	233	214	142
3	150	107	117	120	102	194	206	32	145	247	215	5	224	96	23	141
4	51	16	146	186	241	236	1	110	68	44	121	108	133	235	55	64
5	184	223	125	183	26	153	137	56	171	119	135	88	167	33	242	17
6	70	82	15	191	62	244	45	114	105	25	91	219	161	217	18	3
7	188	124	232	66	199	87	36	198	239	34	185	52	60	9	182	22
8	89	139	21	11	101	77	190	63	179	200	144	29	75	58	10	69
9	28	38	156	178	148	158	218	130	211	209	74	14	123	115	189	80
A	249	93	140	61	35	134	131	4	49	174	76	143	250	42	204	92
B	85	163	221	254	234	196	175	237	129	181	164	39	173	222	170	251
C	65	157	126	245	106	78	210	172	147	31	6	127	230	160	180	30
D	220	195	109	2	253	176	104	53	226	205	192	111	248	118	193	8
E	243	177	20	229	122	90	41	168	99	149	81	59	113	154	228	208
F	252	197	212	152	159	0	155	43	187	12	246	72	202	103	166	132

Table 2: S-box evaluation results and comparison.

S-Box	Min. non-linearity	Mean SAC	SAC offset	BIC-SAC	Min. BIC non-linearity	Max. XOR	LP
Proposed SD	100	0.4976	0.0156	0.4997	100	12	0.0549
AES	112	0.5048	0.02637	0.5046	112	4	0.015625
Madani et al. [20]	-	0.4625	-	0.4969	51.1	-	-
Dridi et al. [21]	102	0.4948	-	0.4991	103.42	10	0.1094
Cavuşoğlu et al [22]	104	0.5039	0.03809	0.5058	98	10	0.0791
Dragan Lambić [23]	106	0.5034	0.02441	0.5014	100	10	0.070557
Alhadawi et al. [24]	106	0.4943	-	0.4982	104.35	10	0.1250
Lai et al. [25]	104	0.5014	-	0.5028	102.75	10	0.1250
Al Solami [26]	106	0.5017	-	0.5026	104	10	0.1094
Xuanping et al. [27]	-	0.4965	-	0.4965	109.36	-	-
Dragan Lambić [28]	108	-	0.02954	-	104	8	0.035156
Liu et al. [29]	104	-	0.03027	-	98	10	0.0625
Guesmi et al. [30]	104	-	0.0293	-	96	10	0.0625
Fatih et al. [31]	100	-	0.03125	-	100	10	0.070557
Guo Chen [32]	102	-	0.03174	-	100	10	0.088135
Lambić et al. [33]	106	-	0.03	-	100	10	0.079

3.2.5. LP property

The LP criteria, as defined in [23, 36, 37] detect any imbalance between the selected input and output bits using two masks a and b. The obtained result after analyzing our S-box is equal to 0.0549 satisfying the requirement ($LP < 0.079$) given in [33].

3.2.6. Discussion

As it is discussed in the previous paragraphs, and presented in Table 2, we can conclude that the proposed technique is suitable for the construction of strong random S-boxes while it satisfies the requirements and offers best results compared the literature similar works.

4. Hardware requirements of proposed architecture

To explore the material performance of the proposed architecture, we used the structural description on VHDL language for low level implementation. The Register-Transfer-Level (RTL) description has been realized on the Xilinx PYNQ-Z2 FPGA prototyping board after synthesis, place and route steps on the the Xilinx Vivado design suite tools (V.2022.1) [13]. To ensure the best functionality of our design, we performed simulation tests at the different levels of design flow, behavioral, post-synthesis functional, post-synthesis timing, post-implementation functional, and post-implementation timing. After the success of these simulations we generated the bit-file and we programmed the FPGA chip.

4.1. Utilization, timing, and power reports analysis

The main information given on the report-utilization generated by the Xilinx Synthesis Technology (XST) after place and route, the timing metrics, and power requirements are presented in Table 3. As we can see, the designed architecture occupies low logic resources on the used Xilinx PYNQ-Z2 xc7z020clg400-1 FPGA device. More precisely, it requires only 1135 (2.13%) Slice LUTs (743 LUT as Logic and 392 LUT as Distributed RAM), 762 (0.72 %) Slice Registers (Register as Flip Flop), and 8 (3.64 %) DSP48E1. The mean of these tree main parameters (2.16 %) show that the available resources are used efficiently. In terms of timing metrics, the design can reach running frequency of 78.62 Mhz according to Equation 6. Where T = 13 ns and WNS = 0.28 ns (Worst Negative Slack, defined in Vivado implementation report. It gives the worst slack of all the timing paths. It is negative if a timing violation is detected in any path and positive, like our study, if all the paths satisfies the timing requirement). Therefore, the 32-bits stream-cipher generation can reach a throughput of 2515.84 Mbps according to Equation 7. If we consider that the architecture will be executed uniformly on the used logic Slices, we define the efficiency parameter according to Equation 8.

$$Max\_Freq = \frac{1}{T - WNS}\ [MHz] \tag{6}$$

$$Throughput = N \times Max\_Freq\ [Mbps] \tag{7}$$

$$Efficiency = \frac{Throughput}{Slices}\ [Mbps/Slices] \tag{8}$$

The power report indicates a total On-Chip consumption of 0.188 W (43 % dynamic and 57 % static). Therefore, in addition to hardware and timing metrics, this low energy requirement of the architecture favorite its utilization on embedded electronic and real-time data protection applications, like smartphone and IoT (Internet of Think) objects or devices.

5. Security evaluation and discussion

To evaluate the security performance of the proposed dynamic S-box-based ZUC stream cipher, we investigated its resilience against cryptanalysis attacks using the most useful tests known for their effectiveness in validating cryptosystems such as NIST statistical tests, keystream uniformity, keystream randomness, entropy, confusion, and diffusion properties, key sensitivity, and key space.

All the simulations have been implemented in Python 3.7 on a standard computer Intel(R) Core(TM) i7-10710U CPU 1.10 GHz operating under Microsoft Windows 10, 64-bit, 16 GB RAM, and 1.6 GHz cpu-speed.

5.1. Uniformity and key-stream distribution analysis

To evaluate the uniformity of a key-stream generated by the proposed algorithm, we encrypted different images (Figures 5(a), 5(b), 5(c), 5(d), 5(e)) of size 512 × 512 pixels using 2097152 generated bits. Then, we expected the histogram distribution of both the plain and encrypted images in each case. As we can notice in Figure 6 (row 2), the encrypted images are uniformly distributed and spatially spread. Unlike plain images following a distribution concentrated on a defined area of pixels, but not on others (see Figure 5, row 2). Therefore, we conclude that the proposed dynamic S-boxes improves the randomness of the generated output key-stream and ciphered data.

5.2. Uniformity and Chi-Square analysis

To confirm statistically the uniformity accurately of the generated key-stream and cipher-text, we explored the Chi-Square value [38] using Equation 9.

$$\chi_{exp}{}^2 = \sum_{i=1}^{N_c-1} \frac{(O_i - E_i)^2}{E_i} \tag{9}$$

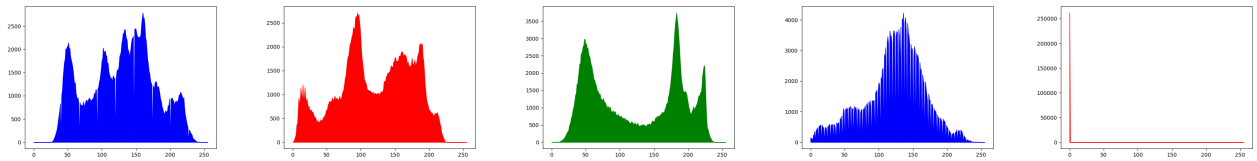
Where $N_c = 2^8 = 256$ is the number of levels, O_i is the calculated occurrence frequency of each gray level, $i \in [0, 255]$ in the histogram of the ciphered image, and E_i is the expected occurrence frequency of the uniform distribution, calculated by $E_i = nb/N_c$. The theoretical value for $\alpha = 0.05$ and $N_c = 256$ is $\chi^2_{th}(255, 0.05) = 293.24$.

The mean value of the experimental Chi-square χ^2_{exp} over 20 cipher images is equal to $\chi^2_{exp} = 263.73$. The obtained value is consistent with the expectations of the definition for this test which considers a uniform cipher-text if the experimental value of its Chi-square is less than the theoretical value, as our case ($\chi^2_{exp} = 263.73 < \chi^2_{th} = 293.24$). According to this analysis, we conclude that the uniformity is confirmed by both the histogram distribution and Chi-square value.

Table 3: FPGA implementation results of the proposed dynamic-Sboxes-based ZUC stream cipher.

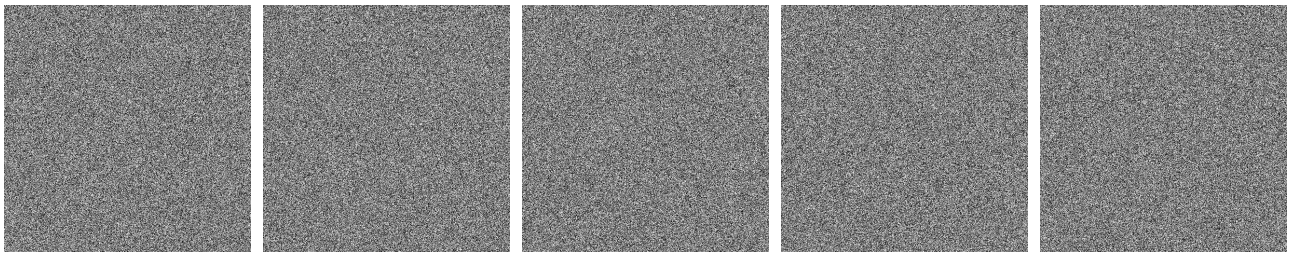
	Parameters	Area Utilization	Area Utilization in %
Board	Family Device	Zynq-7000 7z020-clg400	
Hardware resources	Slice LUTs FFs DSP	374 1135 762 8	(2.81 %) (2.13 %) (0.72 %) (3.64 %)
Time metrics	WNS (ns) Maximum Frequency (MHz) Throughput (Mbps)	0.29 78.68 2515.84	
Efficiency	Efficiency (Mbps/Slices)	6.73	
Consumption	Power (Watts)	0.188	

(a) Lena. (b) Peppers. (c) Lake. (d) Couple. (e) Black.

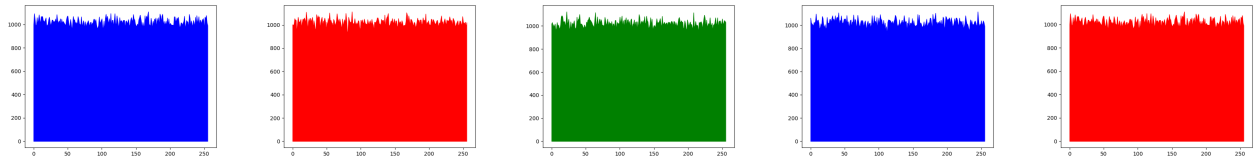


(f) Lena ditribution. (g) Peppers ditribution. (h) Lake ditribution. (i) Couple ditribution. (j) Black ditribution.

Figure 5: Plain images and their distributions.



(a) Lena encrypted. (b) Peppers encrypted. (c) Lake encrypted. (d) Couple encrypted. (e) Black encrypted.



(f) Lena encrypted ditribution. (g) Peppers encrypted ditribution. (h) Lake encrypted ditribution. (i) Couple encrypted ditribution. (j) Black encrypted ditribution.

Figure 6: Encrypted images and their distributions.

5.3. Hamming distance and plain-text sensitivity analysis

To test, the sensitivity to any change on the plain-text, we calculate the average Hamming Distance (HD) between the plain-image (P) and the corresponding cipher-image (C), as given by Equation 10 over 20 different plain images.

$$HD(P,C) = \frac{1}{|N|}\sum_{k=1}^{N}(P[k] \oplus C[k]) \times 100\% \quad (10)$$

Where N is the size in bit of the plain and cipher images.

The obtained results presented in Figure 7 are very close to the optimal value 50%, as defined by the avalanche effect [39] indicating that the probability of bit changes between each ciphered-text and its corresponding plain-text is 50%.

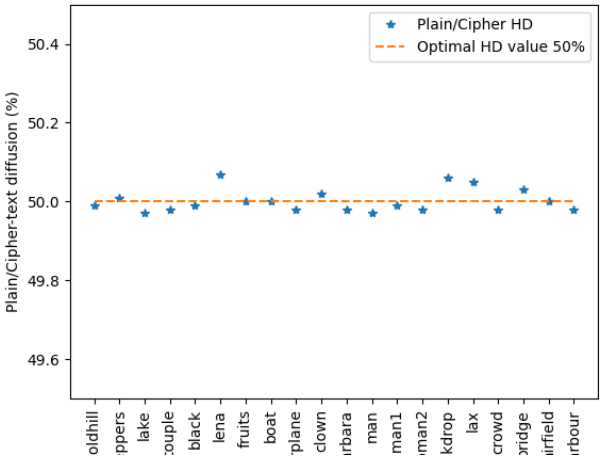


Figure 7: The plain-text sensitivity HD results.

5.4. Hamming distance and secret key sensitivity analysis

Similarly to the previous test, we evaluated the sensitivity to few change on the secret key. The test was performed by ciphering the same plain-image twice using two keys with only one bit of difference to obtain two ciphered-images C_1 and C_2. Then we calculate the HD between C_1 and C_2 using Equation 11 over 100 different secret keys.

$$HD(C_1,C_2) = \frac{1}{|N|}\sum_{k=1}^{N}(C_1[k] \oplus C_2[k]) \times 100\% \quad (11)$$

The obtained results presented in Figure 8 are also very close to the optimal value 50% indicating that a change of only one bit in the secret key leads to a thoroughly different key-stream. This proves the high sensitivity of the proposed ZUC stream to the secret key as defined by the avalanche effect [40] with respect to the confusion property given by Shannon's theory [41, 42]. This means that the complex statistical relationship between the secret key, the plain image, and the encrypted image makes it difficult to recover the secret key even with knowledge of multiple plain-encrypted image pairs.

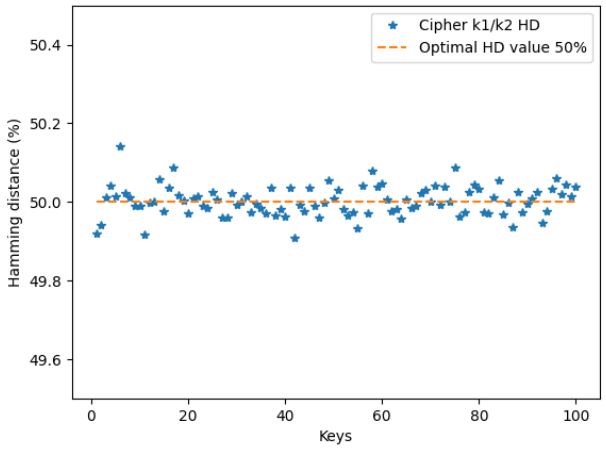


Figure 8: The key sensitivity HD results.

5.5. Key space analysis

The key space of the enhanced ZUC design is improved from 2^{128} to 2^{256} thanks to the use of two dynamic keys to run the chaotic maps and generate the new S-boxes (SD_0 and SD_1). The principle was the combination use of both 128-bits CK and IV to generate the keys KD_1 and KD_2. Therefore, any change in the value of CK or IV leads to the generation of a new S-boxes and a new key-stream, which makes brute-force attacks infeasible.

5.6. NIST statistical tests analysis

For a thorough analysis of the properties of the generated keystream, we used the NIST battery of statistical tests [8, 7, 15, 18, 19]. To explore the fifteenth test, we analyzed a set of 100 generated sequences given by the proposed algorithm. In all the experiments, we set the significance level to 0.01. From the obtained results shown in Table 4, we remark that the proposed ZUC design passes in success all the NIST tests, which prove the high robustness and the best statistical properties of our architecture allowing us to ensure a high-level protection of digital data (text, image, etc.).

5.7. Discussion

As we presented in the above subsections, all the applied experimental results prove the best performance and the enhancement of the generated key-stream. Starting by the uniformity proved by the histogram distribution and the Chi-square value. Then, the sensitivity to any changes in both the plain-text and the secret key proved with respect to the avalanche effect [40] and Shannon's theory [41]. After that, the randomness and the statistical properties proved by the NIST tests. And finally, the complexity of secret key cracking has been doubled by improving the key space from 2^{128} to 2^{256}. Additionally, the high level of the proposed dynamic S-box

Table 4: NIST test results.

Number of test	Type of test	P-Value	Result
1	Frequency (mono-bit) Test	0.437749	success
2	Frequency Test within a Block	0.407566	success
3	Runs Test	0.942123	success
4	Tests for the longest-Run-of-ones in a Block	0.349813	success
5	Binary Matrix Rank Test	0.730751	success
6	Discrete Fourier Transform (Spectral) Test	0.076546	success
7	Non-overlapping Template Matching Test	0.574824	success
8	Overlapping Template Matching Test	0.884123	success
9	Maurer's "Universal Statistical" Test	0.238481	success
10	Linear Complexity Test	0.523428	success
11	Serial Test	0.945384	success
12	Approximate Entropy Test	0.583708	success
13	Cumulative sums Test	0.811180	success
14	Random excursion Test	0.711607	success
15	Random excursion variant Test	0.551820	success

as proved by the main useful criteria (bijection, SAC, non-linearity, BIC, equiprobable input/output XOR distribution, differential approximation probability, maximum expected linear probability) and by the comparison with the literature similar works enforce the whole security of the cryptosystem based on this strong S-box. This means that we have strengthened the resistance of the ZUC stream cipher against cryptanalysis attacks such as brute force attacks, statistical attacks, linear attacks, and differential attacks.

Consequently, we conclude that the combination of the ZUC stream cipher with the proposed dynamic chaotic S-boxes layer increases the data protection for LTE and the new generation of mobile networks.

6. Conclusion

In this article, we have improved the internal architecture of the standardized ZUC stream cipher by combining the original design with a chaos-based generator responsible for generating two dynamic S-boxes (SD_0 and SD_1) in place of the basic static S-boxes (S_0 and S_1). Then, we performed its FPGA-based (Xilinx XC7Z020 ZYNQ platform) implementation using a VHDL description structural language to reach the high performance metrics in terms of material logic resources (Slice LUT, Slice FF, and DSP), and timing requirements (Maximum frequency, WNS, and Throughput). We have also presented the security robustness of the enhanced algorithm as any new proposed cryptosystem.

By analyzing the results obtained, we conclude that the proposed design is adapted to real-time data transmission while achieving a high throughput. In addition, it is suitable for embedded applications while occupying a low area and consuming low energy. Finally, it can ensure a secured transmission of digital data in mobile and IoT networks (it guarantees confidentiality and integrity protections) while resisting brute force, statistical, and differential attacks without modification to the standardized requirements.

In our future work, we will explore how to lighten the computations of the NLF layer while keeping the same level of security. We will also aim to improve the temporal performance to achieve an encryption throughput as close as possible to the order of Gbps.

Conflict of Interest The authors declare no conflict of interest.

Acknowledgment This work was supported by the Bourgogne Franche-Comte region as part of the ANER number 2024PRE00022 project entitled CIAPD.

References

[1] "Specification of the 3GPP Confidentiality and Integrity Algorithms EEA3 & EIA3; Document 1: EEA3 and EIA3 specifications," Technical specification (TS) TS 35.221 V12.0.0, 3GPP, 2014-09.

[2] "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; 3GPP System Architecture Evolution (SAE); Security Architecture," Technical Specification (TS) ETSI TS 133 401 V11.5.0, 3GPP, 2012-10.

[3] "Specification of the 3GPP Confidentiality and Integrity Algorithms EEA3 & EIA3; Document 2: ZUC specification," Technical specification (TS) TS 35.222 V12.0.0, 3GPP, 2014-09.

[4] M. J. AlMashrafi, "A different algebraic analysis of the ZUC stream cipher," in Proceedings of the 4th international conference on Security of Information and Networks (SIN), 139–153, ACM New York, NY, USA ©2011, Sydney, Australia, 2011, doi:10.1145/2070425.2070455.

[5] W. Hongjun, H. Tao, H. Phuong, W. Huaxiong, L. San, "Differential Attacks against Stream Cipher ZUC," in International Conference on the Theory and Application of Cryptology and Information Security, 262–277, ASIACRYPT 2012: Advances in Cryptology, 2012, doi:10.1007/978-3-642-34961-4 17.

[6] F. Lafitte, O. Markowitch, D. Van Heule, "SAT based analysis of LTE stream cipher ZUC," Journal of Information Security and Applications, **22**, 54–65, 2013, doi:10.1016/j.jisa.2014.09.004.

[7] M. Madani, I. Benkhaddra, C. Tanougast, S. Chitroub, L. Sieler, "Enhanced ZUC Stream Cipher Based on a Hyperchaotic Controller System," in The Euromicro Conference on Digital System Design DSD 2017, Work In Progress Session, Vienna, Austria, 30 August-1 September 2017.

[8] M. Madani, C. Tanougast, "Combined and Robust SNOW-ZUC Algorithm Based on Chaotic System," in The International Conference on Cyber Security and Protection of Digital Services (Cyber Security 2018), 1168–1173, IEEE, Glasgow, Scotland, UK, 2018.

[9] F. Özkaynak, "Construction of robust substitution boxes based on chaotic systems," Neural Computing and Applications, **31**, 3317–3326, 2019, doi:10.1007/s00521-017-3287-y.

[10] F. Artuğer, F. Özkaynak, "A method for generation of substitution box based on random selection," Egyptian Informatics Journal, **23**(1), 127–135, 2022, doi:10.1016/j.eij.2021.08.002.

[11] K. Mohamed, M. N. Mohammed Pauzi, F. H. Hj Mohd Ali, S. Ariffin, N. H. Nik Zulkipli, "Study of S-box properties in block cipher," in 2014 International Conference on Computer, Communications, and Control Technology (I4CT), 362–366, 2014, doi:10.1109/I4CT.2014.6914206.

[12] A. Msolli, I. Hagui, A. Helali, "Dynamic S-boxes generation for IoT security enhancement: A genetic algorithm approach," Ain Shams Engineering Journal, **15**(11), 103049, 2024, doi:10.1016/j.asej.2024.103049.

[13] "Zynq-7000 SoC Technical Reference Manual," Ug585 (v1.13), Xilinx, 2021.

[14] "PYNQ Z2 Reference Manual," v1. 1, PYNQ™, 2019.

[15] A. Rukhin, et al, "A Statistical Test Suite for the Random and Pseudorandom Number Generators for Cryptographic Applications," NIST Special Publication 800-22, 2001, Revised: April 2010, doi:http://csrc.nist.gov/rng/SP800-22b.pdf.

[16] "Cid C, Murphy S, Pipir F, Dodd M, ZUC algorithm evaluation repport," Technical report, 2010.

[17] "Knudson LR, Preneel B, Rijman V, Evaluation of ZUC," Technical report, 2010.

[18] M. Madani, I. Benkhaddra, C. Tanougast, S. Chitroub, L. Sieler, "FPGA Implementation of an enhanced SNOW-3G Stream Cipher based on a Hyper-chaotic System," in The 4th international conference on Control, Decision and Information Technologies (CoDIT'17), 1168–1173, IEEE, Barcelona, Spain, 2017.

[19] M. Madani, I. Benkhaddra, C. Tanougast, S. Chitroub, L. Sieler, "Digital Implementation of an Improved LTE Stream Cipher SNOW-3G based on Hyperchaotic PRNG," Security and Communication Networks, Hindawi with John Wiley & Sons, **2017**, 15 pages, 2017, doi:10.1155/2017/5746976.

[20] M. Madani, S. El Assad, C. Tanougast, M. J. Vella, E.-B. Bourennane, O. Deforges, "FPGA-Based Implementation of Enhanced ZUC Stream Cipher Based on Dynamic S-Box," in 2023 International Conference on Engineering and Emerging Technologies (ICEET), 1–6, 2023, doi:10.1109/ICEET60227.2023.10526075.

[21] F. Dridi, S. El Assad, W. El Hadj Youssef, M. Machhout, R. Lozi, "Design, Implementation, and Analysis of a Block Cipher Based on a Secure Chaotic Generator," Applied Sciences, **12**(19), 2022, doi:10.3390/app12199952.

[22] Ü. Cavusoğlu, A. Zengin, I. Pehlivan, S. Kacar, "A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system," Nonlinear Dynamics, **87**(2), 1081–1094, 2017, doi:10.1007/s11071-016-3099-0.

[23] D. Lambić, "A novel method of S-box design based on discrete chaotic map," Nonlinear Dynamics, **87**, 2017, doi:10.1007/s11071-016-3199-x.

[24] H. Alhadawi, M. Zolkipli, M. Ahmad, "A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map," Neural Computing and Applications, **31**, 2019, doi:10.1007/s00521-018-3557-3.

[25] Q. Lai, A. Akgul, C. Li, G. Xu, Ü. Çavusoğlu, "A New Chaotic System with Multiple Attractors: Dynamic Analysis, Circuit Realization and S-Box Design," Entropy, **20**(1), 2018, doi:10.3390/e20010012.

[26] E. Al Solami, M. Ahmad, C. Volos, M. N. Doja, M. M. S. Beg, "A New Hyperchaotic System-Based Design for Efficient Bijective Substitution-Boxes," Entropy, **20**(7), 2018, doi:10.3390/e20070525.

[27] Z. Xuanping, Z. Zhongmeng, W. Jiayin, "Chaotic image encryption based on circular substitution box and key stream buffer," Signal Processing: Image Communication, **29**(8), 902–913, 2014, doi:10.1016/j.image.2014.06.012.

[28] L. Dragan, "A novel method of S-box design based on chaotic map and composition method," Chaos, Solitons & Fractals, **58**, 16–21, 2014, doi:10.1016/j.chaos.2013.11.001.

[29] G. Liu, W. Yang, W. Liu, Y. Dai, "Designing S-boxes based on 3-D four-wing autonomous chaotic system," Nonlinear Dynamics, **82**, 2015, doi:10.1007/s11071-015-2283-y.

[30] R. Guesmi, M. A. Ben Farah, A. Kachouri, M. Samet, "A novel design of Chaos based S-Boxes using genetic algorithm techniques," in 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), 678–684, 2014, doi:10.1109/AICCSA.2014.7073265.

[31] F. Özkaynak, A. B. Özer, "A method for designing strong S-Boxes based on chaotic Lorenz system," Physics Letters A, **374**(36), 3733–3738, 2010, doi:10.1016/j.physleta.2010.07.019.

[32] G. Chen, "A novel heuristic method for obtaining S-boxes," Chaos, Solitons & Fractals, **36**(4), 1028–1036, 2008, doi:10.1016/j.chaos.2006.08.003.

[33] M. Š. Dragan Lambić, Publications de l'Institut Mathématique, (113), 109–115.

[34] T. Cusick, P. Stănică, Cryptographic Boolean Functions and Applications: Second edition, 2017.

[35] A. F. Webster, S. E. Tavares, "On the Design of S-Boxes," in H. C. Williams, editor, Advances in Cryptology — CRYPTO '85 Proceedings, 523–534, Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.

[36] L. Keliher, H. Meijer, "A New Substitution-Permutation Network Cipher Using Key-Dependent S-Boxes," in H. C. Williams, editor, SAC 97, 13–26, 1997.

[37] L. Keliher, "Refined Analysis of Bounds Related to Linear and Differential Cryptanalysis for the AES," volume 3373, 42–57, 2004, doi:10.1007/11506447_5.

[38] S. G. Meintains, Z. HLÁÁVKA, "Goodness-of-Fit Tests for Bivariate and Multivariate Skew-Normal Distribution," Scandinavian Journal of Statistics, **37**(4), 701–714, 2010, http://www.jstor.org/stable/41000416.

[39] D. Han, L. Min, G. Chen, "A Stream Encryption Scheme with Both Key and Plaintext Avalanche Effects for Designing Chaos-Based Pseudorandom Number Generator with Application to Image Encryption," International Journal of Bifurcation and Chaos, **26**(5).

[40] D. Han, L. Min, G. Chen, "A Stream Encryption Scheme with Both Key and Plaintext Avalanche Effects for Designing Chaos-Based Pseudorandom Number Generator with Application to Image Encryption," International Journal of Bifurcation and Chaos, **26**(5), 2016, doi:10.1142/S0218127416500917.

[41] C. Shannon, "Communication Theory of Secrecy Systems," Bell Systems Technical Journal, **28**, 656–715, 1949.

[42] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. Noonan, P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," Inf. Sci, **222**, 323–342, 2013.