Hindawi
Security and Communication Networks
Volume 2021, Article ID 9943713, 10 pages
https://doi.org/10.1155/2021/9943713

Research Article

Lightweight Cryptography for Network-on-Chip Data Encryption

Riadh Ayachi, Ayoub Mhaouch, and Abdessalem Ben Abdelali

Laboratory of Electronics and Microelectronics, University of Monastir, Monastir, Tunisia

Correspondence should be addressed to Riadh Ayachi; riadh.ayachi@fsm.rnu.tn

Received 30 March 2021; Revised 30 April 2021; Accepted 8 May 2021; Published 19 May 2021

Academic Editor: Prosanta Gope

System-on-chip (SoC) is the main processor for most recent applications such as the Internet of things (IoT). SoCs are composed of multiple blocks that communicate with each other through an integrated router. Data routing from a block to another poses many challenges. The network-on-chip (NoC) was used for the transmission of data from a source to a destination with high reliability, high speed, low power consumption, and low hardware occupation. An NoC is composed of a router, network links (NL), and network interface (NI). The main component of the NoC, the NI, is composed of an input/output FIFO, a finite state machine (FSM), pack, and depack modules. Data transmission from a block to another poses a security problem such as secret information extraction. In this paper, we proposed a data encryption framework for NoC based on a light encryption device (LED) algorithm. The main advantages of the proposed algorithm are to reduce the implementation area and to achieve high speed while reducing the power consumption. The proposed encryption framework was simulated Verilog/VHDL on the Xilinx ISE and implemented on the Xilinx Virtex 5 XC5VFX200T. The obtained results have shown that the proposed framework has a smaller area and higher speed compared to existing works. The proposed algorithm has reduced the NI implementation area and enhanced the network performance in terms of speed and security.

1. Introduction

The recent advances in technology have resulted in increasing the number of intellectual property (IP) cores in the same integrated chip. Most SoCs have integrated many cores such as memory units, general-purpose processors, digital signal processors (DSP), and graphics processing units (GPU). Integrating such a big number of cores has raised the problem of data routing for the communication between the IP cores. Besides, the complexity of the chip architecture and the huge number of integrated IPs and connecting those IPs through direct connections elevate the electromagnetic interference which causes more electrical noise. To make the SoC more reliable, a shared bus connection was used to reduce connection wires. Network-on-chip (NoC) [1] has been a solution to ensure communication between IPs without any overlap. The NoC was designed to establish data transmission effectively. It is composed of three main components which are the router, the network interface (NI), and the control fine state machine (FSM). The router is a set of switches used to control the data destination. The NI is used to receive and send data. The FSM is used to control the router and the NI. The NoC solution was very scalable and ensure date communication without any performance degradation of the SoC. However, the use of the NoC introduces security issues such as data extraction, hijacking, and denial server attacks. Data encryption was proposed as a solution for this problem.

Most SoCs contain confident information that must be accessed only by authorized operators. Securing the transmission of that information is a serious action to maintain scalability. Cryptography algorithms were proposed for data inscription. The Advanced Encryption Standard (AES) algorithm [2] was the most used for NoC data encryption [3]. But the AES needs a large implementation area and a lot of computation effort to encrypt the data. Thus, this may increase the implementation area of the SoC and power consumption of the system. Most SoCs such as IoT devices are characterized by a limited implementation area and power. To avoid those hindrances, we proposed the use of lightweight cryptography algorithms.

The main motivation of this work was the limited resources of SoCs that require custom solutions to ensure high reliability. Hardware implementation of NI can enhance the overall performance and reduce the power consumption of the chip which is considered an important constraint in mobile devices. Motivated by the mentioned, we propose a solution based on lightweight cryptography and a custom design.

Lightweight cryptography algorithms [4] are a new set of encryption algorithms designed for real-time processing featured with a small implementation area in addition to very low power consumption. Those algorithms are considered a perfect solution for data encryption in NoCs. Also, the low power consumption of those algorithms allows them to resist physical attacks. In this work, we proposed the use of the light encryption device (LED) algorithm [5] for NoC data encryption.

The LED algorithm is a lightweight cryptography algorithm from the substitution-permutation network (SPN) family. It has two versions, the first one with a key length of 128 bits (LED128) and the second one with a key length of 64 bits (LED64). For LED128, the number of encryption or decryption rounds is 48, and for LED64, it is 32 rounds. In this work, we selected the LED64 for NoC data encryption due to its fast processing and lower implementation area. To make use of the LED64 algorithm in the NoC, the NI was redesigned. The NoC sends data in packages of 32 bits where 16 bits are reserved for the destination address and the other 16 bits are the information to send. Since the LED64 takes an input of 64 bits, we proposed an encoder that transforms the 16 bits to 64 bits for the encryption and a decoder that transforms the 64 bits to 16 bits in the receiver decryption process. These additional encoder and decoder modules have further improved the security level against information extraction.

Furthermore, the NI was improved with a bigger input/output FIFO to speed up the processing and eliminate the data waiting. The proposed NI has five input/output FIFOs each with 32 bits of length. The proposed NI has been designed to reduce the implementation area and to enhance the operating frequency and the network bandwidth. The proposed lightweight cryptography algorithm has a low implementation area and a fast processing speed which allows achieving real-time processing even on low-performance devices such as IoT devices, unlike AES-based solutions that need high-performance computers to achieve real processing and require huge implementation area on hardware.

The novelty of the proposed solution comes from the use of custom-made input/output FIFO that allows the process of data faster and smoother without any delays. Besides, the lightweight cryptography algorithm plays an important role in enhancing the overall performance by reducing the implementation area and accelerating the processing time to achieve real-time processing.

The proposed solution was synthesized on the Xilinx Virtex 5 XC5VFX200T using the Xilinx ISE with VHDL. The proposed NI with the LED64 has only occupied 1420 LUT slices which is 1% of the available slices. The implementation area in terms of LUT-FF pairs was only 660 which does not present a big difference compared to the NI without the encryption algorithm. A maximum frequency of 358 Mhz was achieved. Compared to existing solutions, the proposed NI has a lower implementation area and higher operating frequency with a big margin.

The main contributions of this paper are the following:

(i) Proposing an NI for NoC with high performance and security level

(ii) Proposing a lightweight encryption algorithm for NoC based on the LED64 algorithm

(iii) Proposing a bigger input/output FIFO to avoid data waiting and accelerate the processing time

(iv) Synthetizing the proposed solution on the Xilinx Virtex 5 XC5VFX200T

(v) High performance achieved with a small footprint and high processing speed

(vi) Guaranteeing the use of the proposed NI with limited resources devices.

The rest of the paper is organized as follows: section 2 was reserved for presenting and discussing existing works on NoC data encryption. The proposed approach was presented and detailed in section 3. In section 4, experimental results were demonstrated and discussed. Conclusions are presented in section 5.

2. Related Works

Recent advances in communications have raised many security concerns. Mainly, Internet of things (IoT) devices are weak targets for hackers due to their limited computation resources that prevent the implementation of powerful encryption algorithms. Most IoT devices are based on SoCs where their data must be confidential. Data encryption in NoCs has been widely studied and many solutions have been proposed to achieve a high-security level. Besides, many IoT authentification protocol has been proposed to address the security concerns of data transfer trough servers and clouds.

Oliveira et al. [6] proposed NoC data encryption based on the combination of the AES algorithm and a firewall. The proposed firewall was used as an interconnection between the router, the AES block cipher, and the Processing Elements. It was integrated between the NI and the router to control the communications. The firewall was placed outside of the NI to avoid any modifications in the original architecture. It was used to decide on the sent or received if it will be encrypted/decrypted or not. The data encryption module was based on the AES algorithm. The plain text was loaded using two 64-bit blocks at consecutive clock cycles. Then both keys are injected to start the encryption process. The encryption or decryption process takes 13 clock cycles to generate the final output. The proposed approach requires a total area of 18116 slices. The achieved results proved that the proposed solution requires a huge implementation area which makes it not suitable for embedded SoCs.

An NoC encryption solution based on the AES was proposed in [7]. To reduce the implementation area of the proposed AES, a custom C-S-box was proposed. The new C-S-box has been designed using optimized MI units and four transistors XOR gates. To speed up the processing speed a retiming technique was applied. The retiming technique ails to place registers at each stage of the AES, and then the registers were placed in each round key operation. This technique allows fetching and decoding available data while fetching the next data value. Besides, while the available data are in the execution step, the next data are in the decoding phase and the new data are fetched similarly. Due to adding registers at different places, the proposed approach has reached high speed but suffers from high resource utilization and hog power consumption. The proposed solution cannot be considered for the implementation of limited power devices.

Charles and Mishra [8] proposed to secure data routing in NoC using incremental cryptography [9]. Incremental cryptography has significant improvement over traditional cryptography. It can be used to encrypt documents and videos faster by considering changed values and maintains already encrypted values. Such properties allow to speed up the processing time and achieve a better security level. The hummingbird-2 [10] was used as an encryption algorithm. This algorithm has a block size of 16 bits and a key length of 128-bits. Based on the proposed incremental cryptography technique, data sent by the NoC are compared to the data sent previously using XOR gate. The data were divided into different and similar blocks: different blocks are encrypted and similar blocks are maintained. The proposed solution has enhanced the encryption speed but raised the implementation area.

Tewari et al. [11] proposed a lightweight data encryption protocol to secure radio frequency identification (RFID) communication. The proposed protocol was based on bitwise XOR operation and right shifting and rotation. This protocol was designed to ensure authentification between the server and the RFID tag. The proposed method was ultralight and respects real-time constraints but it was too weak against a variety of attacks and cannot ensure authentification.

A data transfer encryption method was proposed in [12]. The proposed method was designed based on four-image encryption coupled with quaternion Fresnel transform, chaos, and computer-generated hologram. The four images were represented by quaternion algebra and then processed using quaternion Fresnel transform. After that, the processed data were encrypted using Fresnel transform with two virtual independent random phase masks. The proposed method was efficient for image encryption but computationally extensive which makes it unuseful for practice applications.

A smart city authorization method [13] based on blockchain authentification was proposed to grant secured access. The proposed method was developed based on the FIWARE platform. Blockchain technology was proposed to eliminate secure data replica with distributed system problems. The proposed method was computationally extensive and mainly useful for cloud computing and servers.

Li et al. [14] proposed the use of cooperative neural networks (CNN) for image watermarking generation scenarios for smart city applications. First, the gray watermark image was processed. Then, the processed image was embedded as a watermark signal through the block Discrete Cosine Transform. Finally, CNN was used to detect and extract the watermark. Such a method cannot be used with limited resources devices.

A secure environment for big data sharing confidentiality through 6G wireless connectivity was proposed in [15]. The proposed method relies on a combination of powerful algorithms to manage a predefined scenario. The proposed method was proposed for use in high-performance cloud servers.

Al-Qerem et al. [16] proposed a concurrency control protocol for fog and cloud computing and IoT communication. The proposed protocol aims to reduce the communication between the IoT device and the cloud through processing data at the fog node locally. Intensive computations have been performed to evaluate the performances of existing protocols.

Most of the proposed encryption methods for NoC have focused on the security level without considering either the encryption speed of the implementation area. In this work, we proposed a NI design that allows the processing of data faster alongside a lightweight encryption algorithm to ensure data security. More details on the proposed approach will be provided in the next section.

3. Proposed Approach

In this section, we provide a detailed description of the proposed NI and discussed the impact of the use of the lightweight cryptography algorithm on the performance of the NoC.

The NoC is an emerging technology that ensures communication between different IPs in SoCs without any overlap and allows the use of shred bus to reduce communication wires. The typical architecture of the NoC is presented in Figure 1. The NoC is composed of three main parts which are the routers, the network link (physical links), and the network interface (NI). The most important part is the NI which organizes the communication and allows to send and receive data from the IP core.

The proposed NI is suitable for SoCs with a large number of IPs. It can be used for secure and nonsecure IPs. For secure IPs, a lightweight cryptography algorithm was added. The design of the NI has been enhanced with a larger input/output FIFO to eliminate waiting when sending or receiving data. The proposed input/output FIFO is composed of 5 registers; each has a 32-bits length. The proposed NI is presented in Figure 2.

To secure the communication between IPs, an encryption method is based on the LED algorithm. LED is a lightweight encryption algorithm with a low implementation area that makes it suitable for limited computation resources data such as IoT. It was used for data encryption to achieve a low implementation of the NoC and to guarantee real-time processing.

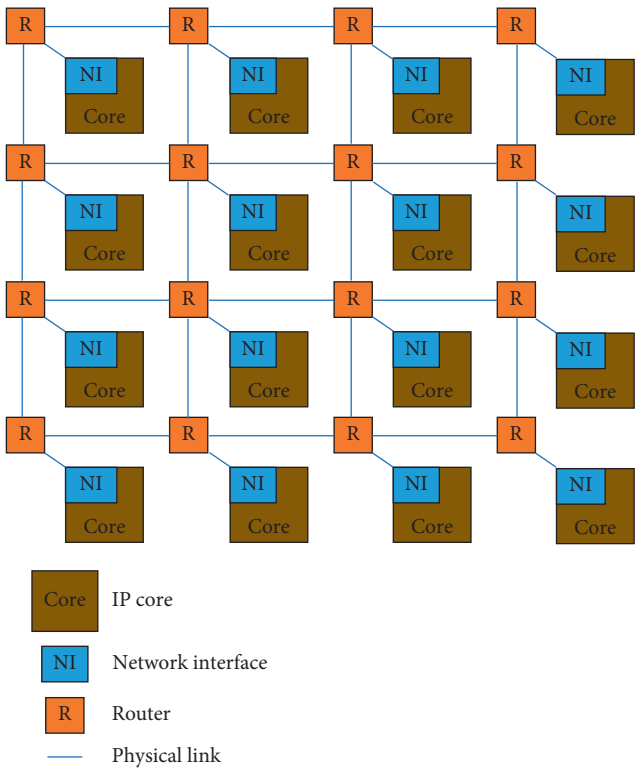

Figure 1: Typical architecture of the NoC.

The LED is an encryption algorithm from the S-PN family. It has two versions based on the key length. The first is LED64 with a key length of 64-bits and the second version is the LED128 with a key length of 128 bits. For LED64, the total number of rounds is 32 and for the LES128, the total number of rounds is 48. In this work, the LED64 was used for data encryption/decryption. For further use, the LED refers to the LED64 version.

The LED algorithm is composed of four main functions which are AddConstants, S-Box, ShiftRows, and MixColumns. The AddConstants function aims to perform Xor between the plain text and a constant matrix. To build the matrix, the size of the key was represented by 8 bytes from Ks_7 to Ks_0. To make the first column of the matrix, a Xor function was performed between the key bytes and their position. For the second column, six-round constant bits (RC_5 to RC_0) were initialized by zero and then updated for each round by a one-bit left shifting and RC_0 takes the result of the Xor between RC_5, RC_4, and one. The constant matrix is presented by 1.

$$\begin{bmatrix} 0 \oplus (Ks_7 \| Ks_6 \| Ks_5 \| Ks_4) & (RC_5 \| RC_4 \| RC_3) & 0 & 0 \\ 1 \oplus (Ks_7 \| Ks_6 \| Ks_5 \| Ks_4) & (RC_2 \| RC_1 \| RC_0) & 0 & 0 \\ 2 \oplus (Ks_3 \| Ks_2 \| Ks_1 \| Ks_0) & (RC_5 \| RC_4 \| RC_3) & 0 & 0 \\ 3 \oplus (Ks_3 \| Ks_2 \| Ks_1 \| Ks_0) & (RC_2 \| RC_1 \| RC_0) & 0 & 0 \end{bmatrix}. \quad (1)$$

The S-Box function takes 16 bits as input and transformed them to another value based on a substitution table. Table 1 represents the substitution tale of the S-Box of the LED algorithm.

The ShiftRows function rotated the input to right. The values of the first, second, and third rows are shifted to the right and the last row is rotated to the first row.

The MixColumns function is a multiplication of the input data by a diffusion matrix MDS. The MDS is represented by 2.

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

The LED block chipper takes an input block with the same size of the key length (64 bits). After 32 rounds, the input plaintext is transformed into a ciphertext. The encryption process of the LED algorithm is defined by the pseudocode in Algorithm 1.

The LED algorithm starts by generating subkeys from the secret key then performs and a round key function through a Xor between the state and the subkeys. This process is repeated after each step. Figure 3 illustrated the encryption process of the LED block cipher.

The LED algorithm was selected for data encryption thanks to its security level, low implementation area, and low power consumption. As mentioned earlier, the LED takes an input data and key with a length of 64 bits but the NoC generates data packages of 32 bits where 16 bits represent the destination address and 16 bits represent the information. So, only 16 bits are available for encryption. To fix this issue, we proposed a decoder that transforms 16-bit input to 64-bit output. At the decryption process, an encoder was used to transform the 64 bits into 16 bits. The proposed solution has enhanced the security level against data retrieval. In case of an attack, if the data was retrieved, the information cannot be interpreted since it has been represented on 64 bits where only 16 bits have the information. The proposed encoder/decoder has allowed for the integration of the encryption algorithm without any modification on NI architecture. After encryption, the ciphertext was concatenated with the 16 bits of the destination address and sent to the router. The proposed NI with the LED block cipher is presented in Figure 4.

In summary, we proposed an NI for NoC with an encryption algorithm to ensure data security. The proposed NI was designed to be suitable for any NoC with a bigger input/output FIFO composed of five registers where each is of 32-bit size. The use of bigger FIFO allows to speed up the processing time while eliminating data waiting. Besides, we implemented a data encryption algorithm based on the LED block cipher. The LED is a lightweight encryption algorithm with a low implementation size and fast processing speed. An encoder/decoder method was added to the NI to convert 16-bit data to 64-bit data and the inverse to allow its encryption by the LED since it takes 64-bit input and output. This addition has enhanced the security level against data retrievals.

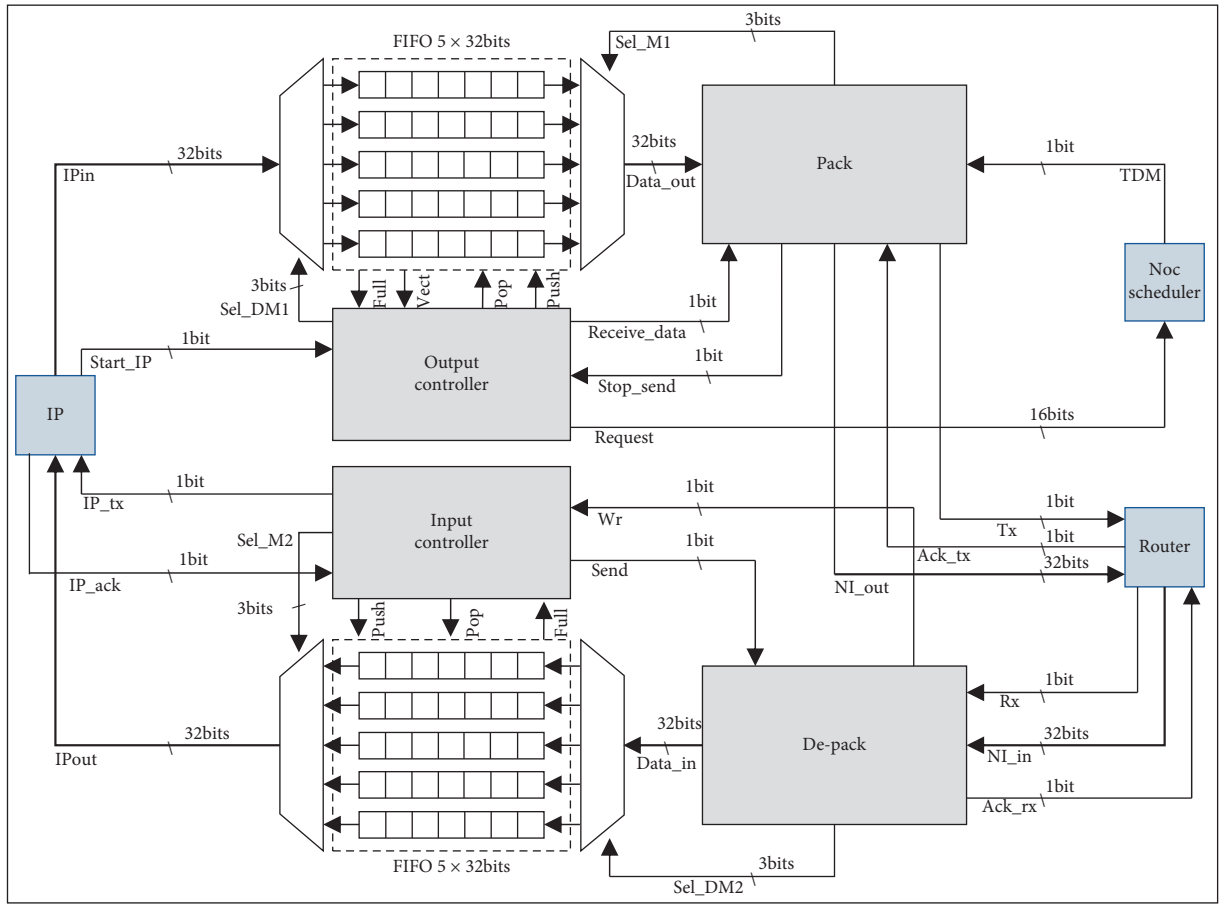

Figure 2: Proposed NI without encryption algorithm.

Table 1: Substitution tale of the S-Box function in 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

4. Experiments and Results

The proposed NI design with encryption algorithm was synthesized on the Xilinx Virtex 5 XC5VFX200T using the Xilinx ISE with VHDL. First, we start by building the NI without the encryption algorithm to determine its implementation area and for comparison purposes against existing works. Figure 5 presents the implementation results of the NI without encryption.

As shown in Figure 5, the proposed NI has a low implementation area with a total of 51% utilization in terms of LUT-FF pairs. In terms of slice register and slice LUTs, the utilization was considered negligible. A maximum frequency of 358 MHz was achieved. To prove the efficiency of the proposed design, a comparative study was conducted with state-of-the-art works that use the same implementation device, the Xilinx Virtex 5 XC5VFX200T. Table 2 presents a comparison with existing works.

As shown in Table 2, the proposed NI design has the lowest LUT-FF pairs that refer to the implementation area. Even when using more registers, the proposed design still requires less area than existing designs.

To secure the data transfer between IPs, the LED block cipher was deployed. An encryption process was implemented at the data generation stage and a decryption process was implemented at the data receiving stage. Figure 6 presents the encryption implementation area.

The LED encryption process requires a very small implementation area with only 81 LUT-FF pairs. The achieved results proved the light size of the LED block cipher which makes it suitable for use in limited resources devices. A maximum frequency of 252.816 MHz allows for running in real time.

At the data receiving stage, the decryption process was implemented. Figure 7 presents the achieved results for the decryption process. The LED decryption process requires only 86 LUT-FF pairs which are approximately equal to the encryption process. A maximum frequency of 225.73 MHz was achieved.

Compared to the AES [7] that requires more than 1300 LUT-FF pairs and works at a maximum frequency of 155 MHz, the LED block cipher is much better and more efficient especially for limited resources devices such as IoT devices.

To ensure data security, the LED was combined with the proposed NI design. The achieved results are presented in

```
Input (P) : A 64-bit plaintext
secret key (K): 64-bits.
Output (C) : A 64-bit ciphertext.
   S ⟵ P
   SK ⟵ GenerateSubkey (K)
   S ⟵ AddRoundKey (S, SK)
   For j = 1 to (r/4) do
      For i = 1 to 4 do
         S ⟵ AddConstants (S)
         S ⟵ S-Box (S)
         S ⟵ ShiftRow (S)
         S ⟵ MixColumns (S)
      End for
      S ⟵ AddRoundKey (S, SK)
   End for
   C ⟵ AddRoundKey (S, SK)
   Return C
```

Algorithm 1: LED encryption process.

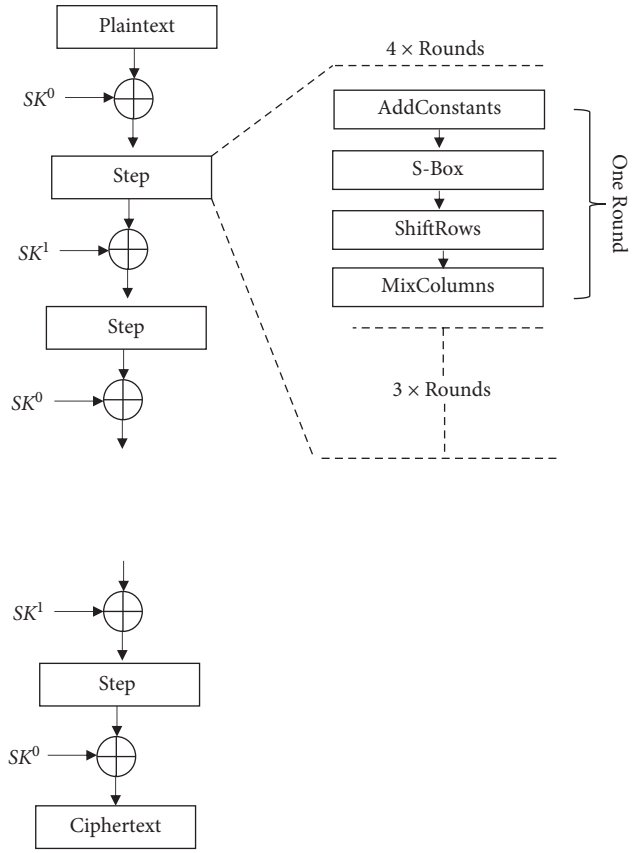

Figure 3: Encryption process of the LED algorithm.

Figure 8. The proposed NI design with the LED block cipher requires a total of 660 LUT-FF pairs and works at a maximum frequency of 221.727 MHz. The achieved results proved that the proposed design is very efficient with a low implementation area and high frequency. The secure NI design has a small difference in implementation area compared to the nonsecure design thanks to the small area of the LED block cipher.

To prove the efficiency of the proposed NI design, we compared the achieved implementation results with existing

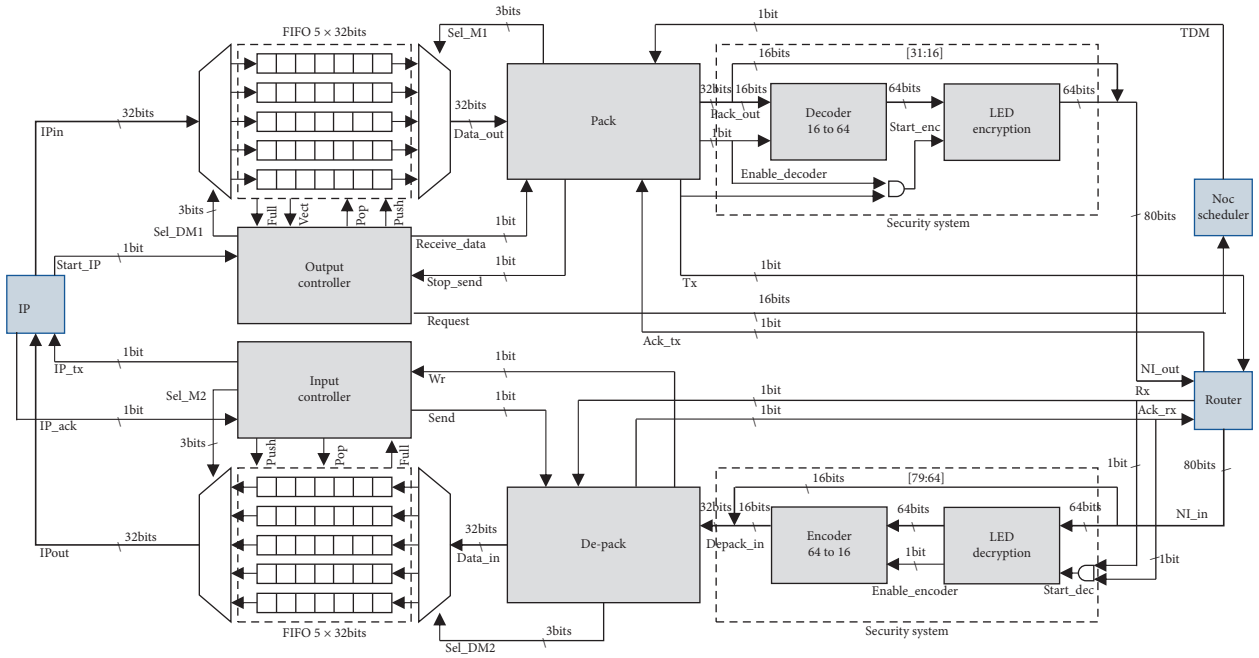

Figure 4: Proposed NI with the LED block cipher.

Top_level Project Status (12/18/2020 –16:13:41)			
Project file:	Led.xise	Parser Errors:	No errors
Module Name:	Top_level	Implementation State:	synthesized
Target Device:	Xc5vfx200t-2ff1738	i) Errors:	No Errors
Project Version:	ISE 14.6	ii) Warnings	3warnings (3 new)
Design Goal:	Balanced	iii) Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	iv) Timing Constraints:	
Environment:	System Setting	v) Final Training Score	

Device utilization summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	144	122880	0%
Number of Slice LUTs	381	122880	0%
Number of fully used LUT-FF pairs	81	444	18%
Number of bonded IOBs	36	960	3%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 5: NI design without encryption.

works that use the Xilinx Virtex 5 XC5VFX200 T device. Table 3 presents a comparison of the proposed secure NI design with existing works.

As shown in Table 3, the proposed secure NI design presents a much lower implementation area compared to existing works based on the AES block cipher. The proposed NI requires 660 LUT-FF pairs while the AES-based NI requires 38342 LUT-FF pairs. The reported results proved the proposed NI is more efficient than those based on the AES and more suitable for IoT devices.

The proposed NI design has been built for general use with any NoC. First, it is characterized with larger input/output FIFO composed of 5 registers; each is 32 bits long used to avoid data waiting. This feature allows to speed up the processing time and avoid long processing delays. Second, to secure the communication between IPs, the LED block cipher was implemented. An encoder/decoder method was added to endure data compatibility with the LED block cipher that uses 64-bit input and output. The proposed NI design has achieved much better results in terms of implementation area and frequency compared to existing works.

Table 2: Comparison of the proposed NI design with existing works.

Ni design	Slice registers	Slice LUTs	LUT-FF pairs
[7]	627	765	1580
[17]	513	1567	880
Proposed	813	629	493

Top_level Project Status (12/18/2020 –16:15:50)			
Project file:	Led.xise	Parser Errors:	No errors
Module Name:	LED_decryption	Implementation State:	synthesized
Target Device:	Xc5vfx200t-2ff1738	i) Errors:	No Errors
Project Version:	ISE 14.6	ii) Warnings	3warnings (0new)
Design Goal:	Balanced	iii) Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	iv) Timing Constraints:	
Environment:	System Setting	v) Final Training Score	

Device utilization summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	144	122880	0%
Number of Slice LUTs	409	122880	0%
Number of fully used LUT-FF pairs	86	467	18%
Number of bonded IOBs	84	960	8%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 6: LED encryption results.

Top_level Project Status (12/18/2020 –16:17:51)			
Project file:	Led.xise	Parser Errors:	No errors
Module Name:	Network_interface	Implementation State:	synthesized
Target Device:	Xc5vfx200t-2ff1738	i) Errors:	No Errors
Project Version:	ISE 14.6	ii) Warnings	1warning (1new)
Design Goal:	Balanced	iii) Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	iv) Timing Constraints:	
Environment:	System Setting	v) Final Training Score	

Device utilization summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	813	122880	0%
Number of Slice LUTs	629	122880	0%
Number of fully used LUT-FF pairs	493	949	51%
Number of bonded IOBs	137	960	14%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 7: LED decryption results.

Top_level Project Status (12/18/2020 –16:04:00)			
Project file:	Led.xise	Parser Errors:	No errors
Module Name:	Network_interface	Implementation State:	synthesized
Target Device:	Xc5vfx200t-2ff1738	i) Errors:	No Errors
Project Version:	ISE 14.6	ii) Warnings	1warning (0new)
Design Goal:	Balanced	iii) Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	iv) Timing Constraints:	
Environment:	System Setting	v) Final Training Score	

Device utilization summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	1101	122880	0%
Number of Slice LUTs	1420	122880	1%
Number of fully used LUT-FF pairs	660	1861	35%
Number of bonded IOBs	233	960	24%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 8: Secure NI design.

Table 3: Comparison of the secure NI design with existing works.

NI design	Slice registers	Slice LUTs	LUT-FF pairs
[7]	5302	15309	19700
[11]	71484	61555	38342
Proposed	1101	1420	660

5. Conclusions

The emerging technology is pushing SoCs to their limit with a huge number of embedded IPs. NoCs were considered a solution to achieve perfect communication between IPs without any overlap and allow the use of shared bus and reduce connection wires. An NoC is composed of three main components which are the network interface (NI), the network links (NL), and the router. The NI is considered the most important component that organizes data sending and receiving. In this paper, we proposed an NI design with a lightweight block cipher to ensure data security. The proposed NI was designed for use with any NoC. It has a large input/output FIFO to avoid data waiting. The LED block cipher was used to encrypt data due to its low implementation area, fast processing speed, and high-security level. The reported results show that the proposed NI design outperforms existing works based on the AES block cipher with a wide range in terms of implementation area and working frequency. The proposed design is more suitable for implementation on limited computation resource devices such as IoT.

Data Availability

The data are available upon request to the corresponding author.

Conflicts of Interest

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

Authors' Contributions

R. Ayachi is responsible for manuscript drafting and revision, A. Mhaouch is responsible for experiment and validation, and A. B. Abdelaali is the project manager.

References

[1] S. Kumar, A. Jantsch, J.-P. Soininen et al., "A network on chip architecture and design methodology," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002*, pp. 117–124, IEEE, Pittsburgh , PA, USA, April 2002.

[2] S. Heron, "Advanced encryption standard (AES)," *Network Security*, vol. 2009, no. 12, pp. 8–12, 2009.

[3] B. S. Kerakalamatti and P. Nagaraj, "Design and implementation of NOC based parallel AES computation," *International Journal of Computer Applications*, vol. 975, p. 8887, 2015.

[4] S. B. Sadkhan and A. O. Salman, "A survey on lightweight-cryptography status and future challenges," in *Proceedings of the 2018 International Conference on Advance of Sustainable*

Engineering and its Application (ICASEA), pp. 105–108, IEEE, Iraq, March 2018.

[5] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 326–341, Springer, Nara, Japan, October 2011.

[6] B. Oliveira, R. Reusch, H. Medina, and F. Moraes, "Evaluating the cost to cipher the NoC communication," in *Proceedings of the 2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS)*, pp. 1–4, IEEE, Puerto Vallarta, Mexico, February 2018.

[7] N. L. Venkataraman and R. Kumar, "An efficient NoC router design by using an enhanced AES with retiming and clock gating techniques," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 12, p. e3839, 2020.

[8] S. Charles and P. Mishra, "Securing network-on-chip using incremental cryptography," in *Proceedings of the 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 168–175, IEEE, Limassol, Cyprus, July 2020.

[9] M. Bellare, O. Goldreich, and S. Goldwasser, "Incremental cryptography and application to virus protection," in *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 45–56, Lasvegas, NV, USA, May 1995.

[10] D. Engels, M.-J. O. Saarinen, P. Schweitzer, and E. M. Smith, "The Hummingbird-2 lightweight authenticated encryption algorithm," in *Proceedings of the International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pp. 19–31, Springer, Nijmegen, The Netherlands, July 2011.

[11] A. Tewari and B. B. Gupta, "Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags," *The Journal of Supercomputing*, vol. 73, no. 3, pp. 1085–1102, 2017.

[12] C. Yu, J. Li, X. Li, X. Ren, and B. B. Gupta, "Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram," *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4585–4608, 2018.

[13] C. Esposito, M. Ficco, and B. B. Gupta, "Blockchain-based authentication and authorization for smart city applications," *Information Processing & Management*, vol. 58, no. 2, Article ID 102468, 2021.

[14] D. Li, L. Deng, B. Bhooshan Gupta, H. Wang, and C. Choi, "A novel CNN based security guaranteed image watermarking generation scenario for smart city applications," *Information Sciences*, vol. 479, pp. 432–447, 2019.

[15] C. L. Stergiou, K. E. Psannis, and B. B. Gupta, "IoT-based big data secure management in the fog over a 6G wireless network," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5164–5171, 2020.

[16] A. Al-Qerem, M. Alauthman, A. Almomani, and B. B. Gupta, "IoT transaction processing through cooperative concurrency control on fog-cloud computing environment," *Soft Computing*, vol. 24, no. 8, pp. 5695–5711, 2020.

[17] H. K. Kapoor, G. B. Rao, S. Arshi, and G. Trivedi, "A security framework for noc using authenticated encryption and session keys," *Circuits, Systems, and Signal Processing*, vol. 32, no. 6, pp. 2605–2622, 2013.