Hindawi
Security and Communication Networks
Volume 2020, Article ID 8860598, 13 pages
https://doi.org/10.1155/2020/8860598

Research Article

Hardware Implementation of Secure Lightweight Cryptographic Designs for IoT Applications

Wajih El Hadj Youssef, Ali Abdelli, Fethi Dridi, and Mohsen Machhout

Faculty of Sciences of Monastir, Electronics and Micro-Electronic Laboratory (LEME), Monastir 5000, Tunisia

Correspondence should be addressed to Wajih El Hadj Youssef; elhadjyoussef.wajih@gmail.com

Received 28 April 2020; Revised 7 November 2020; Accepted 10 November 2020; Published 29 November 2020

Academic Editor: Mamoun Alazab

The recent expansion of the Internet of Things is creating a new world of smart devices in which security implications are very significant. Besides the claimed security level, the IoT devices are usually featured with constrained resources, such as low computation capability, low memory, and limited battery. Lightweight cryptographic primitives are proposed in the context of IoT while considering the trade-off between security guarantee and good performance. In this paper, we present optimized hardware, lightweight cryptographic designs, of 32-bit datapath, LED 64/128, SIMON 64/128, and SIMECK 64/128 algorithms, for constrained devices. Our proposed designs are investigated on Spartan-3, Spartan-6, and Zynq-7000 FPGA platforms in terms of area, speed, efficiency, and power consumption. The proposed designs achieved a high throughput up to 891.99 Mbps, 838.95 Mbps, and 210.13 Mbps for SIMECK 64/128, SIMON 64/128, and LED 64/128 on Zynq-7000, respectively. A deep comparison between our three proposed designs is elaborated on different FPGA families for adequate FPGAs-based application deployment. Test results and security analysis show that not only can our proposed designs achieve good encryption results with high performance and a low reduced cost but also they are secure enough to resist statistical attacks.

1. Introduction

The devices we use every day are becoming connected entities across the planet. The so-called IoT includes technologies combining autonomous embedded sensory objects with communication intelligence. Most of the applications in the IoT have consequently strong real-time requirements and energy limitations [1–3]. Moreover, the IoT can be affected by different classes of security: access to intellectual property, sabotage, espionage, and cyber terrorism in critical infrastructures such as traffic monitoring, smart cities, and Industrial Automation [4, 5].

This imposes to design performant, cryptographic designs that are efficient in terms of security, computational capability, resource occupation, and power consumption. Indeed, designing cryptography systems must deal with the trade-offs between security, performance, and cost [6, 7].

It is generally easy to optimize any two of the three design goals: security and cost, security and performance, or cost and performance; however, it is really difficult to optimize all three design goals at once.

Traditional secure encryption methods are indeed usually calculated intensively with large key sizes which undermine the computation capacity of IoT devices. In the context, lightweight cryptographic primitives are better alternatives while considering the compromise between security guarantee and full performance even if adapted to resource-limited devices. Hence, there is a substantial requirement for designing new lightweight encryption solutions adapted to the IoT-constrained environments [8, 9].

The main focus of this work is to propose an optimized hardware implementation of lightweight cryptographic designs and examine the effect on hardware architectures, the area, power, efficiency, and performance of hardware implementations on low-cost Xilinx FPGA platforms. Three different hardware architectures of LED 64/128, SIMON 64/128, and SIMECK 64/128 algorithms have been proposed in this study. The security level is evaluated by implementing

our designs on diverse types of images. Then, test results and security analysis of the suggested designs are elaborated for attack-resistance proofs.

To the best of our knowledge and based on literature review, this work sets the best performances of hardware lightweight cryptographic cipher architectures. The architectures we have proposed are implemented with 32-bit datapath on different platforms for an adequate device choice where FPGAs are deployed. Furthermore, we quantify the cost of our proposed 32-bit datapath architectures and show the trade-off between the area, throughput, efficiency, and power consumption. The robustness of the introduced lightweight cryptographic designs is shown by implementing it on several types of images. A detailed security analysis has been provided using visual testing, information entropy, and correlation coefficient analysis.

The remainder of this paper is organized as follows: Section 2 discusses previous works related to lightweight cryptographic designs.Section 5 presents the results of hardware implementation on different FPGAs platforms. The obtained results are compared to the state of the art as well as against each other. Security analysis of the elaborated designs is achieved to demonstrate robustness against possible attacks. Section 6 concludes this study.

2. Related Works

In recent years, there was a quick advancement of research and development of lightweight cryptography for implementation on devices with limited resources in IoT environments. The principal objective is to design and employ ultralightweight cryptographic algorithms that can be used in such applications while proving desired security levels.

Generally, cipher implementations targeted for low-resource applications are classified into software and hardware implementation. In the case of software, implementation required memory size of embedded software is considered, for the hardware implementation area, speed, and power consumption are taken into account. These constraints must be respected when it comes to choosing the appropriate security algorithm to be used for resource-limited devices.

Miscellaneous works dealing with both implementations have been made out for lightweight cryptography implementation on constrained devices. In [10], Benadjila et al. explored general software implementations of lightweight ciphers on x86 architectures, with a specific focus on LED, PICCOLO, and PRESENT. They propose new interesting trade-off, with a theoretical cache modeling to better predict which trade-off will be suitable depending on the target processor. Park et al. [11] proposed efficient parallel implementation methods of the SIMECK family block cipher using an Intel AVX2 (Advanced Vector Extension 2) SIMD and an efficient adaptive encryption method to enhance human care service availability based on the SIMECK family block cipher AVX2-optimized implementations which support different data block sizes. In another work, high software performance implementation of SIMON and SPECK is achieved on the AVR family of 8-bit microcontrollers [12]. Kim et al. [13] investigated lightweight features of HIGHT block cipher and presented the optimized implementations of both software and hardware for low-end IoT platforms, including resource-constrained devices (8-bit AVR and 32-bit ARM Cortex-M3) and application-specific integrated circuit (ASIC).

Other existing researches have focused on hardware and lightweight cryptography. Diehl et al. [14] implement six ciphers, AES, SIMON, SPECK, PRESENT, LED, and TWINE, in hardware using register transfer level (RTL) design and in software using the custom reconfigurable processor. These implementations are instantiated in identical Xilinx Kintex-7 FPGAs, enabling direct comparison of throughput, area, throughput-to-area (TP/A) ratio, power, and energy.

Another research presented by Abed et al. [15] proposes implementing, optimizing, and modeling SIMON cipher design for low-resource devices, with an emphasis on energy and power, which are critical metrics for low-resource devices. Several pipelined FPGA implementations of the SIMON 32/64 lightweight cipher were designed and tested with different numbers of hardware rounds per cycle by many scholars.

Ahir et al. [16] proposed reliable and efficient error detection architectures for the block ciphers SIMON and SPECK with acceptable area and power consumption overheads. The fault injection simulations are performed to fix the error detection capabilities of the proposed architectures implemented on the Zynq-7000 FPGA platform. Beaulieu et al. [17] discussed FPGA performance comparisons of SIMON, SPECK, and PRESENT on low-cost Xilinx Spartan-3 FPGAs. In this article, the authors presented the sort of performance that is achieved by SIMON and SPECK on a broad range of existing software and hardware platforms compared to AES and PRESENT.

In another work [18], Dahiphale et al. proposed, implemented, and evaluated the five most efficient datapaths of different data bus sizes of RECTANGLE cipher. All proposed solutions are implemented on different FPGA platforms with the same implementation conditions and the results are compared on every performance metric.

Almost all cited works are interested in optimizing the software or hardware implementation for low area occupation, high-speed calculation, high throughput, or other metrics, but in any work, all performances are respected at the same time neither with a reasonable security level guaranty.

3. Proposed Lightweight Cryptographic Architecture

Table 1 presents lightweight cryptography algorithms' cipher specifications: the block/key size (bits), datapath, and the number of rounds.

3.1. LED-128. The Light Encryption Device (LED) is a 64-bit block cipher based on a substitution-permutation network (SPN). LED is a 64-bit block cipher that can handle key sizes

Table 1: Lightweight cryptography algorithms' cipher specifications.

Algorithm	Block size (bits)	Key size (bits)	Datapath (bits)	Round T
LED	64	128	32	48
SIMON	64	128	32	44
SIMECK	64	128	32	44

from 64 bits up to 128 bits. We denote by LED-x the LED block cipher version that handles x-bit keys [19].

The key schedule of LED is extremely simple as it is almost inexistent, which presents obvious advantages in hardware. This simplicity is also very welcoming for security proofs as we can derive some even for the related-key model. The idea is to just reuse the original key material as is but several times during the computation.

For a 128-bit key, the secret material is divided into two keys $K1$ and $K2$ that are repeatedly and alternatively XORed to the internal state every four rounds of the internal permutation as shown in Figure 1.

The keyed permutation of the LED algorithm is largely inspired by the Advanced Encryption Cipher (AES) structure. Namely, the internal state can be viewed as a 4×4 matrix of 4-bit cells. One round is described by four functions (see Figure 2):

(i) *AddConstants*. This function applies round-dependent constants to each cell of the two first columns.

(ii) *SubCells*. This function applies a 4-bit S-box to every cell of the internal state. We chose to use the very small 4-bit PRESENT cipher S-box.

(iii) *ShiftRows*. This function simply rotates each cell located at row i by i positions to the left.

(iv) *MixColumnsSerial*. This function updates linearly all columns independently. The matrix underlying the MixColumnsSerial layer is Maximum Distance Separable (MDS) to provide maximal diffusion.

In this work, the LED-128 is applied to an internal permutation of 48 rounds. The serialized architecture of LED-128 is described in Figure 3. It contains two registers reserved for the 128-bit key and the 64-bit message, multiplexers (MUX 4/1 and MUX 2/1 on 32 bits and another 2/1 on 64 bits), 5 XOR operations on 32 bits each, a 32-bit S-box substitution function, and a ShiftRows function applied to a data block of 64 bits only.

The 32-bit serial architecture allows all data (message + key) to be loaded in parallel in 32-bit blocks through two "DATA_In" and "Key_in" inputs. This task requires four clock cycles to load all initialization data (128/32 = 4).

3.2. SIMON 64/128. SIMON is one of the recently published lightweight block ciphers from the National Security Agency (NSA) in June 2013 [20].

The SIMON family of lightweight block ciphers is defined for word sizes $n = 16, 24, 32, 48$, and 64 bits. The key is composed of m n-bit words for $m = 2$, 3, and 4 (i.e., the key size mn varies between 64 and 256 bits) depending on the word size n. The block cipher instances corresponding to a fixed word size n (block size $2n$) and key size mn are denoted by SIMON $2n = mn$. In this work, a 32-bit word and a 128-bit key are used as a cipher configuration.

The SIMON block cipher family relies on Addition, word Rotation, denoted as $S^y(x)$ where y is the rotation count, and XOR although it uses AND gates instead of additions. The round functions of SIMON are shown in Figure 4.

For encryption, the SIMON round function can be expressed as

$$R(l, r, k) = \left(\left(S^1(l) \& S^8(l)\right) \oplus S^2(l) \oplus r \oplus k, l\right). \quad (1)$$

For decryption, its inverse is

$$R^{-1}(l, r, k) = \left(r, \left(S^1(r) \& S^8(r)\right) \oplus S^2(r) \oplus l \oplus k\right), \quad (2)$$

where l is the left-most word of a given block, r is the right-most word, and k is the appropriate round key.

The SIMON key schedule function takes the master key and generates a sequence of T key words $(k_0, k_1, k_2, \ldots, k_{T-1})$, where T represents the number of rounds. There are three different versions of the key schedule function, depending on the block size and master key size. In our case, from the initial 128-bit master key, the key schedule generates 44 32-bit sized round keys.

The key schedule function performs two circular shift operations to the right (shift right one and shift right three). The result is XORed with a fixed constant, c, and a constant sequence, zj. There are five sequences for the constant zj, which are version-dependent (i.e., z_0, z_1, z_2, z_6 and z_4). Figure 5 illustrates the key schedule function of SIMON for three master key words (i.e., $m = 2$).

The key expansion function can be expressed as

$$Ki(k, c, zj) = F(ki + 3, ki + 1) \oplus S^{-1}(F(ki + 3, ki + 1)) \oplus ki \oplus c \oplus (zj)i, \quad (3)$$

where

$$F(x, y) = S - 3(x)y. \quad (4)$$

The key schedule employs the constant $c = 2^n - 4 = 0x\mathrm{FF}, \ldots, \mathrm{FC}$ (where $n = 32$ represents the word size parameter).

The round function architecture is composed of two 32-bit size data registers, a 2-input, single-output 32-bit multiplexer, and a combinational circuit containing three 1-bit, 8-bit cyclic shift registers and 2 bits, one AND logic gate, and three XOR logic gates. The results of this circuit are one of the entrances to MUX 2/1 as presented in Figure 6(a). The 128-bit key generation (Figure 6(b)) architecture is composed of 4 blocks of subkeys of 32 bits (key a, key b, key c, and key d), a MUX 2/1 reserved for inputs, and a combinational circuit with $(2n + 1)\mathrm{XOR} + (n - 1)\mathrm{XNOR}$.

Each instance of SIMON uses the familiar Feistel rule of motion. The algorithm is engineered to be extremely small in hardware and easy to serialize at various levels. It is supposed to be more hardware-oriented.

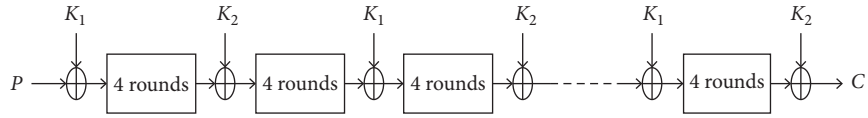


Figure 1: The use of key arrays $K1$ and $K2$ in LED for a 128-bit key array.

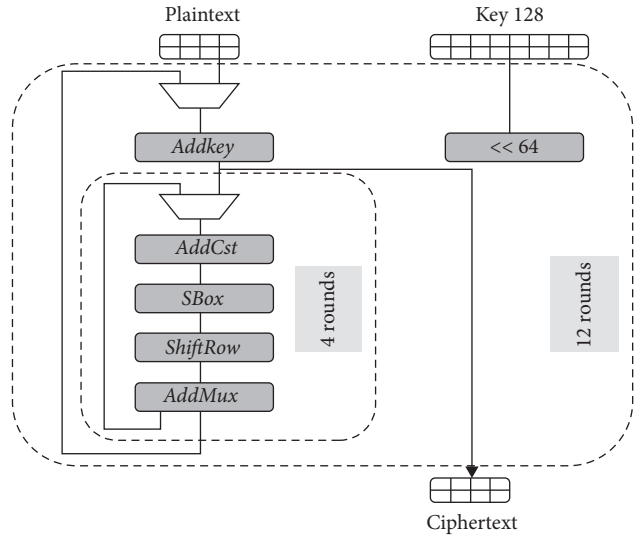


Figure 2: Full datapath architecture for hardware implementation of LED-128.

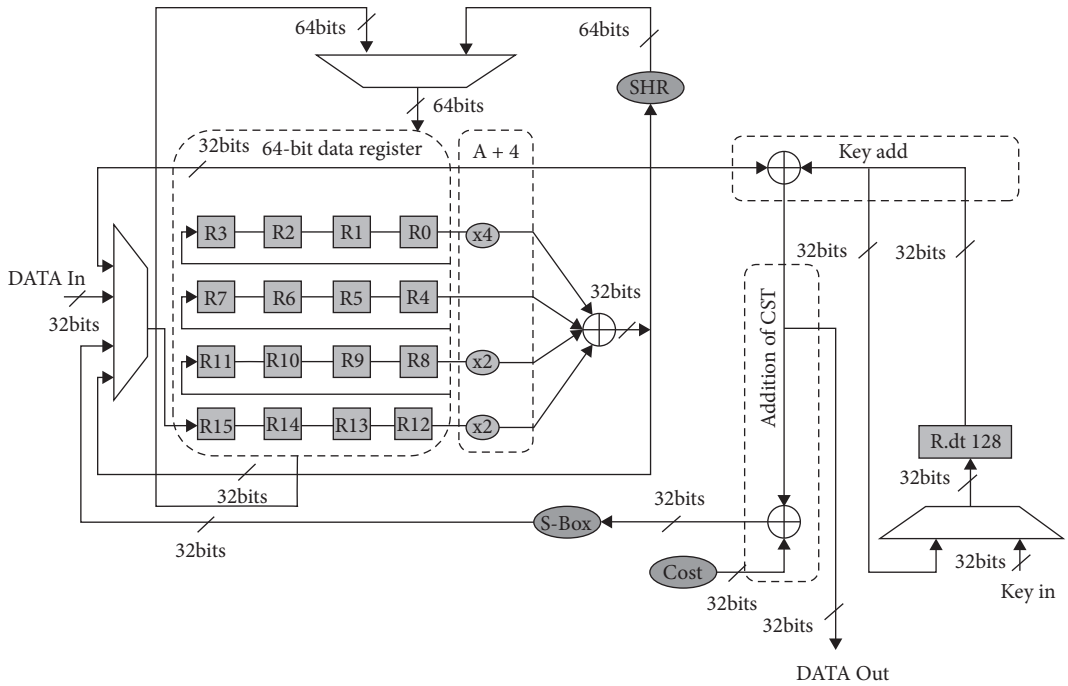


Figure 3: Serial 32-bit datapath architecture of LED-128.

3.3. SIMECK 64/128. SIMECK is a new lightweight block cipher design, proposed at CHES 2015. It is based on combining the design fundamentals of the SIMON and SPECK block cipher [21].

The round function and the key schedule algorithm follow the Feistel structure. The round function of SIMECK is given in Figure 7, where r_i and l_i are, respectively, right word and left word. k_i denotes the ith round key. The ciphertext is the internal state after T rounds.

The SIMECK family of block cipher encryption and decryption round functions has ARX: the bitwise AND ($\odot$), rotation (rotation left, ROLr(), and exclusive-OR ($\oplus$)

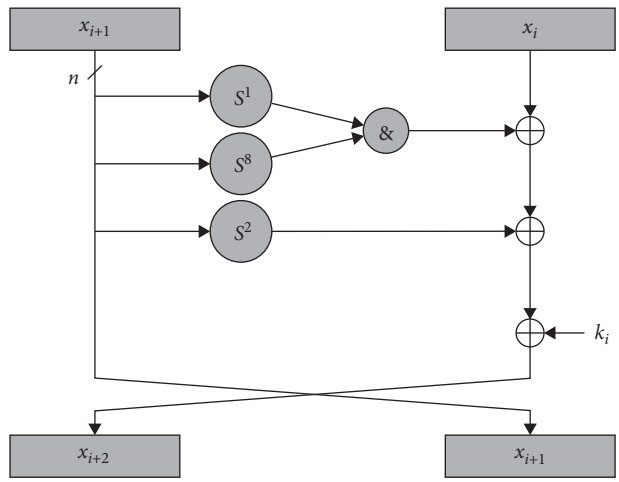


Figure 4: Feistel stepping of the SIMON round function.

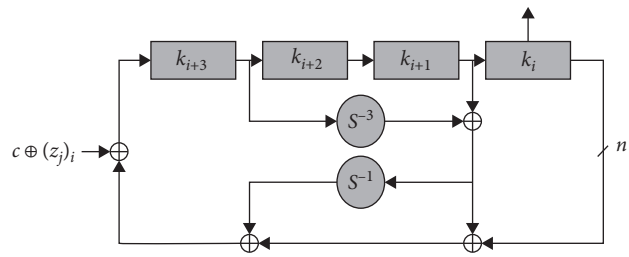
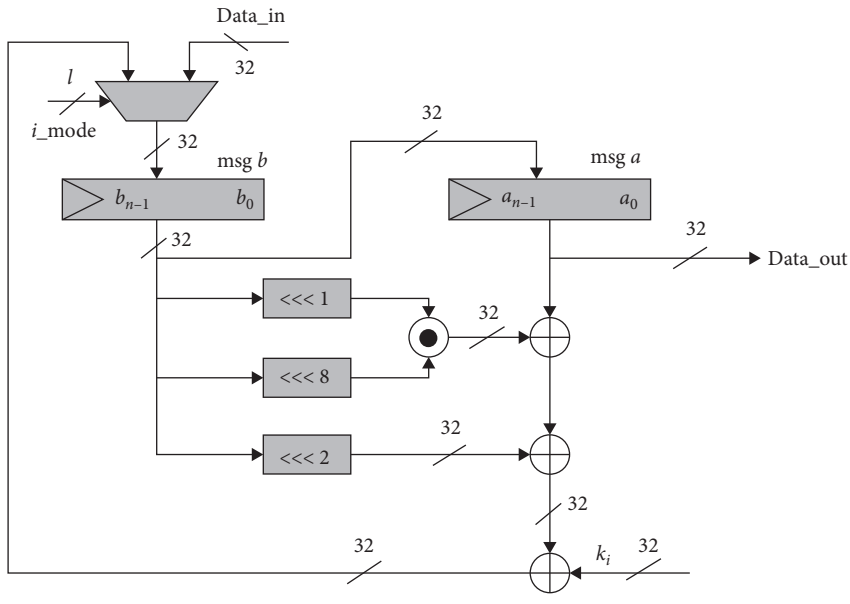
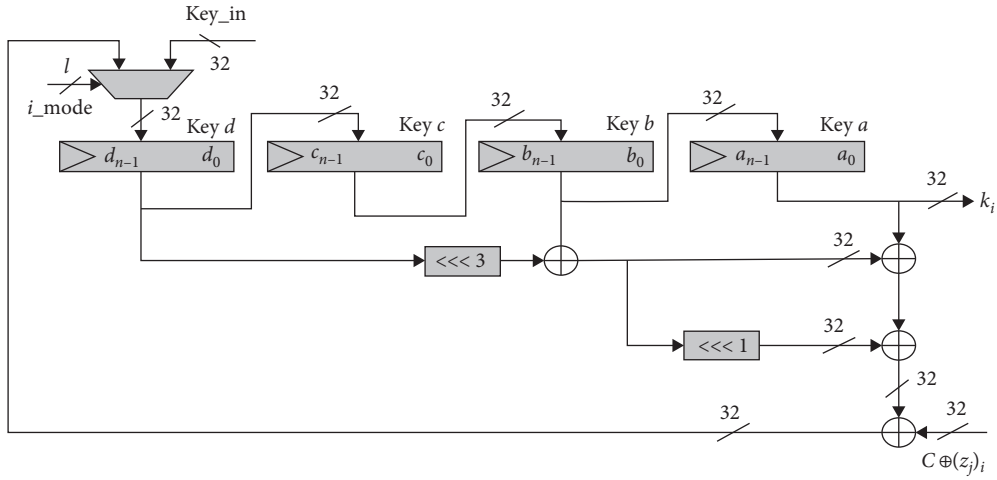


Figure 5: The SIMON, four-word key expansions.



(a)

Figure 6: Continued.



(b)

Figure 6: Parallel 32-bit datapath architecture of SIMON 64/128. (a) 32-bit round function. (b) 128-bit key generation architecture.

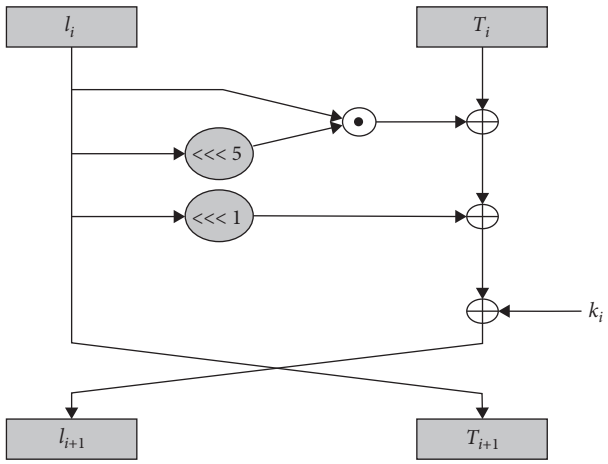


Figure 7: SIMECK round function.

operations. ROLr() function refers to the r-bit left rotation operation.

The round function (of the ith round) is defined as follows:

$$R_{ki}(l_i, r_i) = (r_i \oplus f(l_i) \oplus k_i, l_i). \quad (5)$$

The function f is defined as

$$f(x) = (x \odot \mathrm{ROL}_5(x)) \oplus (\mathrm{ROL}_1(x)). \quad (6)$$

Figure 8 shows the SIMECK family block cipher key schedule as a block diagram. To generate the round key ki from a given master key K, the master key K is first segmented into four words and loaded as the initial states (t_2, t_1, t_0, k_0) of the feedback shift. First, the least significant n-bits of K are loaded into k_0, while the most significant n-bits are put into t_2.

The SIMECK round function $R_{C\oplus(zj)i}$ with a round constant $C \oplus (zj)i$ acts as the round key during each round. The combinational circuit (dashed box in above) in the key schedule of SIMECK in the parallel architecture is composed by $(n + 1)$XOR + $(n - 1)$XNOR + n AND.

Our lightweight block cipher family SIMECK is denoted by SIMECK$2n/mn$, where n is the word size and n is required to be 16, 24, or 32, while $2n$ is the block size and mn is the key size. SIMECK has three instances; in this work, we focus on the SIMECK 64/128 (see Figures 9 and 10).

The combinational circuit (dashed box in above) in the key schedule of SIMECK in the parallel architecture is composed of $(n + 1)$XOR + $(n - 1)$XNOR + n AND.

SIMECK is supposed to perform exceptionally well in both hardware and software. The change in the rotations and the key schedule allow an improved hardware implementation.

Table 2 shows the complexity of our proposed lightweight cryptographic designs.

4. Experimental Results

4.1. Hardware Implementation. In this section, we provide an overview of our proposed hardware implementation results. The area, speed, efficiency, and power consumption performances of the proposed designs are obtained from the implementation of our VHDL code using Xilinx ISE Design Suite 14.7. The areas of the block cipher implementations on FPGA are compared using slices, flip-flops, and lookup tables (LUTs), which are the basic logic block of Xilinx FPGAs. Latency, maximum frequency, and throughput together determine the speed of execution. Efficiency represents throughput-to-area ratio to meet lightweight application requirements.

To get a good insight into the efficiency and performance, our elaborated designs are implemented on three different Xilinx FPGAs: Spartan-3 (XC3S50-5), Spartan-6 (XC6S16-3), and Zynq-7000 (xc7z010-3) families are used as target platforms. The proposed designs have been tested after place and route using simulation to ensure the right functionality.

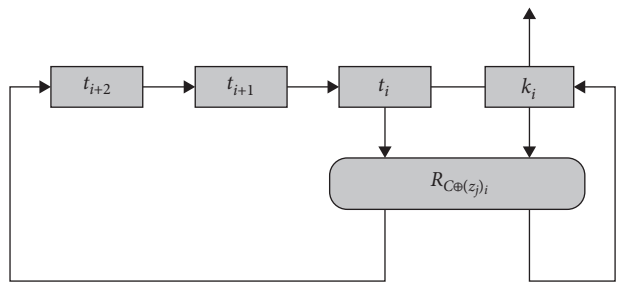


Figure 8: SIMECK key expansion.

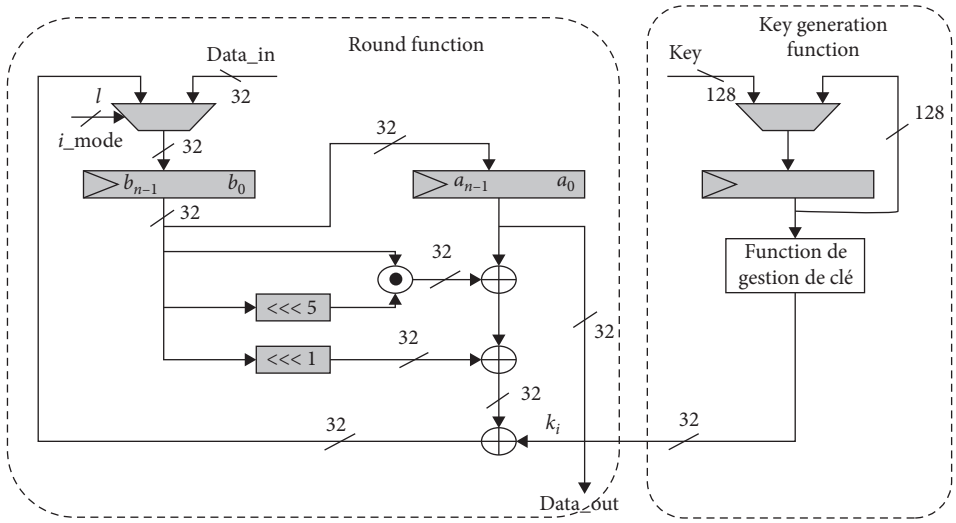


Figure 9: Parallel full datapath architecture SIMECK 64/128.

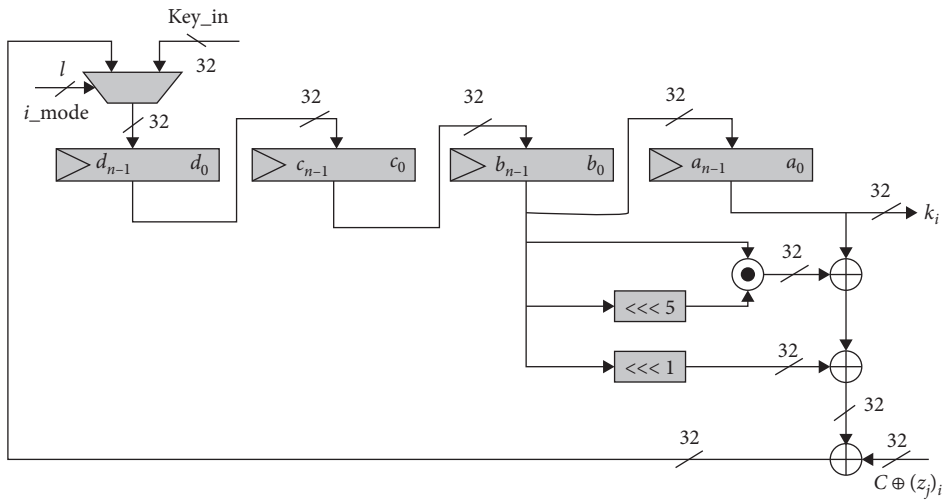


Figure 10: Parallel 32-bit datapath key generation.

Table 2: Complexity of our proposed cryptographic designs.

Algorithm	Flip-flop	XOR	MUX 2/1	MUX 4/1	NOR
LED	320	104	2	1	—
SIMON	192	229	2	—	64
SIMECK	192	197	2	—	64

Table 3 lists the results of our three proposed 32-bit datapath designs. We can conclude that, when implemented in Spartan-3 FPGA, SIMECK is the smallest block cipher implemented with only 399 LUTs plus FFs. This is due to the number of rounds and the change in the rotations and the key schedule. Furthermore, the parallel architecture processes one round of the message in one clock cycle and one round of the key schedule at the same clock cycle. This allows for improved hardware implementation. SIMON is the second smallest block cipher with 416 LUTs plus FFs. Our proposed SIMON and SIMECK designs are very close in terms of throughput and efficiency. The estimated power consumption is very closer to the three proposed architectures.

When using the Spartan-6 FPGA platform, LED is the least consumed algorithm with 452 LUTs plus FFs. The main area cost for SIMON and SIMECK comes from the registers storing the message block and the key. However, SIMON and SIMECK provide better throughput, efficiency, and power consumption.

Unfortunately, few works present results using the two used FPGAs for the three algorithms described in this study, a small number of works on LED and SIMON are made on FPGA platfroms, and not all metrics are treated which made the comparison complex. As shown in Table 3, our proposed 32-bit datapath designs provide more throughput and require less area to implement on both Spartan-3 and Spartan-6 FPGA platforms compared to the state of the art.

In [22–24], the authors use only generic components such as FFs, LUTs, maximum frequency, and throughput. In fact, other design parameters, a trade-off between area and throughput representing the efficiency, and power consumption have to be considered.

No implementations have been undertaken to date to the best of our knowledge for the three studied algorithms on Zynq-7000 with a block size of 64 bits and a key of 128 bits. Only in [25], the authors proposed a reliable hardware architecture for SIMON 48/96 block ciphers by using time redundancy concurrent error detection techniques. They claim that their proposed design has acceptable overheads with very high error coverage. However, the obtained results are very poor and are not considering the constraints of the devices, such as ciphers or lightweight ciphers, where cost, power consumption, energy, and available resources are limited. For this reason, comparison cannot go ahead.

To get the overheads, we compare the implementation results obtained from our proposed lightweight cipher architectures on different FPGAs families. Depending on the design metrics, we can choose the adequate lightweight architecture suitable for the need of the application such as FPGAS-based RFID tags [26] or FPGAS-based wireless sensor nodes [27].

From Figure 11, LED-128 architecture requires less area when implemented on Spartan-6 and Zynq-7000 platforms. SIMECK 64/128 provides better area occupation with 399 LUTs and FFs when implemented on Spartan-3 FPGA.

As shown in Figure 12, we noticed that the Zynq-7000 platform is well suited for resource-constrained environments with high throughput requirements. It provides throughput up to 891.99 Mbps, 838.95 Mbps, and 210.13 Mbps for SIMECK 64/128, SIMON 64/128, and LED 64/128 on Zynq-7000, respectively. SIMECK 64/128 architecture provides the best throughput among the proposed architectures when implemented on the three FPGAs.

From Figure 13, we can conclude that the Zynq-7000 platform is optimized for a good throughput-to-area ratio. We notice also that SIMECK 64/128 is the best suited to meet lightweight application needs when efficiency is considered with 1.73 Mbps/slice on Spartan-3 and 4.45 Mbps/slice on Zync-7000. SIMON 64/128 efficiency presents the highest efficiency with 1.81 Mbps/slice when implemented on Spartan-6 FPGA.

For battery-operated devices, Spartan-3 FPGA is preferable over Spartan-6 and Zynq-7000 platforms as its power consumption is the least with 2 mW for LED-128 and 3 mW for both SIMON 64/128 and SIMECK 64/128. SIMON 64/128 and SIMECK 64/128 architectures provide a far lower power consumption compared to LED-128 when implemented on Spartan-6 with 31 mW and 27 mW, respectively, as depicted in Figure 14.

5. Security Analysis

In this work, statistical analysis has been performed to demonstrate the superior confusion and diffusion properties of the proposed lightweight cryptographic designs against statistical attacks. This is done by performing a series of tests: histogram analysis of the encrypted images, correlation computation of the adjacent pixels in encrypted images, and information entropy calculation [28].

5.1. Histograms of Encrypted Images. In this current section, we apply our introduced lightweight cryptographic designs on several types of images to test their robustness. Three well-known 8-bit greyscale images, Baboon, Barbara, and Lena, with a resolution chart (256×256) are tested as plain images.

The plain images, encrypted Images using the three proposed cryptographic designs of LED, SIMON, and SIMECK algorithms, and their corresponding histograms are presented in Figures 15–17. As can be shown, there is no perceptual similarity between original images and their encrypted equivalents.

As known, the uniform distribution of intensities after the encryption is an indication of desired security. We can see that provided histograms are almost uniform and are significantly different from those of the three original images. Thus, the obtained encrypted images respond well to the diffusion properties and the attacker with the histogram analysis of the encrypted images cannot acquire information from the original images. Furthermore, the results of the histograms of all encrypted images using SIMOM and SIMECK lightweight algorithms are fairly uniform compared to the LED algorithm.

5.2. Correlation Coefficient Analysis. The other statistical test consists of computing the correlation between adjacent

Table 3: Comparison of our proposed 32-bit datapath designs on Spartan-3 and Spartan-6 FPGAs.

Designs	Device	Area (resources)	Speed					Efficiency (Mbps/slices)	Power (mw)
		Slices	Flip-flops	LUTs	Latency (cycles)	Max. freq. (MHz)	Throughput (Mbps)		
LED	Spartan-3 (XC3S50-5)	229	146	432	192	133.76	44.59	0.19	2
SIMON		150	177	239	44	141.89	206.38	1.37	3
SIMECK		140	173	226	44	166.61	242.34	1.73	3
LED [22]		219	227	414	528	128.73	15.6	0.07	—
LED [22]		77	—	148	768	119.19	9.93	0.13	—
LED full width [24]		—	211	970	—	51.7	39.4	—	—
LED serial [24]		—	218	555	—	106.3	4.3	—	—
SIMON [23]		36	—	72	—	136	3.60	0.10	—
LED	Spartan-6 (XC6S16-3)	154	154	298	192	251.28	83.76	0.54	128
SIMON		206	206	267	44	224.46	326.49	1.81	31
SIMECK		206	206	259	44	217.65	316.56	1.75	27
LED full width [24]		—	211	594	—	83.8	63.8	—	—
LED serial [24]		—	217	373	—	142	5.75	—	—

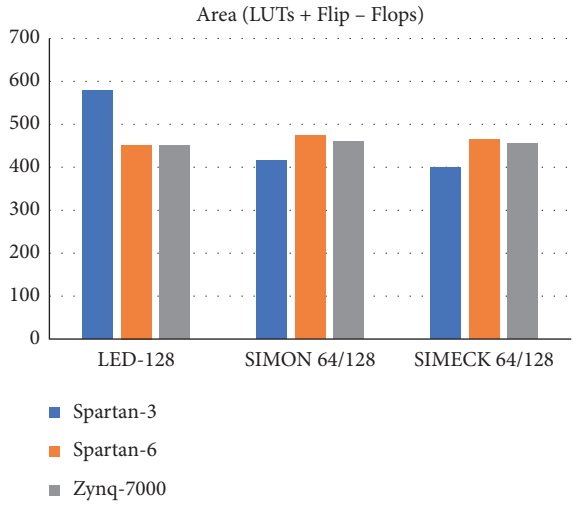


Figure 11: Area comparison of LED, SIMON, and SIMECK.

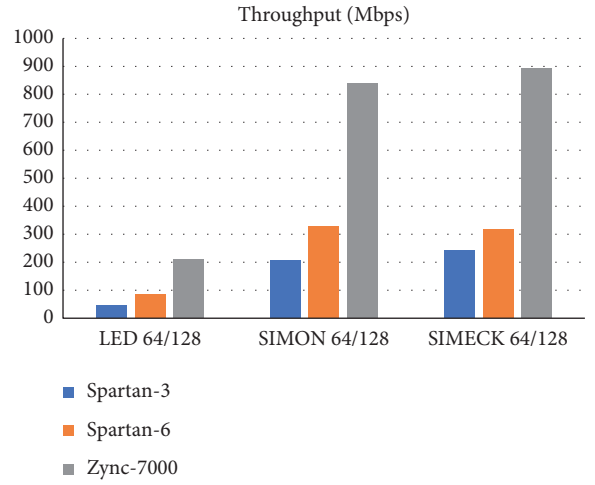


Figure 12: Throughput comparison of LED, SIMON, and SIMECK.

pixels [29]. The coefficient of correlation for each pair is obtained using

$$\rho(X,Y) = \frac{\sum(X-\overline{X})(Y-\overline{Y})}{\sqrt{(X-\overline{X})^2}\sqrt{(Y-\overline{Y})^2}} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{V(X)}.\sqrt{V(Y)}}, \tag{7}$$

where the gray values of any two neighboring pixels of an image are denoted by X and Y; $V(.)$ the variance and Cov $(.)$ the covariance.

This method consists of randomly selecting and calculating the correlation coefficient of adjacent pixels (vertical, horizontal, and diagonal) from the original and the encrypted images separately. In the best case, the correlation coefficient of the original image is equal to one, and the correlation coefficient of the encrypted image is equal to zero.

Table 4 shows the results of horizontal, vertical, and diagonal neighboring pixel correlation coefficients computations of the plain images and the corresponding encrypted images. The above cases show that the values of correlation coefficients of our proposed lightweight cryptographic designs are very close to zero between adjacents. Any linear dependencies are kept between observed pixels in all three directions, which make our designs secure against correlation attacks.

5.3. Information Entropy Analysis. Entropy in the information-theoretic sense is a statistical measure of

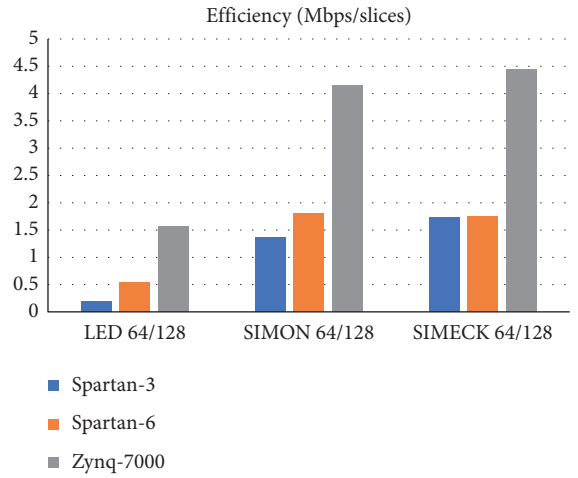


Figure 13: Efficiency comparison of LED, SIMON, and SIMECK.

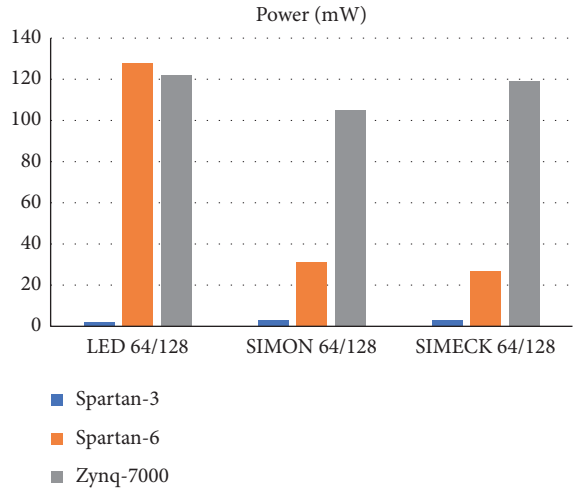


Figure 14: Dynamic power comparison of LED, SIMON, and SIMECK.

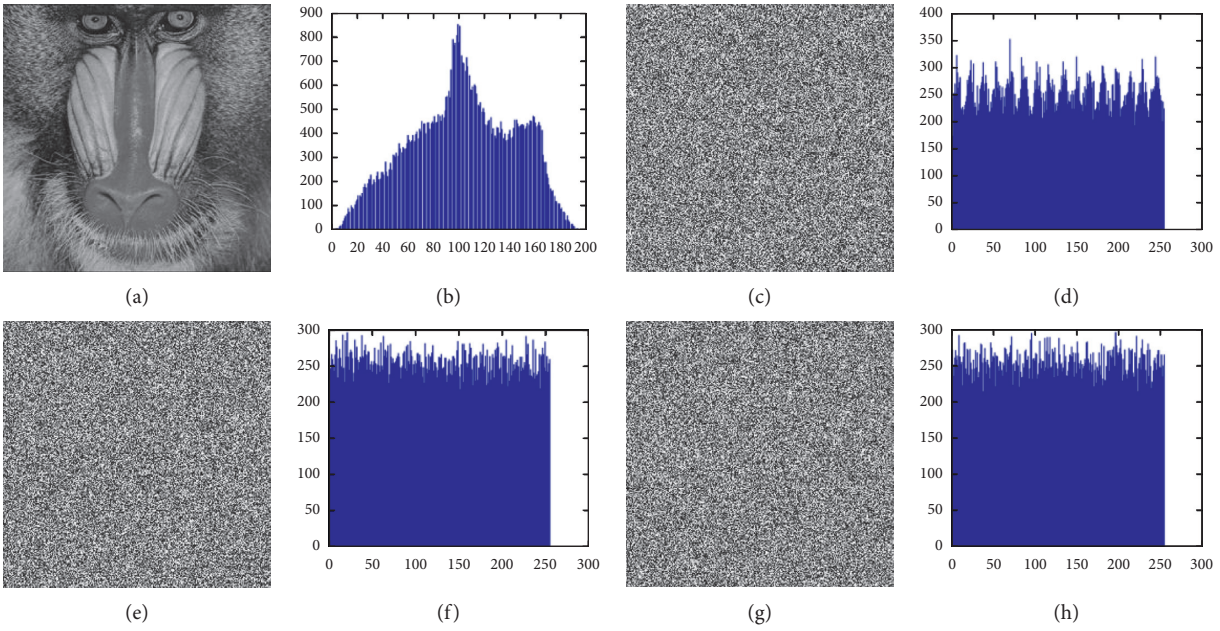


Figure 15: Result of Baboon image. (a) Original image. (b) Histogram of original image. (c) LED-ciphered. (d) Histogram of LED-ciphered. (e) SIMON-ciphered. (f) Histogram of SIMON-ciphered. (g) SIMECK-ciphered. (h) Histogram of SIMECK-ciphered.

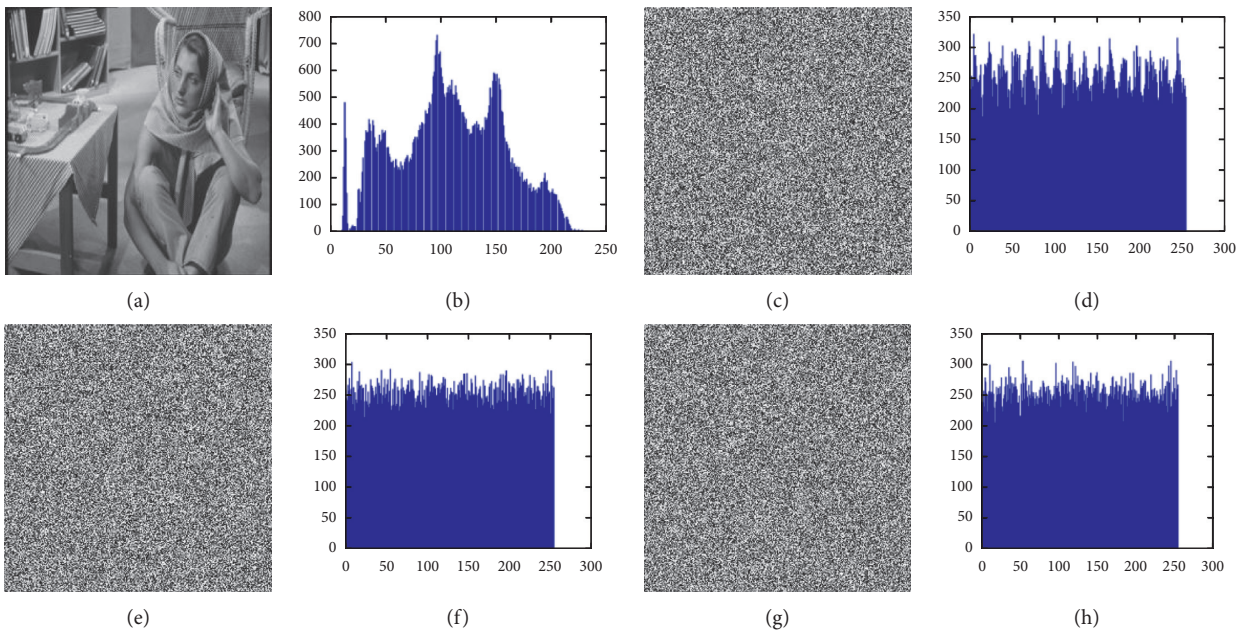


FIGURE 16: Result of Barbara image. (a) Original image. (b) Histogram of original image. (c) LED-ciphered. (d) Histogram of LED-ciphered. (e) SIMON-ciphered. (f) Histogram of SIMON-ciphered. (g) SIMECK-ciphered. (h) Histogram of SIMECK-ciphered.

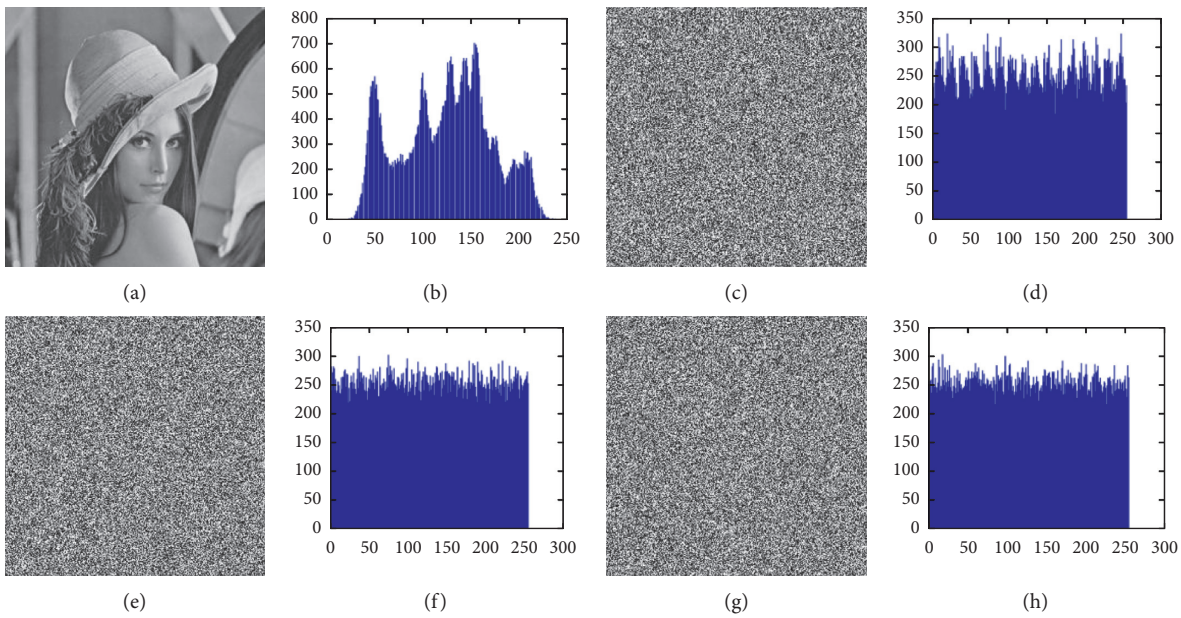


FIGURE 17: Result of Lena image. (a) Original image. (b) Histogram of original image. (c) LED-ciphered. (d) Histogram of LED-ciphered. (e) SIMON-ciphered. (f) Histogram of SIMON-ciphered. (g) SIMECK-ciphered. (h) Histogram of SIMECK-ciphered.

randomness or uncertainty in communication theory [30]; it is defined as follows:

$$H(X) = \sum_{i=0}^{255} p(x_i) \log_2 (x_i), \tag{8}$$

where X is a discrete random variable, $p(xi)$ is the probability density function of the occurrence of the symbol x.

An 8-bit greyscale image can achieve a maximum entropy of 8 bits. From the results in Table 4, it can be seen that the entropy of all encrypted images is close to maximum, depicting an attribute of the algorithm (see Table 5).

As concluded, the obtained information entropy values of the ciphered images are close to the theoretical value of 8. Therefore, it is difficult to conduct a successful attack against our proposed cryptographic designs.

Table 4: Correlation coefficient of two adjacent pixels of plain image and ciphered image.

Correlation	Horizontal	Vertical	Diagonal
Baboon	0.877794	0.834230	0.788141
LED-ciphered Baboon	−0.001537	−0.004250	0.003151
SIMON-ciphered Baboon	−0.001819	−0.010609	0.007425
SIMECK-ciphered Baboon	0.002276	−0.009910	−0.002012
Barbara	0.907829	0.946119	0.883508
LED-ciphered Barbara	−0.002512	0.007238	0.003638
SIMON-ciphered Barbara	−0.004805	−0.006222	−0.002241
SIMECK-ciphered Barbara	−0.012993	−0.011308	0.003706
Lena	0.939403	0.971060	0.931085
LED-ciphered Lena	−0.009072	0.016579	0.000584
SIMON-ciphered Lena	0,000338	−0.012928	−0.003296
SIMECK-ciphered Lena	−0.009522	−0.007509	0.001907

Table 5: Information entropy tests.

Test images	Plain image	LED-ciphered image	SIMON-ciphered image	SIMECK-ciphered image
Baboon	7.310226	7.959743	7.956247	7.958314
Barbara	7.519949	7.957659	7.956241	7.953315
Lena	7.450447	7.956536	7.956232	7.954902

6. Conclusion and Future Work

The Internet of Things (IoT) has become pervasive, with many resources constrained and tiny devices deployed on a large scale and communicating wirelessly with each other and with the Internet at large. Regarding security needs and limited resource properties, the lightweight cryptography is applied to solve this problem.

This article presents hardware implementations and a comparison of three 32-bit datapath lightweight cryptographic designs for LED 64/128, SIMON 64/128, and SIMECK 64/128 algorithms. All implementations' results were compared fairly with previously published works on different FPGA platforms. A deep study of hardware performances and optimizations of lightweight cryptography is elaborated. Better outcomes, compared to the state of the art, were noticed with a low area occupation, high throughput, good efficiency, and low power consumption.

Besides, experimental tests have been carried out with detailed numerical analysis, which shows the robustness of our proposed designs against statistical attack (visual testing). Performance evaluation tests demonstrate that the proposed encryption designs are sufficiently secure against attacks.

As a future work, it will be very interesting to harden our proposed cryptographic designs against possible side-channel attacks such as power analysis and fault injection. In addition, instruction set extensions for lightweight cryptography can be an attractive design option for embedded systems which have a need for security.

Data Availability

The obtained results used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Y. Sun, J. Zhang, R. Bie, and J. Yu, "Advancing researches on IoT systems and intelligent applications," *Personal and Ubiquitous Computing*, vol. 22, no. 3, pp. 449–452, 2018.

[2] P. K. R. Maddikunta, T. R. Gadekallu, R. Kaluri, G. Srivastava, R. M. Parizi, and M. S. Khan, "Green communication in IoT networks using a hybrid optimization algorithm," *Computer Communications*, vol. 159, pp. 97–107, 2020.

[3] R. Ch, T. R. Gadekallu, M. H. Abidi, and A. Al-Ahmari, "Computational system to classify cyber crime offenses using machine learning," *Sustainability*, vol. 12, no. 10, p. 4087, 2020.

[4] M. Henriques and N. Vernekar, "Using symmetric and asymmetric cryptography to secure communication between devices in IoT," in *2017 IEEE International Conference on IoT and Application (ICIOT)*, Nagapattinam, India, June 2017.

[5] M. Numan, F. Subhan, W. Z. Khan et al., "A systematic review on clone node detection in static wireless sensor networks," *IEEE Access*, vol. 8, pp. 65450–65461, 2020.

[6] A. Ragab, "Robust hybrid cryptosystem for protecting smart devices in internet of things (IOT)," Master Thesis, Department of Computer Engineering, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt, 2019.

[7] S. Bhattacharya, S. Somayaji, P. K. R. Maddikunta et al., "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," *Electronics*, vol. 9, no. 2, p. 219, 2020.

[8] A. Ankit Shah and M. Engineer, "A survey of lightweight cryptographic algorithms for IoT-based applications," *Advances in Intelligent Systems and Computing*, vol. 851, pp. 283–293, 2019.

[9] A. A. M. Ragab, A. Madani, A. M. Wahdan, and G. M. I. Selim, "Hybrid cryptosystems for protecting IoT smart devices with comparative analysis and evaluation," in *Proceedings of the Future Technologies Conference (FTC), Advances in Intelligent Systems and Computing*, vol. 1069, Springer, Cham, Switzerland, October 2019.

[10] R. Benadjila, J. Guo, V. Lomné, and T. Peyrin, "Implementing lightweight block ciphers on x86 architectures," in *Proceedings of the Selected Areas in Cryptography (SAC), Lecture Notes in Computer Science*, vol. 8282, Springer, Berlin, Heidelberg, August 2013.

[11] T. Park, H. Seo, S. Lee, and H. Kim, "Secure data encryption for cloud-based human care services," *Journal of Sensors*, vol. 2018, no. 8, pp. 1–10, 2018.

[12] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The Simon and Speck block ciphers on AVR 8-bit microcontrollers," in *Proceedings of the lightweight Cryptography for Security and privacy (LightSec). Lecture Notes in Computer Science*, vol. 8898, Springer, Cham, Switzerland, September 2015.

[13] B. Kim, J. Cho, B. Choi, J. Park, and H. Seo, "Compact implementations of HIGHT block cipher on IoT platforms," *Security and Communication Networks*, vol. 2019, no. 8, pp. 1–10, 2019.

[14] W. Diehl, F. Farahmand, P. Yalla, J. Kaps, and K. Gaj, "Comparison of hardware and software implementations of selected lightweight block ciphers," in *Proceedings of the 27th International Conference On Field Programmable Logic And Applications (FPL)*, pp. 1–4, Ghent, Belgium, September 2017.

[15] S. E. Abed, R. Jaffal, B. Mohd, and M. Alshayeji, "FPGA modeling and optimization of a SIMON lightweight block cipher," *Sensors*, vol. 19, no. 4, p. 913, 2019.

[16] P. Ahir, M. Mozaffari-Kermani, and R. Azarderakhsh, "Lightweight architectures for reliable and fault detection Simon and Speck cryptographic algorithms on FPGA," *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 4, p. 17, 2017.

[17] R. Beaulieu, D. Shors, J. Smith et al., "SPECK: block ciphers for the internet of things," 2018, https://eprint.iacr.org/2015/585.

[18] V. Dahiphale, H. Raut, and G. Bansod, "Design and Implementation of novel datapath designs of lightweight cipher rectangle for resource constrained environment," *Multimedia Tools and Applications*, vol. 78, no. 16, pp. 23659–23688, 2019.

[19] L. Xu, J. Guo, J. Cui, and M. Li, "Key-recovery attacks on LED-like block ciphers," *Tsinghua Science and Technology*, vol. 24, no. 5, pp. 585–595, October 2019.

[20] B. Ray, S. Douglas, J. Smith et al., "The SIMON and SPECK lightweight block ciphers," in *Proceedings of the 52nd Annual Design Automation Conference*, p. 175, ACM, San Francisco, CA, USA, June 2015.

[21] N. Bagheri, "Linear cryptanalysis of reduced-round SIMECK variants," in *Proceedings of the Progress in Cryptology-–INDOCRYPT 2015*, pp. 140–152, Springer, Bangalore, India, December 2015.

[22] N. Nalla Anandakumar, T. Peyrin, and A. Poschmann, "A very compact FPGA implementation of LED and PHOTON," *Progress in Cryptology--INDOCRYPT 2014*, vol. 2014, pp. 304–321, 2014.

[23] A. Aysu, E. Gulcan, and P. Schaumont, "SIMON says: break area records of block ciphers on FPGAs," in *IEEE Embedded Systems Letters*, vol. 6, no. 2, pp. 37–40, 2014.

[24] C. Marchand, L. Bossuet, and K. Gaj, "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*, vol. 45, no. 2, pp. 274–291, 2016.

[25] A. Prashant, M. Mehran, and A. Reza, "Lightweight architectures for reliable and fault detection Simon," *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 4, p. 109, 2017.

[26] M. Feldhofer, M. J. Aigner, T. Baier, M. Hutter, T. Plos, and E. Wenger, "Semi-passive RFID development platform for implementing and attacking security tags," in *Proceedings of the Internet Technology and Secured Transactions (ICITST)*, pp. 1–6, IEEE, London, UK, November 2010.

[27] A. Engel, B. Liebig, and A. Koch, "Feasibility analysis of reconfigurable computing," in *Low-Power Wireless Sensor Applications*, A. Koch, R. Krishnamurthy, J. McAllister, R. Woods, and T. El-Ghazawi, Eds., vol. 6578, pp. 261–268, Springer, Heidelberg, Germany, 2011.

[28] M. H. Abood, "An efficient image cryptography using hash-LSB steganography with RC4 and pixel shuffling encryption algorithms," in *Proceedings of the Annual Conference on New Trends in Information & Communications Technology Applications (NTICT)*, pp. 86–90, Baghdad, Iraq, March 2017.

[29] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Information Sciences*, vol. 222, pp. 323–342, 2013.

[30] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 379–423, pp. 623–656, 1948.