

A Suggested Lightweight Stream Cipher for Internet of Things

Mohammed Salih Mahdi¹, Nidaa Flaih Hassan², Ahmed Saad Mohammed Aloqali³

¹Business Information College, University of Information Technology and Communications, Baghdad, Iraq
mohammed.salih@uoitc.edu.iq

²Computer science Dept. University of Technology, Baghdad, Iraq
1110020@uotechnology.edu.iq

³Computer Department, College of Basic Education, Al-Mustansiriya University Baghdad, Iraq
ahmed.saad@uomustansiriyah.edu.iq

Abstract— Cloud The Internet of Things (IoT) is a dynamic global network framework based on standardized protocols, where interconnected objects like sensors, radio-frequency identification (RFID) devices, and machines connect to the Internet. However, IoT encounters challenges related to resource utilization, energy constraints, memory limitations, and security concerns. This paper proposes an efficient keystream generator that integrates the Salsa20 and ChaCha20 algorithms to tackle these issues. Additionally, it emphasizes securing data exchanged between IoT sensors (such as Arduino IDE, Pulse Heart Rate, and Body Temperature sensors) and a cloud-based backend (with Google Firebase as a real-time database) by encrypting the information using the proposed keystream generator. The results show that the keys produced by the proposed keystream generator are less complex and successfully pass the five standard benchmark tests.

Keywords- IoT, Salsa (20), ChaCha (20), keystream generator, Arduino

1. INTRODUCTION

When it comes to confidentiality and security, IoT faces significant challenges due to several key reasons:

- **First**, IoT extends web Connectivity to various domains, including wireless sensor networks, mobile computing, ambient intelligence, and cloud computing.
- **Second**, the internet is expected to serve as the control node, and all objects are assumed to be connected.
- **Lastly**, since most of these items remain in the public domain, new confidentiality issues may arise [1-10].

The confidentiality architecture of the IoT system can be expanded as follows, As illustrated in Figure 1.

- The first primary criterion is the perceptual layer, which involves acquiring data through sensors in various fields, including agriculture, automobiles, health, earthquake measurement, and industrial automation others.
- The second criterion is, therefore, the network layer that uses technologies such as mobile computing, wireless networks, and the internet to satisfactorily and reliably transport data from the perceptual layer.
- The third criterion is the support layer, a robust platform to assist the fourth criterion through cloud computing. This layer acts as a bridge, connecting the network and application layers below.
- The fourth criterion is the application layer, which allows the delivery of services that strictly correspond to user requirements via platforms like PCs, mobile applications, web interfaces, smart TVs, etc [11-15].

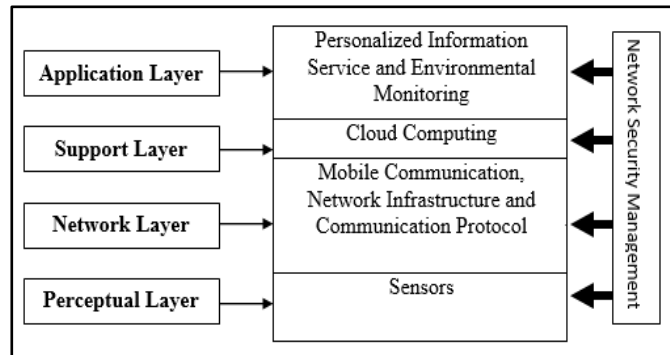


Figure 1: Confidentiality Architecture of IoT.

Several measures need to be considered to minimize the risks of IoT [15-20]:

- Among them, the first is a secure cloud infrastructure, which ensures the security of cloud services for IoT, including backend databases that are typically a major target for attackers.

- The second measure involves secure IoT end devices, a concept whereby IoT devices are made secure from adversaries with deep knowledge of the functionalities of such devices. This will also protect against unauthorized access attempts, such as password breaches.
- The third measure is secure device connectivity, which focuses on securing the communication links between IoT applications and the cloud infrastructure that constitutes the backend.

2. RELATED WORKS

Numerous studies utilize encryption procedures for IoT data. In [21] utilized RC4 stream encryption procedure and updated RC4 according to Hash to supply for securing WSN with IDE Arduino and XBee. In [22] presented an encryption approach for tackling the privacy of IoT according to Diffie Hellman and elliptic curve procedures. Furthermore, the suggested approach assists with minifying link costs and ameliorating fulfillment efficiency. In [23], an efficient and suggested secure procedure for protecting patients' health information in healthcare IoT was offered. Also, the proposed secure procedure minimizes the calculation of total time.

Many different keystream procedures have been suggested to present powerful encryption for IoT data. The keystream procedure in this paper is extended by a hybrid keystream generator based on the combination of salsa 20 and ChaCha 20.

3. MATERIAL AND METHOD

A stream cipher is a symmetric encryption algorithm in which the individual bits or bytes of the plaintext data are encrypted in real-time [24-28]. It is especially suitable for wireless sensor networks, Bluetooth, mobile communication, high resource demands, narrow bandwidth, and low power supply [29-32]. Also, it has been identified that stream ciphers exhibit faster and more efficient processing than block ciphers [33-36]. The two deterministic stream ciphers in this paper are Salsa20 and ChaCha20.

Unlike Salsa20, the ChaCha20 cipher follows the same fundamental design principles but is as fast as Salsa20 with 20 cycles while preserving the same level of security [12]. ChaCha20 is the algorithm of symmetric encryption and has been used by Google [37-39]. Salsa20 is a pseudo-random number generator extended from the eSTREAM portfolio, whereas Phone is an improved Salsa20 that includes amendments to its basic arithmetic for enhanced performance.

The total input size for Salsa20 is 512 bits, as illustrated in Figure 2. These bits serve as seeds, with each being 32 bits in size. The seeds are organized in the following matrix structure and consist of the following components [40-42]:

- Thus, the size of the key ($k_1 \dots k_8$) is presented by 256 bits.
- It should also be known that the total size of nonce (n_1, n_2) and constants ($c_1 \dots c_4$) is 192 bits.
- As mentioned above 64 bits compose the block message counter, b_1, b_2 .

c1	k1	k2	k3		x0	x1	x2	x3
k4	c2	n1	n2		x4	x5	x6	x7
b1	b2	c3	k5	=	x8	x9	x10	x11
k6	k7	k8	c4		x12	x13	x14	x15

Figure 2: Salsa20 inputs.

To encrypt original data using Salsa20, several lightweight operations are performed on the desired input, resulting in a 512-bit keystream. Finally, this keystream is XORed with the original data to achieve the final encryption output.

The first lightweight procedure is Addition, and its functionality is to add two 32-bit values. The second lightweight procedure is Exclusive-OR (XOR), which performs XOR on two 32-bit values. The third lightweight procedure is Rotation, which rotates a 32-bit value by a constant number of bits denoted as $[y \ll e]$. These three relatively light procedures make up a double duty. At the center of this duality is the Four-Quarter Function: It modifies the input column by column, followed by a row-by-row process as shown in Figure 3 [43-45]

Column form	Row form
QR(X 0, X4, X8, X12)	QR(X 0, X 1, X 2, X 3)
QR(X 5, X 9, X 13, X 1)	QR(X 5, X 6, X 7, X 4)
QR(X 10, X 14, X 2, X 6)	QR(X 10, X 11, X 8, X 9)
QR(X 15, X 3, X 7, X 11)	QR(X 15, X 12, X 13, X 14)

Figure 3: FQF of Salsa20

The Four-Quarter Function (FQF) input values include four 32-bit values, and the Four Quarter Function Four 32-bit values output are modified four 32-bit values based on the three lightweight procedures shown in the Figure 4.

$$\begin{aligned}
 b &= b \oplus [(a + d) \ll 7] \\
 c &= c \oplus [(b + a) \ll 9] \\
 d &= d \oplus [(c + b) \ll 13] \\
 a &= a \oplus [(d + c) \ll 18]
 \end{aligned}$$

Figure 4: Procedures in FQF of Salsa20.

The last step in Salsa20 is an addition between the last modified matrix of Salsa20 and the seed input of Salsa20. ChaCha20 is very similar to Salsa20 because ChaCha20 employs the same computation to generate the keystream.

However, the first major one is due to the spatial organization of the matrix seeds, as detailed in Figure 5 [46].

c1	c2	c3	c4		x0	x1	x2	x3
k1	k2	k3	k4		x4	x5	x6	x7
k5	k6	k7	k8	=	x8	x9	x10	x11
b1	b2	n1	n2		x12	x13	x14	x15

Figure 5: ChaCha20 inputs.

The second difference is observed similarly in the assert input processing of the Four-Quarter Function (FQF). In the FQF of ChaCha20, the updating occurs column-wise and then diagonal-wise, as illustrated in the figure below 6.

Column form	Row form
QR(x 0 , x 4 , x 8 , x 12)	QR(x 0 , x 5 , x 10 , x 15)
QR(x 1 , x 5 , x 9 , x 13)	QR(x 1 , x 6 , x 11 , x 12)
QR(x 2 , x 6 , x 10 , x 14)	QR(x 2 , x 7 , x 8 , x 13)
QR(x 3 , x 7 , x 11 , x 15)	QR(x 3 , x 4 , x 9 , x 14)

Figure 6: FQF of ChaCha20.

The third difference is in the formula used for the Four-Quarter Function (FQF), as shown in Figure 7.

$ \begin{aligned} a &= a + b, & d &= (d \oplus a) \ll 16 \\ c &= c + d, & b &= (b \oplus c) \ll 12 \\ a &= a + b, & d &= (d \oplus a) \ll 8 \\ c &= c + d, & b &= (b \oplus c) \ll 7 \end{aligned} $

Figure 7: Procedures in FQF of ChaCha20

In line with the discussion in the paper on Salsa20 and ChaCha20, this paper presents a blended keystream generator combining Salsa20 and ChaCha20 with modified operations to enhance the security of data exchanged between IoT sensors and Google Firebase [47].

4. STRATEGY AND ALGORITHM

The proposed Keystream generator combines the Salsa20 and ChaCha20 mapping, generating a stream key for encrypting IoT information (from IoT sensors). The Salsa20 and ChaCha20 building blocks were evaluated to develop the suggested keystream generator. Therefore, the keys generated by this generator imply fewer steps than those generated by the current complex generator, but the levels of security involved are way higher.

The proposed keystream generator, outlined in Algorithm 1, consists of 10 rounds and can be summarized as follows:

- Even round employs the three lightweight procedures of Salsa20.
- Odd round utilizes the three lightweight procedures of ChaCha20.
- The rotation procedure (7, 9, 13, and 18) in salsa (20) and (16, 12, 8, and 7) in ChaCha20 is changed from a fixed constant to a function of ‘random value’ (y0, y1, y2, and y3) in each round as depicted in Algorithm 2 and Algorithm 3 below

Algorithm (1): Suggested keystream Generator

Input: 512 bit (seed key, nonce, counter and constants) stored in a matrix x ($x_0, x_1, x_2, \dots, x_{15}$)

Output: 512 bit (keystream)

Begin

Step1: round=1

Step2: If (round modular by 2 is not equal to 0) then

Salsa-quarter function (x_0, x_4, x_8, x_{12})

Salsa-quarter function (x_5, x_9, x_{13}, x_{17})

Salsa-quarter function ($x_{10}, x_{14}, x_{18}, x_{22}$)

Salsa-quarter function ($x_{15}, x_{19}, x_{23}, x_{27}$).

Salsa-quarter function (x_0, x_1, x_2, x_3),

Salsa-quarter function (x_5, x_6, x_7, x_4),

Salsa-quarter function (x_{10}, x_{11}, x_8, x_9),

Salsa-quarter function ($x_{15}, x_{12}, x_{13}, x_{14}$).

otherwise

ChaCha-quarter function (x_0, x_4, x_8, x_{12})

ChaCha-quarter function (x_1, x_5, x_9, x_{13})

ChaCha-quarter function (x_2, x_6, x_{10}, x_{14})

ChaCha-quarter function (x_3, x_7, x_{11}, x_{15}).

ChaCha-quarter function (x_0, x_5, x_{10}, x_{15}),

ChaCha-quarter function (x_1, x_6, x_{11}, x_{12}),

ChaCha-quarter function (x_2, x_7, x_8, x_{13}),

ChaCha-quarter function (x_3, x_4, x_9, x_{14}).

Step3: round = round+1

Step4: if (round less than or equal to 10) goto step2

Step5: store the update matrix x in keystream matrix as key

End

Algorithm (2): Salsa-quarter function

Input: Four 32-bit (a, b, c, d),

Output: Update Four 32-bit (a, b, c, d).

Begin

Step1: Generate random number (y_0, y_1, y_2, y_3)

Step2:

$z_1 = b \oplus ((a + d) \lll y_0)$

$z_2 = c \oplus ((x_1 + z_1) \lll y_1)$

$z_3 = d \oplus ((z_1 + z_2) \lll y_2)$

$z_0 = a \oplus ((z_2 + z_3) \lll y_3)$

Step3:

$a = z_0$

$b = z_1$

$c = z_2$

$d = z_3$

End

Algorithm (3): ChaCha-quarter function
Input: Four 32-bit (a, b, c, d)
Output: Update Four 32-bit (a, b, c, d).
Begin Step1: Generate random number (y0, y1, y2, y3) Step2: $a = a + b$ $d = (d \oplus a) \lll y0$ $c = c + d$ $b = (b \oplus c) \lll y1$ $a = a + b$ $d = (d \oplus a) \lll y2$ $c = c + d$ $b = (b \oplus c) \lll y3$ End

Furthermore, an XOR operation is performed on the keystream obtained from algorithm 1 with the 512-bit IoT sensor data to encrypt IoT sensor data.

5. RESULTS AND DISCUSSION

The IoT sensor data of a Pulse Heart Rate, and Body Temperature of a person as input is encrypted using the keystream of the proposed approach and implemented on the Arduino IDE board.

The proposed keystream generator's key randomness characteristics are analyzed using five test statistics described in Table 1. As shown in Table 2, the results affirm that all the requested keys generated by the presented keystream generator incorporate less complicated procedures and have acceptable robustness evaluations; it has passed all five benchmark tests.

Thus, Figure 8 presents the model for exchanging encrypted messages between the cloud level, IoT health sensors, clinicians, and family members based on the keystream generator described above.

Table 1: Equations and Details of the 5 Benchmark Tests [16].

5 - benchmark tests Equations	Information on 5 - benchmark tests
$T1 = \frac{(M0 - M1)^2}{M}$	M0: number of 0's in keystream. M1: number of 1's in keystream. M: total size of keystream.
$T2 = \frac{4}{M-1} ((M11)^2 + (M00)^2 + (M01)^2 + (M10)^2) - \frac{2}{M}(M1^2 + M0^2) + 1$	M11: number of 11's in keystream. M00: number of 00's in keystream. M01: number of 01's in keystream. M10: number of 10's in keystream
$P = \frac{M}{N}$ $T3 = \frac{2^N}{P} \left(\sum_{j=1}^{2^N} M_j^2 \right) - P$	M_j : number of appearance of the j^{th} of length N.
$P_j = \frac{M - j + 3}{2^{(j+2)}}$ $T4 = \left(\sum_{j=1}^N \frac{(B_j - P_j)^2}{P_j} \right) + \left(\sum_{j=1}^N \frac{(G_j - P_j)^2}{P_j} \right)$	N :maximum j for which $P_j \geq 5$. B_j : Amount of blocks (subsequences runs of 1's) of length j in M. G_j : amount of gabs(subsequences runs of 0's) of length j in M.
$A(k) = \sum_{j=0}^{M-k-1} (S_j + S_{j+k}) \text{ Mod } 2$ $T5 = \frac{2 \left(A(k) - \frac{(M-k)}{2} \right)}{\sqrt{(M-k)}}$	k : $1 \leq k \leq [m/2]$

Table 2: Five Benchmark Tests Performance.

5 - benchmark tests	Test value	Threshold	Test value< Threshold
Frequency test T1	0.03	3.841	pass
Serial Test T2	0.645	5.991	pass
Poker Test T3	12.933	14.067	pass
Runs Test T4	4.062	9.487	pass
Autocorrelation T5	1.283	1.96	pass

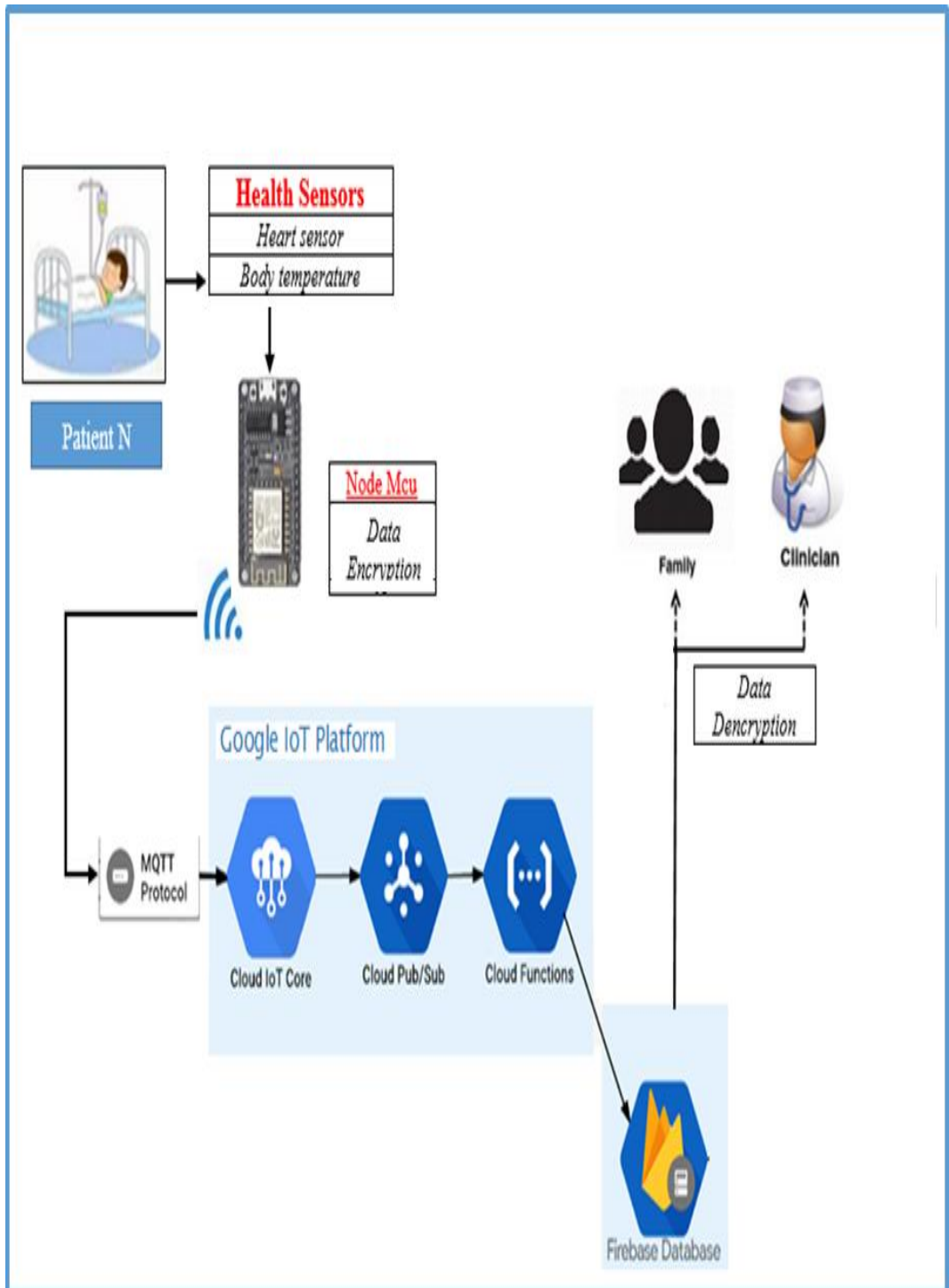


Figure 8: The framework for exchanging encrypted messages using the proposed keystream generator.

6. CONCLUSIONS

As a result, this paper presents a Keystream Generator of Salsa20 integrated with ChaCha20 in a more stable architecture using lightweight techniques for encrypting IoT sensor data. This suggested approach is well suited for IoT security since it solves the following problems often intrinsic to IoT devices: the limited amount of power available for computation and the little storage available on a device. In more detail, it provides security for the information shared between a cloud-level backend (Google Firebase as the real-time database), IoT sensors (Arduino, Pulse Heart Rate, Body Temperature sensors), clinician and family members through the application of the proposed keystream generator by encrypting respective information.

The total key size of 512 bits provided by the proposed keystream generator results in 2^{512} possibilities of the keys. Furthermore, the IoT sensor data changes at every session, and brute-force attacks on the keys are virtually impossible. The proposed keystream generator incurs a relatively small memory for implementation; furthermore, the security performance of the proposed keystream generator is also proven to be fairly good in that it can pass all five benchmark tests.

REFERENCES:

- [1] Hui S., Jiafu W., Caifeng Z. and Jianqi L., "Security in the Internet of Things: A Review ", International Conference on Computer Science and Electronics Engineering, IEEE, 2012.
- [2] G. Yang, J. Xu, W. Chen, Z. H. Qi, and H. Y. Wang, "Security characteristic and technology in the internet of things", Journal of Nanjing University of Posts and Telecommunications (Natural Science), vol. 30, no. 4, Aug, 2010.
- [3] Azeez, Raghad Abdulaali; Abdul-Hussein, Mohammad K; Mahdi, Mohammed Salih; ALRikabi, Haider TH Salim;" Design a system for an approved video copyright over cloud based on biometric iris and random walk generator using watermark technique", Periodicals of Engineering and Natural Sciences (PEN), 10, 1, 178-187, 2021.
- [4] Hassan, Nidaa Flaih; Al-Adhami, Ayad; Mahdi, Mohammed Salih; "Digital speech files encryption based on Hénon and gingerbread chaotic maps", Iraqi Journal of Science,, 830-842, 2022.
- [5] Najm, Hayder. "Data authentication for web of things (WoT) by using modified secure hash algorithm-3 (SHA-3) and Salsa20 algorithm." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12, no. 10 (2021): 2541-2551.
- [6] Mohammed Salih, " Proposed Block Cipher Algorithm with Cloud Computing Based on Keys Generator ", M.Sc. Thesis, University of Technology, 2013.
- [7] Azeez, Raghad Abdulaali; Jamil, Abeer Salim; Mahdi, Mohammed Salih; "A Partial Face Encryption in Real World Experiences Based on Features Extraction from Edge Detection", Int. J. Interact. Mob. Technol., 17, 7, 69-81, 2023.
- [8] Mahdi, Mohammed Salih; ,Proposed Block Cipher Algorithm With Cloud Computing Based on Keys Generator,"University of Technology, Msc Thesis, Baghdad, Iraq", 2013.
- [9] Omran, Alaa Hamza; Abid, Yaser M; Mahdi, Mohammed Salih; GH, Abdul-Majeed; "Design of intelligent controller to minimize the power consumption in smart city based on FPGA", Solid State Technology, 63, 6, 9120-9136, 2020.
- [10] Hassan, Enass Hammadi; Ali, Alyaa Hussein; Shehab, Russul Mohammad; Abd Alrida, Walaa Ahmad; Mahdi, Mohammad Salih; "Using K-mean Clustering to Classify the Kidney Images", Iraqi Journal of Science,, 2070-2084, 2023.
- [11] Hussein, Osama Alaa; Mahdi, Mohammed Salih; "Detection of Rice Plant Leaves Diseases using Data Augmentation and Transfer Learning Technique", 2023.
- [12] Shehab, Russul M., and Alwan M. Alwan. "Improved the sensitivity and limit of detection of surface alloying SERS sensors by controlling mixing ratio of trimetallic (Ag-Au-Pd) nanoparticles." International Journal of Nanoelectronics & Materials 16, no. 2 (2023).

- [13] Mahdi, Mohammed Salih; ,Computer aided diagnosis system for breast cancer using ID3 and SVM based on slantlet transform,QALAAI ZANIST JOURNAL,2,2,142-148,2017.
- [14] Najm, Hayder, Mohammed Salih Mahdi, and Wijdan Rashid Abdhussien. "Lightweight Image Encryption Using Chacha20 and Serpent Algorithm.", Journal of Internet Services and Information Security (JISIS), Volume 14, Issue 4, pages 436-449 (2024).
- [15] Mahdi, Mohammed Salih; Hassan, Nidaa Flaih; "Design of keystream Generator utilizing Firefly Algorithm",Journal of Al-Qadisiyah for computer science and mathematics,10,3,Page 91-99,2018.
- [16] Mahdi, Mohammed Salih; Hassan, Nidaa Falih; "A proposed lossy image compression based on multiplication table",Kurdistan Journal of Applied Research,2,3,98-102,2017.
- [17] Ansaf, Hayder, Sajjad Hussain, Hayder Najm, and O. Hassen. "Face Smile Detection and Cavernous Biometric Prediction using Perceptual User Interfaces (PUIs)." (2021): 62-68.
- [18] Mahdi, Mohammed; Hassan, Nidaa; "A suggested super salsa stream cipher",Iraqi Journal for Computers and Informatics,44,2,5-10,2018,
- [19] Tayyeh, Huda Kadhim; Mahdi, Mohammed Salih; AL-Jumaili, AS Ahmed; "Novel steganography scheme using Arabic text features in Holy Quran",International Journal of Electrical and Computer Engineering,9,3,1910-1918,2019,IAES Institute of Advanced Engineering and Science.
- [20] Farhan, A. K.; Mahdi, M. S.; "Proposal Dynamic Keys Generator for DES algorithms",islamic college university journal,2014,29,25-48,2014.
- [21] Najm, Hayder, Haider Hoomod, and Rehab Hassan. "A New WoT Cryptography Algorithm Based on GOST and Novel 5d Chaotic System." (2021): 184-199.
- [22] Kadhim, Alaa; Salih, Mohammed; "Proposal of new keys generator for DES algorithms depending on multi techniques",Engineering and Technology Journal,32,1B,94-106,2014,University of Technology-Iraq.
- [23] Mahdi, Mohammed Salih; "Proposed Secure Internet of Everything (IoE) in Health Care", "University of Technology, PhD Thesis, Baghdad, Iraq",2018.
- [24] Yahya, Ayat Mohammed, Safa Y. Taha, Russul M. Shehab, Zainab H. Omran, Mustafa A. Jihad, Aqeel Ali Al-Hilali, and Ali M. Ahmed. "The zinc Lysinate-riboflavinate structure colloid particles are a kind of colloid particles." Materials Today: Proceedings (2023).
- [25]Ansaf, Hayder, Hayder Najm, Jasim Mohammed Atiyah, and Oday A. Hassen. "Improved Approach for Identification of Real and Fake Smile using Chaos Theory and Principal Component Analysis." Journal of Southwest Jiaotong University 54, no. 5 (2019).
- [26] Mahdi, Mohammed Salih; Azeez, Raghad Abdulaali; Hassan, Nidaa Falih; "A proposed lightweight image encryption using ChaCha with hyperchaotic maps",Periodicals of Engineering and Natural Sciences (PEN),8,4,2138-2145,2020.
- [27] Mahdi, Mohammed Salih; Abid, Yaser M; Omran, Alaa Hamza; Abdul-Majeed, GH; "A Novel Aided diagnosis schema for covid 19 using convolution neural network",IOP Conference Series: Materials Science and Engineering,1051,1,012007,2021,IOP Publishing
- [28] Mahdi, Mohammed Salih; Hassan, Nidaa Falih; Abdul-Majeed, Ghassan H; "An improved chacha algorithm for securing data on IoT devices",SN Applied Sciences,3,4,429,2021,Springer International Publishing Cham.
- [29] Sami, Zahra Rahim; Tayyeh, Huda Kadhim; Mahdi, Mohammed Salih; "Survey of Iris Recognition using Deep Learning Techniques" ,Journal of Al-Qadisiyah for computer science and mathematics,13,3,Page 47–56-Page 47–56,2021.
- [30] Hassan, Nidaa Flaih; Azeez, Raghad Abdulaali; Mahdi, Mohammed Salih; ,"Video Watermarking, Theories and Robustness Issues",Journal of Al-Ma'moon College,36,2021.
- [31] Khalaf, Mahdi, Hayder Najm, Alaa Abdulhussein Daleh, Ali Hasan Munef, and Ghaith Mojib. "Schema matching using word-level clustering for integrating universities' courses." In 2020 2nd Al-Noor International Conference for Science and Technology (NICST), pp. 1-6. IEEE, 2020.

- [32] Najm, Hayder, Haider K. Hoomod, and Rehab Hassan. "A proposed hybrid cryptography algorithm based on GOST and salsa (20)." *Periodicals of Engineering and Natural Sciences (PEN)* 8, no. 3 (2020): 1829-1835.
- [33] Bakri, Bilal Ibrahim; Abid, Yaser M; Ali, Ghaidaa Ahmed; Mahdi, Mohammed Salih; Omran, Alaa Hamza; Jaber, Mustafa Musa; Jalil, Mustafa A; Kadhim, Roula AJ; "USING DEEP LEARNING TO DESIGN AN INTELLIGENT CONTROLLER FOR STREET LIGHTING AND POWER CONSUMPTION" ,*Eastern-European Journal of Enterprise Technologies*,117,8,2022.
- [34] Mahmood, Amal Ibrahim, Russul M. Shehab, Enas A. Khalid, and Abbas S. Alwan. "Simulation and experimental of Nd-YAG laser surface treatment of biomaterials type 316 L stainless steel." In *AIP Conference Proceedings*, vol. 3232, no. 1. AIP Publishing, 2024.
- [35] Najm, Hayder, H. K. Hoomod, and Rehab Hassan. "Intelligent Internet of Everything (IOE) Data Collection for Health Care Monitor System." *International Journal of Advanced Science and Technology* 29, no. 4 (2020): 2341-2350.
- [36] Hassan, Enass Hammadi; Ali, Alyaa Hussein; Shehab, Russul Mohammad; Mahdi, Mohammed Salih; Abd Alrida, Walaa Ahmad; "Mask Laws to study Texture Features of the Kidney Infection",*Iraqi Journal of Science*,,2261-2270,2023,
- [37] Hussein, Osama Alaa; Mahdi, Mohammed Salih; "REVIEW ON DETECTION OF RICE PLANT LEAVES DISEASES USING DATA AUGMENTATION AND TRANSFER LEARNING TECHNIQUES",*Iraqi Journal for Computers and Informatics*,49,1,30-40,2023.
- [38] Omran, Alaa Hamza; Mahdi, Mohammed Salih; Hashim, Ahmed A; Abdul-Majeed, GH; "Development of Low Cost Intelligent Tracking System Realized on FPGA Technology for Solar Cell Energy Generation",*International Conference on Intelligent Systems Design and Applications*,,1051-1064,2020,Springer International Publishing Cham.
- [39] Shakir, Nooruldeen Subhi; Mahdi, Mohammed Salih; "Using special letters and diacritics in Steganography in holy Quran",*Iraqi Journal for Computers and Informatics*,49,2,1-8,2023,
- [40] Abid, Yaser M; Kaissan, Nada; Mahdi, Mohammed; Bakri, Bilal I; Omran, Alaa; Altaee, Mustafa; Abid, Sura Khalil; "Development of an intelligent controller for sports training system based on FPGA,*Journal of Intelligent Systems*",32,1,20220260,2023,De Gruyter
- [41] Mahdi, Mohammed Salih; Azeez, Raghad Abdulaali; Hassan, Nidaa Flaih , "Video Watermarking, Theories and Robustness Issues",*Journal of Al-Ma'moon College*,,36,,2021,AlMamon University College.
- [42] Ahmed, Ali M., Zainab H. Omran, Safa Y. Taha, Ayat Mohammed Yahya, Mustafa A. Jihad, Russul M. Shehab, and Aqeel Ali Al-Hilali. "Dark matter subhalo disturbance in the tidal field: Towards a new simulations accuracy." *Materials Today: Proceedings* (2023).
- [43] Najm, Hayder, Hayder Ansaf, and Oday A. Hassen. "An Effective Implementation of Face Recognition Using Deep Convolutional Network." *Journal of Southwest Jiaotong University* 54, no. 5 (2019).
- [44] Al-Dabbas, Hadeel Moataz; Salih, Mohammed; "Classification of Brain Tumor Diseases Using Data Augmentation and Transfer Learning",*Iraqi Journal of Science*,,2275-2286,2024.
- [45] Mahdi, HMS; Hassan, NF; Abdul-Majeed, GH; "An improved chacha algorithm for securing data on iot devices", *SN Appl Sci* 3: 429,2021.
- [46] Omran, Alaa; Abid, Yaser; Bakri, Bilal; , "Edge Computing Vs. Cloud Computing: Evaluating Performance, Scalability, and Security in Modern Applications",*CyberSystem Journal*,1,1,1-8,2024.
- [47] Mahdi, Mohammed Salih; "Intrusion Detection Systems Based on RNN and GRU Models using CSE-CIC-IDS2018 Dataset in AWS Cloud" ,*Journal of Al-Qadisiyah for Computer Science and Mathematics*,16,4,141–160-141–160,2024.