Indonesian Journal of Electrical Engineering and Computer Science
Vol. 20, No. 1, October 2020, pp. 287~299
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v20.i1.pp287-299

Design and implementation of pipelined and parallel AES encryption systems using FPGA

Mohamed Nabil[1], Ashraf A. M. Khalaf[2], Sara M. Hassan[3]
[1,2]Electronics and Communications Engineering Department, Faculty of Engineering, Minia University, Egypt
[3]Electronics and Communications Engineering Department, Modern Academy for Engineering and Technology, Egypt

Article Info

Article history:

Received Jan 9, 2020
Revised Mar 15, 2020
Accepted Apr 19, 2020

Keywords:

AES
Encryption
Network
Pipelined
Parallel

ABSTRACT

The information security is one of the most important issues in the design of any communication network. One of the most common encryption algorithms is the Advanced Encryption Standard (AES). The main problem facing the AES algorithm is the high time consumption due to the large number of rounds used for performing the encryption operation. The more time the encryption system consumes to encrypt the data, the more chances the hackers have to break the system. This paper presents two effective algorithms that can be used to solve the mentioned problem and to achieve an effective processing time reduction using pipelined and parallel techniques to perform the encryption steps. These algorithms are based on using certain techniques to make the system able to encrypt many different states (the data will be encrypted) in the same time with no necessity to wait for the previous encryption operation to be completed. These two algorithms are very effective especially for big data size. This paper describes in detail the AES encryption system algorithm and a detailed explanation for the proposed algorithms. Moreover, the research shows the implementation of the three algorithms: the traditional, the pipelined, and the parallel algorithms, and finally a comparison between them.

Corresponding Author:

Mohamed Nabil,
Electronics and Communications Department,
Faculty of Engineering, Minia University, Minia Egypt.
Email: mohammednabel5050@gmail.com

1. INTRODUCTION

The data encryption techniques are algorithms where important data is encrypted, and the only client who can access this data information is the user who has the correct encryption key. The scientists try to develop a new encryption algorithm to prevent hacking; the update of hacking and the continuous discovery of gaps makes the process of the information security more difficult. The Advanced Encryption Standard (AES) is one of the data encryption techniques [1, 2].

The AES is also known as Rijndael algorithm, and it is developed by Vincent Rijmen and Joan Daemen. The United States National Institute of Standards and Technology (NIST) has determined three algorithms of the AES, each has 128 bits block size, but the key length is one of these sizes: 128, 192, or 256 bits. It is a symmetric-key technique that uses the same key for encrypting and the decrypting [3, 4].

There are many algorithms that were developed to provide security and the needed performance. These algorithms are different in their performance, speed, backdoors, and strength. The Data Encryption Standard (DES) is one of these algorithms [5, 6].

The DES was the first algorithm used by NIST. The development of the DES consists of three different-length algorithms. The double DES, the triple DES with two keys, and finally the triple DES with three keys. The crackers have decrypted the DES. One of the attack techniques is the brute force attack that depends on trying as many keys as possible to attack the message. The AES replaced the DES and the

development of the AES was a process to avoid the backdoors of the DES, after the DES was adopted [7, 8]. Blowfish encryption system is another algorithm used instead of DES. Blowfish divides the data into sections of 64 bits and encrypts them separately. It is used in a software like e-commerce for payments security and passwords to protect passwords. It is also known as one of the most flexible encryption algorithms.

Another encryption algorithm is the International Data Encryption Algorithm (IDEA) which uses 64-bit block size and 128-bit key size. It also uses the rounds techniques and adopts a methodology called half rounds where each round uses 6 sub keys of 16 bits. Each of the half rounds uses 4 sub keys [9]. The first 8 sub-keys are extracted from the encryption key while the other 8 are created based on rotation [10]. As mentioned before, many researches try to provide newer algorithms that are more secure than the traditional ones. One of the most important parameters that is taken into account when developing a new encryption algorithm is the speed of this new encryption technique [11, 12]. Therefore, the researchers update the recent encryption techniques to make the encryption process of these techniques faster than before without changing the encryption process itself.

The main problem of the AES encryption system is that it consumes lot of time to perform because of the large number of the rounds used, which makes the system vulnerable [13]. The more time the encryption system consumes to encrypt the data, the more chances the hackers have to break the system. In the next paragraphs, some related researches that were published in the past few years will be mentioned.

In [14],The authors presented an implementation of the AES-Counter (AES-CTR) symmetric cryptographic primitive using the CUDA framework. They presented quantitative data and compared them with the common CPU-based OpenSSL implementation. which is a software library for applications that can secure communications over computer networks. In [15], The authors used a processor with multi-cores to present the implementation of the AES algorithm with different data and task parallelism granularity and to show that AES implementation on a fine-grained, much-core system can achieve high performance throughput per unit of chip area and energy efficiency compared to other software platforms. In [16], The authors proposed a multi-processor arrays to design a parallel AES encryption system and achieve high throughput performance.

This paper provides two different techniques to speed up the processing of the AES encryption and to increase the reliability; the first is the pipelined encryption processing, and the second is the parallel encryption processing. These two techniques are different in the processing time and the used hardware resources. As the speed increases, more resources are consumed. The design, the analysis, and the implementation of these two techniques are discussed in this paper.

The paper is organized as follows: Section 2 describes the AES encryption system and its steps. Section 3 shows the methodology that we use to reach the goal of our research which is speeding up the encryption process. Section 4 discusses the hardware implementation of the normal process and the pipelined process; then, the simulation and results are presented in Section 5. Finally, the conclusion is discussed in Section 6.

2. AES ENCRYPTION

The AES technique consists of 9 main rounds in addition to the initial and final rounds. The 9 main rounds are similar in the processing but different in the used chipper key that is renewed every round. Figure 1 shows the AES Encryption [1, 17, 18].

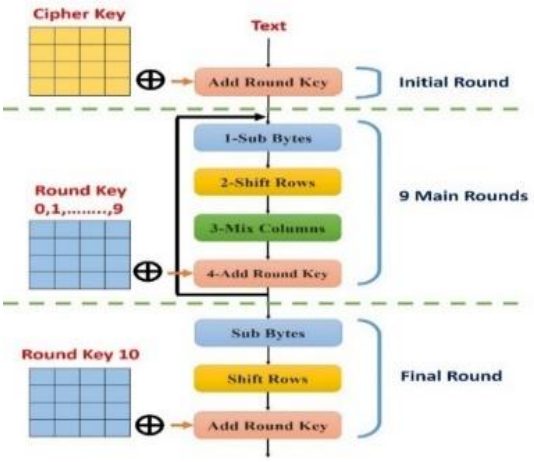

Figure 1. The AES encryption system

Both the state and the key consist of 128 bits. Each round consists of several steps to complete the encryption operation. In the decryption; the inverse steps are performed to obtain the original state [3, 19]. The first round is the initial round where the XOR operation is performed on the state and the key to produce the output of this round. Each round of the next 9 rounds consists of 4 steps that are repeated for every round [20, 21], as shown in the following subsections.

2.1. The subbytes step

The first step is the SubBytes step where every element of the state is converted to another value based on the mapping between the value of the element and the S-Box. The first four bits of the element represent the number of the row of the S-Box while the second four bits represent the number of the column of the S-Box. The value obtained from the S-Box due to the number of the row and the number of the column will replace the previous value of the element [22]. The SubBytes step is shown in Figure 2.

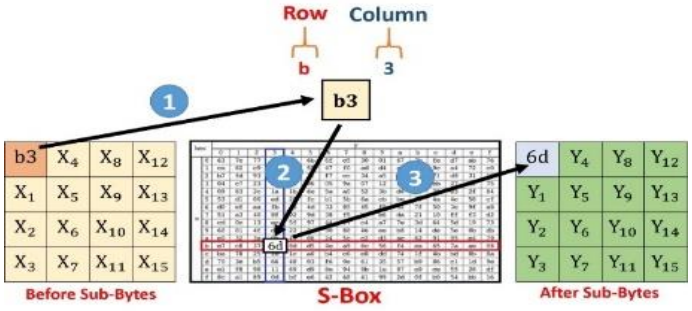

Figure 2. The Subbytes step

2.2. The shiftrows step

The elements of every row will be left shifted except the first row. The number of shifting depends on the number of the row where the first row will not be shifted as we mentioned; the second row will be shifted one time, the third row will be shifted twice and the final row will be shifted three times [23, 24]. Figure 3 shows the ShiftRows step.

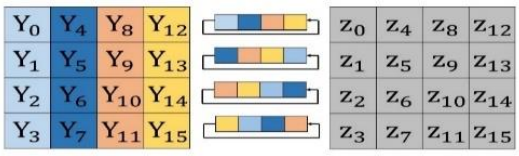

Figure 3. The shiftrows

2.3. The mixcolumns step

In this step, the four numbers of one column are modulo multiplied in Rijndeal's Galois Field by a given matrix as shown in Figure 4 [25, 26].

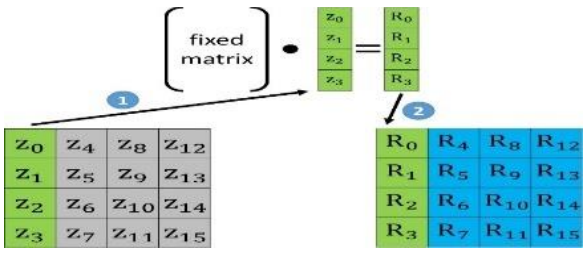

Figure 4. The mixcolumns step

2.4. The addroundkey step

The final step of each round of the main nine rounds is the AddRoundKey step where the XOR operation is performed on the input state and the chipper key to produce the output of each round [27, 28]. Figure 5 shows the AddRoundKey step.

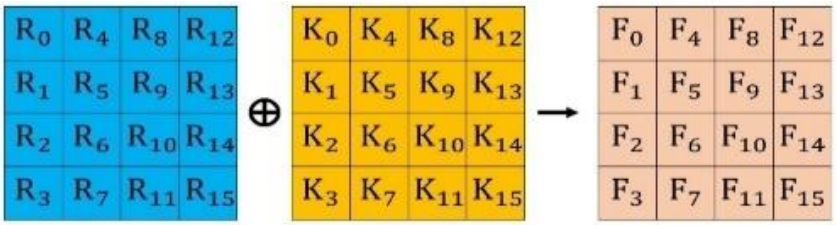

Figure 5. The Addroundkey step

In the final round, the previous steps will be performed except the MixColumns step [29].

- Key Schedule

The key will be updated for each round. First, the last column of the recent key will be down shifted for one time then the SubBytes operation will be performed on this shifted column then it will be XORed with the first column; then, the product will be XORed with certain column of the Rcon array (the number of the column of the Rcon array is the same number of the round) [30, 31]. The output of the previous operations will be the first column of the new key as shown in Figure 6.

The second, third, and fourth columns will be obtained by performing XOR operation on column number x of the recent key and the column number x-1 of the new key to obtain the column number x of the new key.

Figure 7 shows the XOR operation to obtain the second column of the new key [32, 33]. The traditional processing of the encryption is based on the idea of finishing the encryption of the recent 128 bits before starting the encryption of the new 128 bits [34]. This concept is shown in Figure 8.

This normal processing uses a large number of cycles in the encryption of 128 bits state which leads to a long time to encrypt all the required message which in turn makes the system vulnerable and also decreases the reliability, in addition to the large memory that will be used to store the input data till the encryption of the previous 128 bits finishes [13].

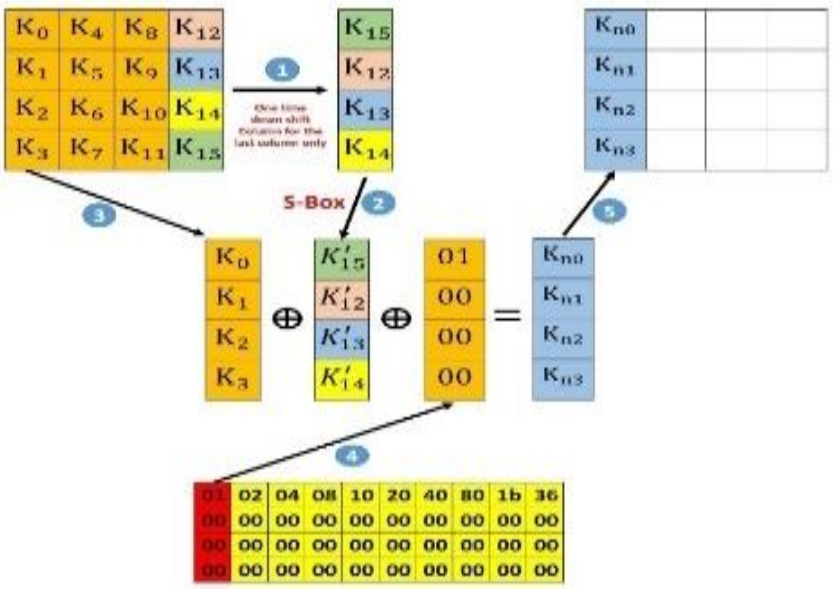

Figure 6. The expansion of the key for the first column

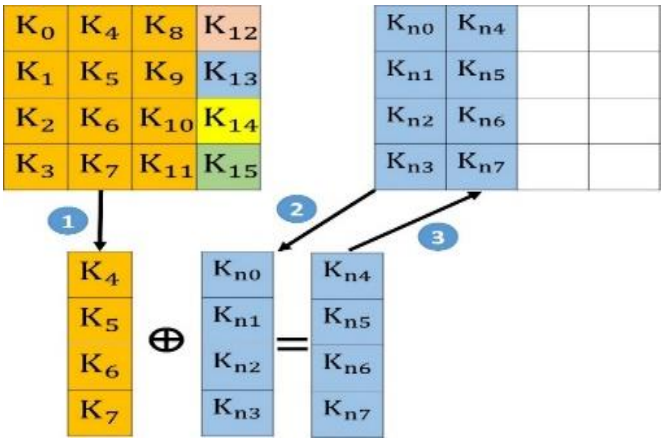

Figure 7. The XOR operation to obtain the second column of the new key

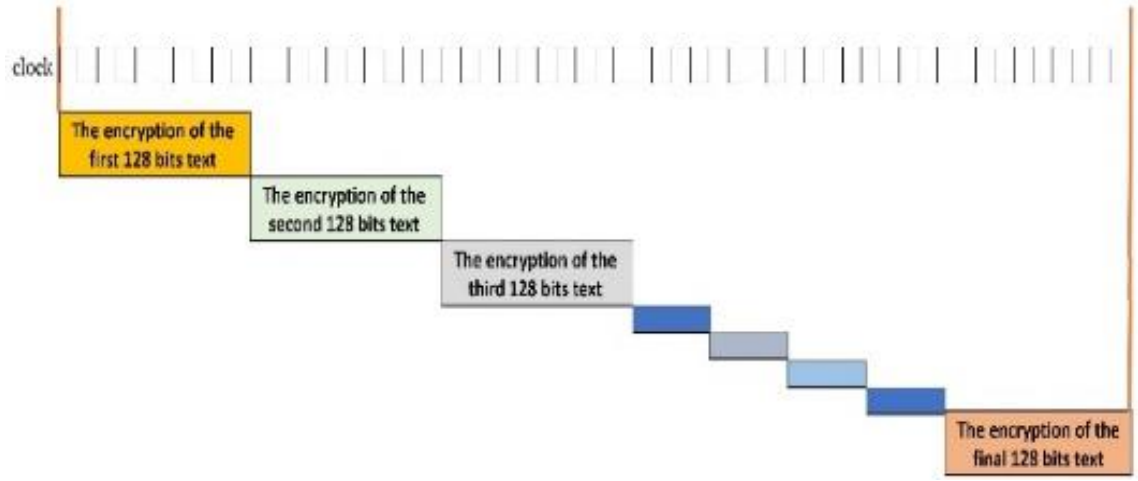

Figure 8. The traditional AES processing

3. THE PROPOSED PIPELINED AND PARALLEL ALGORITHMS

The pipelined algorithm is based on the idea of using most of the resources as much as possible by starting a new encryption process before the previous one is finished; this idea will reduce the processing time specially when there is a large number of the encryption states. The time reduction will be significant. The pipelined processing is shown in Figure 9.

As mentioned in this paper, AES consists of 9 main rounds in addition to the initial and final rounds [35]. So, the total number of rounds is 11 rounds. In the encryption process of the first state, the initial round (R_{in}) is being proceed in the first cycle. In the next cycle, the processing of the second round (R_0) will be performed, and at the same time the beginning of the encryption process for the second state will start, and so on. So in cycle of number 12, the output of the first encryption state will be finished and at the same cycle there are some processes for 12 different states that are being performed. There is an output in each cycle. So it is not necessary to wait for the first state to be finished to start the second state, but the input will be continuous for each cycle [36] as shown in Figure 9 where R refers to the round number.

Each round consists of four operations to encrypt any state [13]. The output of each round is divided into two parts: first, the encrypted state of the certain round; second, the calculated key that will be used in the next round. These two parts are the output of the recent round and the input of the next round. Based on this concept, the output of each round will be transmitted to the next round [34]. The main difference between the classical AES and the pipelined AES is the beginning of a new encryption state [37]. In the traditional process, the start of a new encryption process will begin after all the rounds of the previous encryption end, so each cycle has only one round; but in the pipelined AES every cycle in a new state encryption process will start. This will be effective in the time consumption but will consume more resources because the operation processing that will be performed in one cycle will increase [1, 38].

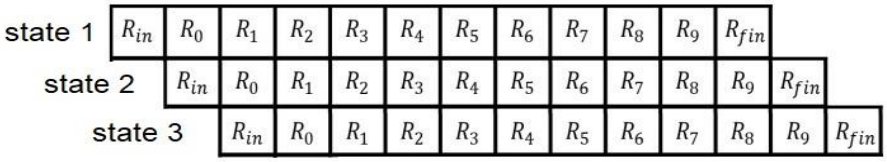

Figure 9. The pipelined AES processing

By using the pipelined algorithm, we will be able to reduce the number of the total cycles that will be used to encrypt the message. This idea makes the system faster and can be used in the communication system [26, 39]. The number of cycles that the system uses in the case of the pipelined processing can be also reduced by using the parallel processing technique which is based on the concept of making the processor work as two or more processors to achieve the encryption task as fast as possible [37, 40].

The code of the project is divided into sub-blocks to make the tracing of the code and the debugging easier. These sub-blocks are divided based on the rounds and the operations in each round. Therefore, there will be sub-block for every operation; so if we trace the code with the output of each sub-block, we can find the error as fast as possible.

As shown in Figure 10, there are 11 processes working in a parallel way. R represents the round number in the encryption process [33]. The parallel processing does not mean the recent round that is not based on the previous round, but it means that the instruction will be sequentially implemented but in the same cycle which consumes a large number of resources.

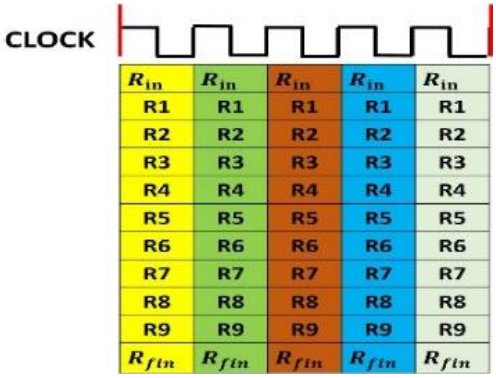

Figure 10. The parallel AES processing

As shown in Figures 9, 10 and 11, the total number of cycles will be reduced in the case of the pipelined and the parallel processing. This improvement makes the system fast and reliable; but, this high speed will make the system use more resources and more logic gates from the used available kit than what the traditional processing will use, through the concept of first in first out [40, 41]. If the used processor has low resources, the traditional processing will be used because in each cycle there is small number of operations that will be performed, so the used time will be high. However, if there is another processor that has more hardware resources, the pipelined processing will be used. this will make more operations to be performed in the same cycle, and in this case the consumed time will be smaller than that of the traditional processing; but the hardware resources will increase [40]. Therefore, in the parallel processing, all the encryption rounds will be performed in one cycle, and this leads to more hardware consumption than in the traditional and the pipelined processing, but the consumed time will be smaller than that of the traditional and the pipelined processing. In this paper, the VHSIC-Hardware Description Language (VHDL) code was optimized to reduce the consumed resources as much as possible, so the Spartan-3A/3AN FPGA Starter Kit is used for all the algorithms; but as seen, the consumed resources in the parallel processing is more than those of the pipelined and the traditional processing [30, 42].

4. HARDWARE IMPLEMENTATION

In the present research, Field Programmable Gate Array (FPGA) implementation and realization of the three methods of the AES will be presented as well. The Xilinx "Spartan-3A/3AN FPGA Starter Kit"

is used. These methods are designed by writing a code using the Xilinx package ISE 14.7 programmed and simulated by using the ISim simulator program. Figure 11 shows the schematic diagram of the traditional processing of the AES encryption system.

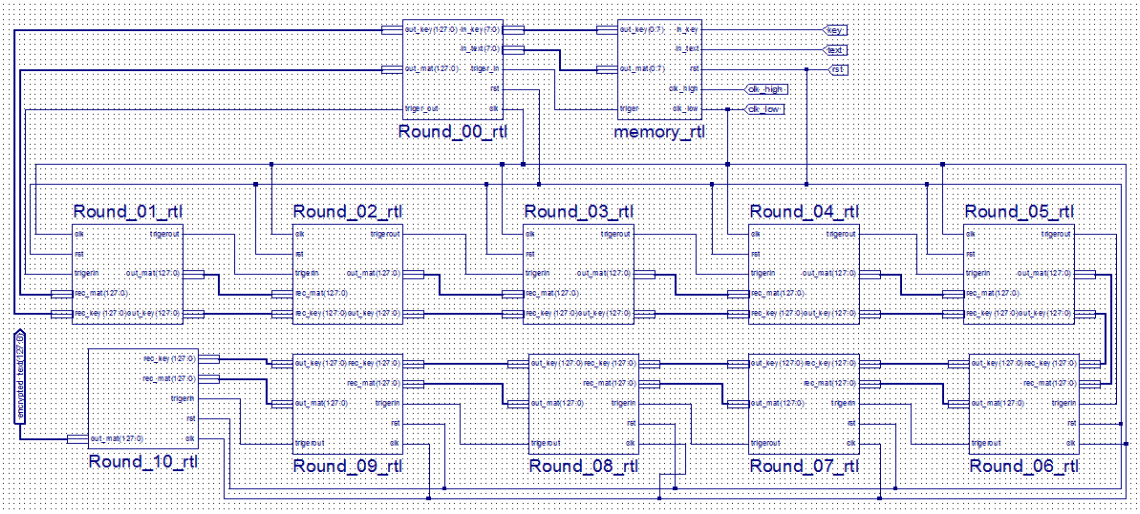

Figure 11. The schematic diagram of the traditional AES encryption system

Figure 15 shows the ISim simulation of the traditional method; as shown, the normal implementation will use 26 cycles to be able to encrypt any 128-bit state. This number of cycles is huge and makes the system very slow which can make the system vulnerable. If we assumed the state that will be encrypted and the given key as the matrices below, hence, the encrypted data will be as the encrypted output matrix shown by simulating the implementation. The simulation output will be as same as the expected results. We can analyze the performance and the speed of our implementation.

$$\text{State} = \begin{bmatrix} 32 & 88 & 31 & E0 \\ 43 & 5A & 31 & 37 \\ F6 & 30 & 98 & 07 \\ A8 & 8D & A2 & 34 \end{bmatrix} \quad \text{Key} = \begin{bmatrix} 2B & 28 & AB & 09 \\ 7E & AE & F7 & CF \\ 15 & D2 & 15 & 4F \\ 16 & A6 & 88 & 3C \end{bmatrix} \quad \text{Encrypted output} = \begin{bmatrix} 39 & 02 & DC & 19 \\ 25 & DC & 11 & 6A \\ 84 & 09 & 85 & 0B \\ 1D & FB & 97 & 32 \end{bmatrix}$$

As shown in Figure 11, there is a memory to store the input data during the processing stages to avoid the loss of any bits; the schematic implementation of the pipelined AES is shown in Figure 12.

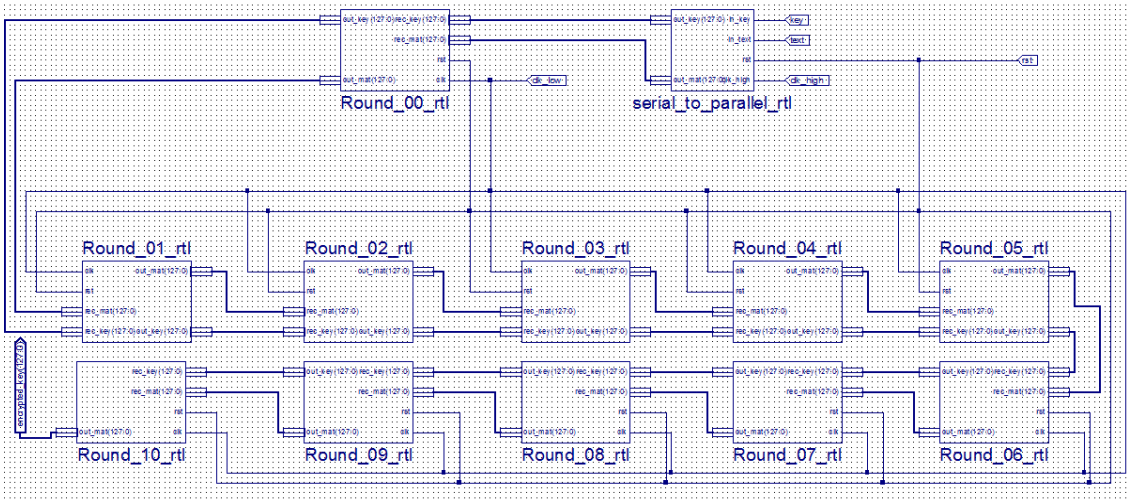

Figure 12. The schematic diagram of the pipelined AES

The ISim simulation of the pipelined implementation is shown in Figure 16. The pipelined processing will reduce the 26 cycles that the encryption process uses in the normal implementation to 12 cycles; however the 12 cycles can be reduced again by using the parallel processing to become just 2 cycles. The schematic diagram of the parallel processing is shown in Figure 13; the parallel methods will speed up the pipelined processing.

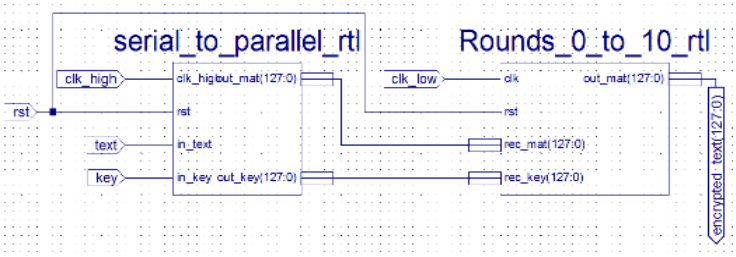

Figure 13. The schematic diagram of the parallel processing of AES

The ISim simulation and the results of the parallel processing are shown in Figure 17. The proposed algorithms need more resources to speed up the encryption process. This is shown in the device utilization summary for the normal, the pipelined and the parallel processes, as shown in Tables 2,3 and 4.

The performance of these processes are shown using the ChipScope (Xilinx ChipScope tool uses the logic analyzer and virtual I/O directly that allows the view of the internal signals) to confirm the design after being downloaded on the used FPGA kit to track the processes behavior of the three algorithms. The ChipScope signals are shown in Figure 18, Figure 19 and Figure 20.

5. SIMULATION AND RESULTS

Figure 14 shows the simulation of the used cycles against the data size. It also shows the number of cycles consumed to encrypt different data sizes. The red line represents the traditional AES algorithm; the black line represents the pipelined algorithm; and the green line represents the parallel algorithm. As the data size increases, the number of consumed cycles will also increase to perform the encryption. Figure 14 will clearly show the high effect of using the pipelined and the parallel algorithms on the numbers of the used cycles especially with big data sizes. As shown in Figure 10, the parallel processing is the fastest method that can be used for encryption. The pipelined method is slower than the parallel one, but it is still faster than the traditional processing. This approves the goal of this paper of increasing the speed of the encryption by using new techniques for the processing.

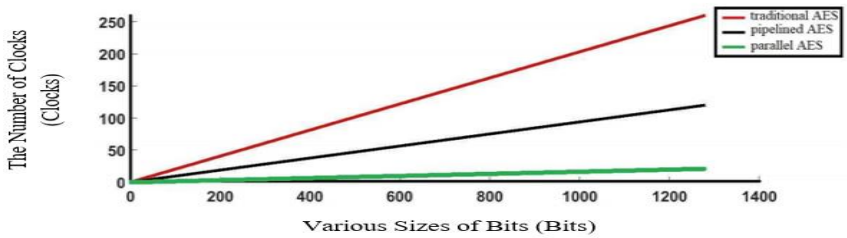

Figure 14. The numbers of cycles for the three methods

Table 1 compares between the result of this paper and the result of the last related work for Bin Liu and Bevan M. Baas that was published under the title “Parallel AES Encryption Engines for Many-Core Processor Arrays” [15]. In this mentioned paper, the authors used a synchronous array of simple processors (multi-core processor arrays) to reach the parallelism. The authors compared their work with other software implementations on programmable processors and did not compare with implementations that contain or are composed of specialized hardware (e.g., ASICs, ASIPs, FPGAs, etc.) that will be included in our paper. Table 1 also shows the operation cycles for the four modes compared to the result of our paper.

Table1. Comparison of the AES throughputs on different platforms

Platform	Throughput (Cycles/Byte)
Pentium 4 561	16
Athlon 64 3500	10.6
Core2 Due E6400	9.19
Core2 Quad Q6600	9.32
Core2 Quad Q9550	7.59
Core i7 920	6.92
AsAp	9.5
This work (pipelined processing)	0.75
This work (parallel processing)	0.125

As shown in Table 1, the result of this paper clearly shows the high effect of using the pipelined and the parallel algorithms on the numbers of the used cycles. The throughput of the pipelined algorithm is 0.75 Cycles/Byte and in the parallel algorithm it is 0.125 Cycles/Byte; on the other hand, the throughput of the AsAP processor (the processor that the authors used in the compared related work) is 9.5 Cycles/Byte, which shows a high effect and performance enhancement especially by using FPGA Figure 15 shows the ISim simulation of the traditional processing and the output will appear after 26 cycles from the input of the first bits. This is a huge number of cycles which will lead to system delay and will present a chance for the hacker to attack, making the system not reliable.

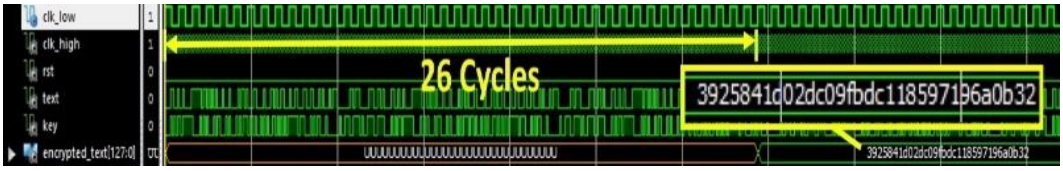

Figure 15. The ISim simulation of the normal AES processing

Figure 16 shows the ISim simulation of the pipelined processing; as shown, the output appears after 12 cycles. This technique will consume more hardware resources from the used kit but it will improve the performance and make the system faster than in the previous implementation. It will also reduce the chances of the hacker to attack the system.

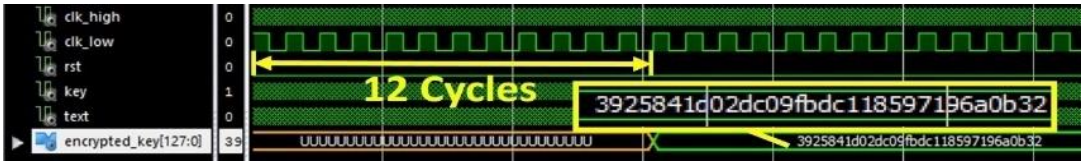

Figure 16. The ISim simulation of the pipelined AES processing

By logic, using parallel processing is the fastest processing method we can use. Figure 17 shows the ISim simulation for the parallel processing. Figure 17 shows also that the parallel method will use only 2 cycles, which will make the system faster than in the two previous methods, which in turn will also make the system more secure and reliable.

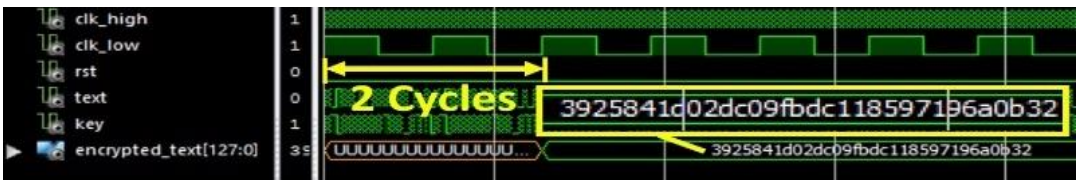

Figure 17. The ISim simulation of the parallel AES processing

To make sure the FPGA design can be used in the real life. The design is downloaded on the kit and the ChipScope tool is used to check the values of the kit output that represent the real implementation. The ChipScope can also be used to check the values that pass between the internal buses of the kit. The real output data is shown in Figure 18, Figure 19 and Figure 20. The red triangular in the three figures refers to the encrypted state of our case study, and the upper red line refers to the start of the encryption processing. The data is presented in the waveform and listing view. As seen in Figure 18, the encryption of our case study will take 26 cycles. This number of cycles is reduced to 12 cycles in the pipelined processing as shown in Figure 19 and it is reduced to 2 cycles in the parallel processing as shown in Figure 20. This output data represents the real data and the real output of the FPGA implementation which approves the goal of this paper and assures the results of our implementation to the reader. The dashed lines refer to the output in zoom out.

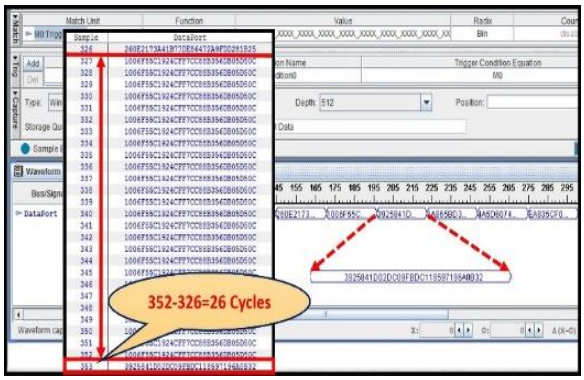

Figure 18. The chipscope analysis of the traditional processing

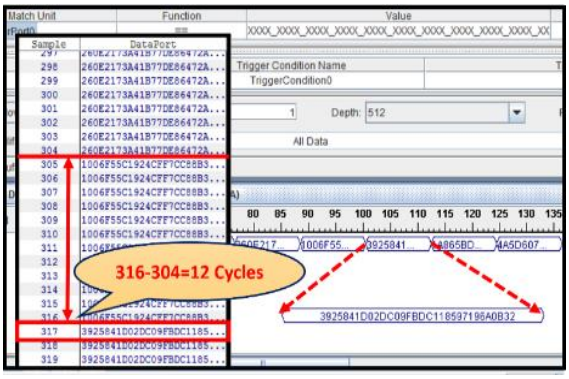

Figure 19. The chipscope analysis of the pipelined processing

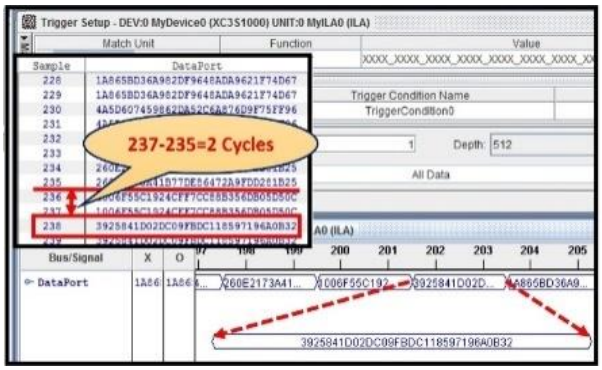

Figure 20. The chipscope analysis of the parallel processing

Tables 2, 3 and 4 show the device utilization summary of the traditional, the pipelined, and the parallel processing. The price of the fast performance comes from the increase of the consumption of logic slices on the used kit; in other words increase of the hardware consumption. As the speed increases, the logic slices consumption also increases.

Table 2. The device utilization summary for the normal processing

Logic utilization	Used number
Number of slice flip flops	5546
Number of 4 input LUTs	2200
Number of occupied slices	45368
Number of slices containing only related logic	22798
Total number of 4 input LUTs	45368
Number of bonded IOBs	133
Number of BUFGMUXs	2
Number of RAMB16BWEs	8

Table 3. The device utilization summary for the pipelined processing

Logic utilization	Used number
Number of slice flip flops	5810
Number of 4 input LUTs	2800
Number of occupied slices	46745
Number of slices containing only related logic	22963
Total number of 4 input LUTs	46745
Number of bonded IOBs	147
Number of BUFGMUXs	2
Number of RAMB16BWEs	8

Table 4. The device utilization summary for the parallel processing

Logic utilization	Used number
Number of slice flip flops	6432
Number of 4 input LUTs	2800
Number of occupied slices	56845
Number of slices containing only related logic	30685
Total number of 4 input LUTs	56845
Number of bonded IOBs	190
Number of BUFGMUXs	2
Number of RAMB16BWEs	8

6. CONCLUSION

The scientists try to update and develop new encryption algorithms to prevent the attacks of the hackers; the development of hacking and the continuous discovery of gaps made the process of the information security more difficult and needed continuous updating. One of these common update methods is the AES. It now became one of the most important encryption techniques.

This paper describes in detail the AES encryption system algorithm and provides time reduction methods based on the pipelined and the parallel processing techniques using the FPGA implementation. The idea presented in this research is the change of the processing methods to make the system use all the resources of the hardware as much as possible to make the system faster than before. The paper succeeded in achieving its goal. The result confirms that the parallel processing is the fastest method that can be used for encryption. The pipelined method is slower than the parallel one, but it is still faster than the traditional processing. This approves the goal of this paper of increasing the speed of the encryption by using new techniques for the processing. As we mentioned, this improvement will be on the account of the hardware resources. The paper also included the implementation of the three algorithms: the traditional, the pipelined, and the parallel techniques, and finally a comparison between the three algorithms and the published related work.

REFERENCES

[1] Jiaming Xu, Ao Fan, Minyi Lu, and Weiwei Shan, "Differential power analysis of 8-bit datapath AES for IoT applications," *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 1470-1473, 2018.

[2] He Fei and, Gao Daheng, "Two kinds of correlation analysis method attack on implementations of Advanced Encryption Standard software running inside STC89C52 microprocessor," *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1265–1269, 2016.

[3] Nidhi Gaur, *et al*," Enhanced AES Architecture Using Extended set ALU at 28nm FPGA," *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 437-440, 2018.

[4] Kaur, Gaurav Raj, and Amanpreet Dheerendra Singh, "Multi RoundSelective Encryption using AES Over Storage Cloud", Global Journalof Computer Science and Technology 13.3, pp.1-8, 2013.

[5] D.Rahul Gandh, V. Kamalakannan, R. Balamurugan,and S. Tamilselvan, "FPGA implementation of enhanced key expansion algorithm for Advanced Encryption Standard," 2014 *International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 409-413, 2014.

[6] Bih-Hwang Lee, Ervin Kusuma Dewi and Muhammad Farid Wajdi, "Data security in cloud computing using AES under HEROKU cloud," *2018 27th Wireless and Optical Communication Conference (WOCC)*, pp.1-5, 2018.

[7] Byung-Yoon Sung, *et al*, "An AES-GCM authenticated encryption crypto-core for IoT security," *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, pp.1-3, 2018.

[8] Pragyanshree Nayak, Sanjeet Kumar Nayak,and Satyabrata Das, "A secure and efficient color image encryption scheme based on two chaotic systems and advanced encryption standard," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 412-418, 2018.

[9] Liting Yu, *et al*, "AES design improvements towards information security considering scan attack," *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering*, pp.322-326, Aug.2018.

[10] Zhang, Xinmiao, and Keshab K. Parhi, "Implementation app.roachesfor the advanced encryption standard algorithm," *IEEE Circuits and Systems Magazine*, vol. 2, no. 4, pp. 24-46, 2018.

[11] Flavius Opritoiu, and Mircea Vladutiu, "Offline self-test architecture for the inversion operation of advanced encryption standard," *2014 IEEE 20th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pp. 263-266, 2014.

[12] Lokireddi Phani Kumar, and A. K. Gupta, "Implementation of speech encryption and decryption using advanced encryption standard," *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 1497-1501, 2016.

[13] Mustafa Emad Hameed, *et al*, "Compression and encryption for ECG biomedical signal in healthcare system," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol 17, no 6, pp. 2826-2833, 2019.
[14] Biagio, A. D., Barenghi, A., Agosta, G. and Pelosi. G, "Design of a parallel AES for graphics hardware using the CUDA framework," *IEEE International Symposium on Parallel & Distributed Processing*, pp. 1-8, 2009.
[15] Bin Liu and Bevan M. Baas, "Parallel AES encryption engines for many-core processor arrays," *IEEE transactions on computers*, vol. 62, no. 3, pp. 536-547, 2013.
[16] A.Anusha and N.Samba Murthy, "Design and analysisof parallel AES encryption and decryption algorithm for multi processor arrays," *IOSR Journal of VLSI and Signal Processing*, vol. 5, no. 1, pp. 1-11, 2015.
[17] Toubal ABDELMOGHNI, *et al*, "Implementation of AES coprocessor for wireless sensor networks," *2018 International Conference on App.lied Smart Systems (ICASS)*, pp.1-5, 2018.
[18] Sattar B. Sadkhan, and Akbal O. Salman, "Fuzzy logic for performance analysis of AES and lightweight AES," *2018 International Conference on Advanced Science and Engineering (ICOASE)*, pp. 318-323, 2018.
[19] Y. Zhang, K. Yang, M. Saligane, D. Blaauw, and D. Sylvester, "Acompact 446 gbps/w aes accelerator for mobile soc and iot in 40nm," *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, pp. 1–2, 2016.
[20] Flevina Jonese D'souza, Dakshata Panchal, "Design and implementation of AES using hybrid app.roach," *2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, pp. 517-521, 2018.
[21] Yue He, *et al*, "Dynamic bandwidth scheduling algorithm for space applications in FC-AE-1553 switching network," *2018 Asia Communications and Photonics Conference (ACP)*, pp.1-3, 2018.
[22] Manh-Hiep Dao, Van-Phuc Hoang, Van-Lan Dao, and Xuan-Tu Tran, "An energy efficient AES encryption core for hardware security implementation in IoT systems," *2018 International Conference on Advanced Technologies for Communications (ATC)*, pp. 301-304, 2018.
[23] Arundhati Joshi, P. K. Dakhole, and Ajay Thatere, "Implementation of S-Box for advanced encryption standard," *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, pp.1-5, 2015.
[24] T. K. Jishamol, and K. Rahimunnisa, "Low power and low area design for advanced encryption standard and fault detection scheme for secret communications," *2013 International Conference on Communication and Signal Processing*, pp. 743- 747, 2013.
[25] Aiguo Bu, Wentao Dai, Minyi Lu, Hao Cai, and Weiwei Shan, "Correlation-based electromagnetic analysis attack using haar wavelet reconstruction with low-pass filtering on an FPGA implementaion of AES," *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE*), pp.197-1900, 2018.
[26] Ayşin Buluş, and Ercan Buluş, "Cipher with AES," *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp.27-30, 2018.
[27] Jiri Petrzela, "Chaotic oscillator based on mathematical model of multiple-valued memory cell," *2018 International Conference on App.lied Electronics (AE)*, pp.1-4, 2018.
[28] Ye Liu, Wei Gong, and Wenqing Fan, "Application of AES and RSA hybrid algorithm in e-mail," *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, pp.701-703, 2018.
[29] Ye Yuan, Yijun Yang,Liji Wu, and Xiangmin Zhang," A high performance encryption system based on AES algorithm with novel hardware implementation," *2018 IEEE International Conference on Electron Devices and Solid State Circuits (EDSSC)*, pp. 1-2, 2018.
[30] P. B. Mane, A. O. Mulani, "High speed area efficient FPGA implementation of AES algorithm," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol 7, no 3, pp. 157-165, 2018.
[31] G. Renuka, V. Usha Shree, P. Chandra Sekhar Reddy, "Comparison of AES and DES algorithms implemented on Virtex-6 FPGA and microblaze soft core processor," *International Journal of Electrical and Computer Engineering (IJECE)*, vol 8, no 5, pp. 3544-3549, 2018.
[32] Jean J. Moadi A, Peyrin T, *et al*, "Bit-Siding: A Generic Technique for Bit-Serial Implementations of SPN-based Primitives," *International Conference on Cryptographic Hardware and Embedded Systems*, pp. 687-707, 2017.
[33] Qi Zhang, and Qun Ding, "Digital image encryption based on advanced encryption standard (AES)," *Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pp. 1218-1221, 2015.
[34] Rupesh Bhandari, Kirubanand V B, "Enhanced encryption technique for secure iot data transmission," *International Journal of Electrical and Computer Engineering (IJECE)*, vol 9, no 5, pp. 3732-3738, 2019.
[35] Mustafa Emad Hameed, Masrullizam Mat Ibrahim, Nurulfajar Abd Manap, Mothana L. Attiah, "Comparative study of several operation modes of AES algorithm for encryption ECG biomedical signal," *International Journal of Electrical and Computer Engineering (IJECE)*, vol 9, no 6, pp. 4850-4859, 2019.
[36] Mohamed Nabil, Adang Suwandi Ahmad, Sarwono Sutikno, Yusuf Kurniawan, Arwin Datumaya Wahyudi Sumari, "Design and implementation of pipelined AES encryption system using FPGA," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 5, pp. 2558-2564, 2020.
[37] Srividya R., Ramesh B., "Implementation of AES using biometric," *International Journal of Electrical and Computer Engineering (IJECE)*, vol 9, no 5, pp. 4266-4276, 2019.
[38] Septafiansyah Dwi Putra, *et al*, "Revealing AES encryption device key on 328P microcontrollers with differential power analysis," *International Journal of Electrical and Computer Engineering (IJECE)*, vol 8, no 6, pp. 5144-5152, 2018.
[39] Heidilyn V. Gamido, Marlon V. Gamido, Ariel M. Sison, "Developing a secured image file management system using modified AES," *Bulletin of Electrical Engineering and Informatics*, vol 8, no 4, pp. 1461-1467, 2019.
[40] Christy Atika Sari, Giovani Ardiansyah, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawantom, "An improved security and message capacity using AES and Huffman coding on image steganography," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol 17, no 5, pp. 2400-2409, 2019.

[41] Septafiansyah Dwi Putra, Mario Yudhiprowira, Sarwono Sutikno, Yusuf Kurniawan, Adang Suwandi Ahmad, "Power analysis attack against encryption devices: a comprehensive analysis of AES, DES, and BC3," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol 17, no 3, pp. 1282-1289, 2019.
[42] Iqbal Ahmed, Fahim Irfan Alam, "Integrity verification for an optimized cloud architecture," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol 18, no 1, pp. 166-173, 2020.

BIOGRAPHIES OF AUTHORS

Mohamed Nabil received the B.E. degree in electronics and communications in 2013. He received M.S. degree in electronics and communications in 2018 from Arab Academy for Science, Technology and Maritime Transport. Currently, He is working towards Ph.D. degree at Minia University at Electronics and Communications Department.
E-mail: mohammednabel5050@gmail.com.

Ashraf A. M. Khalaf (Ph.D.) received his B.S. and M.S. degrees in electrical engineering from Minia University, Egypt, in 1989 and 1994, respectively. He received his Ph.D. in electrical engineering from Graduate School of Natural Science and Technology, Kanazawa University, Japan, in March 2000. He is currently an associate professor at Electronics and Communications Engineering Department, Faculty of Engineering, Minia University, Egypt. E-mail: ashkhalaf@yahoo.com

Sara M. Hassan received the B.S. degree in electrical engineering from Modern Academy for Engineering and Technology, Cairo, Egypt, in 2007, the M. S. degree from Ain Shams University, Cairo, Egypt, in 2013, and the Ph.D. degree from Ain Shams University, Cairo, Egypt, in 2017. She is currently a staff member at Modern Academy for Engineering and Technology, Cairo, Egypt. Her fields of interest include electrical, electronics, design and implementation of communication systems. E-mail: Sara.Hassan@eng.modern-academy.edu.eg.