ISSN: 2093-5374 / E-ISSN: 2093-5382

Image Encryption Using Modified Serpent Algorithm and Harris Hawks Optimization

Mohammed Salih Mahdi[1*], Wijdan Rashid Abdulhussien[2], Hayder Najm[3], and Ahmed Saad Mohammed Aloqali[4]

[1*]Business Information College, University of Information Technology and Communications, Baghdad, Iraq. mohammed.salih@uoitc.edu.iq, https://orcid.org/0000-0003-3177-5469

[2]Department of information Technology, University of Thi-Qar, Thi-Qar, Iraq. wijdan_rashid@utq.edu.iq, https://orcid.org/0000-0001-6804-5553

[3]Department of Computer Techniques Engineering, Imam Alkadhim University College, Baghdad, Iraq. haidernajem@iku.edu.iq, https://orcid.org/0000-0001-9722-4542

[4]Computer Department, College of Basic Education, Al-Mustansiriya University, Baghdad, Iraq. ahmed.saad@uomustansiriyah.edu.iq, https://orcid.org/0009-0001-5293-8856

Received: October 24, 2024; Revised: December 11, 2024; Accepted: January 14, 2025; Published: March 31, 2025

Abstract

Unauthorized data access is rising due to the enhanced expansion of electronic data transfer. Information security is heavily relied on to protect data in storage and during transfer. However, protecting image data from similar means is paramount, mainly because images have become standard information actively used in a virtually endless list of applications and fields. This work proposes an image encryption technique that uses a Harris Hawks Optimization keys generator with a modified serpent algorithm by decreased round numbers and uses a sponge function to create dynamic eight unique substitution boxes (8-S-boxes). The empirical results confirmed that the modified encryption algorithm presents comparable security and superior performance to the classical Serpent. To evaluate the fitness of keys generated by a Harris Hawks Optimization, five standard tests of the National Institute of Standards and Technology successfully surpassed the fitness of the proposed keys generator. Several analytical tests were used to evaluate the encryption methods. These tests included histogram test, correlation test, entropy test, mean squared error, peak signal-to-noise ratio, unified average changing intensity, normalized pixel difference rate, and structural similarity index. The experiments showed that the suggested encryption method significantly improved all these tests. The computed information entropy value of 7.998 is quite near the ideal value of 8, indicating a high level of randomness. Further evidence of solid encryption was provided by a UACI value of 33.979 and an NPCR greater than 99.60%. The effectiveness of the decryption method was also verified.

Keywords: Data Security, Image Encryption, Artificial Intelligence, Nature-inspired Optimization, Serpent Algorithm, Sponge Function, SHA-3, Harris Hawks Optimization.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 16, number: 1 (March), pp. 154-171. DOI: *10.58346/JOWUA.2025.I1.009*

*Corresponding author: Business Information College, University of Information Technology and Communications, Baghdad, Iraq.

1 Introduction

In this digitization age, images have become a necessary data type, and leakage of important or confidential information has frequently occurred. Everyone can transmit digital images via the Internet due to the widespread use of mobile devices. Consequently, the confidentiality of image information has become the primary concern. Scholars have been presented with various encryption algorithms to address the challenge (Qian et al., 2021). Images possess high redundancy, a large amount of image data, and high correlations between neighboring pixels (Najm et al., 2024; Sowmya et al., 2023). Consequently, specific conventional encryption algorithms, such as advanced and data encryption standards, must be more suitable for encrypting the images (Wu & Margarita, 2024). Nevertheless, various suitable encryption schemes have been proposed, such as DNA coding, chaotic systems, and compressing sensing (Omran et al., 2024).

Image encryption protects digital images against unauthorized access and tampering by rendering them unintelligible without a key for decryption. Indeed, encryption algorithms jumble up the data in an image and protect confidentiality and integrity during transmission or storage (Paul et al., 2020). The commonly used techniques for this purpose are AES, chaotic encryption, or selective encryption (Broumandnia, 2020). Image encryption plays a vital role in different applications, ranging from secure communications and military and medical imaging to cloud storage, where privacy and protection against sensitive visual data are prime concerns (Huang, 2024; Aldawoodi, & Bilge, 2024). The challenge in image encryption is how to make a trade-off between computational efficiency due to high-volume data and not degrading the quality of the image through the encryption process (Zendehboudi et al., 2023). Several scientific studies have developed different algorithms and techniques to perform image encryption and decryption to enhance speed, security, and robustness (Azeez et al., 2021; Yakubu et al., 2016).

Serpent is a symmetric key block cipher designed by Ross Anderson, Eli Biham, and Lars Knudsen and submitted as a candidate for the Advanced Encryption Standard competition. It was one of five finalists and is outstanding due to its security and simplicity (Sreevidya & Supriya, 2024). Serpent has a block size of 128 bits and may be used with a key size of 128, 192, or 256 bits (Anderso et al., 1998). It operates on the data in 32 rounds of substitution and permutation. It is designed to be highly secure, taking a conservative approach to providing resistance against differential and linear cryptanalysis. Although Serpent was not selected as the AES standard, it lost the competition. Rijndael is regarded as one of the most secure candidates because of its solid cryptographic properties and margins. The design has put security over speed; hence, it's ideal to implement in those environments where security is a concern (Anderson et al., 1998; Kesavaraj et al., 2023).

A sponge function is a cryptographic primitive; it is a function that can absorb any input and produce any output. This works in two phases: the absorbing phase, during which input data is processed internally, and the squeezing phase, which generates the output (Mohammed, 2023). The technique of operation is in iterations according to a transformation or a permutation of constant size acting on an extensive internal state, which is divided into two parts: the "rate" part, directly influencing input and output data, and the "capacity," for security not directly interacting with the data. The wide application of a sponge function in many modern cryptographic constructions can be found in hash functions and stream ciphers (Mahdi et al., 2021). One particularly well-known example is the Keccak sponge function, which became the base for the SHA-3 cryptographic hash standard. Since a sponge function may offer security and output length adaptability, they find various uses ranging from simple hashing and encryption to message authentication (Najm, 2021; Said et al., 2024).

Harris Hawks Optimization (HHO) is a metaheuristic strategy Heidari and others devised in 2019 (Tripathy et al., 2022; Jacqueline & Ranjith Singh, 2024). It mimics the way several Harris hawks hunt, a bird species that is unique and good at hunting. The algorithm imitates the behaviors of hawks, such as surprise attack, gentle encircle, and harsh encircle, to perform the procedures of search and utilization in the search space. Exploration and exploitation phases ensure that HHO is keen on the general method of seeking answers and, simultaneously, specific improvement of the optimizing solutions, thus making it efficient in solving complicated optimization problems (Li et al., 2024). The versatility and effectiveness of HHO have been revealed through many purposeful employs comprised of engineering design, machine learning, and feature selection. It has attracted appreciation for its capability in solving nonlinear, high-dimensional optimization problems (Mahdi & Hassan, 2018).

This paper presents a new kind of image encryption, a problem that has become more sensitive due to constant technological advancements. Our contributions are multi-fold:

- Innovative Encryption Framework: We propose a novel image encryption system employing the HHO along with the Serpent algorithm of restricted rounds in terms of number. This framework is enhanced further by using a sponge function, which in turn produces eight dynamic substitution boxes (8-S-boxes), thus improving the cryptographic level of security (Abdullah, 2024).
- Rigorous Key Fitness Validation: Five NIST standard tests were used to test the randomness and robustness of the keys generated by HHO. The obtained results successfully overcome all of the benchmarks, thus confirming the applicability of the proposed key generation process.

The remainder of the paper is structured as follows: Section 2 illustrates the related work, Section 3 depicts the Serpent algorithm, the sponge function used, and the Harris Hawks Optimization algorithm, Section 4 describes various facets of the proposed method in detail, Section 5 provides the results and discussion, and Section 6 provides the conclusion of the research.

2 Related Works

A variety of cryptographic techniques are described for image encryption. The authors in (Ibrahim et al., 2023) presented a new computationally efficient approach for a color visual cryptographic scheme using HHO algorithm. This method reduces commonly experienced issues in visual cryptography, such as pixel expansion and relatively low decryption quality, due to the choice of the color levels of the shares. The outcome reveals that the proposed approach improves image quality, encryption security, and computational speed. It yields better results than other benchmark values of NPCR=99.90, UACI=32.00, and PSNR=6.0010.

The authors in (Ahmad et al., 2018) proposed an optimized image encryption based on particle swarm optimization (PSO) and chaotic logistic map, in which the optimized parameter is the correlation between adjacent pixels to yield a highly encrypted image. Hence, it would improve the quality of encryption. It is highly effective in yielding excellent encryption performance regarding pixel distribution, entropy, and resistance to differential attack. Simulation results on benchmark test images prove that the proposed scheme generated flatter histograms with higher entropy values closer to the ideal value of 8 and very good NPCR and UACI scores, demonstrating its robustness in security.

The authors in (Elshoush et al., 2021) proposed a more secure method of encrypting RGB images than Serpent-256-ECB, which uses Lorenz 96 chaotic block key generation. High performance and robust security are achieved with significant improvement compared with traditional Serpent-256-ECB. Varying keys for each block, generated using a chaotic map and parallel computing, increase the

encryption and decryption process speed and reduce the processing time by over 60%. It has also proven to resist statistical and differential attacks, while the correlation coefficients are close to zero and the NPCR and UACI values are increased. Keyspace is expanded, so brute force attack is improbable.

The authors in (Hussain et al., 2023) presented a re-engineered version of the Serpent algorithm for image encryption, including Power Associative loops and Mobius transformation to enhance S-box security. Accordingly, the new scheme enhances complexity and randomness in encryption with a larger key space than traditional Galois Field-based cryptosystems, enhancing robustness against cryptanalytic attacks. A comprehensive analysis involving key sensitivity, correlation, and differential analysis justifies the strength and efficiency of the proposed approach. The technique achieves much better security performance regarding NPCR, UACI, and information entropy while keeping fast execution times.

The authors in (Tahiri et al., 2023) presented a new RGB image encryption method by incorporating spatial and transformation approaches through the modified 3D fractional Henon map and discrete fractional Krawtchouk moments. This paper presents a new hybrid optimization technique, H-SSAOA, which implements the Salp Swarm technique, abbreviated as SSA, and the Arithmetic Optimization Algorithm, abbreviated as AOA, in optimizing the encryption settings. The proposed technique enhances security performance by providing strong encryption keys, effectively defeating statistical and differential attacks. The simulation results represented here reflect the adequacy of the method for maintaining image quality and resistance against different attack vectors. Key sensitivity is guaranteed so that a slight key variation leads to encrypted outputs being notably different from another decryption within illegality rules. Histogram analysis reveals a uniform distribution in encrypted images, defeating statistical attack, while the correlation analysis represents an insignificant correlation between the adjacent pixels. It also resists differential attacks, as confirmed by high values of NPCR and UACI; therefore, it guarantees strong protection against images.

The authors in (Xu et al., 2023) presented an AI-driven image encryption based on compressed sensing incorporated with the Rabinovich hyperchaos system for enhanced image security. In the proposed approach, DWT is used to represent images sparsely. Then, the generated hyperchaotic sequences are employed for diffusion and scrambling to encrypt the images securely. The simulated results are found to be highly secure, with exceptional values in NPCR (0.9978), UACI (0.3337), and information entropy (7.9987). This cipher resists various statistical and differential attacks; it is resistant to any brute-force attack, which essentially protects AI-generated images during transmission.

3 Background

The proposed approach is based on three existing concepts: the serpent algorithm, the sponge function, and Harris Hawks optimization. The subsections preview those three pioneering techniques as a backdrop to the proposed image encryption.

3.1. The Serpent Algorithm

The serpent algorithm is a block cipher designed to offer the most security. Its enhanced algorithm is among the most effective methods of any encryption. The Serpent cipher is a block encryption technique that employs a 128-bit data block and supports key sizes of 128, 192, or 256 bits. It is a 32-round system functioning on four 32-bit words, so the block size is 128 bits. Each round utilizes one of eight 4×4 S-boxes, applied 32 times in tandem. It was designed to perform all operations concurrently. As shown in Figures 1 and 2, It provides for three main functions (Yousif, 2019; Najm et al., 2020; Paul et al., 2020

Mahdi et al., 2018; Najm et al., 2021):

- **Initial Permutation (IP)**

In this stage, the Serpent rearranges the data blocks to prepare them for encryption.

- **Round Function**

This function employs eight S-boxes, referenced as S. As such; the round function is run thirty-two times on data block B. Each cycle has three operations: A key mixing through XOR, 32 rounds using the same 4 × 4 S-box substitution, and a linear transformation referred to as LT. In its place, an additional key combining XOR for the last round is used in the LT.

- **Final Permutation (FP)**

A concluding permutation of bits repositions them correctly, effectively serving as the inverse of the original permutation. FP can be executed using a lookup table or algorithmically by substituting the bit at location (i) with the bit at position (i × 4) mod 127, thus preserving only bits 0 and 127. This ultimate permutation outcome constitutes the definitive ciphertext of the procedure.

- **Key Generation**

The Serpent procedure demands 33 round keys derived from the user's key to carry out all 32 rounds for each block. The algorithm accepts a 256-bit key of the user, then this 256-bit split into eight 32–bit subkeys: w1, w2, w3…, and w8. Then, it generates the 132 intermediate keys. The following process involves expanding the intermediate keys to produce 33 round keys, from which 128-bit blocks are formed using S-boxes.

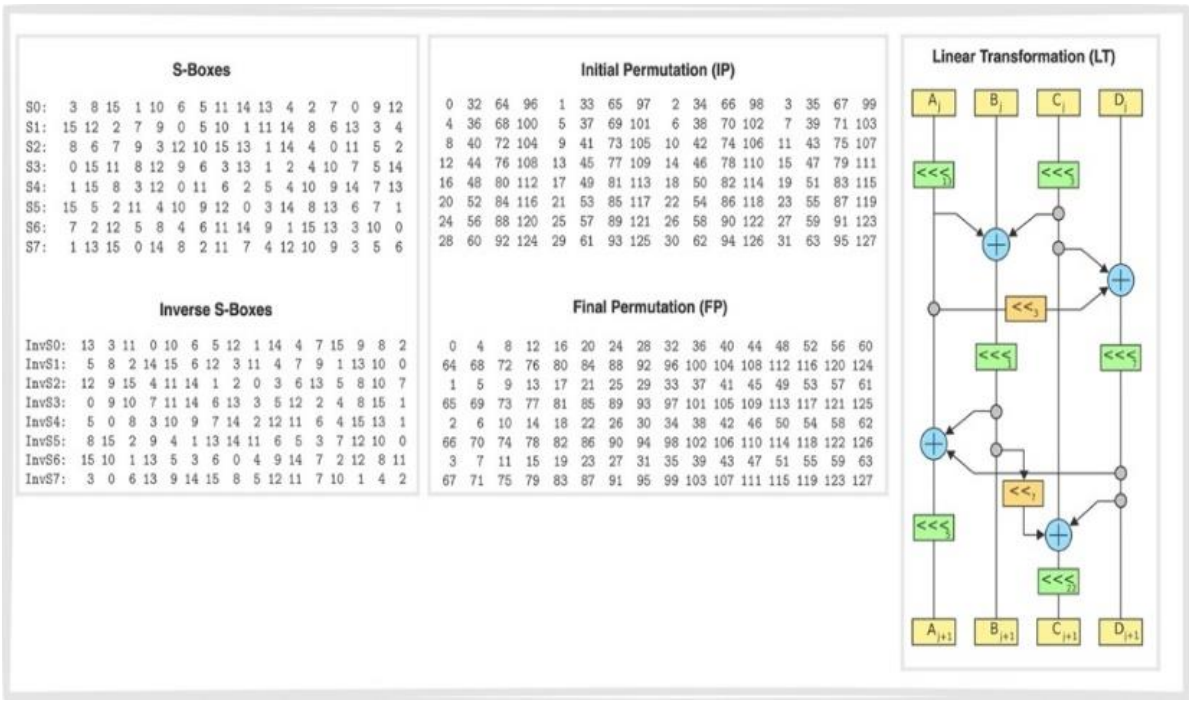


Figure 1: Main Functions of the Serpent Algorithm (Damaj, 2007)

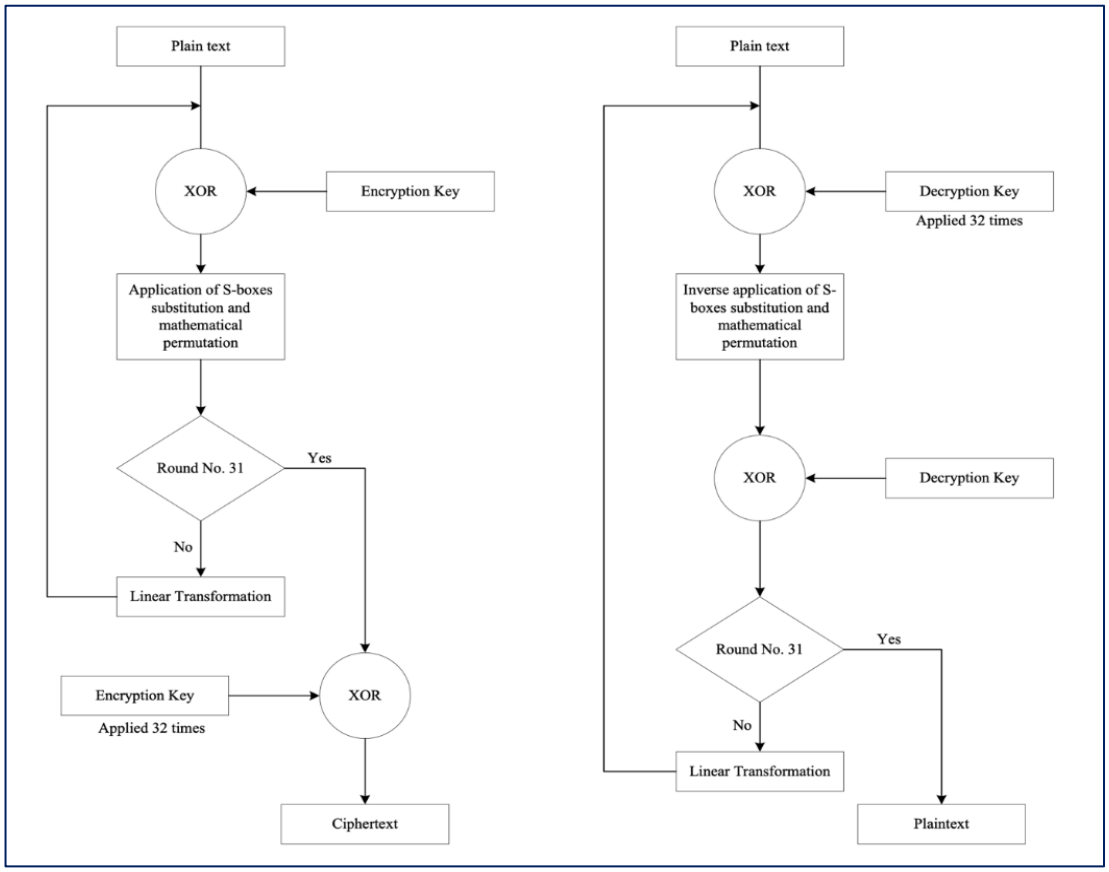


Figure 2: Encryption and Decryption of Serpent Algorithm (Shareef et al., 2022)

3.2. The Sponge Function

The sponge function is a blueprint for cryptography for various cryptographic algorithms, from hash functions and pseudorandom number generators to stream ciphers. Most cryptographic primitives designed in recent years, such as SHA-3, using the Keccak algorithm, are based on the basic underlying principle of the sponge construction methodology. As shown in Figure 3, a sponge function works in two stages (John & Jose, 2023) (Agievich et al., 2016):

- **Absorbing Stage**

Information fed to a function is " absorbed " into its internal state bit by bit or block by block. The function maintains a sizeable internal state (Mahdi et al., 2020). The topmost part of it is easy to touch, but the inner part is more challenging. Now, the input is XORed with the outer part of the state and transformed using a transformation function.

- **Squeezing Stage**

The function starts producing output bits when all the input has been ingested. The outer part of the state creates the output; the same transformation function transforms the internal state for every output produced.

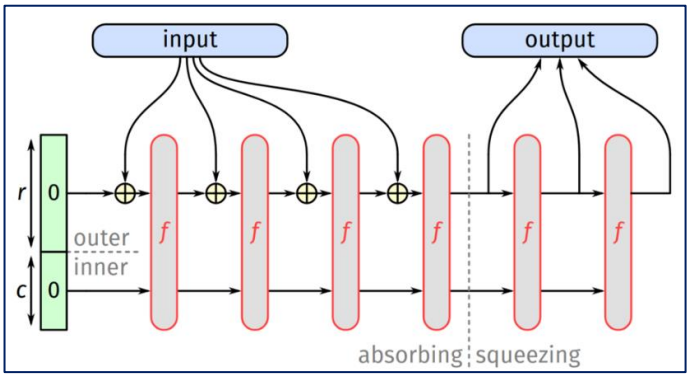


Figure 3: Sponge Function Design (dos Santos & López, 2017)

3.3. Harris Hawks Optimization (HHO)

HHO is an optimization technique inspired by the behavioral modeling of Harris Hawk birds. The cooperative idea among the hawks while hunting their prey is the central concept behind the algorithm. According to this algorithm, a group of Harris hawks attacks the target with diversified directions to attain the element of tactical surprise (Abbasi et al., 2021). The escape pattern of the target is related to the chase model of the Harris hawk. The avian species cooperate when an attack is made. Simultaneously, the leader of the Harris hawks attacks the targeted prey and chases it. Then, in the twinkling of an eye, it vanishes to give space for the next Harris hawk to continue the chase (Albukhnefis et al., 2023)

(Abbas et al., 2022). In this method, the target will be exhausted and caught. Application to constrained problems reveals that the HHO method is better than other algorithms. Also, HHO is a global optimizer that can maintain the balance between the phases of exploitation and exploration. Three steps are included in the HHO algorithm. The first phase is the ability of the algorithm for exploration, which is mathematically described as (Hussien et al., 2022):

$$x(t+1) = \begin{cases} x_{\text{rand}}(t) - r_1 \mid x_{\text{rand}}(t) - 2r_2 x(tt \mid & q \geq 0.5 \\ x_{\text{prey}}(t) - x_a(t) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (1)$$

x(t) is a representation of the current position of Hawk. X (t + 1) denotes the position of Hawk in the next iteration. t, $x_{prey}(t)$ represents the position of prey. r_1, r_2, r_3, r_4, and q are random values between intervals (0,1). $x_{rand}(t)$ represents a randomly selected hawk from the population. LB and UB represent the lower and upper bands, respectively. $x_a(t)$ is Harris Hawk's average position given by (Hussien et al., 2022) (Equation 1):

$$x_a(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t) \quad (2)$$

$x_i(t)$ denotes the position of each Harris Hawk at iteration t. At the same time, N shows the total number of Harris Hawks. The next phase is the exploitation. The energy of the Hawks deteriorates during pursuit and capture. The energy of prey can be modeled as (Equation 2) (Hussien et al., 2022):

$$E = 2E_0\left(1 - \frac{1}{T}\right) \quad (3)$$

E_0 is the energy in the initial stage, T is the maximum number of iterations, and E is the escaped energy. During this process, the exploration happens when $|E_0| \geq 1$, while the exploitation occurs in the case of $|E_0| < 1$ (Equation 3).

The third phase is exploitation, mainly used to improve the local solutions obtained from the previously identified solutions. The attack phase involves the hawks' sudden attack on the prey identified during the previous phase. Four models have been presented for the attack phase, considering the prey's escape and the hawk's pursuit (Hussien et al., 2022). The various phases of the HHO algorithm are represented in Figure 4.

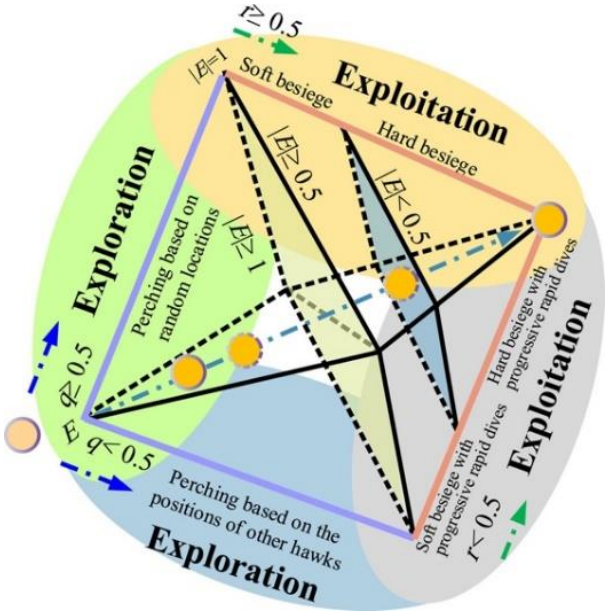


Figure 4: Phases of HHO (Hussien et al., 2022)

The following four strategies give the attacking strategy in the exploitation phase (Tripathy et al., 2021; Kennedy et al., 1995; Khalaf et al., 2020; Azeez et al., 2023; Abdulhussien et al., 2022):

- **Soft Besiege:** Hawks encircle the victim and attempt to besiege it softly. They may do this by changing their positions concerning the prey while keeping enough flexibility to avoid falling into a local minimum.
- **Hard Besiege:** If the victim's vital capacities deteriorate, hawks execute a high-intensity besiege attack and change their positions around the target with high velocity. This strategy stresses a more aggressive approach toward convergent optimal solutions.
- **Soft Besiege with Rapid Dive:** This will simulate scenarios where the prey tries to escape but still has some energy left. The Hawks will practice a mild besiege while performing the swift dive, moving prudently and agilely simultaneously.
- **Hard Besiege with Rapid Dive**: During the stage when the victim is almost exhausted, the hawks perform a fierce attack with a fast dive. This aggressive exploitation strategy converges rapidly toward an optimum solution by minimizing the chances of prey escaping from missing an optimal solution.

4 The Proposed Method

Image encryption is a perfect way of protecting the visual content with the most minor effect on the format, size, and quality. The encryption system is one of the critical elements of modern information security; it imbues an ever-growing need for fast, efficient encryption strategies. The proposed method, Modified Serpent Algorithm with HHO, improves the Serpent algorithm by utilizing the HHO mode.

The HHO was used to generate 128-bit keys to resolve the generating subkeys from the primary key in the serpent algorithm. Take note that you can produce all the keys even before the redesigned serpent is inaugurated.

4.1. Encryption and Decryption Processes

Due to the analytical nature of image encryption with the Serpent algorithm, the encryption process uses HHO to create optimized keys. Each HHO population member in this approach is a candidate key in the form of a 128, 192, or 256-bit binary string, according to the Serpent algorithm's key size. The HHO algorithm, based on the cooperative hunting of Harris Hawks, makes such candidate keys converge toward the one that offers the optimal security of the image encryption. The optimization process allows the probability distribution of variables to be initiated with random keys. These keys are then utilized to encrypt the target image by employing the modified serpent algorithm. A fitness function is used to measure the quality of each key.

In the optimization, HHO adopts different hunting styles of hawks, including soft and hard besiege, to update the keys concerning their fitness score. Subsequent iterations to find the best keys occur until the algorithm approximates a set of keys that best provide enhanced encryption, as shown in Algorithm 1. The final key, identified after the optimization, is used for the modified serpent encryption of the image, which causes the image to be highly secure and encrypted. Specifically, combining HHO with the modified serpent algorithm provides a fresh approach to improving the effectiveness of image encryption to establish extremely secure keys.

Then, the image is read and encrypted using the proposed method through a modified serpent algorithm. The encryption process requires an input of 128-bit plaintext data, which undergoes around 16 rounds before being processed by the function. The IP function handles the 128 bits by amalgamating the data. The following statistics phase involves data processing, which requires using keys provided by the HHO. The XOR function is utilized between the initial derived key and the 128 produced by the function (IP) in the subsequent processing phase. The outcome of the preceding job is partitioned into four pieces, each measuring 128 units in length. These segments are then submitted to a sponge function to generate eight dynamic unique 8-S-boxes based on a key using a combination of a sponge function and the SHA-256 hash algorithm, as shown in Algorithm 2. It's a creative way to implement a pseudo-random S-box generator for cryptographic purposes. Finally, the proposed method produced an encrypted image, as shown in Figure 5.

The decryption process closely resembles the encryption process in terms of parallelism exploitation. However, it employs the reverse sequence of the sub-keys obtained by HHO, the inverse 8-S-boxes from the sponge function, and the inverse linear transformation.

Algorithm 1: HHO_Key_Generation(num_hawks, max_iter, dim, fitness_threshold)

Input:

num_hawks (Number of hawks = **30**).

max_iter (Maximum number of iterations = **500**).

dim (Number of bits in a key = **128**).

fitness_threshold (Threshold for determining strong keys = **1.0**).

Output:

best_key (The best 128-bit key found).

strong_keys (List of strong keys that pass fitness threshold).

```
Step 1: // Initialize Hawks
 hawks ← Initialize num_hawks binary vectors of length dim
 best_hawk ← hawks[0]
 best_fitness ← Objective_Function(best_hawk)
Step 2: //Iterate through maximum iterations
 for iteration from 0 to max_iter do
   for i from 0 to num_hawks do
     r, q ← Generate two random numbers between 0 and 1
     progress ← iteration / max_iter
Step 3: //Exploration or Exploitation Phase
     if r >= 0.5 then
       if progress < 0.5 then
         hawks[i] ← hawks[i] XOR best_hawk
       else
         Randomly select two bits and swap them in hawks[i]
     else
       if q < 0.5 then
         Randomly select num_flips bits and flip them in hawks[i]
       else
         perturbation ← Random binary vector of length dim
         hawks[i] ← best_hawk XOR perturbation
Step 4: //Fitness Evaluation
     fitness ← Objective_Function(hawks[i])      // five tests of NIST
     if fitness == fitness_threshold then
       Add (hawks[i], fitness) to strong_keys
     if fitness > best_fitness then
       best_hawk ← hawks[i]
       best_fitness ← fitness
   // Print progress every 50 iterations, at start, and end
   if iteration % 50 == 0 or iteration == 0 or iteration == max_iter - 1 then
     Print "Iteration", iteration, "Best fitness:", best_fitness
 Step 5: //Convert and save strong keys
 hho_key ← Convert up to 16 strong keys to hexadecimal
 Return best_hawk, strong_keys
```

Algorithm 2: Generating S-Boxes and Their Inverses

Input:

key: A secret key, string type.

output_size: Size of the output of the sponge function, default is 128 bits.

Output:

A set of unique 8-S-boxes and their corresponding inverse 8-S-boxes.

```
Step 1: // Absorbing Phase and Squeezing Phase
Function sponge_function(key, output_size):
        hash_output = SHA256(key)        // Hash the key using SHA-256
        random.seed(hash_output as integer)
        random_output = generate random bits of size (output_size / 8)
Return random_output
Step 2: // Generate Unique 8 S-boxes (dynamic)
Function generate_sboxes(key):
        sboxes = [ ]
        For i = 0 to 7:
          sponge_output = sponge_function(key + str(i), 16 bits)
          sbox = [0, 1, 2, ..., 15]
          random.seed(sponge_output as integer)
          shuffle(sbox)
          append sbox to sboxes
Return sboxes
Step 3: // Generate Inverse S-box (dynamic)
Function generate_inverse_sbox(sbox):
        inverse_sbox = [0] * length(sbox)
        For i = 0 to length(sbox) - 1:
          inverse_sbox[sbox[i]] = i
Return inverse_sbox
```

Algorithm 3: Image Encryption using Modified Serpent with HHO

Input:

Plain-Image: Original Image (128-bit block).

$K_0, K_1, ..., K_{15}$ (Keys generated by HHO)

Output:

Cipher-Image: Encrypted Image.

Step 1: Initial XOR (Key Whitening):

$X = \text{Plain-Image} \oplus K_0$

Step 2: For each round i = 1 to 14:

a. $X = X \oplus K_i$ (XOR with round subkey generated by HHO)

b. X = 8-S-box(X) (Using sponge function)

c. X = Linear Transform(X) (Linear transformation, bit mixing)

d. $X = X \oplus K_{i+1}$ (Further XOR operation)

Step 3: Final XOR with subkey K_{15}:

$\text{Cipher-Image} = X \oplus K_{15}$

End

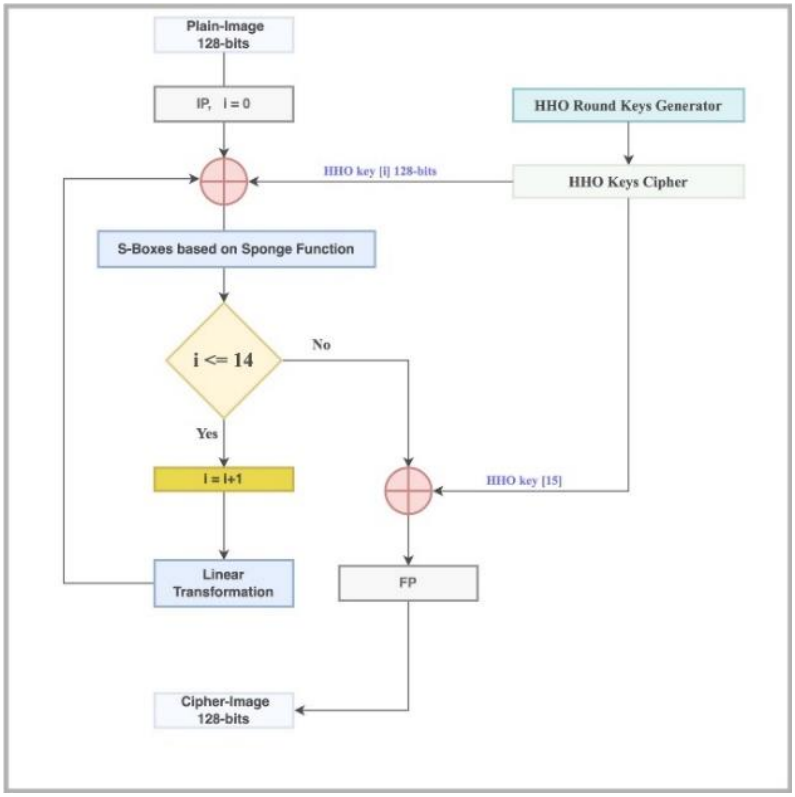


Figure 5: Flow Chart of the Modified Serpent Algorithm Encryption

5 The Results and Discussion

The following experiments are implemented to evaluate the proposed method's results regarding the runtime test, statistical test, differential test, and image cipher quality. The statistical evaluation techniques explain the histogram test and the correlation of nearby pixels. The randomness of the created test data was evaluated using the NIST Statistical Test Suite.

5.1. Runtime Tests

Table 1 below shows the time to encrypt and decrypt the four analyzed images. Furthermore, the dimensions and sizes of the image are given in kilobytes. This makes it possible to compare the encryption time with the decryption time on each image separately.

Table 1: Runtime Tests

Images	**Image-Dim**	**Image Size in Kb**	**Encryption** (sec)	**Decryption** (sec)
Baboon	64*64	8 KB	3.099	3.219
Peppers	128*128	16 KB	3.514	3.622
Cat	256*256	29 KB	3.641	3.741
Dog	512*512	70 KB	4.066	4.176

5.2. Statistical Tests

Statistical tests also prove how resistant the encryption method is against statistical attacks and how strong it is. In this process, critical statistical characteristics of the encrypted pictures will be checked.

An enormous entropy indicates more robust encryption because it is tough for attackers to predict, and hence, it measures pixel entropy, as shown in Table 2. Secondly, to prevent statistical attacks that rely on distinguishing the image pattern, the images' histograms are rechecked to ensure that they are ‘fuzzy,’ as shown in Table 3. Lastly, we employ contingency analysis to determine the correlation (Horizontal, Vertical, and Diagonal) between adjacent pixels. At the same time, this should not be very important between two neighbors for the cipher data in an adequately encrypted image compared to the original, as shown in Table 4. The goal of these tests is to validate the ability of the encryption to defend against statistical vulnerabilities.

Table 2: Entropy Test of the Proposed Method

Image	Entropy
Baboon	7.998
Peppers	7.995
Cat	7.985
Dog	7.997

Table 3: Histogram Test of the Proposed method

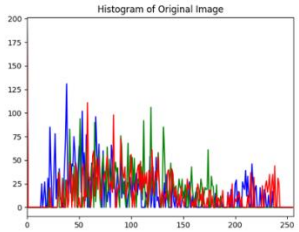
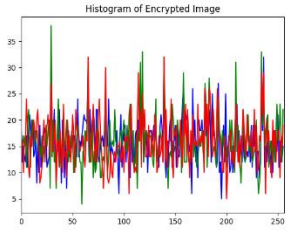
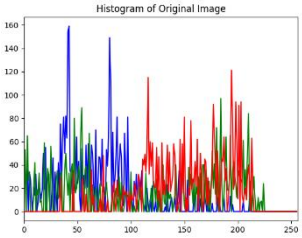
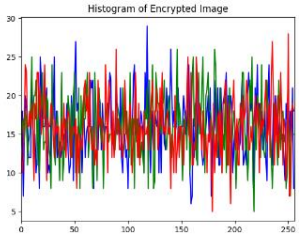
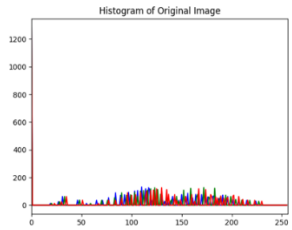
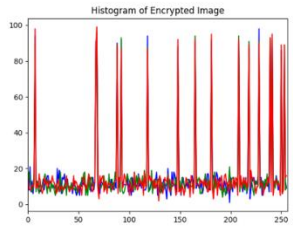
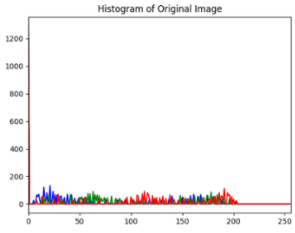
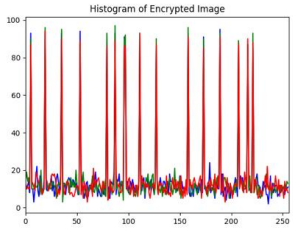
Image	Histogram of Original Image	Histogram of Encrypted Image
Baboon		
Peppers		
Cat		
Dog		

Table 4: Correlation Test of the Proposed Method

Image	Original and Encrypted Image		
	H	V	D
Baboon	0.01027542	-0.00310981	0.02686402
Peppers	-0.03194584	0.01007451	0.01607513
Cat	0.40036088	-0.08423021	-0.05429871
Dog	0.20073041	0.03462675	0.05752362

5.3. Differential Tests

An example of an approach to checking the immunity of cryptographic algorithms from differential tests is presented by observing the changes in the encrypted image when the difference between two adjacent images is merely one pixel. In the differential test, one must consider UACI (Unified Average Changing Intensity) and NPCR (Number of Pixels Change Rate) as shown in Table 5.

Table 5: Differential Test of The Proposed Method

Image	UACI	NPCR
Baboon	31.023	99.381
Peppers	31.669	99.593
Cat	30.271	99.536
Dog	33.979	99.674

5.4. Cipher Image Quality Tests

Prerequisites for comparing the quality of an encrypted and the original image include cipher image quality assessment measures such as mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) as shown in Table 6. Encryption produces significant distortion where its comprehension remains unfazed; thus, these tests are essential.

Table 6: Cipher Image Quality Tests of the Proposed Method

Image	MSE	PSNR	SSIM
Baboon	105.840901	27.884268	0.023399
Peppers	105.250081	27.908579	0.014137
Cat	106.885904	27.841599	0.014930
Dog	104.775309	27.928214	0.021477

5.5. NIST Tests

The National Institute of Standards and Technology (NIST) developed the NIST Statistical Test Suite to determine whether cryptographic sequences are random. A fitness score generated based on this key means that the more tests the key passes, the better the fitness score. To evaluate the randomness (fitness) of keys generated by HHO, five standard tests of NIST successfully surpassed the Fitness of the proposed keys generator, as shown in Table 7.

Table 7: Sample Fitness of the Proposed Key Generator

Tests	P-Value
Frequency	0.855
Frequency within block	0.867
Runs	0.915
Longest Run	0.709
Cumulative Sums	0.812

6 Conclusion

In this study, we proposed an improved Cryptographic technique that uses a modified Serpent algorithm to perform the encryption task and HHO to generate high-quality encryption keys. The applied modifications to the original Serpent algorithm enhanced the security and speed of encryption by using a sponge function to create eight dynamic unique substitution boxes (8 S-boxes) and 16 rounds. Engage the HHO algorithm to extract the most secure and unique cryptographic keys. Furthermore, the method has been challenged with the NIST statistical test suite and other cipher quality criteria (MSE, PSNR, SSIM) that revealed the proposed method's immunity to differential or statistical forms of attacks. HHO integration improved the generation of keys, notably by efficiently mapping critical space to match the word choice and achieve maximum security.

References

[1] Abbas, A. H., Mansour, H. S., & Al-Fatlawi, A. H. (2022). Self-Adaptive Efficient Dynamic Multi-Hop Clustering (SA-EDMC) Approach for Improving VANET's Performance. *International Journal of Interactive Mobile Technologies*, *17*(14). https://doi.org/10.3991/ijim.v16i14.31081

[2] Abbasi, A., Firouzi, B., & Sendur, P. (2021). On the application of Harris hawks optimization (HHO) algorithm to the design of microchannel heat sinks. *Engineering with Computers*, *37*, 1409-1428. https://doi.org/10.1007/s00366-019-00892-0

[3] Abd Elaziz, M., Heidari, A. A., Fujita, H., & Moayedi, H. (2020). A competitive chain-based Harris Hawks Optimizer for global optimization and multi-level image thresholding problems. *Applied Soft Computing*, *95*, 106347. https://doi.org/10.1016/j.asoc.2020.106347

[4] Abdulhussien, W. R., El Abbadi, N. K., & Gaber, A. M. (2021). Hybrid deep neural network for facial expressions recognition. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, *9*(4), 993-1007. http://dx.doi.org/10.52549/ijeei.v9i4.3425

[5] Abdullah, D. (2024). Enhancing cybersecurity in electronic communication systems: New approaches and technologies. *Progress in Electronics and Communication Engineering*, *1*(1), 38-43.

[6] Agievich, S., Marchuk, V., Maslau, A., & Semenov, V. (2016). Bash-f: another LRX sponge function. *Cryptology ePrint Archive*.

[7] Ahmad, M., Alam, M. Z., Umayya, Z., Khan, S., & Ahmad, F. (2018). An image encryption approach using particle swarm optimization and chaotic map. *International Journal of Information Technology*, *10*, 247-255. https://doi.org/10.1007/s41870-018-0099-y

[8] Albukhnefis, A. L., Sakran, A. A., Mahe, A. S., Mousa, M. I., & Mahdi, A. M. (2023). Hybrid Intrusion Detection Systems Based Mean-Variance Mapping Optimization Algorithm and Random Search. *International Journal of Intelligent Engineering & Systems*, *16*(5). https://doi.org/10.22266/ijies2023.1031.47

[9] Aldawoodi, A., & Bilge, H. Ş. (2024). Advancing Sustainable Marine Exploration: Highly Efficient Photonic Radar for Underwater Navigation Systems under the Impact of Different Salinity Levels [Article]. Sustainability (Switzerland), 16(7), Article 2851. https://doi.org/10.3390/su16072851

[10] Anderson, R., Biham, E., & Knudsen, L. (1998). Serpent: A proposal for the advanced encryption standard. *NIST AES Proposal*, *174*, 1-23.

[11] Anderson, R., Biham, E., Knudsen, L., & Technion, H. (1998, August). Serpent: A flexible block cipher with maximum assurance. In *The first AES candidate conference* (pp. 589-606).

[12] Azeez, R. A., Abdul-Hussein, M. K., Mahdi, M. S., & ALRikabi, H. T. S. (2021). 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.

[13] Azeez, R. A., Jamil, A. S., & Mahdi, M. S. (2023). A Partial Face Encryption in Real World Experiences Based on Features Extraction from Edge Detection. *Int. J. Interact. Mob. Technol.*, *17*(7), 69-81.

[14] Bakri, B. I., Abid, Y. M., Ali, G. A., Mahdi, M. S., Omran, A. H., Jaber, M. M., ... & Kadhim, R. A. (2022). Using Deep Learning to Design an Intelligent Controller for Street Lighting and Power Consumption. *Eastern-European Journal of Enterprise Technologies*, *117*(8). 10.15587/1729-4061.2022.260077

[15] Broumandnia, A. (2020). Image encryption algorithm based on the finite fields in chaotic maps. *Journal of Information Security and Applications*, *54*, 102553. https://doi.org/10.1016/j.jisa.2020.102553

[16] Damaj, I. W. (2007). Parallel algorithms development for programmable devices with application from cryptography. *International Journal of Parallel Programming*, *35*, 529-572. https://doi.org/10.1007/s10766-007-0046-1

[17] dos Santos, L. C., & López, J. (2017, November). Pipeline Oriented Implementation of NORX for ARM Processors. In *Anais do XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais* (pp. 2-15). SBC. https://doi.org/10.5753/sbseg.2017.19486

[18] Elshoush, H. T., Al-Tayeb, B. M., & Obeid, K. T. (2021). Enhanced Serpent algorithm using Lorenz 96 Chaos-based block key generation and parallel computing for RGB image encryption. *PeerJ Computer Science*, *7*, e812. https://doi.org/10.7717/peerj-cs.812

[19] Huang, L. (2024). Quantifying Taxation Policy Effectiveness: The Mediating Role of Big Data and the Moderating Influence of Digitalization. *Journal of Internet Services and Information Security, 14*(4), 163-180. https://doi.org/10.58346/JISIS.2024.I4.009

[20] Hussain, S., Asif, M., Shah, T., Mahboob, A., & Eldin, S. M. (2023). Redesigning the serpent algorithm by PA-Loop and its image encryption application. *IEEE Access*, *11*, 29698-29710. https://doi.org/10.1109/ACCESS.2023.3261568

[21] Hussien, A. G., Abualigah, L., Abu Zitar, R., Hashim, F. A., Amin, M., Saber, A., ... & Gandomi, A. H. (2022). Recent advances in harris hawks optimization: A comparative study and applications. *Electronics*, *11*(12), 1919. https://doi.org/10.3390/electronics11121919

[22] Ibrahim, D., Sihwail, R., Arrifin, K. A. Z., Abuthawabeh, A., & Mizher, M. (2023). A novel color visual cryptography approach based on Harris Hawks Optimization Algorithm. *Symmetry*, *15*(7), 1305. https://doi.org/10.3390/sym15071305

[23] Jacqueline, C., & Ranjith Singh, K. (2024). Enriched Deep Neural Network Improved by Chaotic Harris Hawk Optimizer for Prediction of Behavioural Traits of Individuals. *Journal of Internet Services and Information Security, 14*(4), 511-523. https://doi.org/10.58346/JISIS.2024.I4.032

[24] John, A., & Jose, J. (2023). Hash Function Design Based on Hybrid Five-Neighborhood Cellular Automata and Sponge Functions. *Complex Systems*, *32*(2). https://doi.org/10.25088/ComplexSystems.32.2.171

[25] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). ieee. https://doi.org/10.1109/ICNN.1995.488968

[26] Kesavaraj, K., Mithran, N., Venkatesan, M., Vinoth, R., Dhamodharan, N. A., & Swetha, A. (2023, December). Security framework for net gun-equipped unmanned aerial vehicles. In *AIP Conference Proceedings* (Vol. 2914, No. 1). AIP Publishing.

[27] Khalaf, M., Najm, H., Daleh, A. A., Munef, A. H., & Mojib, G. (2020, August). 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. https://doi.org/10.1109/NICST50904.2020.9280318

[28] Li, C., Si, Q., Zhao, J., & Qin, P. (2024). A robot path planning method using improved Harris Hawks optimization algorithm. *Measurement and Control*, *57*(4), 469-482. https://doi.org/10.1177/00202940231204424

[29] Mahdi, M. S., & Hassan, N. F. (2018). Design of keystream Generator utilizing Firefly Algorithm. *Journal of Al-Qadisiyah for computer science and mathematics*, *10*(3), Page-91. https://doi.org/10.29304/jqcm.2018.10.3.441

[30] Mahdi, M. S., Azeez, R. A., & Hassan, N. F. (2020). A proposed lightweight image encryption using ChaCha with hyperchaotic maps. *Periodicals of Engineering and Natural Sciences*, *8*(4), 2138-2145.

[31] Mahdi, M. S., Hassan, N. F., & Abdul-Majeed, G. H. (2021). An improved chacha algorithm for securing data on IoT devices. *SN Applied Sciences, 3*(4), 429. https://doi.org/10.1007/s42452-021-04425-7

[32] Mahdi, M., & Hassan, N. (2018). A suggested super salsa stream cipher. *Iraqi Journal for Computers and Informatics*, *44*(2), 5-10.

[33] Mohammed, R. S. (2023). Design a Lightweight Authentication Encryption Based on Stream Cipher and Chaotic Maps with Sponge Structure for Internet of Things Applications. *International Journal of Intelligent Engineering & Systems, 16*(1). https://doi.org/10.22266/ijies2023.0228.46

[34] Najm, H. (2021). 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*(10), 2541-2551.

[35] Najm, H., Hoomod, H. K., & Hassan, R. (2020). A proposed hybrid cryptography algorithm based on GOST and salsa (20). *Periodicals of Engineering and Natural Sciences*, *8*(3), 1829-1835.

[36] Najm, H., Hoomod, H., & Hassan, R. (2021). A New WoT Cryptography Algorithm Based on GOST and Novel 5d Chaotic System.

[37] Najm, H., Mahdi, M. S., & Abdulhussien, W. R. (2024). Lightweight Image Encryption Using Chacha20 and Serpent Algorithm. *Journal of Internet Services and Information Security (JISIS)*, *14*(4), 436-449. https://doi.org/10.58346/JISIS.2024.I4.027

[38] Omran, A., Abid, Y., & Bakri, B. (2024). Edge Computing Vs. Cloud Computing: Evaluating Performance, Scalability, and Security in Modern Applications. *Cyber System Journal, 1*(1), 1-8. https://doi.org/10.57238/fpj44896

[39] Paul, A., Wang, L., Selvi, S. S. D., & Rangan, C. P. (2020). Non-transferability in proxy re-encryption revisited. *Journal of Internet Services and Information Security*, *10*(3), 1-30.

[40] Qian, X., Yang, Q., Li, Q., Liu, Q., Wu, Y., & Wang, W. (2021). A novel color image encryption algorithm based on three-dimensional chaotic maps and reconstruction techniques. *IEEE Access*, *9*, 61334-61345. https://doi.org/10.1109/ACCESS.2021.3073514

[41] Said, N. M. M., Ali, S. M., Shaik, N., Begum, K. M. J., Shaban, A. A. A. E., & Samuel, B. E. (2024). Analysis of Internet of Things to enhance security using artificial intelligence-based algorithm. *Journal of Internet Services and Information Security, 14*(4), 590-604. https://doi.org/10.58346/JISIS.2024.I4.037

[42] Shareef, S. K., Sridevi, R., Raju, V. R., & Rao, K. S. (2022). Enhancement of Blockchain System in Online Transaction by Detecting Attacks Using an Intelligent Approach Recurrent Neural with Serpent Encryption (RNwSE). *International Journal of Computing and Digital Systems, 12*(1), 867-876.https://dx.doi.org/10.12785/ijcds/120172

[43] Sowmya, C. S., Vibin, R., Mannam, P., Mounika, L., Kabat, S. R., & Patra, J. P. (2023, June). Enhancing Smart Grid Security: Detecting Electricity Theft through Ensemble Deep Learning. In *2023 8th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1803-1810). IEEE. https://doi.org/10.1109/ICCES57224.2023.10192747

[44] Sreevidya, B., & Supriya, M. (2024). Trust based routing – A novel approach for data security in WSN based data critical applications. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 15*(1), 27-41. https://doi.org/10.58346/JOWUA.2024.I1.003

[45] Tahiri, M. A., Karmouni, H., Bencherqui, A., Daoui, A., Sayyouri, M., Qjidaa, H., & Hosny, K. M. (2023). New color image encryption using hybrid optimization algorithm and Krawtchouk fractional transformations. *The Visual Computer*, *39*(12), 6395-6420. https://doi.org/10.1007/s00371-022-02736-3

[46] Tripathy, B. K., Reddy Maddikunta, P. K., Pham, Q. V., Gadekallu, T. R., Dev, K., Pandya, S., & ElHalawany, B. M. (2022). Harris hawk optimization: a survey onvariants and applications. *Computational Intelligence and Neuroscience*, *2022*(1), 2218594. https://doi.org/10.1155/2022/2218594

[47] Wu, Z., & Margarita, S. (2024). Based on Blockchain and Artificial Intelligence Technology: Building Crater Identification from Planetary Imagery. *Natural and Engineering Sciences*, *9*(2), 19-32. https://doi.org/10.28978/nesciences.1567736

[48] Xu, D., Li, G., Xu, W., & Wei, C. (2023). Design of artificial intelligence image encryption algorithm based on hyperchaos. *Ain Shams Engineering Journal*, *14*(3), 101891. https://doi.org/10.1016/j.asej.2022.101891

[49] Yakubu, H. J., Aboiyar, T., & Zirra, P. B. (2016). An improved RSA image encryption algorithm using 1-D logistic map. *International Journal of communication and computer Technologies*, *4*(1), 1-10.

[50] Yousif, I. A. (2019). Proposed A permutation and substitution methods of serpent block cipher. *Ibn AL-Haitham Journal for Pure and Applied Science*, *32*(2), 131-144. https://doi.org/10.30526/32.2.2120

[51] Zendehboudi, M., Azimpour, J., & Gorginpour, H. (2014). Offering a Method for Ensuring Data Storage Security in the Cloud Network by Using Kerberos Algorithm. *International Academic Journal of Science and Engineering*, *1*(2), 75–81.

Authors Biography

Mohammed Salih Mahdi is currently Asst.Prof.Dr. in Business Information College, University of Information Technology and Communications, Iraq. His BSc degree in hiding data in 2010 and his MSc degree in a security of cloud computing in 2012 and his PhD degree in a security of IoE in 2019 from Computer Science department, University of Technology, Iraq.

Wijdan Rashid Abdulhussien is a lecturer at the College of Computer Science and Mathematics - University of Thi Qar, Educational qualifications: Bachelor's degree with first class honors in Computer Science - University of Thi Qar - College of Science, Master's degree in Computer Science - University of Basra - College of Science, PhD in Computer Science - University of Technology.

Hayder Najm is received his BSc degree in computer science from the University of Technology in 2011, received his MSc degree in computer science from the University of Technology in 2014, and received his PhD degree in computer science from the University of Technology in 2022. He is currently working at Imam Al-kadhim Unversity College (IKU), Computer Technique Engineering Department, Wasit, Iraq.

Ahmed Saad Mohammed Aloqali received his bachelor's degree in Software Engineering from the Baghdad College of Economic Sciences University, Iraq, in 2005, and a master's degree from the Faculty of Computers & Informatics, Benha University, Egypt in 2019. He is currently an Assistant Lecturer at Al-Mustansiriya University, College of Basic Education, in the Computer Department. His research interests include machine learning, data mining, and artificial intelligence.