Indonesian Journal of Electrical Engineering and Computer Science
Vol. 22, No. 1, April 2021, pp. 71~82
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v22.i1.pp71-82

Real-time FPGA implementation of concatenated AES and IDEA cryptography system

Sara M. Hassan, Gihan. G. Hamza
[1]Electronics and Communications Engineering Department, Modern Academy for Engineering and Technology, Cairo, Egypt.
[2]National Institute of Standards, Time and Frequency and Microwaves Laboratory, Giza, Egypt

Article Info

Article history:

Received Oct 23, 2020
Revised Dec 14, 2020
Accepted Feb 15, 2021

Keywords:

AES
Decryption
Encryption
IDEA
Rijndael

ABSTRACT

The data encryption is one of the most critical issues in the communication system design. Nowadays, many encryption algorithms are being updated to keep pace with the remarkable progress in the communication field. The advanced encryption standard (AES) is a common algorithm that has proved its efficacy. The main drawback of AES is that it uses too simple algebraic structures, since every block is always encrypted in the same way that makes the hacking process possible if the hacker captures the key and the uses S-Box in the input stage. This especially applies to the unwired communication systems where chances of hacking exceed those found in the wired systems. The paper proposes a security enhancement method that is based on utilizing concatenated AES and international data encryption algorithm (IDEA) algorithms. Upon applying the proposed algorithm, the hacking process becomes a great challenge. The paper incorporates the real-time FPGA implementation of the proposed algorithm in the encryption and the decryption stages. Besides, the paper presents a clear analysis of the system's performance.

Corresponding Author:

Sara M. Hassan
Department of Electronics and Communications Engineering
Modern Academy for Engineering and Technology
Cairo, Egypt
Email: Sara.Hassan@eng.modern-academy.edu.eg

1. INTRODUCTION

The U.S. government selected the advanced encryption standard (AES) for protecting confidential information and encrypting sensitive data [1]. In 1997, the national institute of standards and technology (NIST) began developing AES, announcing that an efficient algorithm for the data encryption standard (DES)-the first algorithm used by NIST-was required. The development of the DES encompasses three different length algorithms: the double DES, the triple DES with two keys and the triple DES with three keys [2]. Hackers have managed to decrypt the DES. Among the fierce hacking attacks is the one that depends on attempting as many keys as possible to illegally access a message. The AES has offered a replacement of the data encryption standard (DES), as the AES was developed for avoiding the drawbacks of the DES [3-5].

Three algorithms of the AES were determined by NIST. The block size of each algorithm is 128 bits, and the key length is one of these sizes: 128, 192 or 256 bits. The key length is a symmetric-key technique which utilizes the same key for encrypting and decrypting [6-8]. One of the main problems that faces the AES encryption system is the long time consumed to perform the encryption and the decryption processes. This is because of the big numbers of rounds used, making the system vulnerable [9-11]. As the time consumed by the

encryption system for encrypting the data increases, the chances of the hackers to break into the system increase. The following paragraphs indicate some relevant researches published in the recent years.

A. Anusha *et al.* [12], the authors propose a multi-processor array for designing a parallel AES encryption system and achieving a high throughput performance. Mohamed Nabil *et al.* [13], the authors present a new technique for increasing the performance speed of the advanced encryption standard based on pipelined processing. Pipelined processing decreased the consumed time, making it less than the time consumed during conducting the normal processing. Mohamed Nabil *et al.* [14], an enhanced processing algorithm based on the parallel processing is presented. The article offers a detailed description of the proposed algorithm methodology. The authors succeeded in speeding up the performance of the advanced encryption standard by using parallel processing. Parallel processing makes the system faster than pipelined processing. However, this enhancement consumes more resources and power. Ritambhara [15], the authors present an enhanced AES algorithm that applies cascading method on a key whose size is 400 bits. For promoting the safety of internet of things' next generation, the article proposes an algorithm that comprises diffusion of AES algorithm. The presented algorithm applies a cascading technique which uses 200 bit plain text and 400 bit key. The algorithm is divided into two equal parts in order to provide different keys at different cascading levels. The technique utilizes only five rounds in place of 10 in the new algorithm in cascaded format. It thereby eliminates mix column twice from the entire AES. This, in turn, decreases time and complexity.

Zuhar Musliyana *et al.* [16], the authors propose an enhancement of the AES based on generating AES key randomly, based on the value of time as a user logs in with a particularly active period. In this technique, the security authentication becomes stronger because of changes in key generating ciphertext changes for each encryption process. This is based on time as a valuable benchmark. The authors introduce enhancement of the system by changing the key with time. This paper proposes an enhancement by using two concatenated various systems with three different keys. This can be considered another face of the previously mentioned paper.

Another problem faces AES: due to the simplicity of its rounds, the hacking process is possible if the hacker captures the key and the S-Box in the input stage. This especially occurs in unwired communication systems, where chances of hacking exceed those of wired systems. The themes of security systems as well as encryption techniques are daily scalable. Thus, security researchers conduct plenty of experiments for finding new techniques to enhance security systems [17-19]. Security researchers target determining hackers by developing security systems and techniques. This paper is an attempt to update one of the strongest encryption systems employed in information security. The paper presents a new technique that could be applied for enhancing the security of advanced encryption systems. The research proposes a new technique that is based on a design of concatenated AES and IDEA cryptography systems. Hacking processes become a big challenge upon applying the new algorithm [20-22].

The block size of IDEA is 64-bit, and the key size is 128-bit. IDEA utilizes the rounds techniques, applying a methodology named half-rounds in which every round uses 6 sub-keys of 16-bit. Every half-round uses 4 sub-keys [23, 24]. The encryption key is used to extract the first 8 sub-keys, whereas the other 8 keys are created on basis of the rotation [25]. The question that arises now is why do we use IDEA? IDEA has been successful in countering a lot of cipher attack methods. IDEA has proven immune in certain conditions, and it has not revealed algebraic or linear weaknesses [26-28]. In 2005, advanced packet exchange technology (APEX) proclaimed that IDEA algorithm outperformed many encryption algorithms which are being used nowadays by several governments and European countries [29, 30].

As previously mentioned, the traditional processing steps of the AES cryptography system have become one of the most famous reasons that make the system vulnerable to hacking in case the hacker can access the system. This is possible if it is assumed that someone can access a certain communication system and capture the key. A hacker would then be able to decrypt any data because the processing steps are known.

This paper can be considered a method for changing the normality of the AES system and increasing the security level of the system. This is done by using two different concatenated cryptography systems with different keys, increasing the possible values and making the prediction process impossible. The study is divided into seven sections. Section 1 includes an introduction. Section 2 is a description of the AES. Section 3 offers a depiction of the IDEA. Section 4 demonstrates the methodology applied for achieving the study's objective. Section 5 manifests the implementation of the hardware. Section 6 presents the simulation and the results. Finally, section 7 includes the conclusion of the paper.

2. THE AES ALGORITHM

2.1. AES Encryption

The AES encryption algorithm offers an encryption and decryption of data in blocks of 128 bits. Figure 1 is a demonstration of an AES algorithm simplified overview. As shown in Figure 1, the plain text is

the sensitive data which is required to encrypt [31]. The algorithm key is a 128-bit secret variable created by an algorithm that will be used for encrypting the input text. A key expansion process is performed at each round; in this process, a series of new "round keys" are derived by using the Rijndeal key schedule algorithm. The encrypted output is called cipher text after passing through all the encryption rounds shown in Figure 1. The encryption processes are demonstrated on the left-hand side and the decryption processes are shown on the right-hand side [32, 33].

A series of mathematical transformations that use the plaintext and the secret key as a starting point are performed by the actual AES cipher [34]. The mathematic transformations will be performed according to Figure 1 and will be discussed below. The first process is called AddRoundKey. Every round key combines with the plaintext utilizing the additive XOR algorithm demonstrated in Figure 2. As shown in Figure 2 the input plain text bits represented by elements of the matrix (A) is XORed with the 128 bits key to get the matrix (Z) [35].

The second process is the SubByte process in which a substitution table is used to perform the substitution of the resulting data [36]. Each element will be replaced by another depending on the value of this element. The number of rows in the S-Box is represented by first 4 bits of the element; the number of columns is represented by the second 4 bits [37]. The element in the s-box that is equivalent to this row and this column replaces the old element in the text as shown in Figure 3. As shown in the figure each element in the matrix (a) is divided into two 4-bits blocks that refer to the number of row and the number of column in the S-Box. The new element takes place the old element to get the matrix (b) [38].

The third process is called ShiftRows. In this process, each byte in the 4x4 column of 16 bytes–making up a 128-bit block–is shifted to the right as presented in Figure 4. As shown in the figure, there is no shifting in the first row, there is one left element shifting in the second row, there are two left elements shifting in the third row and there are three left elements shifting in the final row [39].

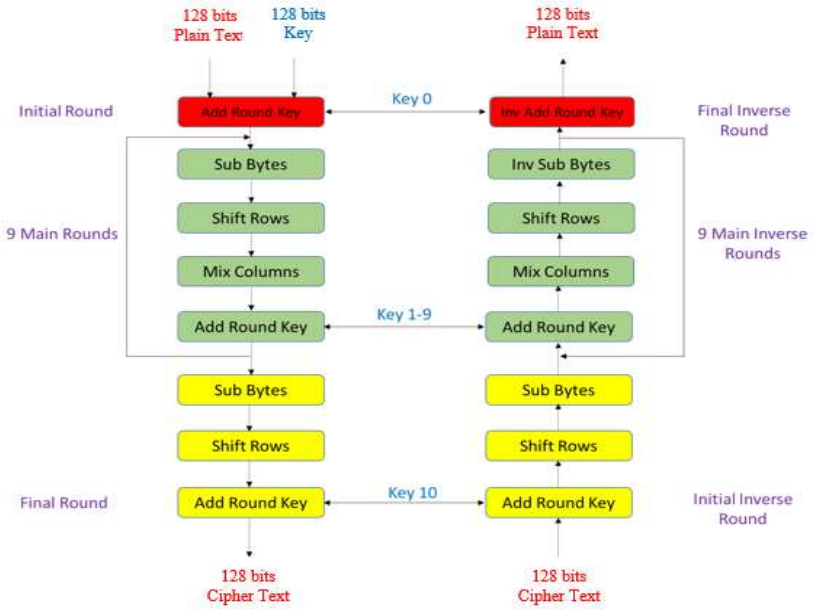

Figure 1. The AES algorithm

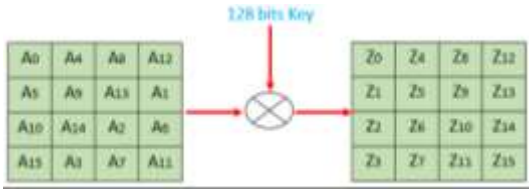

Figure 2. The AddRoundKey process

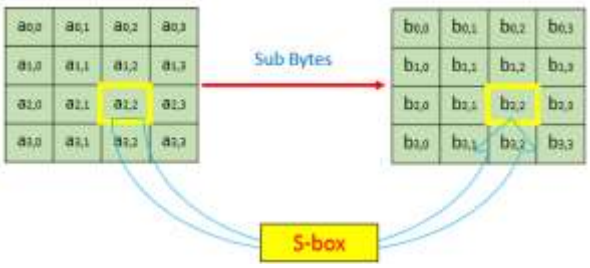

Figure 3. The SubByte process

The fourth process is the MixColumns process. The modulo multiplication is performed using the C(X) 4x1matrix as shown in Figure 5. As shown in the figure, the matrix (a) is modulo multiplicated by c(x) to get the matrix (b) [40]. These processes are repeated for nine main times, besides the initial round that consists of the AddRoundKey process as well as the final round that does not include the MixColumns process. As shown in Figure 1, every round is re-encrypted via one of the round keys that are generated during the process key expansion. Every one of the added rounds reduces chances of shortcut attacks [41].

Figure 4. The ShiftRows process

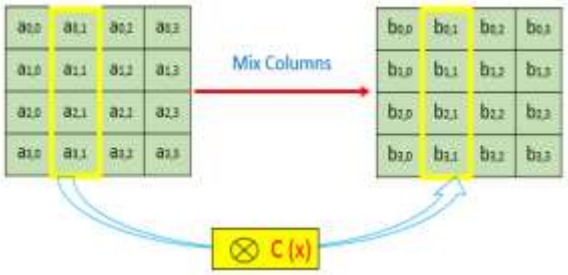

Figure 5. The MixColumns process

2.2. AES Decryption

As shown in Figure 1, the decryption of AES is simple. The previously mentioned steps are reversed. The first step is inverting the round key. Certainly, the original secret key should be known for reversing the process utilizing every inverse round key [42]. The InvAddRoundKey is performed on the encrypted text and the key to obtain this pre-encrypted text. This is a simple step, where the XOR operation is executed on both the encrypted text as well as the round key as illustrated in Figure 6 [43].

The InvMixColumns process, in which the modulo multiplication is performed using the inverse C(X) 4x1matrix, is shown in Figure 7 [44]. The InvShiftRows, where each byte in the 4x4 column of 16 bytes constitutes a 128-bit block, is shifted to the left to obtain the original text as shown in Figure 8. As shown in the figure, there is no shifting in the first row, there is one right element shifting in the second row, there are two right elements shifting in the third row and there are three right elements shifting in the final row [45]. The InvSubByte process, in which the inverse substitution of the resultant data using an inverse S-Box is executed, is demonstrated in Figure 9 [46].

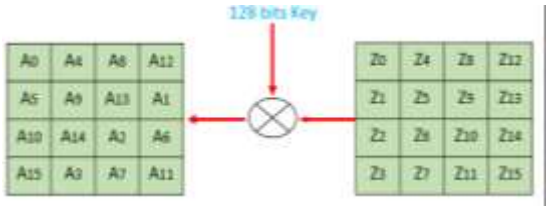

Figure 6. The InvAddRoundKey proces

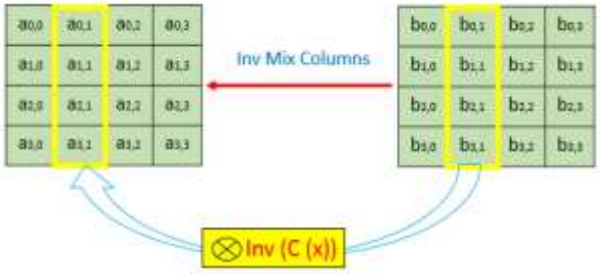

Figure 7. The InvMixColumns process

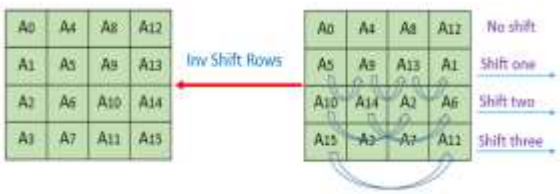

Figure 8. The InvShiftRows process

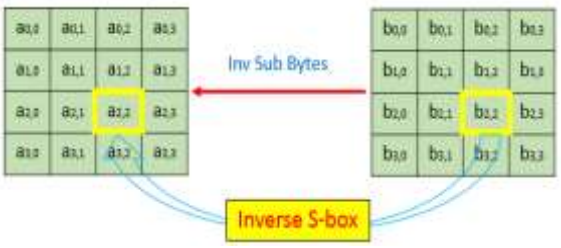

Figure 9. The InvSubByte process

3. THE IDEA ALGORITHM

IDEA works on 64-bit blocks utilizing a 128-bit key, and it is made up of a chain of eight similar transformations as well as an output transformation (the half-round) [47]. Both encryption and decryption processes are reversely identical. Much of the security of IDEA is derived via interleaving operations from diverse sets (modular addition, modular multiplication and bitwise exclusive OR (XOR), and these are algebraically "incompatible" in some sense). The following is a detailed description of the operators that work on 16-bit quantities [48].

Figure 10 shows the encryption procedure of the IDEA. Bitwise XOR (exclusive OR) operation is represented by a circled plus (⊕). A boxed plus (⊞) represents addition modulo 216. Multiplication modulo 216+1 is represented by a circled dot (⊙). A final "half-round" follows the eight rounds. The output transformation is demonstrated below (the swap at the end of the last round is canceled by the swap of the middle two values; thereby, no net swap is found as shown in Figure 11) [49].

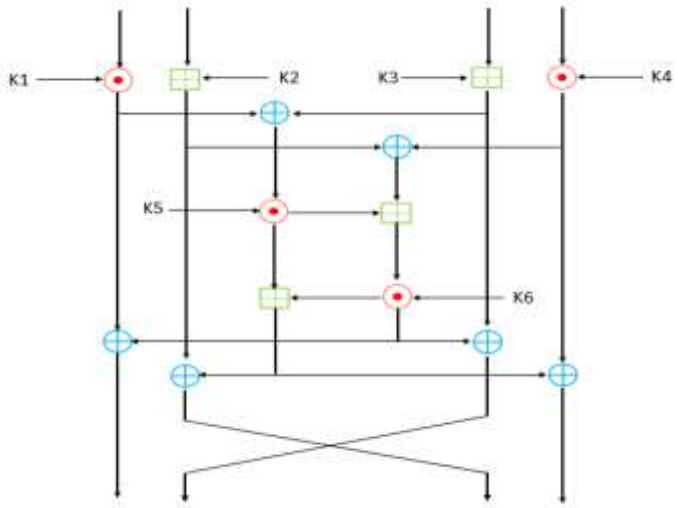

Figure 10. The first eight rounds of the IDEA encryption algorithm

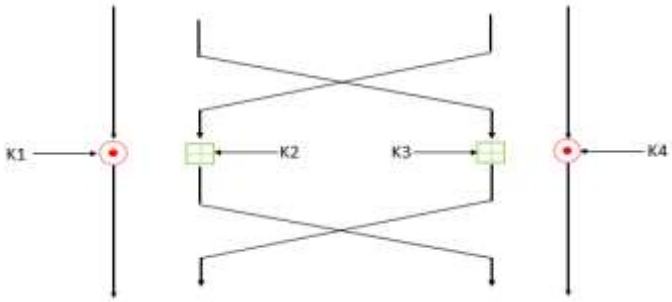

Figure 11. The final round of the IDEA encryption algorithm

3.2. IDEA Decryption

Decryption and encryption operate similarly. However, the round keys order is inverted, and the odd rounds' sub keys are inverted. For example, the inverse of K49-K52 replaces the values of sub keys K1-K4 for the respective group operation. For decryption, K47 and K48 should replace K5 and K6 of each group [49].

4. THE PROPOSED CONCATENATED AES AND IDEA SYSTEM

As illustrated in Figure 12, the proposed system is based on utilizing a concatenated system which is made up of two stages of encryption and decryption. The encryption's first stage is the AES algorithm. The AES encrypts the 128-input plain text by using certain cipher key. Then the encrypted 128 bits are divided into

two 64-bit groups. The first 64 bits are transmitted to the first IDEA encryption block, and the second 64 bits are transmitted to the second IDEA encryption block. The encryption procedure is performed on the two 64 bits using two different encryption keys as shown in Figure 12. The output data of the two IDEA is stored into a 128 buffer, and then the data is transmitted via the channel. Of course, the data is affected, since it is subjected to errors because of the channel noise. This issue can be solved by using an error detection and error correction algorithm (coding system). There are many coding algorithms that can used to perform error detection and error recovery, but this issue is not in the scope of this article. Therefore, it will not be discussed in this paper.

As shown in Figure 12, there are three 128 bits keys. The AES key, the first IDEA key and the second 128 bits key render the hacking process a big challenge for hackers. Figure 13 shows the decryption system. It consists of two concatenated algorithms in different orders. The first decryption stage represents the two IDEA decryption algorithms. The data that is received form the channel is stored into a buffer that specifies the data for each IDEA system. The first 64 bits are transmitted to the first IDEA decryption block, and the second 64 bits are transmitted to the second IDEA decryption block. The decryption procedures are performed on the two 64 bits, and the output of each block is transmitted to the AES decryption block, which performs the decryption process on the 128 bits to obtain the original 128 plain text.

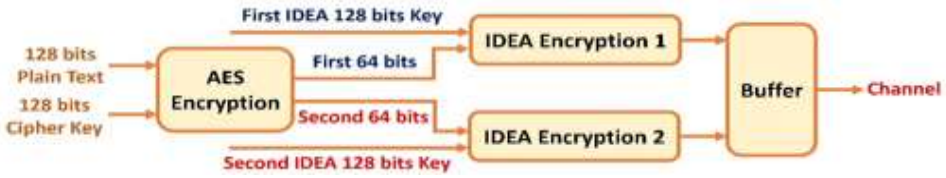

Figure 12. The proposed encryption system's block diagram

Figure 13. The proposed decryption system's block diagram

5. HARDWARE IMPLEMENTATION

In this paper, the implementation and realization of the proposed system are presented via the FPGA. The FPGA utilized is the Xilinx "Spartan-3A/3AN FPGA Starter Kit". Figure 14 shows the implementation kit that is used. The proposed system is designed through writing VHDL codes utilizing the Xilinx package ISE 14.7 program, and the codes are simulated via the ISim simulator program. The testing strategy is based on connecting the FPGA kit to the laptop by using a JTAG cable. The data is transmitted to the kit using a JTAG cable, the encryption process is performed, and the result is transmitted to the laptop to check these results on the ChipScope package (Xilinx software that makes the users can check the data passing through the FPGA paths). The encrypted data is transmitted again to the kit, the decryption process is performed, and the output is transmitted to the laptop to check the result that should be equal to the original data.

Figure 15 illustrates the ISE 14.7 of the proposed encryption system's schematic diagram. The presented encryption system consists of AES encryption system and two parallel branches of IDEA encryption system. The system's input consists of the plain text (the text that needs to be encrypted), the cipher key for the AES encryption, the first IDEA key for the first IDEA encryption system and the second IDEA key for the second IDEA encryption system. The clock is used to perform the synchronization needed to make the components of the circuit operate correctly together in a certain protocol. The clock cycle used in this project is 50 Mhz, and it is a built-in clock chip in the used FPGA Kit. Finally, the rst input is required to reset all the parameters of the system to their default values if needed. The output of this schematic diagram is the encrypted text that is transmitted via the channel.

Figure 16 illustrates the ISE 14.7 of the proposed decryption system's schematic diagram. The decryption system comprises AES decryption system and two parallel branches of IDEA decryption system. The system's input includes the encrypted text (the text that needs to be decrypted to obtain the original text), and the clock that will be used to perform the synchronization needed to make the components of the circuit

operate correctly in a certain protocol. Besides, the clock cycle used in this project is 50 Mhz, and it is the same clock cycle used in the encryption (the transmitter) to obtain a synchronization environment between the transmitter and the receiver. Finally, the rst input is required to reset all the parameters of the system to their default values if needed. The proposed encryption system's ISim simulation is illustrated in Figure 17. The output of this implementation is the original text. The proposed decryption system's ISim simulation is illustrated in Figure 18.

Figure 14. The Xilinx "Spartan-3A/3AN FPGA Starter Kit"

The enhanced performance is illustrated utilizing the ChipScope (Xilinx ChipScope uses the logic analyzer and virtual input/output directly, and it allows viewing the internal signals). This is done to confirm the design after it is downloaded on the utilized FPGA kit for proving the proposed algorithm's behavior. Figure 18 and Figure 19 illustrate the ChipScope outputs.

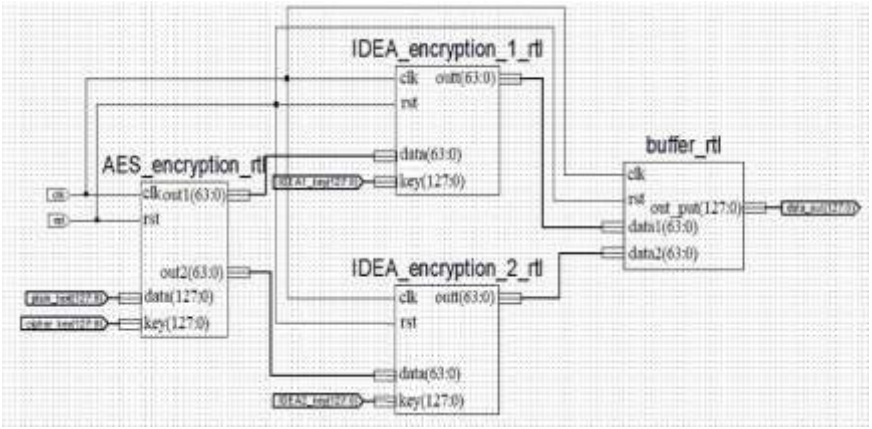

Figure 15. The ISE 14.7 of the proposed encryption system's schematic diagram

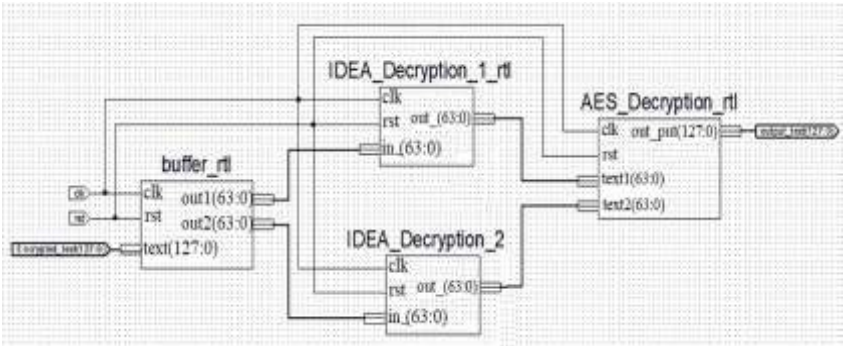

Figure 16. The ISE 14.7 of the proposed decryption system's schematic diagram

6. SIMULATION AND RESULTS

Figure 17 illustrates the encryption system's ISim simulation. The system's input comprises the clk, rst, plain_text, cipher_key, IDEA1_key and IDEA2_key. Data_out, which is surrounded by a yellow rectangle, is the output of the encryption system; and this output represents the encrypted text. Figure 18 illustrates the proposed decryption system's ISim simulation. The input of the system is the encrypted_text. The output_text, which is surrounded by a yellow rectangle, is the output of the system; and it represents the original text before the encryption process. As expected, the decrypted output is the same system input (plain text).

To prove that, the FPGA design could be utilized in real life. This is achieved through downloading the design on the kit. The ChipScope tool is used for checking the values of the kit output, which represent the real data. The values which pass between the internal buses of the kit are checked via ChipScope. Figure 19 shows the encryption system's real output data. Figure 20 illustrates the decryption system's real output data. These output values have the same simulation output, which proves the validity of the paper's proposal, and confirms the result of the study's implementation.

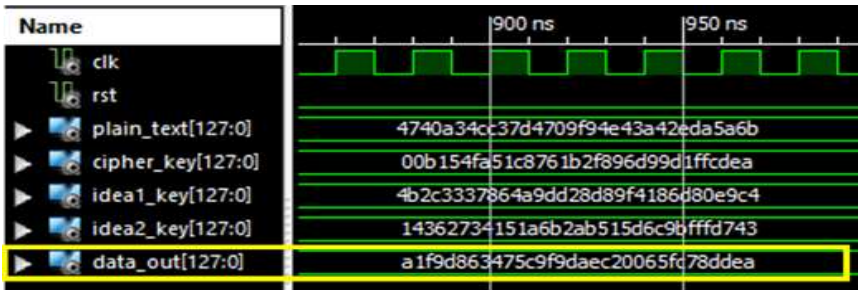

Figure 17. The proposed encryption system's ISim simulation

Figure 18. The proposed decryption system's ISim simulation

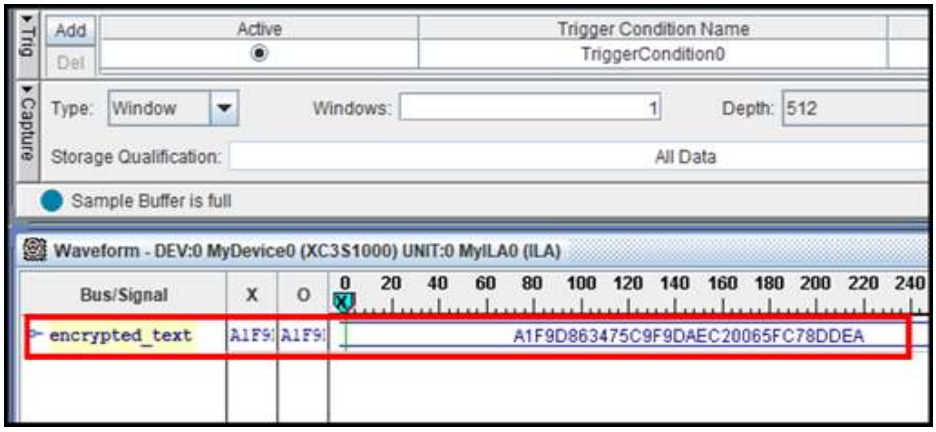

Figure 19. The encryption's ChipScope analysis

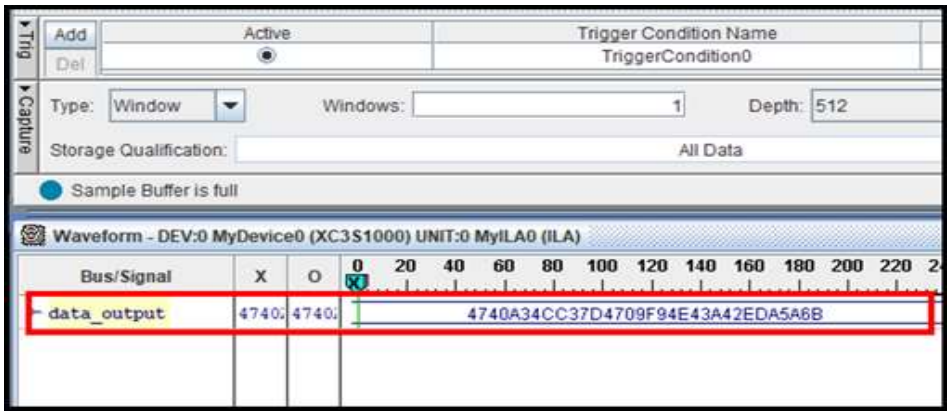

Figure 20. The decryption's ChipScope analysis

Table 1 illustrates the proposed system's device utilization summary. By comparing the hardware resources of this paper shown in Table 1 with the utilized hardware table in [14], it is evident that the proposed concatenated algorithm consumes more hardware resources than the normal processing. This result is predicated because the paper uses two different concatenated algorithms. However, this hardware consumption can be accepted in order to improve the security of the system.

The last related work set in comparison to this paper was published in the ICCCA [15]. In the author's proposal, the efficiency of AES algorithm has been improved via utilizing sequential 200-bit plain text as input and 400 bits as key in cascaded format. This is due to AES-AES cascading and the division of 400-bit key into two 200-bit key, which makes the system less susceptible to severe attacks. This improved the performance of the AES but made the system unreliable due to the very high resource consumption and the high complexity of the system. In this paper, the second stage is replaced by IDEA system, which is simpler than the AES and consumes less resources. Thus, the paper reaches the optimum solution to make the complexity and the resources consumption as low as possible, rendering the system more reliable. Zuhar Musliyana *et al.* [16], the authors introduce normal AES processing with changing the key by the time. The dynamic key generation leads to certain delays that can be a drawback in the proposed improvement. The enhancement in this paper occurs by using fixed keys but different processing steps and certain bits arrangements that are known to both the transmitter and the receiver.

Table 1. The proposed system's device utilization summary

Logic utilization	Used number
Slice Flip Flops Number	8653
4 input LUTs Number	3524
Occupied Slices Number	65896
Slices containing only related logic Number	28965
4 input LUTs Total Number	65896
Bonded IOBs Number	768
BUFGMUXs Number	2
RAMB16BWEs Number	8

7. CONCLUSION

The development of hacking and the persistent discoveries of gaps in systems have made the process of information security more challenging, and it is in continuous need of updating. Two of the common algorithms for securing data are the AES and the IDEA algorithms. This paper offers a detailed description of the AES cryptography algorithm and introduces a new technique which can be utilized for increasing the performance of the advanced encryption standard security. The basis of the new technique is designing a concatenated AES and IDEA encryption system. As mentioned before, in many cases the IDEA algorithm outperformed numerous encryption algorithms that are currently being used by several governments and European countries. The proposed algorithm poses a big challenge for hackers. This is due to the high complexity of the encryption and decryption processes, since three different keys are used for the encryption process. In addition, the targeted information, such as the bits format and the arrangement of these bits in the encrypted text (known for the receiver), is hard to obtain by hackers. This paper has succeeded in achieving its objective as the results shown in the simulation and the results section have proven. The paper demonstrates

the real-time FPGA implementation of the presented algorithm in the encryption and the decryption stages. The study also offers a clear analysis of the performance of the proposed system. The proposed concatenated technique in this paper makes the system more secure than the normal one (AES cryptography), the technique is also more reliable for the application systems that need the highest security levels to counter hacking attacks (such as military systems). On the other hand, the technique consumes more resources and time. Therefore the future works target improving the performance of the proposed system by using parallel or pipelined processing techniques to decrease the time consumed. Another objective would be attempting to consume fewer hardware resources. Subsequently, the system complexity will decrease.

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, doi: 10.1109/TrustCom/BigDataSE.2018.00205.

[2] Nidhi Gaur, Anu Mehra, and Pradeep Kumar, "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, doi: 10.1109/SPIN.2018.8474090.

[3] Madhumita Panda, *et al.*, "Performance Evaluation of Symmetric Encryption Algorithms for Information Security," *International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)*, vol. 4 no. 11, pp. 37-41, 2017.

[4] Flevina Jonese D'souza, and Dakshata Panchal, "Design and Implementation of AES using Hybrid Approach," *2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, pp. 517-521, 2018, doi: 10.1109/PEEIC.2018.8665586.

[5] Subhi R. M. Zeebaree, *et al.*, "DES encryption and decryption algorithm implementation based on FPGA," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 774-781, 2020, doi: 10.11591/ijeecs.v18.i2.pp774-781.

[6] 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, doi: 10.1109/ATC.2018.8587500.

[7] Adnan Ibrahem Salih, Ashwaq Alabaichi, and Ammar Yaseen Tuama, "Enhancing advance encryption standard security based on dual dynamic XOR table and mixcolumns transformation," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 19, no. 3, pp. 1574-1581, 2020, doi: 10.11591/ijeecs.v19.i3.pp 1574 -1581.

[8] Archana Mishra and Saurabh Sharma, "Design and Implementation of High Speed AES Algorithm for Data Security," *International journal of engineering sciences & research technology*, vol. 5, no. 8, pp. 325-337, 2016.

[9] Toubal Abdelmoghni, Ould Zmirli Mohamed, Bengherbia Billel, Maazouz Mohamed, and Lachenani Sidahmed, "Implementation Of Aes Coprocessor For Wireless Sensor Networks," *2018 International Conference on Applied Smart Systems (ICASS)*, pp. 1-5, 2018, doi: 10.1109/ICASS.2018.8652006.

[10] Shraddha More and Rajesh Bansode, "Implementation of AES with Time Complexity Measurement for Various Input," *Global Journal of Computer Science and Technology: ENetwork, Web & Security*, vol. 15, no. 4, 2015.

[11] Heidilyn V. Gamido, *et al.*, "Implementation of a bit permutation-based advanced encryption standard for securing text and image files," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 19, no. 3, pp. 1596-1601, 2020, doi: 10.11591/ijeecs.v19.i3.pp1596-1601.

[12] A.Anusha and N.Samba Murthy, "Design andAnalysis of Parallel AES Encryption and Decryption Algorithm for Multi Processor Arrays," *IOSR Journal of VLSI and Signal Processing*, vol 5, no. 1, 2015.

[13] Mohamed Nabil, Ashraf A. M. Khalaf, and Sara M. Hassan, "Design and Implementation of Pipelined AES Encryption System using FPGA," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 5, pp. 2565-2571, 2020.

[14] Mohamed Nabil, Ashraf A. M. Khalaf, and Sara M. Hassan, "Design and Implementation of pipelined and parallel AES Encryption Systems Using FPGA," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 20, no. 1, pp. 287-299, 2020, doi: 10.11591/ijeecs.v20.i1.pp287-299.

[15] Ritambhara, Alka Gupta, and Manjit Jaiswal, "An enhanced AES algorithm using cascading method on 400 bits key size used in enhancing the safety of next generation internet of things (IOT)," *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 422-427, 2017, doi: : 10.1109/CCAA.2017.8229877.

[16] Zuhar Musliyana, Teuku Yuliar Arif and Rizal Munadi, "Security Enhancement of Advanced Encryption Standard (AES) using Time-Based Dynamic Key Generation," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 18, pp. 8347-8350, 2015.

[17] Samar Zaineldeen1 and Abdelrahim Ate, "Improve the security of transfer data file on the cloud by executing hybrid encryption algorithms," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 20, no. 1, pp. 521-527, 2020, doi: 10.11591/ijeecs.v20.i1.pp521-527.

[18] Felicisimo V. Wenceslao, Jr., *et al.*, "Enhancing the Performance of the Advanced Encryption Standard (AES) Algorithm Using Multiple Substitution Boxes," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 10, no. 3, pp. 496-501, 2018.

[19] Mustafa Emad Hameed, Masrullizam Mat Ibrahim, and Nurulfajar Abd Manap, "Review on improvement of advanced encryption standard (AES) algorithm based on time execution, differential cryptanalysis and level of security," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 1, pp. 139-145, 2018.
[20] Amrutha T. V., and N. R. Prashanth, "Enhanced Key Expansion Algorithm for Advanced Encryption Standard using Different S-Box Implementation on FPGA," *Global Research and Development Journal for Engineering*, vol. 1, no. 5, pp. 112-117, 2016.
[21] Prashant Kumar Dey and Tarun Kumar Dey, "Analysis of The Security of AES, DES, 3DES and IDEA NXT Algorithm," *International journal of engineering sciences & research* technology, vol. 4, no. 10, pp. 177-181, 2015.
[22] Zoran Hercigonja, and Druga gimnaija, "Comparative Analysis of Cryptographic Algorithms," *International Journal of Digital Technology & Economy*, vol. 1, no. 2, pp. 127-134, 2016.
[23] Harish Kumar, Vikas Jadhav, Shefali Sonavane, and R.K. Sharma "Cryptanalytic Attacks on International Data Encryption Algorithm Block Cipher," *Defence Science Journal*, vol. 66, no 6, pp. 582-589, 2016, doi: 10.14429/dsj.66.10798
[24] Byung-Yoon Sung, Ki-Bbeum Kim, and Kyung-Wook Shin, "An AES-GCM authenticated encryption crypto-core for IoT security," *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1-3, 2018, doi: 10.23919/ELINFOCOM.2018.8330586.
[25] 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, doi: 10.1109/ICOASE.2018.8548832
[26] 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, doi: 10.1109/ICACCI.2018.8554728.
[27] Ahmed Nashaat Shakir, *et al.*, "Study and Design of an Encryption Algorithm for Data Transmitted Over the Network by the IDEA and RSA," *International Journal of Computer Applications*, vol. 176, no. 4, pp. 35-41, 2017.
[28] Shaik Rasheeda, Krishna Dharavathu and M. S. Anuradha, "FPGA Implementation of Simplified IDEA and IDEA algorithm," *International journal of basic and applied research*, vol. 8, no. 11, pp. 271-282, 2018.
[29] Shailaja Acholli, and Krishnamurthy Gorappa Ningappa, "VLSI Implementation of Hybrid Cryptography Algorithm Using LFSR Key," *International journal of intelligent Engineering & Systems*, vol. 12, no. 4, pp. 10-19, 2019, doi: 10.22266/ijies2019.0831.02.
[30] Pushpalatha G. S., Harshitha N. G., Rashmi C., Rashmi P. K., and Preksha S., "Performance Analysis of Idea Algorithm on FPGA for Data Security," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 7, pp. 2635-2638, 2019, doi: 10.22214/ijraset.2019.5436.
[31] 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, doi: 10.1109/RTEICT.2016.7808081.
[32] Ako Muhamad Abdullah, *et al.*, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data," *Cryptography and Network Security Journal*, vol. 16, no. 1, pp. 1-12, 2017.
[33] 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. 1897-1900, 2018, doi: 10.1109/TrustCom/BigDataSE.2018.00288.
[34] Liting Yu, Dongrong Zhang, Liang Wu, Shuguo Xie, Donglin Su, and Xiaoxiao Wang, "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, 2018, doi: 10.1109/TrustCom/BigDataSE.2018.00056.
[35] 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, doi: 10.1109/CompComm.2016.7924907.
[36] Ayşin Buluş, and Ercan Buluş, "Cipher with AES," *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp. 27-30, 2018, doi: 10.1109/UBMK.2018.8566543.
[37] Bih-Hwang Lee, and 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, doi: 10.1109/WOCC.2018.8372705.
[38] Jiri Petrzela, *et al.*, "Chaotic Oscillator Based on Mathematical Model of Multiple-Valued Memory Cell," *2018 International Conference on Applied Electronics (AE)*, pp. 1-4, 2018, doi: 10.23919/AE.2018.8501458.
[39] 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, doi: 10.1109/EDSSC.2018.8487056.
[40] 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, doi: 10.1109/VLSIC.2016.7573553.
[41] Jérémy Jean, Amir Moradi, Thomas Peyrin, and Pascal Sasdrich, "Bit-Siding: A Generic Technique for Bit-Serial Implementations of SPN-based Primitives," *2017 International Conference on Cryptographic Hardware and Embedded Systems. Springer*, pp. 687-707, 2017.

[42] Subhadeep Banik, Andrey Bogdanov, and Francesco Regazzoni, "Compact circuits for combined AES encryption/decryption," *Journal of Cryptographic Engineering*, vol. 9, no. 1, pp. 69-83, 2019, doi: 10.1007/s13389-017-0176-3.
[43] Qi Zhang, and Qun Ding, "Digital Image Encryption Based on Advanced Encryption Standard (AES)," *2015 5th International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pp. 1218-1221, 2015, doi: 10.1109/IMCCC.2015.261.
[44] Saeideh Sheikhpour, Ali Mahani, and Nasour Bagheri, "Reliable advanced encryption standard hardware implementation: 32- bit and 64-bit data-paths," *Microprocessors and Microsystems Journal*, vol. 81, pp. 1-16, 2020.
[45] Yue He, Liqian Wang, Yueying Zhan, Suzhi Cao, and Xiao Luo, "Dynamic Bandwidth Scheduling Algorithm for Space Applications in FC-AE-1553 Switching Network," *2018 Asia Communications and Photonics Conference (ACP)*, pp. 1-3, 2018.
[46] 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, doi: 10.1109/ICIS.2018.8466380.
[47] V. S. Prajwal, and K. V. Prema, "User Defined Encryption Procedure For IDEA Algorithm," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1668-1671, 2018, doi: 10.1109/ICACCI.2018.8554699
[48] Mark Kristian C. Ledda, Bobby D. Gerardo, and Alexander A. Hernandez, "Enhancing IDEA Algorithm using Circular Shift and Middle Square Method," *2019 17th International Conference on ICT and Knowledge Engineering (ICT&KE)*, pp. 3931-3944, 2019, doi: 10.1109/ICTKE47035.2019.8966827.
[49] Karim Shahbazi, Mohammad Eshghi, and Reza Faghih Mirzaee b., "Design and implementation of an ASIP-based cryptography processor for AES, IDEA, and MD5," *International Journal of Engineering Science and Technology*, vol. 20, no. 4, pp. 1308-1317, 2017.

BIOGRAPHIES OF AUTHORS

Sara M. Hassan (Ph.D.) received her B.Sc. degree in Electrical Engineering from Modern academy for engineering and technology, Cairo, Egypt, in 2007. She received her M. Sc. and Ph.D. degrees in electrical engineering from Ain Shams University, Cairo, Egypt, in 2013 and 2017 respectively. 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 for communication systems.
E-mail: Sara.Hassan@eng.modern-academy.edu.eg.

Gihan. G. Hamza is an Associate Professor at the National Institute for Standards (NIS), Egypt. She received her B.Sc., M.Sc., and Ph.D degrees in Communications and electronics from Ain Shams University, Faculty of Engineering, in Cairo, Egypt in 1998, 2004, and 2010, respectively. Her research interests are time and frequency dissemination through using GPS receivers.
E-mail: gihan_gomah@yahoo.com