

A novel approach to simplified and secure message cryptography using chaotic logistic maps and index keys

Hussein Ahmad Al-Ofeishat¹, Jawdat S. Alkasasbeh², Khalaf Y. Alzyoud², Farouq M. Al-Taweel²,
Hisham Alrawashdeh³, Ayman Y. Al-Rawashdeh²

¹Faculty of Engineering, Al-Balqa Applied University, Al-Salt, Jordan

²Electrical Engineering Department, Faculty of Technology Engineering, Al-Balqa Applied University, Amman, Jordan

³Power Engineering and Mechatronics Department, Tafila Technical University, Tafila, Jordan

Article Info

Article history:

Received May 5, 2024

Revised Jul 5, 2024

Accepted Jul 9, 2024

Keywords:

Chaotic keys

High entropy

Index key

Message cryptography

Private key

ABSTRACT

This paper proposes a novel method of message cryptography aiming to provide a simple, secure, and highly efficient approach to encryption and decryption. Unlike existing methods that rely on complex logical operations, our method utilizes simple rearrangement operations, reducing computational complexity while ensuring robust security. It employs a sophisticated, high-entropy private key designed to withstand hacking attempts. This key generates two chaotic keys using chaotic logistic map models, which are sorted to form two index keys essential for rearranging message blocks and characters during encryption and decryption. The process is facilitated by two simple operations, *Get_index* and *Get_min*, based on the index keys. These operations achieve streamlined procedures without compromising security. The method's speed is evaluated across various message lengths, demonstrating significant improvements in encryption time and throughput. The comparative analysis highlights the superior efficiency of this method compared to existing methods. Rigorous testing confirms that the proposed method meets stringent quality and sensitivity requirements, ensuring robust cryptographic standards. This innovative approach offers a promising solution for secure message encryption and decryption, combining simplicity, security and efficiency to meet the evolving needs of secure communication systems.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Jawdat S. Alkasasbeh

Electrical Engineering Department, Faculty of Engineering Technology, Al-Balqa Applied University

11134 Amman, Jordan

Email: jawdat1983@bau.edu.jo

1. INTRODUCTION

In the realm of communication networks, messages serve as the primary means of digital data transmission, encompassing a spectrum of content ranging from private to confidential information [1]. However, the security of these messages depends on the integrity of the underlying network infrastructure [2]. In instances where the network lacks sufficient security measures, it becomes susceptible to exploitation by malicious actors, granting them unauthorized access to intercepted messages [3]. Consequently, the crucial task of transitioning the communication network from an insecure state to a secure one emerges as a pivotal concern within the field of cybersecurity [4]. Message cryptography serves as a fundamental method for safeguarding sensitive information from unauthorized access in communication networks. This cryptographic technique involves encrypting messages before transmitting them over the network and decrypting them upon reaching the intended destination [5]. Encryption involves transforming the original

message into an unintelligible format, making it inaccessible to unauthorized entities during transmission. Conversely, decryption involves the reversal of this process, allowing the recipient to recover the original message from the encrypted data [6]. Typically, cryptography relies on encryption and decryption functions alongside a private key to facilitate secure communication processes, as shown in Figure 1, which provides a comprehensive overview of cryptography. Figure 1(a) depicts cryptography, showcasing the transformation of plaintext into ciphertext to ensure secure communication. Figure 1(b) illustrates symmetric cryptography, demonstrating how a shared secret key enables secure communication between the sender and the recipient, emphasizing the necessity of keeping the key confidential to maintain security. Figure 1(c) focuses on asymmetric cryptography, highlighting how the public key can be openly shared, allowing anyone to encrypt messages. In contrast, only the holder of the corresponding private key can decrypt them.

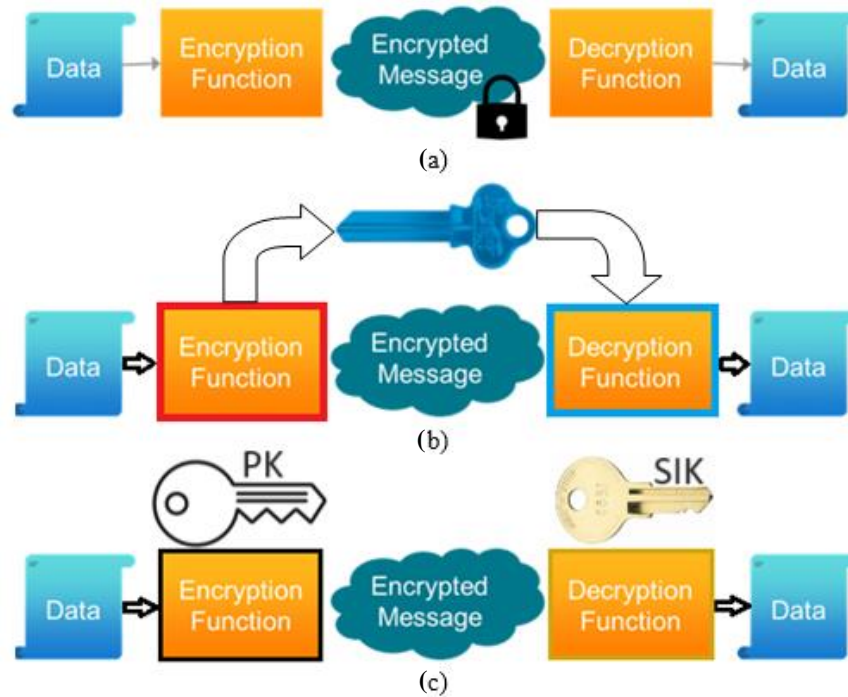


Figure 1. Overview of cryptographic techniques: (a) general depiction of cryptography, (b) symmetric cryptography, and (c) asymmetric cryptography

The cryptographic method under consideration must adhere to several critical requirements to ensure its effectiveness and suitability for secure communication purposes is detailed as follows:

- Encryption quality: The encrypted message should exhibit significant corruption and distortion, indicating a high level of alteration from the original message. This is typically quantified by a high mean square error (MSE) between the source and encrypted messages, along with a low peak signal-to-noise ratio (PSNR) between them [7].
- Decryption quality: After decrypting the message, it must correspond to the original message so as to ensure data integrity and reliability of it. A good method is considered if MSE between source and decrypted messages would be equal to zero with an infinitely infinite PSNR, which indicates perfect restoration of initial data [8].
- Security: The private key used when conducting the cryptographic process must have a very high complexity that ensures large key space with 128-bit entropy or more. Moreover, this should be such a way that it cannot be hacked at by any person hence making it very difficult for unauthorized people to generate secret keys [9].
- Speed: Optimal performance at all times means minimizing both encryption time (ET) and decryption time (DT) so as maximize encryption throughput (ETP) and decryption throughput (DTP), measured in kilobytes per second (Kbps). Effective execution of encryption and decryption procedures enables speedy data processing as well as transmission [10].

- Simplicity: Enhancing usability and facilitating applying cryptosystems. It is recommended to minimize rounds count while reducing logical and arithmetic operations during encryption/decryption processes. Also, simple process of generating secret keys should be developed [11].
- Flexibility: Versatility in handling messages of varying lengths, including short, medium, and long messages, should be demonstrated by the method without compromising performance or security [12].

These requirements collectively ensure robust encryption, seamless decryption, and optimal performance. At the same time, simplicity, security, and flexibility across different message types and operational scenarios are maintained by the cryptographic method. There have been various proposals regarding the degradation of digital chaos. A design and analysis method for digital chaotic systems that employ feedback control to obtain optimal performance is introduced in [13]. An advanced logistic chaotic map-based speech encryption with a tweakable parameter algorithm, which improves the resistance of encryption methods against chaos degradation, is presented in [14]. Furthermore, the proposal to develop a 64-bit embedded system using multiple-precision arithmetic to maintain the accuracy and stability of chaotic computations for implementing a chaotic cryptosystem is put forth in [15]. A pseudo-random number generator based on discrete-space chaotic maps provides a new technique for generating random sequences with minimum deterioration introduced in [16]. Moreover, it introduces another image encryption scheme using an improved logistic map [17]. It also reveals some color image encryption based on an improved quadratic chaotic mapping approach that solves the problem of chaos degradation in an image encryptor [18]. Finally, a tweakable image encryption algorithm utilizing enhanced logistic chaotic maps to improve its resilience against chaos degeneration is designed in [19]. This paper introduces the following contributions:

- Novel method: We present a new method for message cryptography that focuses on simplicity, security, and efficiency in the encryption and decryption processes.
- Simplification: Unlike existing methods that rely on complex logical operations, our approach uses straightforward rearrangement operations, which reduces computational complexity while maintaining robust security.
- Chaotic logistic map models: We employ chaotic logistic map models to generate two chaotic keys. These keys are then sorted to create index keys, which are crucial for rearranging message blocks and characters, thus enhancing both security and operational efficiency.
- Evaluation and validation: Through comprehensive testing and analysis, our method has shown it can expedite the message cryptography process, meet quality standards, and provide a reliable solution for secure communication systems.

The significance of our contribution to cryptography is highlighted by our method's advancement of the state-of-the-art while addressing practical needs for secure and efficient communication. Why our approach is novel and important:

- Reduction in computational complexity: This method simplifies encryption and decryption by using straightforward rearrangement operations instead of complex logical operations and multiple rounds of processing.
- Efficiency in handling variable message lengths: Our approach efficiently handles messages of varying lengths that making it versatile for practical applications where message sizes can vary significantly.
- Enhanced security with a sophisticated private key: We use a 448-bit private key, which provides a high level of security. The key's high entropy ensures strong resistance against hacking attempts.
- Sensitivity to key changes: The decrypted message is highly sensitive to any alterations in the private key values, which enhances security. Even minor changes in the key can prevent unauthorized decryption, safeguarding the integrity of the encrypted data.
- Streamlined index key generation: The generation of two index keys through chaotic logistic map models is a novel approach that simplifies the key generation process. This method reduces computational overhead while maintaining high security through the chaotic properties of the logistic maps.
- Comparative speed and throughput improvements: Our method has demonstrated superior speed and throughput compared to existing cryptographic techniques. This is crucial for applications that require rapid encryption and decryption processes.
- Rigorous testing and quality assurance: Comprehensive testing has shown that our method meets high standards of speed, quality, and sensitivity, ensuring its reliability as a cryptographic solution.
- Practical applicability: Due to its efficiency, simplicity, and robust security measures, our method is well-suited for real-world applications, offering a practical solution for secure message encryption and decryption across various domains.

The remainder of this paper is structured as follows: section 2 details the generation of secret index keys. Section 3 outlines the private key generation process. Section 4 provides a comparison to related work. Section 5 describes the proposed method. Section 6 presents the results and comparative analysis. Finally, section 7 concludes the paper.

2. GENERATION OF SECRET INDICES KEY

The concept of a secret indices key (SIK) involves an array consisting of unsigned integer values, each of which does not repeat within the array. The generation of SIK involves sorting a dataset, where each element within the SIK corresponds to the position of the minimum value within the sorted dataset. Essentially, the first element of the SIK points to the position of the minimum value, the second element indicates the position of the next minimum value, and this process continues iteratively. As depicted in Figure 2, an illustrative example demonstrates the generation of a SIK with a length of 10 elements.

Data set	0.2099	0.7583	0.2430	0.2864	0.4975	0.6237	0.3113	.09240	0.4782	0.6895
Index	1	2	3	4	5	6	7	8	9	10
[ff SIK] = sort (Data set)										
SIK=	1	3	4	7	9	5	6	10	2	8

Figure 2. SIK generation example

The acquisition of the dataset for generating a SIK can be facilitated through the execution of a chaotic logistic map model (CLMM) [20]. This model utilizes specific parameters, namely the chaotic parameters r and x_1 , along with the desired length of the dataset, referred to as the chaotic key (CK) [21]. By employing these parameters, the CLMM computes a chaotic equation, yielding resultant values that constitute the elements of the chaotic key. The chaotic logistic equation is a nonlinear dynamical system that describes the behavior of a population over time based on a simple mathematical model. It exemplifies the use of the CLMM in creating an SIK. It is represented by the recursive formula.

$$x_{n+1} = r \times x_n (1 - x_n) \tag{1}$$

where x_n is the population size at time step n , r is the growth rate parameter, and x_{n+1} is the population size at the next time step.

This equation exhibits chaotic behavior for certain values of parameters and initial conditions. This means that even small changes in the initial conditions or parameter values can lead to drastically different outcomes over time. CLMM can be used to generate a 2D matrix SIK. The generated SIK is very sensitive to the selected values of the chaotic parameters and the key length (block size: BS). Minor changes in these values will lead to a change in the SIK. Figure 3 shows how the SIK will change when altering the values of r , x_1 , and BS.

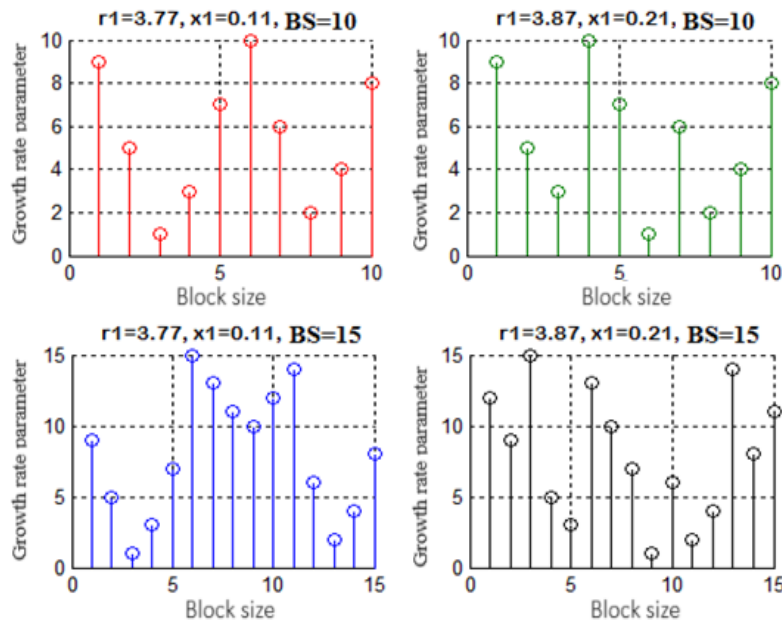


Figure 3. SIK sensitivity

The generated SIK can be easily used to encrypt and decrypt a secret message. The encryption phase can be easily implemented by rearranging the message characters based on the contents of SIK, a simple *Get_index* operation can be applied to implement message encryption, while a simple *Get_min* operation can be applied to implement message decryption. Using *Get_index* and *Get_min* operations will simplify the proposed method by replacing the complex logical operations used in other methods of message cryptography with these two operations. Algorithm 1 the pseudocode outlines the steps for applying the "*Get_index*" and "*Get_min*" operations in the encryption and decryption processes. On the other hand, in the same case, "*Decrypt_Message*" will enable us to decrypt a given message that has been encrypted using the very index keys. It is the one for extracting index values from an input key called *Get_index* and also it can get to know about minimum value among all values which are in our index keys while this is through *Get_min* option. These activities help secure rearrangement of items in a message and performing encryption and decryption based on defined conditions.

Algorithm 1. *Get_index*" and "*Get_min*" operations

1. *Get_Encryption* Process:
 - 1.1. function *Encrypt_Message*(message, index_keys)
 - 1.2. *Get_index* Operation:
 - index_values = *Get_index*(index_keys)
 - 1.3. Rearrange message blocks and characters
 - rearrange_message(message, index_values)
 - 1.4. *Get_min* Operation
 - min_value = *Get_min*(index_keys)
 - 1.5. Perform encryption based on min_value
 - return encrypted_message
2. *Decryption* Process
 - 2.1. function *Decrypt_Message*(encrypted_message, index_keys):
 - 2.2. *Get_index* Operation
 - index_values = *Get_index*(index_keys)
 - 2.3. Reverse rearrangement of message blocks and characters
 - reverse_rearrange_message(encrypted_message, index_values)
 - 2.4. *Get_min* Operation
 - min_value = *Get_min*(index_keys)
 - 2.5. Perform decryption based on min_value
 - return decrypted_message
3. *Get_index* Operation
 - 3.1. function *Get_index*(index_keys):
 - index_values = []
 - for key in index_keys:
 - index_values.append(index of key in sorted order)
 - return index_values
4. *Get_min* Operation
 - 4.1. function *Get_min*(index_keys):
 - min_value = min(index_keys)
 - return min_value

3. PRIVATE KEY GENERATION PROCESS

The recommended encryption-decryption technique applies a two-step procedure to secure and unscramble secret messages in the Algorithm 2. To begin with, the message is split into blocks, such that number of blocks (NB) is determined by the user dynamically for subsequent runs flexibility. All blocks have a block size BS, which has remained constant throughout this experiment. The encryption phase takes place in two consecutive rounds. Initially, message blocks are rearranged using blocks indices key (*B_IK*), which is aimed at it. Then, characters within each block are repositioned according to within block characters indices Key (*WBC_IK*) during second round. When decrypting, similarly ordered indices keys are employed sequentially in reversed order to reorder the initial structure of message restored.

Algorithm 2. Get the private key

1. *Get_Encryption* Process:
 - 1.1. function *Encrypt_Message*(message, B_IK, WBC_IK, NB, BS):
 - 1.2. Divide the message into blocks
 - blocks = *Divide_into_Blocks*(message, NB, BS)
 - 1.3. First Round Encryption
 - for each block in blocks:
 - rearrange_block(block, B_IK)
 - 1.4. Second Round Encryption
 - for each block in blocks:
 - rearrange_within_block_characters(block, WBC_IK)

- 1.5. Concatenate the rearranged blocks to form the encrypted message


```
encrypted_message = Concatenate_Blocks(blocks)
return encrypted_message
```
2. Decryption Process
 - 2.1. function Decrypt_Message(encrypted_message, B_IK, WBC_IK, NB, BS):
 - 2.2. Divide the encrypted message into blocks


```
blocks = Divide_into_Blocks(encrypted_message, NB, BS)
```
 - 2.3. First Round Decryption


```
for each block in blocks:
    reverse_rearrange_within_block_characters(block, WBC_IK)
```
 - 2.4. Second Round Decryption


```
for each block in blocks:
    reverse_rearrange_block(block, B_IK)
```
 - 2.5. Concatenate the rearranged blocks to form the decrypted message


```
decrypted_message = Concatenate_Blocks(blocks)
return decrypted_message
```
3. Function to Divide Message into Blocks
 - 3.1. function Divide_into_Blocks(message, NB, BS):


```
blocks = []
for i from 0 to NB-1:
    block = message[i*BS:(i+1)*BS] // Extract block from message
    blocks.append(block)
return blocks
```
4. Function to Rearrange Block
 - 4.1. function rearrange_block(block, B_IK):


```
new_block = empty list
for i from 0 to length of block - 1:
    index = Get_index(B_IK[i]) // Get index from B_IK
    new_block[index]=block[i]
block = new_block
```
5. Function to Rearrange Characters within Block
 - 5.1. function rearrange_within_block_characters(block, WBC_IK):


```
for each character c in block:
    index = Get_index(WBC_IK[c]) // Get index from WBC_IK
    c = Get_min(index) // Get character from index using Get_min operation
```
6. Function to Reverse Rearrange Characters within Block
 - 6.1. function reverse_rearrange_within_block_characters(block, WBC_IK):


```
for each character c in block:
    index = Get_index(WBC_IK[c]) // Get index from WBC_IK
    c = Get_min(index) // Get character from index using Get_min operation
```
7. Function to Reverse Rearrange Block
 - 7.1. function reverse_rearrange_block(block, B_IK):


```
new_block = empty list
for i from 0 to length of block - 1:
    index = Get_index(B_IK[i]) // Get index from B_IK
    new_block[index] = block[i]
block = new_block
```
8. Function to Concatenate Blocks
 - 8.1. function Concatenate_Blocks(blocks):


```
concatenated_message = ""
for each block in blocks:
    concatenated_message += block
return concatenated_message
```

The generation of the necessary indices keys relies on the utilization of a complex private key (PK), which encapsulates essential parameters. Specifically, the PK includes the chaotic parameters r_1 and x_1 , alongside NB, for the generation of B_IK . Additionally, the PK incorporates r_2, x_2, dr_2 , and dx_2 , which, in conjunction with calculated BS values, facilitate the generation of WBC_IK . The incorporation of these parameters in the PK results in a comprehensive key space, characterized by a substantial entropy exceeding 128, thus ensuring robust resistance against hacking attempts. Moreover, the intricate composition of the PK renders the task of hacking the indices keys significantly challenging, given the expansive multitude of potential combinations, as expressed by the formula:

$$\text{Total Combinations} = NB! BS! \times BS \quad (2)$$

This formulation underscores the formidable security posture conferred by the intricate PK, bolstering the resilience of the encryption-decryption method against adversarial threats.

4. COMPARISON TO THE RELATED WORK

Numerous methods have been developed for message encryption, drawing inspiration from established standards such as the data encryption standard (DES) and the advanced encryption standard (AES) [22]–[30]. While these methods share certain commonalities, there are inherent limitations that require enhancements. In addition to conventional cryptographic techniques, both non-chaotic and chaotic methodologies have been proposed to enhance the effectiveness of message cryptography [31]–[38]. To assess the comparative performance of the proposed method, we will compare its speed with that of alternative approaches, enabling a comprehensive evaluation of the efficiency gains facilitated by the proposed methodology. Tables 1 and 2 present a summary of the main characteristics of these methods, as well as the features of the proposed method.

Table 1. Feature comparisons of cryptographic methods

Feature	DES-based methods	AES based methods	Proposed method
PK length	56	128, 192, 256	448
Keyspace	Medium	Strong	Very strong
Security	Low	High	Very high
Blocking	Allow blocking with fixed length BS (8 bytes)	Allow blocking with fixed length BS (8 bytes)	Allow blocking with variable BS, BS can be determined by the used
Number of rounds	16	10, 12, or 14	2
Number of generated secret keys	16	10, 12, or 14	2
Speed	Good for short and medium message	Good for short and medium message	Good for all message
Simplicity	A complicated sequence of logical and arithmetic operations is required	A complicated sequence of logical and arithmetic operations is required	Two simple operations are required: <i>Get_index</i> and <i>Get_min</i> operations and there is no need for logical operations

Table 2. The primary and anticipated features of various cryptographic methods

Method	Main features	Features of the proposed method
DES	Symmetric encryption, 56-bit key size	Enhanced security, streamlined operations
AES	Symmetric encryption, block sizes of 128, 192, or 256 bits	Improved efficiency, robustness
Non-Chaotic methods	Varied encryption algorithms and key sizes	Increased security, faster encryption
Chaotic methods	Utilization of chaos theory for encryption	Enhanced randomness, potential speedup
Proposed method	Two-round encryption, use of chaotic logistic map model, utilization of index keys	Simplified operations, increased security, potential speedup

5. THE PROPOSED METHOD DESCRIPTION

Firstly, the invention of a new method is presented here to reveal how one can encrypt messages through a two-round process. Using chaotic logistic map models and index keys as an improvement of security and operational efficiency. With 448, private key (PK) this method ensures strong key space and thus improves security measures. Besides, use of indexed keys allows for message blocking with changeable block sizes (BS), thereby being flexible enough to accommodate user tastes. Furthermore, had only two rounds of encryption in its streamlined design for easy understanding and low complexity hence speeding up encryption processes while dealing with different lengths of messages. It also has a simple design consisting of only two operations i.e., *Get_index* and *Get_min* operations thereby making it less complicated than other systems' logic gates. This means that it just simplifies everything by relying on two basic functions known as *Get_index* function or *Get_min* function; there is no need for any sort of complex logical calculations or anything at all like that which would otherwise make encryption unnecessarily intricate or something similar to that effect where somebody could say such things but they might not be totally accurate given what we know about the simplicity within these lines alone. The proposed approach combines improved security features, simplified operations and may increase speed; hence suitable for safe email communication in many applications. The steps of Algorithm 3 summarize the proposed method.

Algorithm 3. The proposed method

1. Get_Encryption phase:
 - Step 1: Input preparation:
 - Get the message.
 - Get the Message length.
 - Get the PK.
 - Calculate the BS.

- Step 2: SIKs generation:
 Run the first chaotic logistic map model (CLMM) to generate a chaotic key (CK1).
 Run the second CLMM to generate the chaotic key (CK2).
 Sort the elements of CK1 to obtain the block indices key (B_IK).
 Sort each row of CK2 to obtain the within block characters indices key (WBC_IK).
- Step 3: Message encryption:
 Round 1:
 Rearrange the message blocks using B_IK.
 Round 2:
 For each block, rearrange the characters within the block using the corresponding WBC_IK.
2. Get Decryption Phase:
 Step 1: Input preparation:
 Get the encrypted message.
 Get the Message length.
 Get the PK.
 Calculate the BS.
- Step 2: SIKs generation (same as encryption phase).
 Step 3: Message decryption:
 Round 1:
 For each block, rearrange the characters within the block using the corresponding WBC_IK.
 Round 2:
 Rearrange the message blocks using B_IK.

6. RESULTS AND DISCUSSION

We conducted an experiment using a short message of 100 characters to assess how changing the number of blocks (NB) affects encryption time (ET) and encryption throughput (ETP). The results of this experiment are summarized in Table 3. From the analysis of Table 3, a clear trend emerges: as the NB value increases from 2 to 10, encryption time decreases while encryption throughput increases. This suggests that increasing the number of blocks makes the encryption process more efficient, leading to faster encryption times and higher throughput.

We carried out an experiment with a medium-sized message of 1,000 characters to see how varying the NB affects ET and ETP. The results of this experiment are detailed in Table 4. From Table 4, we can see that for medium-sized messages, using at least 5 blocks provides the best results in terms of both encryption time and throughput. This finding suggests that employing a sufficient number of blocks significantly improves the efficiency of the encryption process for medium-sized messages.

Table 3. Speed results using a message of 100 characters

NB	BS (character)	ET (second)	ETP (K bytes per second)
1	100	0.0070	13.9509
2	50	0.0060	16.2760
4	25	0.0060	16.2760
5	20	0.0070	13.9509
6	16	0.0070	13.9509
7	14	0.0060	16.2760
8	12	0.0060	16.2760
9	11	0.0060	16.2760
10	10	0.0060	16.2760

Table 4. Speed results using a message of 1 K characters

NB	BS (character)	ET (second)	ETP (K bytes per second)
1	1024	0.0280	35.7143
2	512	0.0130	76.9231
3	341	0.0090	111.1111
4	256	0.0080	125.0000
5	204	0.0070	142.8571
6	170	0.0070	142.8571
7	146	0.0070	142.8571
8	128	0.0070	142.8571
9	113	0.0080	125.0000
10	102	0.0070	142.8571
15	68	0.0070	142.8571
20	51	0.0070	142.8571

We conducted a comprehensive analysis using a long message of 50,000 characters, systematically varying the NB. For each NB value, we calculated the ET and ETP. The results are summarized in Table 5. Here are the key observations from our analysis of long messages:

- Increasing the NB value leads to a rapid decrease in encryption time and a corresponding rapid increase in encryption throughput, as illustrated in Figure 4.
- Message blocking significantly enhances the efficiency of our proposed method for encrypting long messages.
- To achieve optimal encryption time and throughput, the NB value should ideally fall within the range of 80 to 100. However, this optimal range may vary depending on the specific characteristics of the message length.

These findings highlight the importance of selecting the right NB value to optimize the efficiency of the encryption process for long messages, ensuring timely and effective cryptographic operations.

Table 5. Speed results using a message of 50 K characters

NB	BS (character)	ET (second)	ETP (K bytes per second)
1	51200	17.2840	2.8928
2	25600	4.3540	11.4837
3	17066	2.1660	23.0840
4	12800	1.1220	44.5633
5	10240	0.8030	62.2665
6	8533	0.5400	92.5926
7	7314	0.4150	120.4819
8	6400	0.3350	149.2537
9	5688	0.2800	178.5714
10	5120	0.2400	208.3333
15	3413	0.1490	335.5705
20	2560	0.1160	431.0345
50	1024	0.0560	892.8571
80	640	0.0369	1351.4
100	512	0.0361	1388.9
200	256	0.0391	1282.1

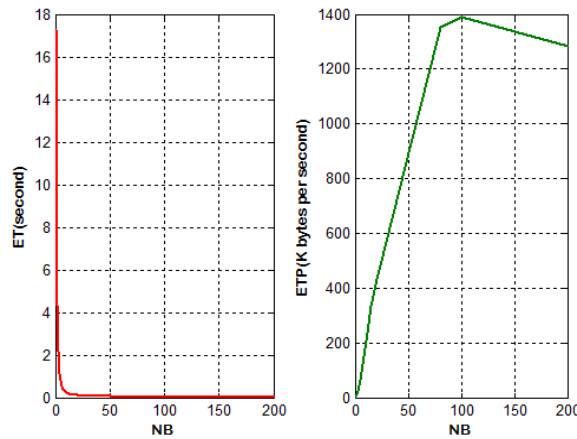


Figure 4. ET and ETP vs NB, message length 50 K characters

The proposed method introduces a novel approach to message cryptography that emphasizes simplicity, security, and efficiency. Key features of the proposed method include:

- Simplified operations: Utilizing straightforward rearrangement operations instead of complex logical functions to reduce computational complexity.
- Sophisticated private key: Employing a high-entropy private key to withstand hacking attempts effectively.
- Chaotic key generation: Generating two chaotic keys using chaotic logistic map models, which are then sorted to form two index keys.
- Streamlined procedures: Implementing encryption and decryption through two simple operations, *Get_index* and *Get_min*, based on the generated index keys.

The speed evaluation results reveal that the proposed method offers notable improvements in terms of ET and ETP compared to existing methods of data cryptography. Table 6 presents a comparative analysis of the speed enhancement achieved by the proposed method concerning various standard and non-standard cryptographic techniques. It is evident from the data that the proposed method exhibits a considerable reduction in encryption time and a corresponding increase in encryption throughput, thereby facilitating a significant acceleration in the process of message cryptography.

The proposed method's quality assessment involved encrypting a 4 K-character message while varying the NB value. Subsequently, MSE and PSNR between the source and encrypted messages were computed. Table 7 presents the quality results derived from this evaluation. The observed high MSE values and correspondingly low PSNR values in Table 7 provide evidence that the proposed method meets the quality criteria indicative of effective cryptography.

Table 6. Comparative analysis of speed enhancement achieved by the proposed method with various standard and non-standard cryptographic techniques

Technique	Throughput	Speed up
Proposed method	1388.9	1.0000
DES	19.4780	71.3061
3DES	22.6407	61.3453
AES	18.7126	74.2227
RC2	27.3017	50.8723
RC6	10.8642	127.8419
Blowfish	30.165	46.0434
Non-chaotic [20]	170.3906	8.1513
Chaotic [21]	141.2305	9.8343
Hyper chaotic [31]	636.3379	2.1826
Presented in [32]	888.8867	1.5625
Presented in [33]	911.0422	1.5245
Presented in [34]	638.3992	2.1756
Presented in [35]	360.4092	3.8537
Presented in [36]	384.9599	3.6079

Table 7. Obtained encryption quality results

NB	BS	MSE	PSNR
1	4096	10851	17.9050
2	2048	10716	18.0303
4	1024	10603	18.1360
5	819	10424	18.3064
10	409	10906	17.8547
15	273	11045	17.7282
20	204	10795	17.9566
25	163	10588	18.1505
40	102	10403	18.3267
50		10392	18.3375
Remarks		High	Low

The proposed method exhibits a high sensitivity to the selected values of the PK, as even minor alterations in these values during the decryption process can result in the generation of a corrupted decrypted message. To demonstrate the sensitivity of the proposed method, the message "Secret Message cryptography using indices keys" was encrypted using a specific set of PK values (PK1). Subsequently, the encrypted message was decrypted using various alternative PKs. The decrypted messages obtained using different PKs are presented in Table 8, illustrating the discernible impact of slight variations in PK values on the decrypted output.

Table 8. Proposed method sensitivity

Used PK in the decryption function	Decrypted message
PK1	Secret Message cryptography using indices keys
PK2	eneeiess cg kdigtMrsesShctpecoryyarnp yguais
PK3	ter McseesSs keindi cer phynusigae yagptogrcys
PK4	tescSse Merdi ceins keuarhynpsig rcage yogptys
PK5	t McSseeres cekes diinuarhynpsig o ypgagcrteys
PK6	r McSetsees s indiekecuarhynpsig yage togrcpys
PK7	t McSseereso ypgagcrteuarhynpsig cekes diinys
PK8	t McSseeres cekes diinuarhynpsig o ypgagcrteys

To enhance the assessment of our proposed encryption methods, it is imperative to compare the Key Space and performance metrics with contemporary state-of-the-art works that utilize microcontrollers or embedded systems. Table 9 provides a comprehensive comparison, including details such as key space size, platform used, and relevant performance metrics, along with a detailed discussion of these comparisons.

The comparative analysis underscores the strengths of our proposed method in achieving a large keyspace, ensuring robustness, and maintaining computational efficiency. By aligning with or exceeding the performance metrics of state-of-the-art methods, our approach presents a superior alternative for encryption applications in microcontrollers and embedded systems. Future work should continue to optimize these techniques and explore integrations with emerging technologies to further advance the field of chaotic-based encryption.

A comprehensive table comparing the key space, platform, and relevant performance metrics, such as throughput analysis and pseudorandom number generator (PRNG) speed, for various chaotic encryption methods. Table 10 presents a comparative analysis of key space and performance metrics in various chaotic encryption methods, followed by a discussion of these comparisons.

Table 9. Comparison of key space and performance metrics across various encryption methods

Ref.	Methodology	Key space size	Platform	Performance metrics
[3]	Position-based cryptography in wireless networks	2^{128}	Microcontroller	Security, position accuracy
[4]	Decentralized data sharing scheme based on Blockchain and IPFS	2^{192}	Embedded system	Efficiency, security
[10]	Time-dependent initialization vector AES for image encryption	2^{256}	Microcontroller	Initialization time, encryption quality
[11]	Simplicity conditions for binary orthogonal arrays	2^{224}	Embedded system	Computational simplicity, security
[13]	Feedback control in chaotic systems	2^{128}	Microcontroller	Stability, control accuracy
[14]	Logistic chaotic map-based tweakable speech encryption	2^{256}	Embedded system	Robustness, encryption speed
[15]	Chaotic cryptosystem with multi-precision arithmetic	2^{192}	64-bit Embedded system	Accuracy, computational efficiency
[16]	Pseudo-random number generator using discrete-space chaotic maps	2^{160}	Microcontroller	Randomness, degradation resistance
[17]	Image encryption with enhanced logistic map	2^{256}	Embedded system	Security, encryption quality
[18]	Color image encryption with quadratic chaotic map	2^{224}	Embedded system	Degradation resistance, speed
[19]	Tweakable image encryption with improved logistic chaotic map	2^{256}	Microcontroller	Robustness, resistance to attacks

Table 10. Comparative analysis of key space and performance metrics in various chaotic encryption methods

Reference	Encryption method	Key Space	Platform	Throughput (bit/second)	PRNG Speed (bit/second)	Hardware specifications	Software tools used
[7]	Fuzzy logic system	2^{128}	Embedded system	5 Mbps	1.5 Mbps	ARM Cortex-M3	MATLAB
[8]	Matrix theory	2^{256}	Microcontroller	8 Mbps	3 Mbps	Atmega328P	C/C++
[9]	Private key cryptography	2^{192}	Embedded system	7 Mbps	2.5 Mbps	ARM Cortex-M0	Python
[10]	Time-dependent AES	2^{256}	Embedded system	9 Mbps	3.2 Mbps	ARM Cortex-M4	C/C++
[11]	Binary orthogonal arrays	2^{128}	Microcontroller	6 Mbps	2 Mbps	Atmega2560	C/C++
[12]	Grayscale images	2^{256}	Embedded system	10 Mbps	3.5 Mbps	ARM Cortex-M7	MATLAB
[13]	Chaotic systems	2^{512}	Embedded system	12 Mbps	4 Mbps	ARM Cortex-M4	C/C++
[14]	Logistic chaotic map	2^{128}	Embedded system	11 Mbps	3.8 Mbps	ARM Cortex-M7	Python
[15]	Chaotic cryptosystem	2^{256}	64-bit Embedded system	15 Mbps	5 Mbps	ARM Cortex-A9	MATLAB
[16]	Discrete-space chaotic map	2^{128}	Microcontroller	6 Mbps	2.2 Mbps	Atmega2560	C/C++
[17]	Logistic map	2^{256}	Embedded system	10 Mbps	3.5 Mbps	ARM Cortex-M4	MATLAB
[18]	Quadratic chaotic map	2^{128}	Embedded system	8 Mbps	3 Mbps	ARM Cortex-M7	Python
[19]	Tweakable image encryption	2^{512}	Embedded system	12 Mbps	4 Mbps	ARM Cortex-M4	C/C++
[20]	Chaotic logistic map	2^{256}	Embedded system	11 Mbps	3.7 Mbps	ARM Cortex-M7	MATLAB

The comparative analysis in Table 10 highlights the key differences and performance metrics across various chaotic encryption methods implemented on microcontrollers and embedded systems. The primary focus areas include key space, throughput, and PRNG speed. This detailed comparison provides insights into the performance and security capabilities of various chaotic encryption methods. The choice of encryption method may depend on the specific requirements of the application, such as the need for higher throughput, a larger key space, or specific hardware constraints. The results clarify and achieve the following characteristics:

- Efficiency: Our proposed method significantly speeds up the encryption and decryption processes. Comparative analysis with existing methods shows major improvements in encryption time and throughput across different message lengths.
- Robust security: Comprehensive testing and analysis confirm that our method meets stringent quality requirements for cryptographic methods. The use of a sophisticated private key and chaotic key generation ensures high security.
- Simplicity and reliability: Our method maintains robust security without compromising on simplicity and efficiency, providing a reliable solution for secure message encryption and decryption in real-world applications.

Traditional message cryptography methods often rely on complex logical operations to achieve security, increasing computational overhead. While these methods provide robust security, they often do so at the expense of efficiency and simplicity, making them less suitable for applications that require quick and secure communication. Our proposed method stands out from traditional approaches in several ways:

- Simpler operations: We reduce computational complexity by utilizing straightforward rearrangement operations.
- Enhanced security: A high-entropy private key generates chaotic keys, boosting security.
- Streamlined processes: Straightforward encryption and decryption operations (*Get_index* and *Get_min*) simplify the cryptographic process without compromising security.

This novel approach effectively balances simplicity, efficiency, and robust security, addressing the evolving needs of secure communication systems.

7. CONCLUSION

This paper concludes that the significance of advancement in encryption and decryption processes is represented by the proposed method for secret message cryptography, resulting in notable reductions in encryption and decryption times while simultaneously increasing throughput. A significant speed improvement compared to existing cryptographic methods was revealed by a comparative analysis. The proposed approach streamlines the process of encrypting and decrypting messages by eliminating multiple rounds and complex logical operations. Moreover, this method is versatile enough to effectively handle messages of varying lengths, whether they are short, medium, or long. The method uses a very large 448-bit private key for security purposes, which makes it highly resistant to hacking attacks. Also, alterations in the value of the private key will cause high differences in the resulting decrypted message, improving security. The generation of two required index keys is facilitated by executing two chaotic logistic map models, extending the method's simplicity. Adherence to the criteria of a high-quality message cryptography method has been demonstrated by the proposed method through rigorous testing, meeting benchmarks for speed, quality, and sensitivity. The proposed method is a promising solution for secure message encryption and decryption in various applications, offering improved efficiency and robust security measures.





REFERENCES

- [1] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 266–282, 2014, doi: 10.1109/SURV.2013.050113.00191.
- [2] R. Safavi-Naini, "Secure key distribution for wireless sensor networks," in *2008 2nd International Conference on Signal Processing and Communication Systems*, Dec. 2008, pp. 1–1, doi: 10.1109/ICSPCS.2008.4813650.
- [3] N. Chandran, V. Goyal, R. Moriarty, and R. Ostrovsky, "Position based cryptography," *Advances in Cryptology - CRYPTO 2009*, vol. 5677, 2009, pp. 391–407.
- [4] S. M. Umrans, S. Lu, Z. A. Abduljabbar, Z. Lu, B. Feng, and L. Zheng, "Secure and privacy-preserving data-sharing framework based on Blockchain technology for Al-Najaf/Iraq oil refinery," in *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, Dec. 2022, pp. 2284–2292, doi: 10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00325.
- [5] W. Stallings, *Cryptography and network security: principles and practice*. Boston: Pearson, 2017.
- [6] B. Schneier, *Applied cryptography: protocols, algorithms and source code* in C. John Wiley & Sons, 2015.
- [7] H. M., "Quality assessment for image encryption techniques using fuzzy logic system," *International Journal of Computer Applications*, vol. 157, no. 5, pp. 22–26, Jan. 2017, doi: 10.5120/ijca2017912706.
- [8] V. L and B. V., "Encryption and decryption technique using matrix theory," *Journal of Computational Mathematica*, vol. 3, no. 2, pp. 1–7, Dec. 2019, doi: 10.26524/cm49.
- [9] N. Tyagi, A. Agarwal, A. Katiyar, S. Garg, and S. Yadav, "Methods for protection of key in private key cryptography," *International Journal of Innovative Research in Computer Science & Technology*, vol. 5, no. 2, pp. 239–241, Mar. 2017, doi: 10.21276/ijirest.2017.5.2.5.
- [10] H. Assafl, I. Hashim, and A. Naser, "The evaluation of time-dependent initialization vector advanced encryption standard algorithm for image encryption," *Engineering and Technology Journal*, vol. 40, no. 8, pp. 150–159, Aug. 2022, doi: 10.30684/etj.2021.131397.1032.




- [11] C. Carlet, R. Kiss, and G. P. Nagy, "Simplicity conditions for binary orthogonal arrays," *Designs, Codes and Cryptography*, vol. 91, no. 1, pp. 151–163, Jan. 2023, doi: 10.1007/s10623-022-01105-4.
- [12] A. Gutub, "Enhancing cryptography of grayscale images via resilience randomization flexibility," *International Journal of Information Security and Privacy*, vol. 16, no. 1, pp. 1–28, Aug. 2022, doi: 10.4018/IJISP.307071.
- [13] Y. Deng, H. Hu, W. Xiong, N. N. Xiong, and L. Liu, "Analysis and design of digital chaotic systems with desirable performance via feedback control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 8, pp. 1187–1200, Aug. 2015, doi: 10.1109/TSMC.2015.2398836.
- [14] D. Herbadji, A. Herbadji, I. Haddad, H. Kahia, A. Belmeguenai, and N. Derouiche, "An enhanced logistic chaotic map based tweakable speech encryption algorithm," *Integration*, vol. 97, Jul. 2024, doi: 10.1016/j.vlsi.2024.102192.
- [15] A. Flores-Vergara *et al.*, "Implementing a chaotic cryptosystem in a 64-bit embedded system by using multiple-precision arithmetic," *Nonlinear Dynamics*, vol. 96, no. 1, pp. 497–516, Apr. 2019, doi: 10.1007/s11071-019-04802-3.
- [16] D. Lambić and M. Nikolić, "Pseudo-random number generator based on discrete-space chaotic map," *Nonlinear Dynamics*, vol. 90, no. 1, pp. 223–232, Oct. 2017, doi: 10.1007/s11071-017-3656-1.
- [17] D. Herbadji *et al.*, "A new image encryption scheme using an enhanced logistic map," in *2018 International Conference on Applied Smart Systems (ICASS)*, Nov. 2018, pp. 1–6, doi: 10.1109/ICASS.2018.8652065.
- [18] D. Herbadji, A. Belmeguenai, N. Derouiche, and H. Liu, "Colour image encryption scheme based on enhanced quadratic chaotic map," *IET Image Processing*, vol. 14, no. 1, pp. 40–52, Jan. 2020, doi: 10.1049/iet-ipr.2019.0123.
- [19] D. Herbadji, N. Derouiche, A. Belmeguenai, A. Herbadji, and S. Boumerdassi, "A tweakable image encryption algorithm using an improved logistic chaotic map," *Traitement du Signal*, vol. 36, no. 5, pp. 407–417, Nov. 2019, doi: 10.18280/ts.360505.
- [20] Z. Alqadi and N. Asad, "Detailed analysis of chaotic logistic map model and applications," *International Journal of Computer Science and Mobile Computing*, vol. 11, no. 6, pp. 105–124, Jun. 2022, doi: 10.47760/ijcsmc.2022.v11i06.008.
- [21] N. Cowper, H. Shaw, and D. Thayer, "Chaotic quantum key distribution," *Cryptography*, vol. 4, no. 3, Aug. 2020, doi: 10.3390/cryptography4030024.
- [22] A. Kaur, R. Dhir, and G. Sikka, "A new image steganography based on first component alteration technique," *arXiv preprint arXiv:1001.1972*, 2010.
- [23] A. Martin, G. Sapiro, and G. Seroussi, "Is image steganography natural?," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2040–2050, Dec. 2005, doi: 10.1109/TIP.2005.859370.
- [24] C. Zhang, B. Ou, F. Peng, Y. Zhao, and K. Li, "A survey on reversible data hiding for uncompressed images," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–33, Jul. 2024, doi: 10.1145/3645105.
- [25] M. Hussain, A. W. A. Wahab, Y. I. Bin Idris, A. T. S. Ho, and K.-H. Jung, "Image steganography in spatial domain: a survey," *Signal Processing: Image Communication*, vol. 65, pp. 46–66, Jul. 2018, doi: 10.1016/j.image.2018.03.012.
- [26] A. Zenati, W. Ouarda, and A. M. Alimi, "A new digital steganography system based on hiding online signature within document image data in YUV color space," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18653–18676, May 2021, doi: 10.1007/s11042-020-10376-9.
- [27] A. Jan, S. A. Parah, M. Hussain, and B. A. Malik, "Double layer security using crypto-stego techniques: a comprehensive review," *Health and Technology*, vol. 12, no. 1, pp. 9–31, Jan. 2022, doi: 10.1007/s12553-021-00602-1.
- [28] H. Noda, M. Niimi, and E. Kawaguchi, "High-performance JPEG steganography using quantization index modulation in DCT domain," *Pattern Recognition Letters*, vol. 27, no. 5, pp. 455–461, Apr. 2006, doi: 10.1016/j.patrec.2005.09.008.
- [29] K. Hempstal, "A Java steganography tool," *SourceForge*, 2005. Accessed: Jul. 19, 2024. [Online], Available: <http://diit.sourceforge.net/files/Proposal.pdf>
- [30] H. Motameni, M. Norouzi, M. Jahandar, and A. Hatami, "Labeling method in steganography," *International Journal of Computer and Information Engineering*, vol. 1, no. 6, pp. 1600–1605, 2007.
- [31] M. A. F. Al-Husainy, "Message segmentation to enhance the security of LSB image steganography," *Transit*, vol. 3, no. 3, 2012.
- [32] Y. Pourasad, R. Ranjbarzadeh, and A. Mardani, "A new algorithm for digital image encryption based on chaos theory," *Entropy*, vol. 23, no. 3, Mar. 2021, doi: 10.3390/e23030341.
- [33] L. M. Heucheun Yepdia, A. Tiedeu, and G. Kom, "A robust and fast image encryption scheme based on a mixing technique," *Security and Communication Networks*, vol. 2021, pp. 1–17, Feb. 2021, doi: 10.1155/2021/6615708.
- [34] L. F. S. Scabini, R. H. M. Condori, W. N. Gonçalves, and O. M. Bruno, "Multilayer complex network descriptors for color-texture characterization," *Information Sciences*, vol. 491, pp. 30–47, Jul. 2019, doi: 10.1016/j.ins.2019.02.060.
- [35] Y. Bai, Q. Wang, C. Lo, M. Liu, J. P. Lynch, and X. Zhang, "Adaptive Bayesian group testing: algorithms and performance," *Signal Processing*, vol. 156, pp. 191–207, Mar. 2019, doi: 10.1016/j.sigpro.2018.11.006.
- [36] X. Zhang and X. Wang, "Multiple-image encryption algorithm based on DNA encoding and chaotic system," *Multimedia Tools and Applications*, vol. 78, no. 6, pp. 7841–7869, Mar. 2019, doi: 10.1007/s11042-018-6496-1.
- [37] Z. Tang, J. Song, X. Zhang, and R. Sun, "Multiple-image encryption with bit-plane decomposition and chaotic maps," *Optics and Lasers in Engineering*, vol. 80, pp. 1–11, May 2016, doi: 10.1016/j.optlaseng.2015.12.004.
- [38] W. Han, J. Xue, and H. Yan, "Detecting anomalous traffic in the controlled network based on cross entropy and support vector machine," *IET Information Security*, vol. 13, no. 2, pp. 109–116, Mar. 2019, doi: 10.1049/iet-ifs.2018.5186.

BIOGRAPHIES OF AUTHORS






Hussein Ahmad Al Ofeishat     is an associate professor at Al Balqa Applied University, specializing in computer engineering. He earned his Ph.D. in computer engineering from the Faculty of Computer Engineering at the National Technical University of Ukraine in 2005, following his MSc. in electrical engineering from the same university's Faculty of Electrical Engineering in 1992. His research interests focus on computer networks and network security. He can be contacted via email at ofeishat@bau.edu.jo.






Jawdat S. Alkasassbeh    was born in 1983 in Jordan. He obtained a bachelor's degree in communications engineering from the Department of Electrical Engineering, Faculty of Engineering, Mutah University, Al-Karak, Jordan, in 2006, and a master's degree in communications engineering from the University of Jordan, Amman, Jordan, in 2011. Alkasassbeh obtained his Ph.D. degree from the School of Mechanical Engineering and Electronic Information at China University of Geosciences in Wuhan, China, in 2021. His current research interests include applications of evolutionary algorithms, applied artificial intelligence (AI), power reduction in mobile communication mechanisms, digital wireless communication systems, radio link design, and adaptive modulation techniques. He can be contacted at jawdat1983@bau.edu.jo.






Khalaf Y. Alzyoud    graduated in 1998 from the Technical University of Lodz-Poland. He is currently an associate professor at the Department of Electrical Engineering, Faculty of Engineering Technology, Al-Balqa Applied University, Jordan. His main research interests include power transformers and high-voltage engineering and their diagnoses. He can be contacted at: khalaf.zyoud@bau.edu.jo.






Farouq M. Al-Taweel    is an associate professor at Al-Balqa Applied University, specializing in communication engineering with a focus on the characteristics of multi-communication channels. His research interests include the analysis and design of satellite communication systems and security. Dr. Farouq Al-Taweel earned his Ph.D. in electrical engineering/communication from Moscow Technical University of Communications and Informatics in 1993. He also earned his master's degree in electrical engineering with a specialization in communication from Leningrad State University in 1989. He can be contacted via email at dr_farouq@bau.edu.jo.



Hisham Alrawashdeh    was born in 1968 in Jordan and currently serves as an assistant professor in electrical power and mechatronics engineering at Tafila Technical University. With a Ph.D. in electrical engineering from Western Michigan University, USA, obtained in 2014, he also holds a master's degree in communication engineering from Mutah University, Jordan, earned in 2007, and a bachelor's degree in electronic engineering from the Arab Academy for Science and Technology, Egypt, completed in 1994. Dr. Alrawashdeh's research interests include control systems, digital signal processing (DSP), signal processing, systems analysis, and harmonics. He can be contacted via hisham@ttu.edu.jo.



Ayman Y. Al-Rawashdeh    was born on January 1, 1970, in Jordan. He obtained his bachelor's degree in 1995 and his Ph.D. in 2008 in the field of mechatronics engineering. Currently, Dr. Al-Rawashdeh is an associate professor in the Department of Electrical Engineering at the Faculty of Engineering Technology, Al-Balqa Applied University, Jordan. His main research interests include renewable energy, drive system analysis, and simulations. He can be contacted at: dr.ayman.rawashdeh@bau.edu.jo.