







## Efficient Design for a Hardware Implementation of the LED Block Cipher

Ayoub Mhaouch<sup>1\*</sup>, Wajdi Elhamzi<sup>2</sup>, Abdessalem Ben Abdelali<sup>1</sup>, Mohamed Atri<sup>3</sup>

<sup>1</sup>Laboratory of Electronics and Microelectronics (EμE), Faculty of Sciences of Monastir, University of Monastir, Monastir 5000, Tunisia

<sup>2</sup>College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-kharj 11942, Saudi Arabia

<sup>3</sup>College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia

Corresponding Author Email: [ayoubmhaouch46@gmail.com](mailto:ayoubmhaouch46@gmail.com)

<https://doi.org/10.18280/jesa.560502>

### ABSTRACT

**Received:** 7 May 2023

**Revised:** 11 October 2023

**Accepted:** 19 October 2023

**Available online:** 31 October 2023

#### Keywords:

*lightweight cryptography, hardware implementation, LED block cipher, key fob, high-performance, security analysis, low-resource*

Recently, most modern cars can be controlled with the Remote Keyless System (RKS). A Remote Keyless System consists of a key fob that communicates wirelessly with the car transceiver that is used to control and secure access to the vehicle. The keyless entry systems are based on RFID communication and are equipped with limited computation and power resources. It was proven to be a target for cybercriminals. In this work, we proposed the use of the LED block cipher for data encryption-decryption of the keyless entry systems. The proposed hardware architecture of the LED algorithm was optimized to fit the limited resources of the keyless entry systems. The proposed 8-bit sequential architecture contains 627 LUTs+FFs. To further reduce hardware resources, we proposed a 4-bit architecture for the MixColumns sub-function, which gives good results in occupying hardware resources by reducing it to 597 LUTs+FFs. This leads to a good effect on the area implementation and power consumption of keyless entry systems. Thus, the proposed LED hardware architecture may be applied to lightweight applications that demand a high level of secrecy such as the key fob. In order to adopt the proposed design of the LED block cipher as a security system for keyless entry systems, we examine the security of the proposed LED architecture using five metrics; entropy analysis, histogram analysis, correlation analysis, NPCR, and UACI. As a result, the LED block cipher has a good ability to encrypt any data against any attack.

## 1. INTRODUCTION

Modern cars are becoming smarter through a collection of automotive systems such as braking, power, locks, and so on. As the number of functionalities in vehicles explodes, the software needed to manage that functionality will highly increase, presenting the best opportunities for hackers looking to attack vehicles. As important systems, key fobs are becoming more complex. In addition to unlocking the vehicle and enabling the ignition, key fobs can be used to control windows and mirrors, set seats, and enable radio [1]. As an example, Tesla key fobs can be used to start automated parking and parking.

Recently, keyless vehicles have been increasingly targeted by burglars. Theft of automobiles is accomplished via the reprogramming of remote-entry keys [2]. For this reason, several insurance companies have rejected the insurance for this issue. Keyless entry systems are a system that has an RFID (Radio-Frequency IDentification) tag for wireless communication to the vehicle. As a result of RFID technology, keyless entry systems are becoming more vulnerable to several security threats [3]. It is vulnerable to a Scan Attack, Playback Attack, Two-Thief Attack, Challenge Forward Prediction Attack, and Dictionary Attack [4]. Also, On-Board-Diagnose (OBD) key programmers are another risk to Remote Keyless Entry Systems (RKES). For systems that employ the rolling code technique, the scanning attack is very effective [5]. It is possible to launch the scan attacks against these systems, by

sending several codes to the automobile transceiver and making sure they match the transceiver's code. Using the OBD key programmers [6], can be used to unlock vehicles, where an attacker can replicate all key information to a keyless entry system that has not been programmed.

To solve these problems, an encryption system is an efficient security system to ensure data protection for the keyless entry system. The mentioned features make a keyless entry system encryption an attractive target for cyberattacks. So, hackers will try to break the encryption and clone the RKES. Glocker et al. [4] proposed the symmetric encryption algorithms as a solution to the keyless system. As a result, the use of symmetric encryption algorithms in the keyless system gives a good level of security to protect private data. However, it hurts hardware resources and power consumption.

As the keyless entry systems are based on a RFID transponder to communicate with the vehicle, the available computation resources are very limited which prevents the implementation of powerful encryption algorithms [6]. So, the invention of lightweight cryptography algorithms is considered a powerful solution. In the last decade, a large number of research work has already been carried out to propose a lightweight security system that is suitable for resource-constrained applications. Lightweight encryption was provided as a security solution for highly constrained devices and embedded systems [7]. In this field, several lightweight encryption algorithms have been proposed, for instance, PRESENT [8], LED [9], Piccolo [10], KATAN [11],

and SPARK [12].

The main objective of this work is to propose an efficient hardware architecture of the lightweight block cipher for implementation on a keyless entry system. To that end, we propose a hardware-serial architecture to improve the LED block cipher in terms of area implementation for use as a security system on a keyless entry system. The LED algorithm was designed especially for embedded devices with limited computation resources (e.g., Memory, area, and power). The LED block cipher uses input data of 64 bits and a key length of 128 bits. The main idea of this work was to read the data by 8 bits through 16 cycles. Then, to speed up the processing time, an 8-bit architecture of the MixColumns process was proposed to reduce the latency.

The following is a summary of the paper's contributions:

- LED block cipher proposal as a security system to protect private vehicle information in a keyless entry system.
- Propose two hardware architectures of the LED block cipher to select an efficient architecture.
- Test the security level of the proposed LED algorithm with five metrics: NPCR, UACI, entropy, histogram, and correlation.

The rest of the paper is organized as follows: The description of LED Block Cipher is presented in Section 2. The proposed hardware designs of the LED algorithm are detailed in Section 3. The performance and the security analysis of the proposed designs are presented in Section 4. Finally, the conclusion is drawn in Section 5.

## 2. DESCRIPTION OF LED BLOCK CIPHER

Already an important aspect of keyless entry systems, an LED block cipher is a lightweight block cipher designed for resource-constrained environments, making it suitable for small devices such as key fobs and smart cards. The LED algorithm provides efficiency and a lightweight design, making it suitable for keyless entry systems with limited resources. However, its security is limited by its small block size, and it may not be appropriate for applications requiring high levels of security. When LED block cryptography is used to ensure the security of keyless entry systems, the keyless

entry systems are robust against side-channel and similar attacks.

LED (lightweight encryption devices) is a lightweight encryption algorithm based on an S-PN (substitution-permutation network). It was proposed by Guo et al. [9] in 2011. LED is an algorithm that needs 64 bits to implement the encryption or decryption process. LED block cipher can be used with two versions of key lengths (46-bit or 128-bit). To generate the ciphertext, the encryption process for the LED algorithm requires 32 clock cycles or 48 clock cycles for a 64-bit or 128-bit key length. The encryption and decryption process of the LED algorithm is shown in Figure 1.

In order to understand how the data processing through the LED algorithm, Figure 2 shows the graph of a single LED encryption round, in which the encryption process has five sub-functions: AddRoundKey, AddConstants, S-Boxes, ShiftRows, and MixColumns.

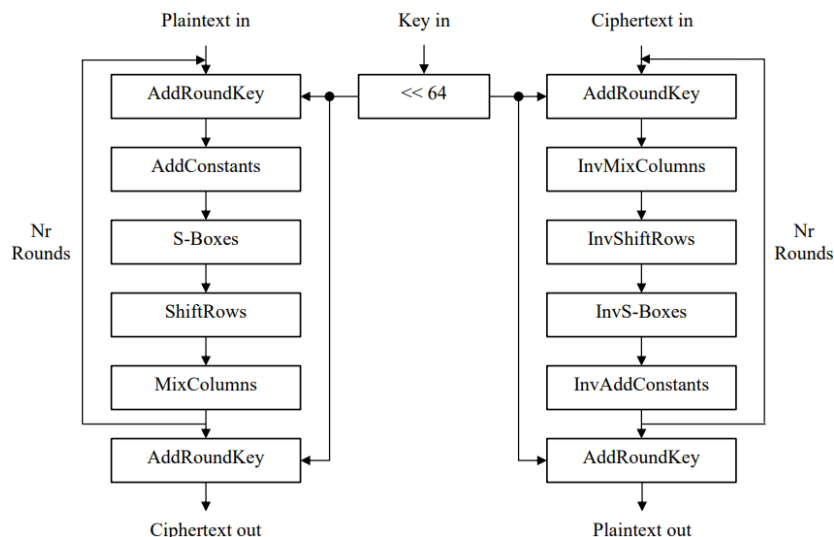
In each round, the input data **M** (i.e., the 64-bit data path) consisting of 16 blocks (**A0**, through **A15**) is arranged in a four-by-four bits matrix. The key addition layer (AddRoundKey) is the first operation of the LED encryption process, it is to add a 64-bit of the subkey **Sk<sub>i</sub>** with a 64-bit of state. The two inputs are combined through a bit XOR operation, where the XOR operation is equal to addition. Then, the round constants **RCST** are combined with the output of the first process. The round constant **RCST** is defined as follows in (1):

$$RCST = \begin{pmatrix} 0 \oplus (K_{s_7} \parallel K_{s_6} \parallel K_{s_5} \parallel K_{s_4}) & (RC_5 \parallel RC_4 \parallel RC_3) & 0 & 0 \\ 1 \oplus (K_{s_7} \parallel K_{s_6} \parallel K_{s_5} \parallel K_{s_4}) & (RC_2 \parallel RC_1 \parallel RC_0) & 0 & 0 \\ 2 \oplus (K_{s_3} \parallel K_{s_2} \parallel K_{s_1} \parallel K_{s_0}) & (RC_5 \parallel RC_4 \parallel RC_3) & 0 & 0 \\ 3 \oplus (K_{s_3} \parallel K_{s_2} \parallel K_{s_1} \parallel K_{s_0}) & (RC_2 \parallel RC_1 \parallel RC_0) & 0 & 0 \end{pmatrix} \quad (1)$$

As shown in Figure 2, the third layer in each round is the 4-bit Substitution operation. The Substitution or S-box layer can be viewed as a row of 16 parallel S-boxes. Each element of the state **A<sub>i</sub>** is replaced by another element **B<sub>i</sub>** using lookup tables with special mathematical properties, as given in Table 1.

**Table 1.** 4-bit Substitution table for the LED algorithm

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2



**Figure 1.** The encryption and decryption process of the LED algorithm

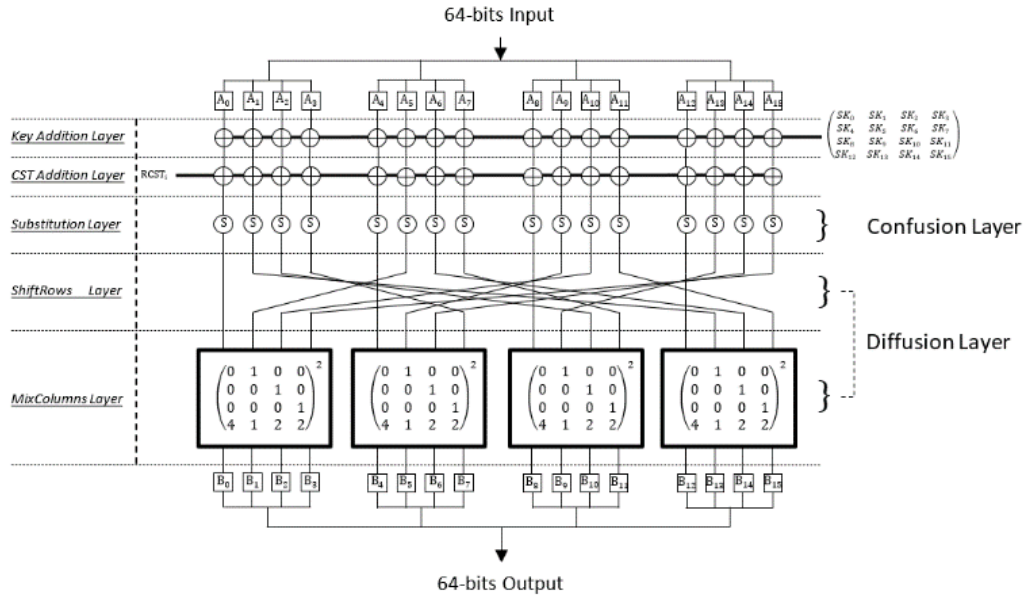


Figure 2. Graph of a single LED encryption round

The diffusion layer of the LED algorithm consists of two sub-layers: ShiftRows and MixColumn transformations. The diffusion layer is an important layer to enhance the security level of any encryption algorithms, it is the influence of individual bits on the entire state.

The ShiftRows operation is a cyclic shift, shifting the second row of the state matrix cyclically by two elements to the right. The third and fourth rows are cyclically shifting two and three items to the right, respectively.

The MixColumn operation is the main diffusion element in the LED algorithm where every change in a 4-bit input directly affects the 16-bit output. Each 16-bit column of the state after the ShiftRows operation is considered a vector and multiplied by the **MDS** matrix. The matrix **MDS** contains constant entries, as presented in Eq. (2).

$$MDS = \begin{pmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 1 & 2 & 2 \end{pmatrix}^2 \quad (2)$$

### 3. PROPOSED DESIGN OF THE LED ALGORITHM

The encryption process of the LED algorithm consists of three blocks: an encryption round block, a control block, and a key scheduler. The key scheduler is a process that is used to generate 48 subkeys for 48 LED rounds using the original input key, which has a length of 128 bits. Due to the key required for key whitening in the final key addition layer, the number of subkeys is equal to the number of rounds plus one. As a result, for a 128-bit of key length, the key scheduler process generates 48+1 subkeys, each of 64 bits. The subkeys are alternatively equal to the left part or the right part for the original key.

As mentioned in Section 2, the round block of the LED algorithm consists of five layers: 1-AddRoundKey, 2-AddConstants, 3-S-Box, 4-ShiftRows, and 5-MixColumns. All operations require 64 bits of input to generate 64 bits of output, so one round operation takes one clock cycle. In order to propose an efficient hardware design for the LED block

cipher, a hardware architecture for the LED algorithm (128-bit key length) is designed. The proposed hardware architecture is illustrated in Figure 3.

In the proposed architecture, the round block is designed on 8-bit for the datapath encryption, in which most of the LED sub-functions are reduced to 8 bits (AddRoundKey, AddConstants, and S-Box). Thus, these processes take 8 clock cycles. ShiftRows is a process that requires only 64-bit input to generate 64-bit output. For this reason, the ShiftRows process is performed on a one-clock cycle that is separate from other processes.

The MixColumns layer requires a minimum of 16-bit input to be multiplied by the MDS matrix. So, it needs sixteen clock cycles.

$$\begin{pmatrix} C_0 \\ C_4 \\ C_8 \\ C_{12} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 1 & 2 & 2 \end{pmatrix}^2 \times \begin{pmatrix} S_0 \\ S_4 \\ S_8 \\ S_{12} \end{pmatrix} \quad (3)$$

To achieve efficient hardware implementation of the LED algorithm, two architectures are designed for the MixColumns process. Figure 3 shows the proposed architectures of the MixColumns process for the LED algorithm.

As shown in Figure 4, the 4-bit architecture of the MixColumns process takes 16 clock cycles, which affects the performance of the proposed hardware architecture of the LED algorithm. As a result, one round operation of the proposed hardware architecture of the LED algorithm takes 25 clock cycles and requires 1200 total cycles to generate the ciphertext. To reduce the latency of this process, we propose an 8-bit architecture for the MixColumns process. The proposed design takes 8 clock cycles, which can reduce the total latency of the proposed LED design to 17 cycles, and requires 816 total cycles to generate the ciphertext.

The 8-bit design for the LED block cipher was carefully considered based on several factors that make it advantageous for our specific application. While there are various design possibilities, this design offers key benefits over others:

**Efficiency for Resource-Constrained Devices:** Our primary consideration was the efficiency of the design,

especially for resource-constrained devices commonly found in keyless entry systems. The 8-bit design minimizes computational overhead, making it well-suited for devices with limited processing power and memory.

**Real-Time Requirements:** Keyless entry systems often require real-time responsiveness, and the 8-bit design ensures rapid encryption and decryption. This is crucial for the

seamless and quick operation of such systems.

**Compatibility and Integration:** Many existing keyless entry systems utilize 8-bit microcontrollers. Choosing an 8-bit design ensures compatibility and smooth integration with these systems, minimizing the need for extensive hardware upgrades.

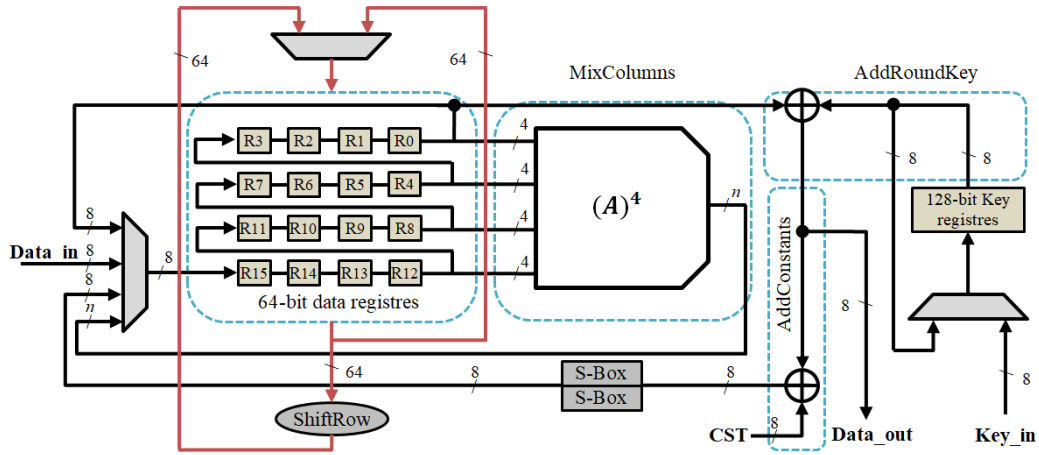


Figure 3. Proposed efficient hardware architecture of the LED algorithm

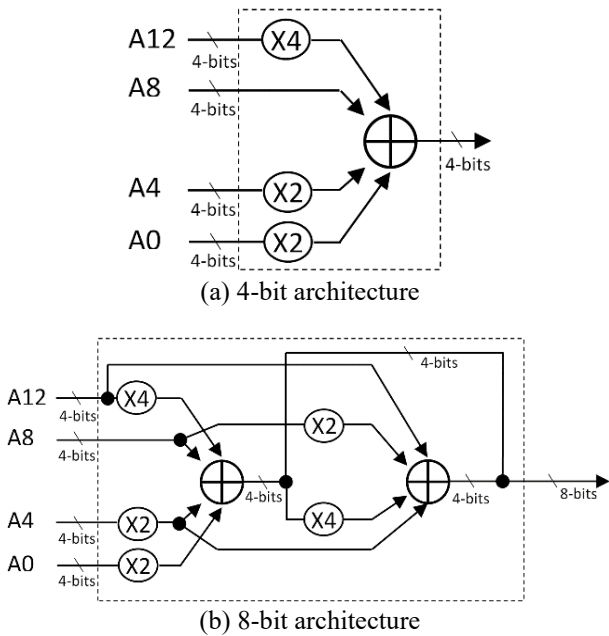


Figure 4. The proposed architecture of the MixColumns process for the LED algorithm

## 4. SECURITY AND PERFORMANCE ANALYSIS

### 4.1 Security analysis

To protect sensitive data, it is required to deploy cryptography algorithms to encrypt data. However, the recent IoT applications are featured with limited resources devices that require lightweight algorithms. In this part, we evaluate the security of the LED algorithm against a recognized statistical analysis. One of the most significant methods of representing information is an image, which is frequently

utilized in fields including telemedicine, medical imaging, and military communication. Over unsecured networks, images are often shared between two parties. As a result, experts and academics are now studying how to safeguard image data from being intercepted, copied, and destroyed. The process of changing the original picture into one that is unintelligible, unrecognizable in appearance, disordered, and unsystematic is known as image encryption. In order to evaluate the security level provided by the LED algorithm (128-bit key length), we perform a statistical study of the different evaluation properties of the encrypted image by the LED algorithm, using 5 metrics: Number of Pixel Change Rate (NPCR), Unified Average Changing Intensity (UACI), image entropy, image histogram, and Correlation analysis.

A robust encryption algorithm is sensitive to a light input fluctuation, even to a one-bit change in the input image. A little change in the input image should result in a large change in the encrypted image as a prerequisite for the image encryption technique to resist the differential attack (NPCR & UACI). The number of pixels change rate (NPCR) and unified average changing intensity (UACI), are used to assess the impact of a one-pixel change in the encrypted image.

NPCR (Number of Pixel Change Rate) is a metric used to evaluate the sensitivity of the encryption algorithm to changes in the input data. It quantifies the percentage of differing pixel values in two ciphertexts generated from plaintexts with a one-bit difference. A high NPCR value, approaching 100%, indicates that the encryption algorithm is highly sensitive to changes in the plaintext, which is generally desirable for cryptographic algorithms. NPCR is calculated using the following equation [13]:

$$NPCR = \frac{\sum_{i=1}^m \sum_{j=1}^n \theta(i, j)}{m \times n} \times 100\% \quad (4)$$

$$\theta(i, j) = \begin{cases} 1 & \text{if } c_1(i, j) \neq c_2(i, j) \\ 0 & \text{if } c_1(i, j) = c_2(i, j) \end{cases}$$

- $n$  is the height of the image in pixels.
- $m$  is the width of the image in pixels.
- $C1$  and  $C2$  represent the corresponding pixel values in two different ciphertexts generated from plaintexts with a one-bit difference.

UACI (Unified Average Changing Intensity) is another important metric used to assess the diffusion properties of an encryption algorithm. It measures the average intensity change in the encrypted image when a one-bit change is made in the plaintext. A lower UACI value indicates that the encryption algorithm achieves a better diffusion effect with smaller intensity changes in the ciphertext. UACI is determined using the following equation [14] :

$$UACI = \frac{\sum_{i=1}^m \sum_{j=1}^n |c_1 - c_2|}{m \times n \times 255} \times 100\% \quad (5)$$

$$\theta(i, j) = \begin{cases} 1 & \text{if } c_1(i, j) \neq c_2(i, j) \\ 0 & \text{if } c_1(i, j) = c_2(i, j) \end{cases}$$

- $n$  is the height of the image in pixels.
- $m$  is the width of the image in pixels.
- $C1$  and  $C2$  represent the corresponding pixel values in two ciphertexts generated from plaintexts with a one-bit difference.

To evaluate the effect of changing a single pixel in the input image on the encrypted image, we performed the analysis using both NPCR and UACI for different images. Table 2 provides a summary of the achieved NPCR and UACI values. According to the results, even with a one-bit difference in the input images, the proportion of pixels altered in the encrypted images is greater than 90.6% for NPCR and greater than 30.9% for UACI, which indicates that the LED algorithm is sensitive to even little changes in the input image.

**Table 2.** Results of the NPCR and UACI metrics for the LED algorithm

	Peppers	Boats	Baboon
NPCR	90.782%	90.639%	91.125%
UACI	31.344%	30.984%	31.152%

In cryptographic security analysis, entropy is a useful measure used to measure the randomness of data output to create an effective cryptographic system. In image processing, entropy is used to measure how much information is contained in the image output of the cryptography algorithm. In general, it is impossible to extract any significant properties from data that have a full entropy. To hide private information, a highly secure algorithm must get high entropy values close to 8. The entropy provides insight into the level of uncertainty. A high entropy value often indicates a great deal of uncertainty, and the ability to predict the following extracted values is greatly increased by a low entropy value. For the security of hash functions and cryptographic systems, entropy analysis is highly recommended. Entropy methods are introduced by Shannon which can be calculated as follows [15]:

$$e(x) = \sum_{i=0}^M P(x_i) \times \log_2(P(x_i)) \quad (6)$$

- $n$  is the number of different pixel values in the image.
- $P(x_i)$  is the probability of occurrence of pixel value  $x_i$  in the image.

where,  $e(x)$  is the entropy of the cipher image,  $P(x_i)$  is the probability of the image pixel intensity  $x_i$ , and  $M$  is the total pixel of the cipher image  $x$ . The entropy values of the tested images are shown in Table 3. The average entropy value for the different encrypted images is  $7.99 \approx 8$ . The results obtained indicate that information leakage from encrypted images is negligible, which means that the encryption algorithm LED is secure against entropy-based attacks.

The histogram analysis is particularly used to lower the likelihood of image attacks and to conceal the private information in input images [16]. The histogram is used to assess how well the encryption process is working. It displays the distribution of gray pixel values in the image, which should be fairly similar to a uniform distribution. In most cases, the recovered histogram of the encrypted image differs greatly from the original image. When the histogram of the encrypted image is completely different and flat than the histogram of the input image, successful attacks might not be feasible in this situation. This type of analysis is highly recommended to reduce the risk of attacks, as the histogram of the encrypted image should be close to uniform distribution. Figures 5-7 illustrate the histogram analysis of the original images and their encrypted using the LED algorithm (128-bit key length). The histogram results of the encrypted images are approximated by a uniform distribution. They are completely different and flat than the histogram of the input image. The uniformity is presented by the chi-square test in Eq. (7):

$$\chi^2 = \sum_{k=1}^{256} \frac{(v_k - 256)^2}{256} \quad (7)$$

where represents the observed occurrence frequencies of each gray level (0–255),  $k$  represents the total number of gray levels (256), and the expected occurrence frequency of each gray level is (256). When comparing the histograms of the encrypted images with those of the input images, we observed several differences in the histogram results between the plain-image and encrypted images. In the case of input images, the histograms often displayed peaks and valleys, indicating varying levels of pixel value concentration. However, the histograms of the encrypted images exhibited a more uniform distribution. The absence of prominent peaks in the encrypted image histograms suggests that the LED algorithm effectively disperses pixel values, minimizing any patterns that could be exploited.

Correlation analysis is used to assess how similar two neighboring pixels are to one another. A low correlation value between adjacent pixels is required to get a secure system. A numerical measure preceding a relationship between two variables is shown by the correlation metric. The link between the original data and its encryption should be eliminated via good encryption. As a result, no meaningful information can be recovered and the link between the plaintext and its encryption cannot be established. In this study, we determined the relationship between the original image and its encrypted output using the LED algorithm. The correlation coefficient is calculated as [17]:



$$r_{u,v} = \frac{\text{cov}(u,v)}{\sqrt{D(u)D(v)}},$$

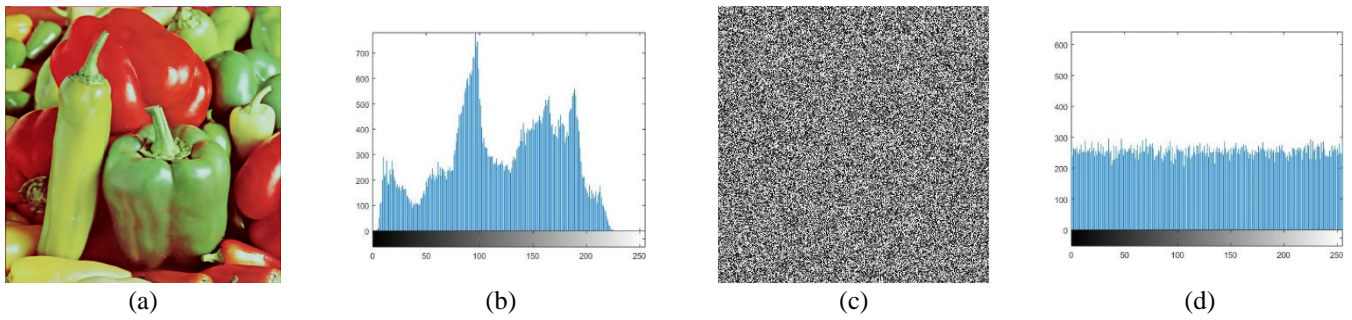
$$D(u) = \frac{1}{N} \sum_{i=1}^N \left( u_i - \frac{1}{N} \sum_{i=1}^N u_i \right)^2, \quad (8)$$

$$\text{cov}(u,v) = \frac{1}{N} \sum_{i=1}^N (u_i - E(u))(v_i - E(v)),$$

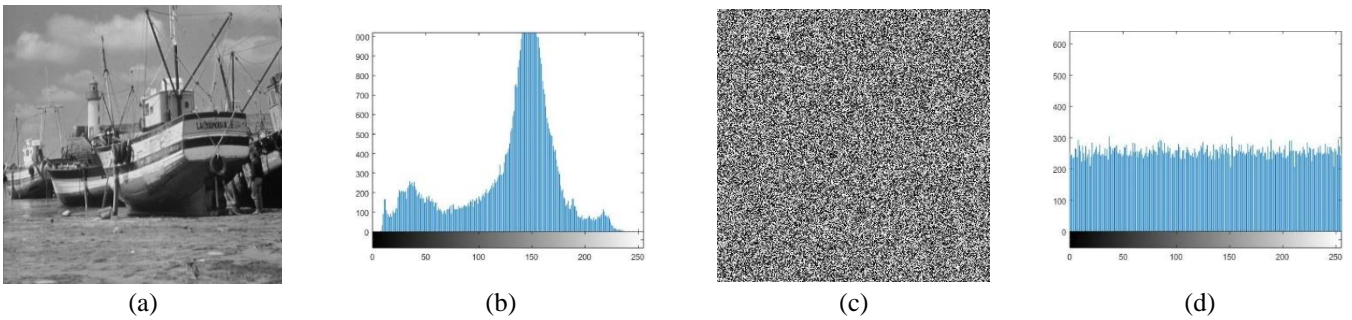
$$E(u) = \frac{1}{N} \sum_{i=1}^N u_i$$

where,  $u$  and  $v$  are gray-scale values of two adjacent pixels in the image. For the perfect cipher, the correlation  $r(u,v)$  should be equal to 0 and it will be equal to 1 for the worst cipher. This criterion is best explained by the theory of Shannon [17]. Table 3 shows the correlation coefficient values for three encrypted images (Pepper image, Boat image, and Baboon image) using the above formulas and the correlations of adjacent pixels are illustrated in Figure 8.

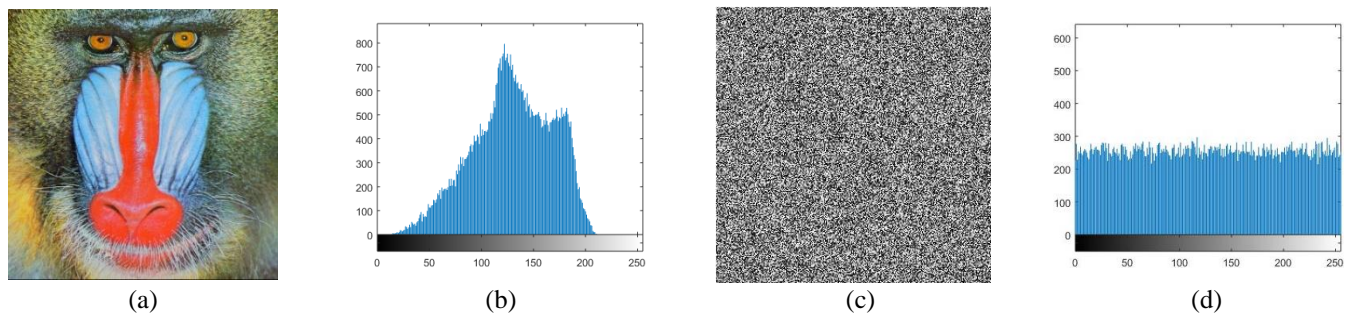
#### 4.2 Performance analysis



**Figure 5.** Histogram results of Pepper image for the LED algorithm. (a) Pepper plain-image, (b) histogram of Pepper plain-image, (c) Pepper cipher-image, and (d) histogram of Pepper cipher-image



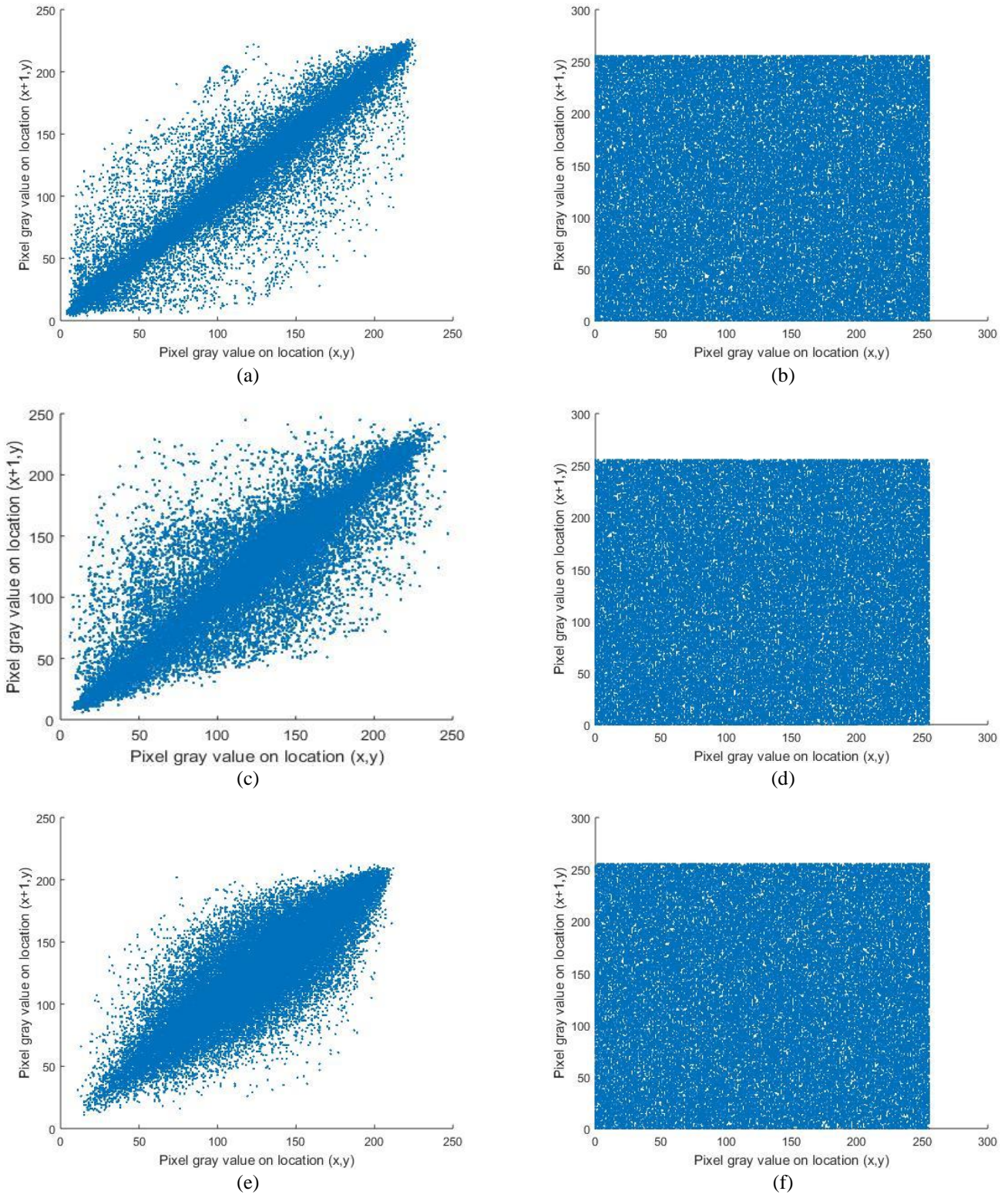
**Figure 6.** Histogram results of Boat image for the LED algorithm. (a) Boat plain-image, (b) histogram of Boat plain-image, (c) Boat cipher-image, and (d) histogram of Boat cipher-image



**Figure 7.** Histogram results of Baboon image for the LED algorithm. (a) Baboon plain-image, (b) histogram of Baboon plain-image, (c) Baboon cipher-image, and (d) histogram of Baboon cipher-image

A hardware implementation of the proposed designs of the LED algorithm was carried out using VHDL description language. The Xilinx ISE 14.7 was used to simulate the proposed implementations. To evaluate the performance of the proposed designs, we used a Spartan-3 FPGA device. Table 4 presents the obtained results for the proposed designs on the Spartan-3 FPGA and an existing FPGA implementation for some block cipher algorithms.

To evaluate the performance of the proposed implementations, we focus on the area consumption expressed as the number of slices, flip-flops (FFs), and LUTs consumed for each design. Also, the speed performance is evaluated and expressed in terms of latency (clock cycles) and throughput (Mbps). The proposed 4-bit MixColumns have a higher latency than the proposed 8-bit MixColumns, which directly affects the performance of the proposed hardware implementation of the LED algorithm. The proposed designs have approximately the same Slices for resources as those used in FPGAs (Spartan-3).



**Figure 8.** Correlations of adjacent pixels for (a) Pepper plain-image, (b) Pepper cipher-image, (c) Boat plain-image, (d) Boat cipher-image, (e) Baboon plain-image, (f) Baboon cipher-image

**Table 3.** Correlation and entropy results for Peppers, boats, and baboons of size  $256 \times 256$

	Peppers		Boats		Baboon	
	Plain-image	Cipher-image	Plain-image	Cipher-image	Plain-image	Cipher-image
$e(x)$	7.577	7.998	7.158	7.998	7.228	7.998
$r_{u,v}$	0.964	0.001	0.927	0.002	0.874	0.002



**Table 4.** Results of various hardware implementations of lightweight encryption algorithms on FPGA

Design	Size (bits)		Area				Speed			Efficiency	FPGA Device
	Key	Datapath	Slices	FFs	LUTs	LUTs+FFs	Clock Cycles	Freq. (MHz)	Throughput (Mbps)	Eff. (Mbps/slices)	
<b>Proposed 4-bit</b>	128	8	210	212	385	597	1200	102.89	5.48	0.02	
<b>Proposed 8-bit</b>	128	8	222	216	411	627	816	123.22	9.66	0.04	
<b>Piccolo [18]</b>	128	4	265	260	442	702	496	45.85	5.92	0.02	
<b>Piccolo [18]</b>	128	64	397	207	757	964	31	81.82	168.9	0.49	Spartan-3
<b>AES [19]</b>	128	8	393	-	-	-	534	-	16.86	0.04	XC3S50-5
<b>Klein [20]</b>	80	4	-	194	597	791	-	116	26	-	
<b>Lilliput [20]</b>	80	4	-	205	592	797	-	119.2	28	-	
<b>Piccolo [21]</b>	128	8	271	260	512	772	248	47.83	12.34	0.04	
<b>Piccolo [21]</b>	128	16	281	241	532	773	124	47.63	24.58	0.08	
<b>Piccolo [21]</b>	128	32	301	248	575	823	62	48.23	49.78	0.16	
<b>PRESENT V1 [22]</b>	80	16	271	145	524	669	250	141.26	36.16	0.13	Spartan-3E
<b>PRESENT V2 [22]</b>	80	16	256	98	478	576	132	132.19	64.09	0.25	500FG320-5

The results of FPGA implementations for the proposed LED architectures are compared with other implementations of Piccolo, PRESENT, AES, Klein, and Lilliput block ciphers in this section. The proposed designs of the LED algorithm have fewer hardware resources compared to other hardware implementations of block ciphers algorithms, which indicates that the proposed designs have low power consumption compared to those designs.

## 5. CONCLUSIONS

To propose an efficient hardware implementation of the LED algorithm, we have proposed an 8-bit serial architecture with two designs of MixColumns(4-bit and 8-bit). The proposed designs were implemented on the FPGA (Spartan-3 XC3S50 device). The proposed designs have different latencies to perform the encryption process. From the result in Table 4, the proposed serial architecture with 8-bit MixColumns is the best design. It presents a higher performance and efficiency compared to the proposed serial architecture with 4-bit MixColumns. Compared to existing block cipher implementations, the proposed designs are characterized by lower energy consumption thanks to the lower use of resources.

To evaluate the level of security of the LED algorithm, we used five metrics such as entropy analysis, histogram analysis, correlation analysis, NPCR, and UACI. As a result, the LED algorithm has a good ability to encrypt any data against any attack. The 8-bit designs offer a higher speed compared to the 8-bits ones. Indeed, the first one speed's is 9.66 compared to 5.48 for the second design.

In future work, we plan to implement the efficient design (the proposed serial architecture with 8-bit) of the LED algorithm on the RKS to ensure the security of vehicle data. A key fob that wirelessly connects with the automobile transceiver used to regulate and secure access to the vehicle makes up a Remote Keyless System. The RFID-based keyless entry systems have a finite amount of processing power and energy available to them. It has been established that cybercriminals are after it.

## REFERENCES

- [1] Vincent, L., Chevret, G. (2008). Customer identification device, keyless access system for vehicle, vehicle sharing system including such a device and methods using such a device. Patent WO2008044093 A, 1.
- [2] Lee, D. (2014). Keyless cars 'increasingly targeted by thieves using computers'. Internet: www. bbc. com/news/technology-29786320.
- [3] Garcia, F.D., Oswald, D., Kasper, T., Pavlidès, P. (2016). Lock it and still lose it—On the (In)Security of automotive remote keyless entry systems. In 25th USENIX security symposium (USENIX Security 16).
- [4] Glocker, T., Mantere, T., Elmusrati, M. (2017). A protocol for a secure remote keyless entry system applicable in vehicles using symmetric-key cryptography. In 2017 8th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, pp. 310-315. <https://doi.org/10.1109/ICICS.2017.7921990>
- [5] Moradi, A., Kasper, T. (2009). A new remote keyless entry system resistant to power analysis attacks. In 2009 7th International Conference on Information, Communications and Signal Processing (ICICS), Macau, China, pp. 1-6. <https://doi.org/10.1109/ICICS.2009.5397727>
- [6] Migacz, L., Feng, X., Conrad, M. (2021). Automotive security and theft prevention systems: State of the art. In 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). pp. 806-812. <https://doi.org/10.1109/DASC-PiCom-CBDCom-CyberSciTech52372.2021.00134>
- [7] Makanju, A., Zincir-Heywood, A.N., Milios, E.E. (2011). A lightweight algorithm for message type extraction in system application logs. IEEE Transactions on Knowledge and Data Engineering, 24(11): 1921-1936. <https://doi.org/10.1109/TKDE.2011.138>
- [8] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., Seurin, Y., Vikkelsoe, C. (2007). PRESENT: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds) Cryptographic Hardware and Embedded Systems - CHES 2007. CHES



2007. Lecture Notes in Computer Science, vol 4727. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-540-74735-2\\_31](https://doi.org/10.1007/978-3-540-74735-2_31)
- [9] Guo, J., Peyrin, T., Poschmann, A., Robshaw, M. (2011). The LED block cipher. In: Preneel, B., Takagi, T. (eds) Cryptographic Hardware and Embedded Systems – CHES 2011. CHES 2011. Lecture Notes in Computer Science, vol 6917. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-23951-9\\_22](https://doi.org/10.1007/978-3-642-23951-9_22)
- [10] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T. (2011). Piccolo: An ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds) Cryptographic Hardware and Embedded Systems – CHES 2011. CHES 2011. Lecture Notes in Computer Science, vol 6917. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-23951-9\\_23](https://doi.org/10.1007/978-3-642-23951-9_23)
- [11] De Cannière, C., Dunkelman, O., Knežević, M. (2009). KATAN and KTANTAN—A family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds) Cryptographic Hardware and Embedded Systems - CHES 2009. CHES 2009. Lecture Notes in Computer Science, vol 5747. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-04138-9\\_20](https://doi.org/10.1007/978-3-642-04138-9_20)
- [12] Dave, J., Faruki, P., Laxmi, V., Zemmari, A., Gaur, M., Conti, M. (2020). Spark: Secure pseudorandom key-based encryption for deduplicated storage. *Computer Communications*, 154: 148-159. <https://doi.org/10.1016/j.comcom.2020.02.037>
- [13] Shannon, C.E. (1949). Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4): 656-715. <https://doi.org/10.1002/j.1538-7305.1949.tb00928.x>
- [14] Behnia, S., Akhshani, A., Mahmodi, H., Akhavan, A.J.C.S. (2008). A novel algorithm for image encryption based on mixture of chaotic maps. *Chaos, Solitons & Fractals*, 35(2): 408-419. <https://doi.org/10.1016/j.chaos.2006.05.011>
- [15] Wu, Y., Zhou, Y., Saveriades, G., Agaian, S., Noonan, J.P., Natarajan, P. (2013). Local Shannon entropy measure with statistical tests for image randomness. *Information Sciences*, 222: 323-342. <https://doi.org/10.1016/j.ins.2012.07.049>
- [16] Kwok, H.S., Tang, W.K. (2007). A fast image encryption system based on chaotic maps with finite precision representation. *Chaos, Solitons & Fractals*, 32(4): 1518-1529. <https://doi.org/10.1016/j.chaos.2005.11.090>
- [17] Borujeni, S.E., Eshghi, M. (2007). Design and simulation of encryption system based on PRNG and Tompkins-Paige permutation algorithm using VHDL. In *Proceedings of the International Conference on Robotics, Vision, Information and Signal Processing*, pp. 63-67.
- [18] Mhaouch, A., Elhamzi, W., Atri, M. (2020). Lightweight hardware architectures for the piccolo block cipher in FPGA. In *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Sousse, Tunisia, pp. 1-4. <https://doi.org/10.1109/ATSIP49331.2020.9231586>
- [19] Kaps, J.P., Sunar, B. (2006). Energy comparison of AES and SHA-1 for ubiquitous computing. In: Zhou, X., et al. *Emerging Directions in Embedded and Ubiquitous Computing. EUC 2006. Lecture Notes in Computer Science*, vol 4097. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/11807964\\_38](https://doi.org/10.1007/11807964_38)
- [20] Marchand, C., Bossuet, L., Gaj, K. (2017). Area-oriented comparison of lightweight block ciphers implemented in hardware for the activation mechanism in the anti-counterfeiting schemes. *International Journal of Circuit Theory and Applications*, 45(2): 274-291. <https://doi.org/10.1002/cta.2288>
- [21] Mhaouch, A., Elhamzi, W., Abdelali, A.B., Atri, M. (2022). Optimized Piccolo lightweight block cipher: Area efficient implementation. *Traitement du Signal*, 39(3): 805-814. <https://doi.org/10.18280/ts.390305>
- [22] Lara-Nino, C.A., Morales-Sandoval, M., Diaz-Perez, A. (2016). Novel FPGA-based low-cost hardware architecture for the PRESENT block cipher. In *2016 Euromicro Conference on Digital System Design (DSD)*, Limassol, Cyprus, pp. 646-650. <https://doi.org/10.1109/DSD.2016.46>